

**SEMINARIO DE INVESTIGACIÓN EN ROBÓTICA MÓVIL  
AUTÓNOMA**

**DANIEL RICARDO CARRILLO DUARTE  
GARY ALEJANDRO SÁNCHEZ AMAYA  
JONATHAN FABIAN NIÑO CASTILLO**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS  
ESCUELA DE INGENIERÍA MECÁNICA  
BUCARAMANGA**

**2013**

**SEMINARIO DE INVESTIGACIÓN EN ROBÓTICA MÓVIL  
AUTÓNOMA**

**DANIEL RICARDO CARRILLO DUARTE  
GARY ALEJANDRO SÁNCHEZ AMAYA  
JONATHAN FABIAN NIÑO CASTILLO**

**Trabajo de grado para optar al título de ingeniero mecánico**

**Director:  
CARLOS BORRÁS PINILLA  
Ing. Mecánico**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS  
ESCUELA DE INGENIERÍA MECÁNICA  
BUCARAMANGA**

**2013**

## AGRADECIMIENTOS

*A Dios en primer lugar por ser mi guía y mi fortaleza, por acompañarme en los buenos y malos momentos y por colmarme diariamente de tantas bendiciones.*

*A mi mamá y a mi papá, ellos son mi más grande tesoro, mi familia, mi hogar, mi apoyo incondicional, el motivo de mi felicidad y a quienes más amo en el mundo. Gracias a ellos por toda su entrega como padres, amigos, consejeros y maestros, por enseñarme, corregirme, cuidarme y por estar presentes en todo momento de mi vida, gran parte de este proyecto es gracias al fruto de su amor y de sus grandes esfuerzos.*

*A mi tío Gabriel quien fue un extraordinario apoyo durante el transcurso de mi vida universitaria y quien ha sido como mi segundo padre; gracias por su compañía, sus consejos, su buena energía y alegría que lo caracteriza en todo momento.*

*A mis profesores, compañeros de carrera y sobre todo a mis compañeros de proyecto Jonathan y Gary quienes son también mis amigos y su esfuerzo y trabajo en equipo hicieron posible el desarrollo y conclusión de este proyecto.*

Daniel Ricardo Carrillo Duarte

## AGRADECIMIENTOS

*A Dios primeramente por darme la oportunidad de realizar y culminar este proyecto de una forma satisfactoria. A Él le debo todo.*

*A mis padres, pues ellos son el motor que me impulsa a seguir adelante en mi existencia y sin su apoyo y esfuerzo el desarrollo de este trabajo no hubiese sido posible.*

*A mi Whicky, ese ser maravilloso que ha estado conmigo por muchos años y quien me llena de sonrisas cada día aún en momentos de dificultad.*

*A mis profesores, compañeros de carrera y en especial a mis compañeros de proyecto por hacer parte de este proceso.*

Gary Alejandro Sánchez Amaya

## AGRADECIMIENTOS

*A Dios, por estar conmigo a cada paso, por fortalecer mi corazón e iluminar mi mente y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo el periodo de estudio.*

*A mis padres por el esfuerzo realizado y el apoyo en mis estudios, de no ser así no hubiese sido posible.*

*A mi familia y amigos que me brindan el apoyo, la alegría y me dan la fortaleza necesaria para seguir adelante.*

Jonathan Fabian Niño Castillo

---

# ÍNDICE GENERAL

INTRODUCCIÓN	58
<b>I SEMINARIO DE INVESTIGACIÓN</b>	<b>59</b>
1. GENERALIDADES	60
1.1. ORGANIZACIÓN DEL SEMINARIO DE INVESTIGACIÓN	61
1.2. TEMA DEL SEMINARIO DE INVESTIGACIÓN . . . . .	62
1.3. DIRECCIÓN DEL SEMINARIO DE INVESTIGACIÓN . . .	62
1.4. METODOLOGÍA DEL SEMINARIO DE INVESTIGACIÓN	63
2. PLANEACIÓN	64
2.1. ESTUDIO BIBLIOGRÁFICO . . . . .	66
3. EJECUCIÓN	71
3.1. ACTAS . . . . .	71
3.2. TALLERES DE APLICACIÓN . . . . .	72
3.3. ROBOT LEVI . . . . .	73
4. FINALIZACIÓN	77
4.1. PRESENTACIONES . . . . .	77
4.2. DOCUMENTO . . . . .	77

<b>II</b>	<b>MEMORIAS Y RESULTADOS</b>	<b>80</b>
<b>5.</b>	<b>INTRODUCCIÓN A LOS ROBOTS MÓVILES</b>	<b>81</b>
5.1.	HISTORIA . . . . .	81
5.2.	LA PALABRA ROBOT . . . . .	86
5.3.	ORIGEN Y DESARROLLO DE LA ROBOTICA . . . . .	87
5.4.	LA ROBÓTICA EN COLOMBIA . . . . .	98
5.4.1.	Universidad Nacional . . . . .	101
5.4.2.	Universidad del Valle . . . . .	103
5.4.3.	Pontificia Universidad Javeriana . . . . .	104
5.5.	LA ROBÓTICA EN LA UNIVERSIDAD INDUSTRIAL DE SANTANDER . . . . .	105
5.5.1.	Diseño y construcción de una plataforma robótica cooperativa basada en un sistema embebido para la gestión y manejo de inventarios . . . . .	105
5.5.2.	Diseño y construcción de un prototipo de robot móvil autónomo de exploración, 2009 . . . . .	105
5.5.3.	Robot cartesiano neumático para el laboratorio de sistemas mecatrónicos de la escuela a de ingeniería mecánica. diseño y construcción, 2010 . . . . .	106
5.5.4.	Diseño y construcción de un helibot 3 dof como herramienta de investigación y desarrollo para el laboratorio de sistemas dinámicos e ingeniería del control . . . . .	107
5.5.5.	Diseño y construcción de una plataforma de entrenamiento para helicóptero a escala 60 . . . . .	107
5.5.6.	Diseño y construcción de un prototipo de robot scara con tres grados de libertad . . . . .	108
5.5.7.	Seminario de investigación en robotica industrial . . . . .	109
5.5.8.	Diseño y construcción de un prototipo experimental de un vehiculo electrico autobalanceado unipersonal veau . . . . .	110
5.6.	DEFINICIÓN DE ROBÓTICA . . . . .	110
5.7.	CLASIFICACION DE LOS ROBOTS . . . . .	111
5.7.1.	Según su cronología . . . . .	111
5.7.2.	Según su morfología . . . . .	113
5.7.3.	Según su grado de inteligencia . . . . .	115
5.7.4.	Según su capacidad de movimiento . . . . .	117
5.7.5.	Según su aplicación . . . . .	117

<b>6. ESTADO DEL ARTE DE LOS ROBOTS MÓVILES</b>	<b>123</b>
6.1. METODOLOGÍA DE DISEÑO . . . . .	123
6.2. LA PERCEPCIÓN Y CONOCIMIENTO DE LA SITUACIÓN	123
6.3. CONTROL DE ROBOTS MÓVILES . . . . .	124
6.3.1. Control de formación . . . . .	124
6.3.2. Control de robots terrestres no Holonómicos . . . . .	125
6.3.3. Control de vehículos aéreos no tripulados. . . . .	125
6.4. ROBOTS MÓVILES . . . . .	125
6.4.1. Aplicaciones de robots móviles . . . . .	125
6.4.2. Tecnologías utilizadas en los sistemas motrices . . . . .	131
6.4.3. Robots terrestres . . . . .	133
6.4.4. Robots biomiméticos . . . . .	141
6.4.5. Robots espaciales . . . . .	142
6.4.6. Robots acuáticos . . . . .	147
6.4.7. Robots Aereos. . . . .	154
6.5. COMPAÑÍAS FABRICANTES DE ROBOTS MÓVILES EN LA ACTUALIDAD . . . . .	158
6.5.1. Robots de inspección, exploración, rescate y defensa . .	159
6.5.2. Robots de entretenimiento y aprendizaje . . . . .	161
6.5.3. Robots de vigilancia y seguridad . . . . .	162
6.5.4. Robots en sectores de construcción y minería . . . . .	162
6.5.5. Robots en sectores ganadero, agrícola y forestal . . . . .	163
6.5.6. Robots para limpieza y mantenimiento . . . . .	163
6.5.7. Robots para cuidados y ayuda a personas . . . . .	164
<b>7. LOCOMOCIÓN DE LOS ROBOTS MÓVILES</b>	<b>165</b>
7.1. INTRODUCCIÓN . . . . .	165
7.2. GENERALIDADES DE LA LOCOMOCIÓN . . . . .	166
7.2.1. Niveles en la locomoción . . . . .	166
7.2.2. Tipos de locomoción . . . . .	167
7.2.3. Diseño y Locomoción de robots móviles . . . . .	167
7.2.4. Las cuestiones clave para la locomoción . . . . .	168
7.3. ROBOTS MÓVILES CON PIERNAS . . . . .	170
7.3.1. Configuraciones de las piernas y la estabilidad . . . . .	170
7.3.2. Ejemplos de locomoción de robots de piernas . . . . .	172
7.4. ROBOTS MÓVILES CON RUEDAS . . . . .	179
7.4.1. Locomoción de ruedas: el espacio de diseño . . . . .	179

7.4.2.	Locomoción de ruedas: estudios del caso . . . . .	184
7.5.	<b>ROBOTS TREPADORES</b> . . . . .	188
7.5.1.	Introducción . . . . .	188
7.5.2.	Mecanismos de locomoción de los robots trepadores . . . . .	190
7.5.3.	Mecanismos de adhesión de robots trepadores . . . . .	199
7.6.	<b>BIROBOTS</b> . . . . .	209
7.6.1.	Locomoción de arrastre . . . . .	209
7.6.2.	Locomoción de piernas (Caminadores) . . . . .	209
7.6.3.	Locomoción mediante saltos . . . . .	212
7.6.4.	Locomoción en el agua (nadadores) . . . . .	212
7.6.5.	Locomoción de vuelo . . . . .	213
7.6.6.	Robots ápodos . . . . .	215
<b>8.</b>	<b>CINEMÁTICA DE LOS ROBOTS MÓVILES</b>	<b>227</b>
8.1.	<b>GENERALIDADES DE LA CINEMÁTICA.</b> . . . . .	227
8.2.	<b>SUPUESTOS CONSIDERADOS</b> . . . . .	227
8.2.1.	Configuraciones de ruedas . . . . .	228
8.2.2.	Algunos conceptos previos . . . . .	233
8.3.	<b>MODELOS Y RESTRICCIONES CINEMÁTICAS</b> . . . . .	235
8.3.1.	Representando la posición del robot . . . . .	235
8.3.2.	Modelos cinemáticos de avance . . . . .	237
8.3.3.	Restricciones cinemáticas de la ruedas . . . . .	240
8.3.4.	Restricciones Cinemáticas del Robot . . . . .	249
8.3.5.	Algunos ejemplos de modelos y restricciones cinemáticas del robot . . . . .	251
8.4.	<b>MANIOBRABILIDAD DEL ROBOT MÓVIL</b> . . . . .	254
8.4.1.	Grado de movilidad . . . . .	255
8.4.2.	El grado de maniobrabilidad . . . . .	259
8.4.3.	Maniobrabilidad del Robot . . . . .	260
8.5.	<b>ESPACIO DE TRABAJO DEL ROBOT MÓVIL</b> . . . . .	262
8.5.1.	Grados de libertad . . . . .	263
8.5.2.	Robots Holonómicos . . . . .	264
8.5.3.	Ruta de acceso y consideraciones de trayectoria . . . . .	266
8.6.	<b>MÁS ALLÁ DE LA CINEMÁTICA BÁSICA</b> . . . . .	268
8.7.	<b>CONTROL DE MOVIMIENTO</b> . . . . .	268
8.7.1.	Introducción . . . . .	268

8.7.2.	Los modelos de control . . . . .	270
8.7.3.	Control de lazo abierto (Siguiendo una trayectoria) . . .	273
8.7.4.	Control de retroalimentación . . . . .	274
8.8.	<b>CINEMATICA DE ROBOTS RODANTES . . . . .</b>	<b>279</b>
8.8.1.	Robots con ruedas convencionales . . . . .	280
8.8.2.	Los robots con ruedas omnidireccionales . . . . .	286
<b>9.</b>	<b>DINÁMICA DE LOS ROBOTS MÓVILES . . . . .</b>	<b>291</b>
9.1.	<b>ROBOTS MÓVILES HILARE . . . . .</b>	<b>291</b>
9.1.1.	Coordenadas generalizadas y velocidades generalizadas .	293
9.1.2.	La aceleración de los centros de masa del robot y las ruedas	294
9.1.3.	Diagramas de cuerpo libre y cinético . . . . .	295
9.1.4.	Las ecuaciones de movimiento . . . . .	295
9.1.5.	El papel de la ecuación de restricción No holonómica . .	298
9.1.6.	La forma de primer orden de las ecuaciones dinámicas .	299
9.2.	<b>DINÁMICA DE ROBOTS MÓVILES NO HOLONOMICOS .</b>	<b>299</b>
9.3.	<b>DINÁMICA DE ROBOTS RODANTES . . . . .</b>	<b>304</b>
9.3.1.	Robots con ruedas convencionales . . . . .	304
9.3.2.	Robots con ruedas omnidireccionales . . . . .	314
<b>10.</b>	<b>PERCEPCIÓN EN LOS ROBOTS MÓVILES . . . . .</b>	<b>325</b>
10.1.	<b>SENSORES PARA ROBOTS MÓVILES . . . . .</b>	<b>325</b>
10.1.1.	Clasificación de los sensores . . . . .	325
10.1.2.	Caracterización del rendimiento del sensor . . . . .	326
10.1.3.	Sensores de Ruedas/motor . . . . .	331
10.1.4.	Detectores de rumbo . . . . .	332
10.1.5.	Acelerómetros . . . . .	334
10.1.6.	Unidad de medición inercial (IMU) . . . . .	336
10.1.7.	Balizas de tierra . . . . .	337
10.1.8.	Alcance activo . . . . .	338
10.1.9.	Sensores de movimiento y velocidad . . . . .	350
10.2.	<b>FUNDAMENTOS DE LA VISIÓN POR COMPUTADOR . .</b>	<b>351</b>
10.2.1.	La cámara digital . . . . .	352
10.2.2.	Formación de la imagen . . . . .	354
10.2.3.	Cámaras omnidireccionales . . . . .	362
10.2.4.	Estructura de estéreo . . . . .	368
10.2.5.	Estructura de movimiento . . . . .	378

10.2.6. Movimiento y flujo óptico . . . . .	382
10.2.7. Seguimiento de Color . . . . .	385
10.3. <b>FUNDAMENTOS DEL PROCESAMIENTO DE IMÁGENES</b> . . . . .	388
10.3.1. Filtrado de Imagen . . . . .	388
10.3.2. Detección de bordes . . . . .	392
10.3.3. Cálculo de similitud de imagen . . . . .	398
10.4. <b>EXTRACCIÓN DE CARACTERÍSTICAS</b> . . . . .	399
10.5. <b>EXTRACCIÓN DE CARACTERÍSTICAS DE IMAGEN: DETECTORES DE PUNTO DE INTERÉS</b> . . . . .	401
10.5.1. Introducción . . . . .	402
10.5.2. Propiedades del detector de característica ideal . . . . .	402
10.5.3. Detectores de Esquina . . . . .	404
10.5.4. Invariancia a los cambios geométricos y fotométricos . . . . .	407
10.5.5. Detectores de mancha (Blob). . . . .	414
10.6. <b>RECONOCIMIENTO DE LUGAR</b> . . . . .	420
10.6.1. Introducción . . . . .	420
10.6.2. De la bolsa de características a las palabras visuales . . . . .	420
10.6.3. Reconocimiento de ubicación eficiente usando un archivo invertido . . . . .	422
10.6.4. Verificación geométrica para el reconocimiento de lugar robusto . . . . .	423
10.6.5. Otras representaciones de imágenes para el reconocimiento lugar . . . . .	423
10.7. <b>EXTRACCIÓN DE CARACTERÍSTICAS BASADA EN DATOS DE ALCANCE (LÁSER, ULTRASONIDOS)</b> . . . . .	427
10.7.1. Línea de montaje . . . . .	427
10.7.2. Seis algoritmos de extracción de línea . . . . .	432
10.7.3. Características de histograma de alcance . . . . .	441
10.7.4. Extracción de otras características geométricas . . . . .	442
<b>11. LOCALIZACIÓN EN LOS ROBOTS MÓVILES</b> . . . . .	<b>444</b>
11.1. <b>GENERALIDADES</b> . . . . .	444
11.2. <b>EL DESAFÍO DE LA LOCALIZACIÓN: EL RUIDO Y ALIASING</b> . . . . .	445
11.2.1. Ruido del sensor . . . . .	446
11.2.2. Aliasing del sensor . . . . .	446
11.2.3. Ruido del actuador Effector . . . . .	446

11.2.4. Un modelo de error de estimación de la posición odométrica . . . . .	447
11.3. LOCALIZAR O NO LOCALIZAR: LOCALIZACIÓN BASADA EN NAVEGACIÓN VERSUS SOLUCIONES PROGRAMADAS . . . . .	450
11.4. REPRESENTACIÓN DE LA CREENCIA . . . . .	454
11.4.1. Creencia de hipótesis única . . . . .	454
11.4.2. Creencia de múltiples hipótesis . . . . .	454
11.5. REPRESENTACIÓN DEL MAPA . . . . .	457
11.5.1. Representaciones continuas . . . . .	457
11.5.2. Estrategias de descomposición . . . . .	459
11.6. LOCALIZACIÓN PROBABILÍSTICA BASADA EN MAPAS . . . . .	461
11.6.1. Generalidades . . . . .	461
11.6.2. El problema de localización de robot . . . . .	461
11.6.3. Conceptos básicos de la teoría de la probabilidad . . . . .	463
11.6.4. Terminología . . . . .	465
11.6.5. Los ingredientes de la localización basada en mapa probabilístico . . . . .	466
11.6.6. Clasificación de los problemas de localización . . . . .	467
11.6.7. Localización de Markov . . . . .	468
11.6.8. Localización del filtro de Kalman . . . . .	478
11.7. CONSTRUCCIÓN AUTÓNOMA DE MAPAS . . . . .	488
11.7.1. Generalidades . . . . .	488
11.7.2. SLAM: La localización simultánea y el problema de asignación . . . . .	489
11.7.3. Definición matemática de SLAM . . . . .	490
11.7.4. SLAM del Filtro de Kalman Extendido (EKF) . . . . .	492
11.7.5. SLAM Visual con una sola cámara . . . . .	493
11.7.6. Discusión sobre EKF SLAM . . . . .	496
11.7.7. SLAM basada en Gráficos . . . . .	496
11.7.8. Filtro de partículas SLAM . . . . .	498
<b>12. PLANIFICACIÓN Y NAVEGACIÓN EN LOS ROBOTS MÓVILES</b>	<b>500</b>
12.1. GENERALIDADES . . . . .	500
12.2. COMPETENCIAS PARA LA NAVEGACIÓN: PLANIFICACIÓN Y REACCIÓN . . . . .	501
12.3. PLANIFICACIÓN DE RUTA . . . . .	502

12.3.1.	Búsqueda gráfica . . . . .	503
12.3.2.	Planificación de trayectorias de campo potencial . . . . .	512
12.4.	<b>EVASIÓN DE OBSTÁCULOS</b> . . . . .	515
12.4.1.	Algoritmo de Bug . . . . .	516
12.4.2.	Histograma de campo vectorial . . . . .	516
12.4.3.	Técnica de banda de burbuja . . . . .	519
12.4.4.	Técnicas de velocidad de curvatura . . . . .	520
12.4.5.	Enfoques dinámicos de ventana . . . . .	521
12.4.6.	El enfoque de Schlegel para evasión de obstáculos . . . . .	523
12.4.7.	Diagrama de proximidad . . . . .	523
12.4.8.	Método de gradiente . . . . .	524
12.4.9.	Adición de restricciones dinámicas . . . . .	524
12.5.	<b>ARQUITECTURAS DE NAVEGACIÓN</b> . . . . .	525
12.5.1.	Modularidad para la reutilización e intercambio de código . . . . .	525
12.5.2.	Localización de control . . . . .	525
12.5.3.	Técnicas para la descomposición . . . . .	529
12.5.4.	Estudios de casos: arquitecturas escalonadas de robot . . . . .	533
<b>13.</b>	<b>FUTURO DE LA ROBÓTICA MÓVIL AUTÓNOMA</b>	<b>539</b>
13.1.	<b>TENDENCIAS ACTUALES Y FUTURAS</b> . . . . .	539
13.1.1.	Robótica situada . . . . .	539
13.1.2.	Robótica basada en el comportamiento de conducta . . . . .	541
13.1.3.	Robótica cognitiva . . . . .	542
13.1.4.	Robótica Evolutiva . . . . .	543
13.1.5.	Robótica Biomimética, Biorrobótica o Robótica Inspirada Biológicamente . . . . .	548
13.2.	<b>EVOLUCIÓN ARTIFICIAL</b> . . . . .	548
13.2.1.	Operadores Genéticos . . . . .	548
13.2.2.	Evaluación de Calidad . . . . .	552
13.2.3.	Hipersuperficie o Paisaje de Calidad . . . . .	552
13.2.4.	Técnicas de Evolución Artificial . . . . .	553
13.2.5.	Estructuras de Control para Robótica Evolutiva . . . . .	556
13.3.	<b>¿QUÉ SE PUEDE EVOLUCIONAR EN ROBÓTICA?</b> . . . . .	558
13.3.1.	Evolución de los controladores de comportamientos. Tipos de estructuras de control . . . . .	559
13.3.2.	Evolución del hardware o morfología del robot . . . . .	563

13.3.3. Evolución del modelo de entorno dentro del mecanismo cognitivo . . . . .	565
13.4. ¿CUÁL PUEDE SER EL CAMINO A SEGUIR EN LA ROBOTICA MOVIL AUTONOMA? . . . . .	569
13.4.1. Estructuras de control más complejas . . . . .	570
13.4.2. Mejora del proceso de evolución . . . . .	571
13.4.3. Independencia de la plataforma del robot, comunicación y cultura . . . . .	571
13.4.4. Percepción activa y atención . . . . .	574
13.4.5. Hacia comportamientos más deliberativos. Tratamiento de información temporal . . . . .	575
13.4.6. Consideraciones finales . . . . .	575
<b>CONCLUSIONES</b>	<b>578</b>
<b>RECOMENDACIONES</b>	<b>582</b>
<b>BIBLIOGRAFÍA</b>	<b>583</b>
<b>ANEXOS</b>	<b>589</b>

---

# ÍNDICE DE FIGURAS

1.1. Metodología seminario de investigación . . . . .	63
2.1. Esquema general de la Robótica Móvil Autónoma . . . . .	65
3.1. Ejemplo de un acta . . . . .	73
3.2. Entorno gráfico de la simulación cinemática en MATLAB . . . . .	75
3.3. Código en MATLAB del taller de simulación cinemática . . . . .	75
3.4. Robot LEVI . . . . .	76
4.1. Ejemplo de diapositiva en el inicio de una sesión . . . . .	78
4.2. Ejemplo de diapositiva en las sesiones . . . . .	78
4.3. Textmaker: Editor de texto en L <sup>A</sup> T <sub>E</sub> X. . . . .	79
5.1. Reloj hidráulico de Ctesibius, y pilas para beber agua de Herón de Alejandría . . . . .	82
5.2. El gallo de Estrasburgo . . . . .	82
5.3. León mecánico de Da Vinci . . . . .	83
5.4. Pato de Jacques Vaucanson . . . . .	83
5.5. La pianista y el escritor de Pierre Jaquet Droz . . . . .	84
5.6. Karakuri . . . . .	85
5.7. Jugador de ajedrez de Leonardo Torres Quevedo . . . . .	85
5.8. Telekino . . . . .	86
5.9. I robot, Isaac Asimov . . . . .	87
5.10. R. C Goertz del Argonne National Laboratory . . . . .	88
5.11. Sonda Surveyor-3 . . . . .	88

5.12. Robot Shakey . . . . .	89
5.13. Brazo de Stanford . . . . .	90
5.14. Robot PUMA . . . . .	90
5.15. Robot móvil Lunakhod y robot que imita a las serpientes . . . . .	91
5.16. Robot IrB6 y brazo robótico a bordo de la sonda Viking-1 . . . . .	91
5.17. Robot Scara . . . . .	92
5.18. Wasubot . . . . .	92
5.19. a) Robot móvil Sojourner. b) Robot mascota AIBO de Sony . . . . .	93
5.20. (Izquierda) robot P2. (Derecha) robot P3 . . . . .	94
5.21. (Izquierda) robot ASIMO. (Derecha) aspiradora autónoma ROOMBA . . . . .	94
5.22. Guardrobo D1 . . . . .	95
5.23. Robot NUVO . . . . .	95
5.24. Robot violinista . . . . .	96
5.25. Robot Vinegar . . . . .	96
5.26. Robot Kobian . . . . .	97
5.27. HRP-4C . . . . .	97
5.28. Robot Manoi . . . . .	98
5.29. Diseñador Hiroshi Ishiguro con su clon robótico Geminoid . . . . .	98
5.30. GEMINOID DK . . . . .	99
5.31. Robots de la empresa Hanson Robotics . . . . .	99
5.32. Robot vigilante . . . . .	100
5.33. Mano artificial . . . . .	100
5.34. Robot UN-PIER . . . . .	101
5.35. Robot AGV . . . . .	102
5.36. UN-ROVER . . . . .	102
5.37. UN-BOT . . . . .	103
5.38. UNROCA-III . . . . .	103
5.39. Robot SCARA 4DOF . . . . .	104
5.40. Robot Arcadio . . . . .	105
5.41. Versión final del robot . . . . .	106
5.42. Robot cartesiano . . . . .	106
5.43. Planta dinámica de pruebas . . . . .	107
5.44. Estructura mecánica HeliT60 . . . . .	108
5.45. Prototipo de robot SCARA . . . . .	108
5.46. Robot AL5B Lynxmotion . . . . .	109
5.47. Fotografía del modelo final VEAU . . . . .	110
5.48. Robot de primera generación . . . . .	111

5.49. Robot de segunda generación . . . . .	112
5.50. Robots de tercera generación . . . . .	112
5.51. Robots de cuarta generación . . . . .	113
5.52. Robots de quinta generación . . . . .	113
5.53. Ejemplos de robots poliarticulados . . . . .	114
5.54. Ejemplos de robots móviles . . . . .	114
5.55. Ejemplos de robot androide . . . . .	115
5.56. Ejemplos de robots zoomórficos . . . . .	116
5.57. Ejemplos de robots híbridos . . . . .	116
5.58. Clasificación de los robots según su grado de inteligencia . . . . .	117
5.59. Robots de exploración . . . . .	118
5.60. Locomoción en robots móviles de exploración terrestre . . . . .	119
5.61. Categorías de los robots utilizados en la enseñanza . . . . .	120
5.62. Mapa conceptual de la robótica . . . . .	121
5.63. Mapa conceptual de los robots móviles autónomos . . . . .	122
6.1. Robot Spirit (NASA) . . . . .	128
6.2. Submarino SAR, de la compañía Ifremer . . . . .	129
6.3. Robot móvil para escardar . . . . .	130
6.4. Silla de ruedas avanzada: IBOT . . . . .	131
6.5. Comparativa de modelos de locomoción . . . . .	132
6.6. Autonomous Underwater Vehicles (AUVs) . . . . .	133
6.7. Tractor autónomo para agricultura . . . . .	134
6.8. Robot SAAPIN, guiado por un sistema de visión. . . . .	134
6.9. Robocrop, tractor robotizado de la empresa Tillett and Hague Techno- logy Ltd . . . . .	135
6.10. Robots MOSRO (interior) y OFRO (exterior) . . . . .	136
6.11. Robot mSecurit de MoviRobotics. . . . .	136
6.12. Robot ofensivo Riotbot de TECHNOROBOT. . . . .	137
6.13. Robot AUNAV. . . . .	137
6.14. Robot militar TALON de FORTER MILLER en configuraciones Sword) . . . . .	138
6.15. Robot PACKBOT . . . . .	138
6.16. BigDog de la Boston Dynamics. . . . .	139
6.17. Robot Tirant III . . . . .	139
6.18. Robots autónomos del proyecto CROMAT . . . . .	140
6.19. Silla de ruedas autónoma desarrollada por Simuzhu Corporation. . . . .	140
6.20. Un robot escalador de muro de una sola ventosa. . . . .	141

6.21. Tri-leg Waalbot. (a) Modelo CAD (b) Fibras con tallos para lograr una adherencia en seco . . . . .	142
6.22. Stickbot . . . . .	143
6.23. Un innovador robot escalador de muro basado en el efecto Bernoulli. . .	143
6.24. Robot Lunakhod. . . . .	145
6.25. Robot Lunakhod-2 . . . . .	146
6.26. Moon Buggy . . . . .	146
6.27. Robots Spirit Opportunity, de la NASA . . . . .	147
6.28. Robot Opportunity saliendo de una duna. . . . .	147
6.29. Robot submarino con turbinas. . . . .	149
6.30. Robot acuático con aletas. . . . .	149
6.31. Robot marino con movimiento ondulante. . . . .	150
6.32. Robot capaz de caminar sobre el agua. . . . .	150
6.33. Submarino RAS. . . . .	151
6.34. Robot Nautilo. . . . .	151
6.35. Robot Xanthos, AUV. . . . .	151
6.36. Robot AN/BLQ-11. . . . .	152
6.37. Robot Penguin B3. . . . .	152
6.38. Robot AQUA. . . . .	153
6.39. Robot Madeleine. . . . .	153
6.40. Robot Salamandra. . . . .	154
6.41. Robot andador sobre el agua. . . . .	154
6.42. Robot de tipo helicóptero. . . . .	155
6.43. Insecto robótico capaz de volar. . . . .	155
6.44. Robot del proyecto MFI. . . . .	156
6.45. DragonFly de WowWee. . . . .	156
6.46. Robots voladores. . . . .	157
6.47. UAV PREDATOR. . . . .	157
6.48. UAV AirRobot . . . . .	158
6.49. UAV “Talarion” de EADS-CASA. . . . .	158
6.50. UAV X-47B. . . . .	158
7.1. Mecanismos de locomoción utilizados en los sistemas biológicos . . . . .	166
7.2. Un sistema de caminado bípedo puede ser aproximado por un polígono de rodadura, con pasos iguales de longitud $d$ . . . . .	167
7.3. La potencia específica en comparación con la velocidad alcanzada de varios mecanismos de locomoción . . . . .	168

7.4. Ejemplos de robots con diferentes efectores para la locomoción terrestre	169
7.5. Disposición de las piernas de varios animales	170
7.6. Dos ejemplos de piernas con tres grados de libertad	171
7.7. Dos modos de marcha con cuatro patas.	171
7.8. Caminar estático con seis patas. Un trípede formado por tres patas siempre existe	173
7.9. El Raibert Hopper	174
7.10. El saltamontes de pierna de arco	174
7.11. El Sony SDR-4X II	175
7.12. Robot humanoide P2 de Honda, Japan	175
7.13. El robot humanoide WABIAN-RIII en la Universidad Waseda en Japón	176
7.14. El Spring Flamingo desarrollado en el MIT	177
7.15. AIBO, el perro artificial de Sony, en Japón	177
7.16. Titan VIII, un robot cuadrúpedo desarrollado en el Tokyo Institute of Technology	178
7.17. Lauron II, una plataforma hexápoda desarrollada en la Universidad de Karlsruhe, Alemania	178
7.18. Genghis Khan, uno de los robots caminantes más famosos del MIT, utiliza servomotores comunes como sus actuadores	179
7.19. Los cuatro tipos básicos de ruedas.	180
7.20. Navlab I, el primer vehículo de carretera autónomo que dirige y controla el acelerador usando sensores de visión y radar. Desarrollado en la CMU	181
7.21. Cye, un robot doméstico disponible comercialmente que puede aspirar y hacer entregas en el hogar y es construido por Aethon Inc	184
7.22. Configuración Síncrona	185
7.23. El Tribolo diseñado en el EPFL (Instituto Federal Suizo de Tecnología de Lausana, Suiza)	186
7.24. El robot Carnegie Mellon Uranus, un robot omnidireccional con cuatro ruedas suecas motorizadas de 45°	186
7.25. Los Nomad XR4000 de Nomadic Technologies tenían un arreglo de cuatro ruedas locas para el movimiento holonómico.	187
7.26. El microrover Nanokhod, desarrollado por von Hoerner y Sulger GmbH y el Instituto Max Planck, de Mainz, para la Agencia Espacial Europea (ESA)	188
7.27. Shrimp, un robot todo terreno con excelentes habilidades pasivas de escalada (EPFL)	188

7.28. El Rover personal, demostrando escalada de obstáculos usando un cambio continuo de COM . . . . .	189
7.29. RAMR 1 caminando (trepando) en una superficie . . . . .	190
7.30. Robots trepadores NINJA I y NINJA II . . . . .	191
7.31. Robot REST I soldando un casco de barco . . . . .	192
7.32. Prototipos de ALICIA 1 y 2 . . . . .	193
7.33. Vehículo de tres ruedas sincronizadas . . . . .	193
7.34. WallWalker trabajando en la ventana . . . . .	194
7.35. Estructura principal de Cleanbot II . . . . .	194
7.36. Movimiento del TRIPILLAR desde el piso hasta el techo . . . . .	195
7.37. Robot trepador GECKO III (comercializado ampliamente) . . . . .	196
7.38. Sky Cleaner 1, 2 y 3 y el mecanismo de traslación . . . . .	196
7.39. Grit blasting robot . . . . .	197
7.40. Robot escalador soportado por cables . . . . .	197
7.41. Robots trepadores por cables . . . . .	198
7.42. El robot de limpieza de fachada SYRIUSc . . . . .	199
7.43. ALICIA 3 pasando a través de los obstáculos . . . . .	199
7.44. Robot trepador ROMA II . . . . .	200
7.45. DEXTER escalando una ventana . . . . .	201
7.46. Robot limpiador TITO . . . . .	201
7.47. Robot escalador de pared para la inspección de tanques de aceite . . . . .	202
7.48. Prototipo de robot trepador móvil . . . . .	203
7.49. Robot trepador ROMA I . . . . .	203
7.50. Robot MATS y su estación de acoplamiento (DS) . . . . .	204
7.51. Stickybot, un robot trepador experimental . . . . .	205
7.52. Robot escalador inspirado en un gecko . . . . .	205
7.53. Spinybot y espinas miniatura . . . . .	206
7.54. Pruebas de un robot escalador de pared . . . . .	207
7.55. Mecanismo de movimiento de RRX en dirección longitudinal . . . . .	207
7.56. (a) Salamandra Robótica, un robot que se arrastra, nada, y camina (b) Un robot inspirado en un gusano submarino diseñado para rastrear a través de los intestinos . . . . .	209
7.57. (a) El robot langosta de la Northeastern University (b) El robot de cricket de la Case Western Reserve University (c) El robot zancudo de agua. (d) BigDog de Boston Dynamics (e) RunBot de la Universidad de Stirling . . . . .	210
7.58. Robot Sprawlette, a) Una simple antena montada en el robot Sprawlette, b) Un dispositivo táctico más avanzado . . . . .	211

7.59. El robot saltador GRILLO . . . . .	213
7.60. La cola del RoboTuna del MIT . . . . .	213
7.61. El robot Madeleine . . . . .	214
7.62. a) MFI, b) Mentor . . . . .	214
7.63. Evolución de los robots serpiente de la familia ACM (Active Cord Mechanism). Hirose-Fukushima Robotics Lab . . . . .	215
7.64. La serpiente robótica del CMU . . . . .	216
7.65. Los prototipos amphibot I y II del grupo de bioinspirada del EPFL . . . . .	217
7.66. Wombot (izquierda), diseñado por Conrad. S5, Miller (derecha) . . . . .	218
7.67. Robots auto-propulsados (serpentine robots) en el Hirose-Fukushima Robotics Lab . . . . .	219
7.68. Diferentes prototipos de robots ápodos con propulsión: GMD-snake, Makro, Swarm-bot, OmniThread OT-8 y OT-4 . . . . .	220
7.69. El robot JL-I . . . . .	220
7.70. Los módulos de Polypod y Polybot . . . . .	222
7.71. Polybot. Diferentes configuraciones de Polybot G1 . . . . .	222
7.72. Diferentes configuraciones de Polybot . . . . .	223
7.73. Módulos y distintas configuraciones de CkBot . . . . .	223
7.74. Módulos y diferentes configuraiones del robot M-TRAN . . . . .	224
7.75. Módulo Conro y diferentes configuraciones . . . . .	225
7.76. El módulo de Superbot y diferentes configuraciones . . . . .	225
7.77. Los robots modulares Yamor y Molecube . . . . .	226
8.1. Diferentes configuraciones de ruedas . . . . .	229
8.2. Configuración diferencial. . . . .	230
8.3. Configuración síncrona. . . . .	231
8.4. Configuración tipo triciclo. . . . .	231
8.5. Configuración Ackerman. . . . .	232
8.6. Configuración omnidireccional. . . . .	232
8.7. Configuración omnidireccional. . . . .	233
8.8. El marco de referencia global y el marco de referencia local del robot. . . . .	236
8.9. El robot móvil alineado con un eje global. . . . .	237
8.10. Un robot de conducción diferencial en su marco de referencia global. . . . .	238
8.11. Una rueda fija estándar y sus parámetros. . . . .	241
8.12. Una rueda orientable estándar y sus parámetros. . . . .	244
8.13. Una rueda loca y sus parámetros. . . . .	245
8.14. Ruedas castor de una Silla de oficina. . . . .	245

8.15. Rueda Sueca y sus parámetros. . . . .	247
8.16. Una rueda esférica y sus parámetros. . . . .	248
8.17. Un robot Omnidireccional de tres ruedas desarrollado por la Universidad Carnegie Mellon. . . . .	252
8.18. El marco local de referencia y los parámetros detallados para la rueda 1.	253
8.19. (a) Coche de cuatro ruedas con dirección Ackerman. (b) Bicicleta. . . .	256
8.20. (a) Robot diferencial con dos ruedas motorizadas individualmente y una rueda loca, e.g., el robot Pigmalión en EPFL. (b) triciclo con dos ruedas fijas estándar y una rueda estándar orientable, e.g., Piaggio minitransporter. . . . .	256
8.21. Centro Instantáneo de rotación. . . . .	260
8.22. Centro Instantáneo de rotación. . . . .	260
8.23. Los cinco tipos básicos de configuraciones de tres ruedas. . . . .	261
8.24. Configuración de conducción sincronizada . . . . .	262
8.25. Ejemplo de la trayectoria de un robot omnidireccional. . . . .	266
8.26. Ejemplo de la trayectoria del robot similar a la figura 8.25, con dos ruedas motrices. . . . .	267
8.27. Caso típico para el control de retroalimentación de un robot móvil. . .	268
8.28. Robots móviles Monociclo y Tipo carro. . . . .	269
8.29. Analogía del carro / monociclo con remolque. . . . .	269
8.30. Las variables de configuración. . . . .	273
8.31. Control de lazo abierto de un robot móvil basado en líneas rectas y segmentos de trayectoria circulares. . . . .	274
8.32. Cinemática de un robot y sus marcos de intereses. . . . .	275
8.33. Caminos resultantes cuando el robot se encuentra inicialmente en el círculo unitario en el plano $x, y$ . . . . .	279
8.34. Un robot rodante de 2-DOF: (a) Su disposición general, y (b) Detalle de las ruedas accionadas. . . . .	281
8.35. Presentación de los vectores unitarios fijados a la plataforma y al soporte.	284
8.36. (a) La Rueda Mecanum, (b) Su vista lateral. . . . .	287
8.37. El rodillo activo de la $i$ -ésima rueda. . . . .	288
8.38. El diseño de la $i$ -ésima rueda con respecto a la plataforma robot. . . .	289
9.1. Robot móvil Hilare . . . . .	292
9.2. El diagrama de cuerpo libre detallado para un robot móvil Hilare . . .	295
9.3. El diagrama de cuerpo libre y cinético para un robot móvil Hilare como un único sistema . . . . .	296

9.4. El diagrama de cuerpo libre y cinético de la rueda derecha . . . . .	296
9.5. Un robot rodante de 2 DOF: (a) Su disposición general, y (b) Detalle de las ruedas accionadas. . . . .	304
9.6. Robot rodante con ODWs en arreglo $\Delta$ . . . . .	316
9.7. Una vista del robot de tres ruedas con ruedas Mekanum en un arreglo $\Delta$ . . . . .	317
10.1. Ejemplos de robots con sistemas de sensores múltiples: (a) HelpMate del Transition Research Corporation, (b) B21 del Real World Interface, (c) BIBA Robot, BlueBotics SA . . . . .	326
10.2. Codificador de cuadratura óptico de la rueda . . . . .	331
10.3. Brújula digital: Se muestran los sensores tales como el sensor de efecto Hall analógico/digital . . . . .	332
10.4. Giroscopio mecánico de dos ejes . . . . .	333
10.5. Acelerómetros: (a) Principio de funcionamiento del acelerómetro mecánico, (b) Un acelerómetro MEMS producido por Sandia National Laboratories, (c) Un acelerómetro MEMS comercial . . . . .	335
10.6. (a) Diagrama de bloques IMU, (b) Un IMU comercial producido por Xsens	336
10.7. Cálculo de la posición y rumbo basado en GPS . . . . .	337
10.8. Cálculo de la posición y rumbo basado en GPS . . . . .	338
10.9. Distribución de intensidad típica de un sensor ultrasónico . . . . .	340
10.10 Lecturas típicas de un sistema ultrasónico: (a) 360° de exploración; (b) Resultados de diferentes primitivas geométricas. . . . .	341
10.11 Esquema de telemetría láser mediante medición de cambio de fase . . . . .	341
10.12 Rango de estimación mediante la medición del desplazamiento de fase entre señales transmitidas y recibidas . . . . .	342
10.13 (a) Esquema del sensor de alcance láser con espejo giratorio, (b) Sensor de nivel de barrido de EPS Technologies Inc., (c) Sensor Industrial láser de 180 grados de Sick Inc., Alemania . . . . .	343
10.14 (a) Imagen rango típico de un sensor láser de rangos 2D con un espejo giratorio. (b) Stanley, el coche autónomo de Stanford ganó el 2005 Darpa Grand Challenge, (c) El Smarter, el coche autónomo desarrollado en la ASL (ETH Zurich). (d) Una vista de cerca a los Sicks utilizados en el Smarter. (e) Una nube de puntos láser 3D construida a partir de los Sicks giratorios . . . . .	344

10.15(a) La unidad Velodyne HDL-64E cuenta con 64 rayos láser y gira hasta 15 Hz para recopilar los datos (imagen cortesía de Velodyne). (d) El principio de trabajo del Ibeo Alasca XT (c), el cual utiliza un haz de cuatro capas de láser. El Alasca y Velodyne fueron utilizados por el equipo CMU Tartan Racing (b) . . . . .	345
10.16(a) La ZCAM producida por el desarrollador israelí sistemas 3DV, (b) La Swiss Ranger SR3000 producida por la empresa suiza MESA, (c) Imagen de alcance de una silla capturada con una cámara de tiempo de vuelo .	346
10.17 Principio de la triangulación de laser 1D . . . . .	347
10.18 Sensor de triangulación óptica Sharp . . . . .	347
10.19(a) Principio de triangulación activa de dos dimensiones. (b) Otras estructuras ligeras posibles. (c) Esquemática 1D del principio. . . . .	348
10.20 Efecto Doppler entre dos objetos móviles (a) o un objeto en movimiento y fijas a (b) . . . . .	350
10.21 CCD chips y cámaras CCD comercialmente disponibles. . . . .	351
10.22 Una cámara CMOS comercialmente disponible y de bajo costo. . . . .	353
10.23 Filtro Bayer de color de matriz . . . . .	353
10.24 Representación de la óptica de la cámara y su impacto en la imagen . .	354
10.25 Dos imágenes de la misma escena tomada con una cámara en dos posiciones de enfoque diferentes . . . . .	355
10.26(a) Cuando $d \gg f$ y $d \gg l$ la cámara puede ser modelada como una cámara estenopeica. (b) La cámara oscura en un dibujo del matemático Reinerus Gemma-Frisius. . . . .	355
10.27(a) Modelo de cámara estenopeica utiliza para representar las cámaras de perspectiva estándar. (b) El modelo de agujero de alfiler . . . . .	356
10.28 Cambio de coordenadas entre la cámara y el marco de referencia del mundo	359
10.29 Ejemplo de distorsión radial de la lente: (a) Ninguna distorsión, (b) Distorsión de barril, (c) acerico . . . . .	361
10.30 Imágenes de la Caja de herramientas de calibración de la cámara para Matlab desarrollada por JY Bouguet . . . . .	361
10.31 Cámara dioptrica (por ejemplo, ojo de pez), (b) Cámara catadióptrica, (c) una ejemplo de cámara polydióptrica producida por Immersive Media	363
10.32(a) La lente de ojo de pez de Omnitech Robótics proporciona un campo de visión de 190 grados. (b) Una cámara miniatura catadióptrica construida en el ETH Zurich (c) La cámara Mufly construida por CSEM. Además, se utiliza un CCD polar (d) donde los píxeles están dispuestos radialmente	363

10.33(a) Una cámara catadióptrica omnidireccional usando un espejo hiperbólico. (b) Lente de ojo de pez Nikon FC-E8. . . . .	364
10.34Ejemplos de cámaras centrales y no centrales . . . . .	364
10.35Cámaras catadióptricas centrales pueden ser construidas usando espejos hiperbólicos y parabólicos. . . . .	365
10.36Proyecciones de cámaras centrales . . . . .	366
10.37Modelo de proyección unificado para cámaras centrales catadióptricas de Geyer y Daniilidis . . . . .	366
10.38(Izquierda) El STH-MDCS3 de Videre Design utiliza sensores CMOS. (Derecha) El Bumblebee2 de Point Grey utiliza sensores CCD. . . . .	369
10.39Geometría de cámara idealizada para visión estéreo . . . . .	370
10.40Visión estéreo: caso general . . . . .	371
10.41Un par estéreo. Puntos correspondientes son proyecciones del mismo punto de escena. . . . .	373
10.42Geometría epipolar: (a) Segmento de línea epipolar correspondiente a un rayo, (b) Conjunto correspondiente de líneas epipolar y su plano epipolar	375
10.43Rectificación de un par estéreo: (a) Imagen original, (b) Compensación de la distorsión de la lente, (c) Compensación de rotación y traslación, (d) Después de la rectificación epipolar, las líneas epipolares aparecen alineadas y horizontales . . . . .	375
10.44Un ejemplo de mapa de disparidad calculado a partir de las dos imágenes superiores . . . . .	377
10.45Una cámara estéreo de Videre Design en el robot camarón desarrollado en la ASL . . . . .	377
10.46Un ejemplo de estructura de movimiento: los puntos de imagen sobresalientes se extraen y se hacen coincidir en varios fotogramas. . . . .	378
10.47(Parte superior izquierda) Un resultado ejemplo de odometría visual con su mapa relacionado (inferior) obtenido utilizando una sola cámara omnidireccional montada en el techo del vehículo (derecha). . . . .	383
10.48Movimiento de la esfera o la fuente de luz . . . . .	383
10.49Marcadores de colores en la parte superior de los robots de fútbol de STeam Engine . . . . .	386
10.50Ejemplos de extracción de plano adaptativa del suelo . . . . .	387
10.51Ilustración del concepto de filtrado espacial, (c) imagen de entrada, (d) la imagen de salida después de la aplicación de filtro de promedio. . . . .	389
10.52(a) Foto de una lámpara de techo. (b) Calculo de los bordes a partir de (a). . . . .	392

10.53Ejemplo de función Paso de la forma de la segunda derivada y el impacto del ruido. . . . .	393
10.54(a) Una función gaussiana. (b) La primera derivada de una función gaussiana. . . . .	393
10.55(a) Función bidimensional gaussiana. (b) Filtro Vertical. (c) Filtro horizontal. . . . .	394
10.56(a) Ejemplo de una imagen de borde; (b) Supresión Nonmaxima de (a).	395
10.57Ejemplo de extracción de características basada en visión con las diferentes etapas de procesamiento: (a) Datos brutos de imagen; (b) Imagen filtrada mediante un filtro Sobel; (c) Uso de umbral, selección de píxeles de borde (d) Supresión nonmaxima. . . . .	396
10.58(a) El número de pixeles con una magnitud de gradiente específica en la imagen de la figura 10.42(b). (b) Igual que (a), pero con escala logarítmica.	397
10.59La tubería perceptual: a partir de lecturas de los sensores a modelos de conocimiento. . . . .	399
10.60Representación del entorno y modelado: (a) basado en funciones (continuo métrico), (b) red ocupación (métrica discreta). Cortesía de Sjur Vestli. . . . .	400
10.61(a) Panorama construido a partir de múltiples imágenes superpuestas utilizando software Autostitch, (b) Primer paso, (c) Segunda etapa, (d) Dos imágenes de ejemplo donde las características no fueron detectadas	401
10.62(a) Región “plana” (b) “Borde” (c) “Esquina” . . . . .	403
10.63(a) Esta elipse se construye a partir de la matriz de segundo momento y visualiza las direcciones de cambio de intensidad más rápidas y más bajas, (b) La clasificación de esquina y los bordes de acuerdo con Harris y Stephens. . . . .	403
10.64Extracción de las esquinas de Harris, (a) Imagen original, (b) la función Cornerness, (c) Las esquinas de Harris se identifican como máximos locales de la función cornerness . . . . .	405
10.65Modelos de cambios de imagen. . . . .	406
10.66(a) Detector de Harris es invariante a la rotación de la imagen (b) A la inversa, no es invariante a escala de la imagen . . . . .	408
10.67Tasa de repetición de detector de Harris. Comparación con Harris-Laplace.	409
10.68Análisis de imagen a diferentes escalas . . . . .	409
10.69Intensidad media como una función del tamaño de la región. (a) Imagen original (b) Imagen recalibrada (cambio de tamaño). . . . .	410
10.70(a) – (b) son funciones malas de escala invariante. (c) es una buena. . .	411

10.71	Comparación entre el Laplaciano de Gauss y diferencia de Gauss. . . .	411
10.72	Respuesta del operador de registro en dos puntos correspondientes a partir de dos imágenes tomadas a diferentes escalas. . . . .	412
10.73	Cálculo de la elipse afín invariante en dos imágenes relacionadas por una transformación afín. . . . .	413
10.74	(a) Detector SUSAN compara pixeles dentro de una región circular, mientras que FAST (b) los compara sólo en un círculo. . . . .	414
10.75	La extracción de características SIFT y MSER de la imagen ejemplo utilizada para el detector de Harris en la figura 10.64 . . . . .	415
10.76	(a) Imágenes gaussianas borrosas en diferentes escalas, (b) Imágenes de Diferencia de Gauss, (c) Selección de punto clave (keypoint) . . . . .	416
10.77	(a) Asignación de Orientación, (b) Algunas características SIFT con orientación y escala detectada. . . . .	417
10.78	Fracción de puntos clave SIFT correctamente emparejados como una función del ángulo de punto de vista. . . . .	418
10.79	Partición del espacio descriptor de características: Cada celda representa una palabra visual . . . . .	421
10.80	Reconocimiento de ubicación basado en palabras visuales mediante un sistema de archivo invertido . . . . .	422
10.81	Seis histogramas de ID de la imagen de arriba. A 5 x 5 filtro de suavizado se convolucionan con cada banda antes de histografía. . . . .	424
10.82	Dos imágenes panorámicas y sus secuencias de huellas digitales asociadas.	426
10.83	Tres secuencias de cadena reales . . . . .	427
10.84	Estimación de una línea en el sentido de mínimos cuadrados. Los parámetros del modelo $r$ (longitud de la perpendicular) y $\alpha$ (su ángulo con el eje de abscisas) describen únicamente una línea. . . . .	428
10.85	Línea extraída a partir de mediciones de distancias por láser (+) . . . .	431
10.86	Clustering: encontrar segmentos vecinos de una línea común. . . . .	432
10.87	División y fusión implementada de la manera Iterativa ajuste-punto-extremo . . . . .	433
10.88	División y fusión aplicado a una tomografía láser 2D (cortesía de B Jensen).	434
10.89	Una ventana deslizante con Tamaño $N_f = 3$ se utiliza en este ejemplo .	435
10.90	Principio de funcionamiento de RANSAC . . . . .	436
10.91	Transformada de Hough: a) Espacio de imagen. b) Parámetro de espacio de Hough. . . . .	438
10.92	Detección de puerta usando la transformada de Hough. . . . .	439
10.93	Histograma de ángulo. . . . .	441

10.94	Varias características geométricas en un pasillo único, incluyendo puertas y discontinuidades en la anchura del pasillo. . . . .	442
10.95	Proceso de extracción de características planas . . . . .	443
11.1.	¿Dónde estoy?, determinación de la posición en el medio ambiente. . . . .	444
11.2.	Esquema general para la localización del robot móvil. . . . .	445
11.3.	El movimiento de un robot diferencial-drive. . . . .	448
11.4.	Crecimiento de la incertidumbre de posición para movimiento en línea recta. . . . .	451
11.5.	Crecimiento de la incertidumbre de posición para movimiento circular ( $r = const$ ). . . . .	451
11.6.	Una muestra del entorno. . . . .	452
11.7.	Una arquitectura para navegación basada en comportamiento. . . . .	453
11.8.	Una arquitectura para navegación basada en mapas. . . . .	453
11.9.	La representación de la creencia con respecto a la posición del robot (1D) en mapas continuos y discretos (teselados). . . . .	455
11.10	Tres ejemplos de hipótesis única de posición utilizando diferentes representaciones del mapa. . . . .	456
11.11	Ejemplo de seguimiento de múltiples hipótesis . . . . .	456
11.12	Una representación continua utilizando polígonos como obstáculos ambientales . . . . .	458
11.13	Ejemplo de una representación de línea valor continuo de EPFL. (a) Mapa real. (b) Representación con un conjunto de líneas infinitas. . . . .	458
11.14	Ejemplo de descomposición de celda exacta. . . . .	459
11.15	Descomposición fija del mismo espacio, el pasaje estrecho desaparece. . . . .	460
11.16	Ejemplo de descomposición adaptada (celda variable aproximada) de un entorno. . . . .	460
11.17	Una representación topológica de una zona interior de oficinas. . . . .	461
11.18	En la robótica probabilística, las creencias acerca de la configuración del robot se representan como funciones de densidad de probabilidad. . . . .	462
11.19	Algoritmo general para la localización de Markov. . . . .	469
11.20	Ilustración del algoritmo de localización de Markov. . . . .	471
11.21	Localización de Markov usando un mapa de red. . . . .	472
11.22	La matriz 3D de representación del estado creencia utilizada en la localización de Markov . . . . .	474
11.23	Dervish explorando su entorno. . . . .	475

11.24	Un entorno de oficina geométrica (izquierda) y su análogo topológico (derecha). . . . .	475
11.25	Un entorno realista topológico interior. . . . .	477
11.26	Típica aplicación del filtro de Kalman . . . . .	478
11.27	Aplicación del algoritmo de filtro de Kalman para localización de robots móviles. . . . .	480
11.28	Propagación de la densidad de probabilidad de un robot móvil. . . . .	481
11.29	La fusión de la densidad de probabilidad de dos estimaciones. . . . .	483
11.30	Esquema para la localización del robot móvil de filtro Kalman. . . . .	485
11.31	Un robot autónomo debería ser capaz de seguir cambios en los ambientes para la localización. . . . .	489
11.32	Ilustración del problema SLAM. . . . .	490
11.33	SLAM de EKF con escáner láser 3D. . . . .	493
11.34	Parches de características de imagen y regiones de búsqueda predicha a partir del marco anterior utilizando un modelo de velocidad constante movimiento. . . . .	496
11.35	Pasos principales de V-SLAM de Davison. . . . .	497
11.36	Ilustración evocativa de la construcción gráfica. . . . .	498
11.37	SLAM de EKF estándar . . . . .	498
12.1.	El espacio físico (a) y la configuración del espacio (b) para un brazo robótico planar con dos articulaciones . . . . .	502
12.2.	Gráfico de visibilidad. . . . .	504
12.3.	Diagrama de Voronoi . . . . .	505
12.4.	Ejemplo de descomposición de celda exacta. . . . .	505
12.5.	Estado de enrejado de 16 direcciones construido para un rover de exploración planetaria. . . . .	506
12.6.	Principio de funcionamiento de la búsqueda de primero amplitud . . . . .	508
12.7.	Un ejemplo de la transformada de la distancia y la trayectoria resultante como es generada por la función NF1. . . . .	509
12.8.	Principio de funcionamiento de la búsqueda primero en profundidad . . . . .	509
12.9.	Principio de funcionamiento del algoritmo $A^*$ . . . . .	510
12.10	Comparación del número de celdas expandidas para $D^*$ y en cualquier momento $D^*$ . . . . .	511
12.11	La evolución de un RRT. (a) 45 iteraciones. (b) 2345 iteraciones . . . . .	512
12.12	Típico campo potencial generado por el objetivo atrayente y dos obstáculos	513

12.13	Comparación entre un campo potencial clásico y un campo potencial extendido . . . . .	515
12.14	Algoritmo <i>Bug1</i> con H1, H2, puntos de alcance, y L1, L2, puntos de salida	517
12.15	Algoritmo <i>Bug2</i> con H1, H2, puntos de alcance, y L1, L2, puntos de salida	517
12.16	Histograma polar . . . . .	518
12.17	Ejemplo de direcciones bloqueadas e histogramas polares resultantes: (a) Robot y los obstáculos que lo bloquean, (b) Histograma polar, (c) histograma polar enmascarado . . . . .	519
12.18	La forma de las burbujas alrededor del vehículo . . . . .	519
12.19	Una banda de burbuja típica . . . . .	520
12.20	Curvaturas tangentes para un obstáculo . . . . .	521
12.21	El enfoque dinámico de ventana . . . . .	522
12.22	Distancias $l_i$ resultantes de la curvatura $i$ , donde el robot gira alrededor de $M$ . . . . .	523
12.23	Descomposición temporal genérica de una arquitectura de navegación .	529
12.24	Muestra de descomposición temporal de cuatro niveles de una navegación simple de un robot móvil. . . . .	531
12.25	Ejemplo de una descomposición serial pura . . . . .	532
12.26	Ejemplo de una descomposición paralela pura . . . . .	532
12.27	El ejemplo básico arquitectónico utilizado a lo largo de este estudio . .	534
12.28	Una arquitectura de navegación escalonada general basada en una descomposición temporal . . . . .	535
12.29	Una arquitectura de dos niveles para la planificación offline . . . . .	536
12.30	Una arquitectura de planificación episódica de tres niveles . . . . .	537
12.31	Una arquitectura integrada de planificación y ejecución en el que la planificación no es más que un paso de ejecución en tiempo real . . . . .	538
13.1.	Robótica tradicional . . . . .	541
13.2.	Robótica basada en el comportamiento . . . . .	541
13.3.	Sistemas de programación tradicionales . . . . .	543
13.4.	Escenario de trabajo de los algoritmos genéticos . . . . .	544
13.5.	Esquema del proceso de un algoritmo genético . . . . .	545
13.6.	Estructura de una neurona . . . . .	546
13.7.	Esquema RNA . . . . .	547
13.8.	Esquema de un nodo de la RNA . . . . .	547
13.9.	Representación de cromosoma y gen. . . . .	549
13.10	El valor 1 corresponde al alelo de esta posición del gen . . . . .	549

13.11	Representación de un genotipo binario y su posterior decodificación a fenotipo . . . . .	550
13.12	Cruce de un punto . . . . .	550
13.13	Cruce de dos puntos . . . . .	551
13.14	Cruce uniforme . . . . .	551
13.15	Representación hipotética de un paisaje de calidad . . . . .	553
13.16	Ejemplo de predicción de entradas con PE. . . . .	555
13.17	Robots Khepera utilizados por Nolfi y Floreano . . . . .	556
13.18	Efecto de la Reina Roja . . . . .	556
13.19	Arquitectura Multi-Agent ESP . . . . .	557
13.20	Esquema del funcionamiento una neurona biológica . . . . .	558
13.21	Dos de las funciones de activación de mayor utilización, (sigmoïdal binaria y sigmoïdal bipolar) . . . . .	558
13.22	Esquema de funcionamiento de una neurona artificial. Las entradas de esta neurona están dadas por X1, X2 y X3. . . . .	559
13.23	Sistema de clasificación utilizado por Dorigoy Colombetti . . . . .	560
13.24	Funciones de pertenencia a cada uno de los conjuntos borrosos en los valores de los sensores (izquierda) y de los motores (derecha) de ambas ruedas de un Khepera. . . . .	561
13.25	Proceso de inferencia cuando se activan diferentes reglas que actúan sobre una misma variable de control . . . . .	561
13.26	Organización en capas de los nodos de una RNA . . . . .	562
13.27	Ejemplo de Lund y Hallam de conexión directa de los valores de los seis sensores frontales y de los dos traseros de infrarrojos del robot Khepera con los nodos de entrada del controlador neuronal. . . . .	562
13.28	Dos criaturas virtuales de Sims luchando por la posesión de un bloque .	563
13.29	Ejemplos de criaturas virtuales de Sims evolucionadas para nadar en un entorno acuoso . . . . .	564
13.30	Khepera con “oídos” que simula un grillo hembra y RoBat, una simulación de la morfología de recepción de los ecos de sonido emitidos por un murciélago . . . . .	565
13.31	Ejemplo de estructura final en el trabajo de Lipson y Pollack (2000) . .	565
13.32	Módulos de la arquitectura cognitiva con aprendizaje automático del entorno . . . . .	566
13.33	Evolución del error cuadrático medio del modelo de mundo respecto al mundo real según interactúa el robot con el mundo . . . . .	567

13.34	Camino seguido por el robot en simulación (izquierda) y en el mundo real (derecha) cuando aún no ha aprendido a alcanzar un objeto (iteraciones de la 53 a la 57) . . . . .	567
13.35	Camino seguido por el robot en simulación y en el mundo real cuando ya ha aprendido a alcanzar un objeto (iteraciones a partir de la 420) . .	568
13.36	Esquema del método de modelización utilizado por Nordiny . . . . .	569
13.37	Red de “gas” de Husbands. Cada nodo emite un “gas” que modula propiedades de los nodos afectados . . . . .	570
13.38	Evolución de calidades al considerar un cambio de plataforma entre un robot Khepera y un robot Koala . . . . .	572
13.39	Niveles de sensores y actuadores virtuales . . . . .	572
13.40	Robot Hermes en el proceso de detectar la posición exacta de un objeto con un único sensor de IR . . . . .	573
13.41	Reutilización de un comportamiento de seguir paredes en una arquitectura jerárquica . . . . .	573
13.42	Ejemplo de necesidad de percepción activa provocada por la existencia de aliasing en la percepción con los infrarrojos del Khepera . . . . .	574
13.43	Trayectorias del robot Rug Warrior en una tarea de seguimiento de luz que utiliza un único fotosensor en simulación (izquierda) y en la realidad con diferentes niveles de luz ambiente (centro y derecha) . . . . .	575
13.44	Trayectorias del Khepera en 60 pruebas, en el giro de la unión T previamente indicado por el foco de luz . . . . .	576
13.45	Trayectorias del Khepera móvil cuando se aproxima al robot central, evitando los choques . . . . .	576

---

# ÍNDICE DE CUADROS

1.1. Formación Ing. Carlos Borrás Pinilla . . . . .	62
2.1. Temas y subtemas del seminario de investigación . . . . .	68
3.1. Asignación de roles para cada una de las sesiones . . . . .	72
3.2. Talleres de aplicación: MATLAB y/o SIMULINK . . . . .	74
7.1. Configuraciones de ruedas para vehiculos rodantes (2 y 3 ruedas) . . . . .	182
7.2. Configuraciones de ruedas para vehiculos rodantes (4 ruedas) . . . . .	183
7.3. Configuraciones de ruedas para vehiculos rodantes (6 ruedas) . . . . .	184
7.4. Lista de los robots escaladores de acuerdo a los campos industriales, con tareas de la aplicación . . . . .	208
10.1. Clasificación de los sensores usados en aplicaciones de robótica móvil. (A, Activo; P,Pasivo; P/A, pasivo/activa; PC, propioceptivos; EC, exte- roceptivos) . . . . .	327
10.2. Comparación de los detectores de características: propiedades y rendi- miento. . . . .	420
10.3. Valores de Medición . . . . .	430
10.4. Comparación de los algoritmos para la extracción de la línea de datos de láser 2D. . . . .	440
11.1. Tabla de certeza de Dervish. . . . .	476
12.1. Enfoques de evasión de obstáculos (Parte 1) . . . . .	526
12.2. Enfoques de evasión de obstáculos (Parte 2) . . . . .	527

12.3. Enfoques de evasión de obstáculos (Parte 3) . . . . .	528
---	-----

---

# ÍNDICE DE ANEXOS

ANEXO A. Otras técnicas de filtrado de imagen.....	589
ANEXO B. Otros ejemplos de sistemas de localización.....	596
ANEXO C. Pruebas y talleres.....	604

## GLOSARIO

- **ACTUADOR:** Un actuador es un dispositivo capaz de transformar energía hidráulica, neumática o eléctrica en la activación de un proceso con la finalidad de generar un efecto sobre un proceso automatizado. Este recibe la orden de un regulador o controlador y en función a ella genera la orden para activar un elemento final de control como, por ejemplo, una válvula.
- **ADAPTABILIDAD:** Capacidad de acomodarse o ajustarse una cosa a otra. La adaptabilidad asociada a un material hace referencia a aquellos materiales que se amoldan más fácilmente a otros elementos.
- **AERODINÁMICA:** Rama de la mecánica de fluidos que estudia las acciones que aparecen sobre los cuerpos sólidos cuando existe un movimiento relativo entre éstos y el fluido que los baña, siendo éste último un gas y no un líquido, caso éste que se estudia en hidrodinámica.
- **AERODINO:** Un aerodino es una aeronave capaz de generar sustentación. Los aerodinos se dividen en dos grandes grupos: las aeronaves de ala fija y las aeronaves de alas móviles.
- **ALGORITMO:** En matemáticas, ciencias de la computación y disciplinas relacionadas, un algoritmo es un conjunto preescrito de instrucciones o reglas bien definidas, ordenadas y finitas que permite realizar una actividad mediante pasos sucesivos que no generen dudas a quien deba realizar dicha actividad.
- **ANTROPOMORFISMO:** Es la atribución de características y cualidades humanas a animales, objetos o fenómenos naturales.
- **ÁPODO:** Nombre que suele darse a una persona o animal, en sustitución del propio, normalmente tomado de sus peculiaridades físicas o de alguna otra circunstancia. Falto de pies.
- **ARENISCA:** La arenisca o psamita es una roca sedimentaria de tipo detrítico, de color variable, que contiene clastos de tamaño arena.
- **ASIMETRÍA:** Se refiere a la propiedad de determinados cuerpos, funciones matemáticas y otros tipos de elementos en los que, al aplicarles una regla de transformación efectiva, se observan cambios respecto al elemento original. Surge una discordia cuando no somos capaces de reconocer qué parte es la original de la asimetría.

- **ASINTÓTICA:** En matemática, se le llama asíntota a una línea recta que se aproxima continuamente a otra función o curva; es decir que la distancia entre las dos tiende a cero, a medida que se extienden indefinidamente.
- **AUTÓMATA:** Máquina que imita la figura y los movimientos de un ser animado.
- **AUV:** Por sus siglas en inglés “Autonomous Underwater Vehicle”, en español llamado “Vehículo Submarino Autónomo” (VSA), es un robot que viaja bajo el agua sin necesitar de un piloto humano.
- **BIOMIMÉTICA:** Es el estudio de la estructura y función de los sistemas biológicos como modelos para el diseño y la ingeniería de los materiales y las máquinas.
- **BIÓNICA:** Es la aplicación de soluciones biológicas a la técnica de los sistemas de arquitectura, ingeniería y tecnología moderna.
- **BOMBA HIDRÁULICA:** Una bomba hidráulica es una máquina generadora que transforma la energía (generalmente energía mecánica) con la que es accionada en energía hidráulica del fluido incompresible que mueve.
- **BOGIE:** El boje o bogie es un conjunto de dos o tres pares de ruedas, según modelo, montadas sobre sendos ejes próximos, paralelos y solidarios entre sí, que se utilizan en ambos extremos de los vehículos de gran longitud destinados a circular sobre rieles.
- **CAVITACIÓN:** Es un efecto hidrodinámico que se produce cuando el agua o cualquier otro fluido en estado líquido pasa a gran velocidad por una arista afilada, produciendo una descompresión del fluido debido a la conservación de la constante de Bernoulli.
- **CELOSÍA:** En ingeniería estructural, una celosía es una estructura reticular de barras rectas interconectadas en nudos formando triángulos planos (en celosías planas) o pirámides tridimensionales (en celosías espaciales).
- **CENTRO DE GRAVEDAD:** Es el punto de aplicación de la resultante de todas las fuerzas de gravedad que actúan sobre las distintas porciones materiales de un cuerpo, de tal forma que el momento respecto a cualquier punto de esta resultante aplicada en el centro de gravedad es el mismo que el producido por los pesos de todas las masas materiales que constituyen dicho cuerpo.

- **CHASIS:** El chasis o chasis del automóvil consta de un bastidor que integra entre sí y sujeta tanto los componentes mecánicos, como el grupo motopropulsor y la suspensión de las ruedas, incluyendo la carrocería de un vehículo terrestre.
- **CILIOS:** Son unos orgánulos exclusivos de las células eucariotas, que se caracterizan por presentarse como apéndices con aspecto de pelo que contienen una estructura central altamente ordenada, constituida generalmente por más de 600 tipos de proteínas, envuelta por el citosol y la membrana plasmática.
- **CLUSTER:** (a veces castellanizado como clúster) es un término inglés encontrado en varios tecnicismos. La traducción literal al castellano es racimo, conjunto, grupo o cúmulo.
- **COGNICIÓN:** Está íntimamente relacionada con conceptos abstractos tales como mente, percepción, razonamiento, inteligencia, aprendizaje y muchos otros que describen numerosas capacidades de los seres superiores, aunque estas características también las compartirían algunas entidades no biológicas según lo propone la inteligencia artificial.
- **COJINETE:** Un cojinete en ingeniería es la pieza o conjunto de ellas sobre las que se soporta y gira el árbol transmisor de momento giratorio de una máquina.
- **COMPRESOR:** Un compresor es una máquina de fluido que está construida para aumentar la presión y desplazar cierto tipo de fluidos llamados compresibles, tal como lo son los gases y los vapores.
- **CONFIGURACIÓN ACKERMAN:** Es la configuración que estamos habituados a ver en los vehículos convencionales, y por lo tanto es una configuración muy probada y estable. Se basa en una estructura de cuatro ruedas colocadas en dos ejes, donde sólo las dos ruedas delanteras permiten un giro sobre el eje.
- **CONTROLABILIDAD:** Es una propiedad importante de un sistema de control, y juega un papel crucial en muchos problemas de control, como la estabilización de sistemas inestables, o el control óptimo.
- **CONTROL DE LAZO ABIERTO:** Es aquel sistema en que solo actúa el proceso sobre la señal de entrada y da como resultado una señal de salida independiente a la señal de entrada, pero basada en la primera. Esto significa que no hay retroalimentación hacia el controlador para que éste pueda ajustar la acción de control.

- **CONVOLUCIÓN:** En matemáticas y, en particular, análisis funcional, una convolución es un operador matemático que transforma dos funciones  $f$  y  $g$  en una tercera función que en cierto sentido representa la magnitud en la que se superponen  $f$  y una versión trasladada e invertida de  $g$ .
- **CROMINANCIA:** La crominancia es el componente de la señal de vídeo que contiene las informaciones del color. Por otra parte, luma de la luz o brillo.
- **DERRAPE:** Patinazo o desviación lateral de un vehículo.
- **DIGITALIZACIÓN O DIGITALIZAR:** Consiste en la transcripción de señales analógicas en señales digitales, con el propósito de facilitar su procesamiento (codificación, compresión, etc.) y hacer la señal resultante (la digital) más inmune al ruido y otras interferencias a las que son más sensibles las señales analógicas.
- **DIRECCIONALIDAD:** Cualidad de direccional, que sirve para dirigir u orientar hacia una cosa o acción o en una dirección.
- **EFECTOR:** Se aplica al órgano de un animal que se encarga de la respuesta de este frente a un estímulo determinado.
- **ELASTÓMERO:** Los elastómeros son aquellos polímeros que muestran un comportamiento elástico. El término, que proviene de polímero elástico, es a veces intercambiable con el término goma, que es más adecuado para referirse a vulcanizados.
- **EMBODIMENT:** Es un término que hace referencia a la posición en la ciencia cognitiva y la filosofía de la mente haciendo hincapié en el papel que juega el cuerpo en la formación de la mente.
- **ENDOSCOPIO:** El endoscopio es un instrumento (producto sanitario) en forma de tubo, que contiene una luz y una óptica que permite la visualización del interior de un órgano hueco o una cavidad corporal, introduciéndolo mediante un agujero natural o una pequeña incisión quirúrgica.
- **ESCABROSO:** Terreno desigual, lleno de tropiezos y accidentes.
- **ESCARDAR:** Quitar o arrancar las hierbas nocivas de los sembrados
- **ESCARPADO:** Abrupto, accidentado y áspero al que es difícil acceder.
- **ESPECULAR:** Del espejo o semejante a él. Que se refleja en un espejo.

- **FERROMAGNETISMO:** El ferromagnetismo es un fenómeno físico en el que se produce ordenamiento magnético de todos los momentos magnéticos de una muestra, en la misma dirección y sentido. Un material ferromagnético es aquel que puede presentar ferromagnetismo.
- **FILTRO DE KALMAN:** Es un algoritmo desarrollado por Rudolf E. Kalman en 1960 que sirve para poder identificar el estado oculto (no medible) de un sistema dinámico lineal además funciona cuando el sistema está sometido a ruido blanco aditivo.
- **GECKO:** Los gecónidos, guecos, gecos, gembas, tuqueques, tutecas, salamanquesas o cuijas (Gekkonidae) son una familia de saurópsidos (reptiles) escamosos, que incluye especies de tamaño pequeño a mediano que se encuentran en climas templados y tropicales de todo el mundo.
- **GEOMETRÍA EPIPOLAR:** Es la geometría de la visión estéreo. Cuando dos cámaras ven una escena 3D a partir de dos posiciones distintas, hay una serie de relaciones geométricas entre los puntos 3D y sus proyecciones sobre las imágenes 2D que conducen a las restricciones entre los puntos de la imagen.
- **GIROSCOPIO:** Aparato consistente en un disco circular que gira sobre un eje libre y demuestra la rotación del globo terrestre.
- **GPS:** Global Positioning System, en español Sistema de posicionamiento global, es un sistema global de navegación por satélite que permite determinar en todo el mundo la posición de un objeto, una persona o un vehículo con una precisión hasta de centímetros (si se utiliza GPS diferencial), aunque lo habitual son unos pocos metros de precisión.
- **GRADOS DE LIBERTAD:** Por sus siglas en inglés DOF (Degrees of Freedom) el número de grados de libertad en un sistema se refiere al número de parámetros independientes que definen la configuración su estructura.
- **HEURÍSTICA:** Se puede definir heurística como un arte, técnica o procedimiento práctico o informal, para resolver problemas.
- **HEXAÉDRICO:** Pertenece o relativo a un poliedro de seis caras.
- **HIDROFÓBICO:** Es la propiedad de una molécula que tiene la tendencia a repeler o no disolverse en el agua (o en otras sustancias polares).

- **HIPERESTÁTICO:** En estática, una estructura es hiperestática o estáticamente indeterminada cuando está en equilibrio pero las ecuaciones de la estática resultan insuficientes para determinar todas las fuerzas internas o las reacciones.
- **HISTOGRAMA:** Es una representación gráfica de una variable en forma de barras, donde la superficie de cada barra es proporcional a la frecuencia de los valores representados.
- **IMAGEN ESPECULAR.** Es la imagen mediante la reflexión de la luz en una superficie especular, donde los rayos incidentes se reflejan con un ángulo igual al de incidencia (ambos tomados con respecto a la perpendicular al plano en ese punto).
- **INCLINÓMETRO:** También conocido como escoliómetro es un instrumento usado por topografía, por la aviación y por los navíos para medir la inclinación del plano con respecto de la horizontal (superficie terrestre).
- **INCOMPLETITUD:** Los teoremas de incompletitud de Gödel son dos célebres teoremas de lógica matemática demostrados por Kurt Gödel en 1931. Ambos están relacionados con la existencia de proposiciones indecidibles en ciertas teorías aritméticas.
- **IRRADIANCIA:** Es la magnitud utilizada para describir la potencia incidente por unidad de superficie de todo tipo de radiación electromagnética.
- **JOYSTICK:** Palanca de mando, es un dispositivo de control de dos o tres ejes que se usa desde una computadora o videoconsola, pasando por grúas hasta aviones de caza o transbordadores espaciales.
- **LAMPREA:** Los hiperoartios (Hypertoartia) son una clase de agnatos o peces sin mandíbulas, conocidos vulgarmente con el nombre de lampreas.
- **LASTRE:** Es un sistema que desplaza en el plano vertical u horizontal, aumenta o disminuye, el valor de masa del centro de gravedad de un cuerpo frente a un medio, en el caso de las embarcaciones y sumergibles en el agua.
- **LOCOMOCIÓN:** En términos específicos, la palabra locomoción hace referencia al movimiento que realiza una persona, un animal, un microorganismo, un aparato o máquina para moverse de un lugar a otro, para trasladarse en el espacio. La locomoción varía en términos de forma, estructura, velocidad y otros elementos de acuerdo al tipo de sujeto al que hagamos referencia.

- **LÓGICA DIFUSA:** Es una extensión de la lógica booleana que admite valores lógicos intermedios entre FALSO (0) y VERDADEIRO (1); por ejemplo el valor medio “TALVEZ” (0,5). Esto significa que un valor lógico difuso é un valor cualquiera en el intervalo de valores entre 0 e 1, por tanto esta lógica no exige valores exactos, soporta valoraciones subjetivas e incluso valoraciones lingüísticas para los parámetros del problema y las incorpora apropiadamente a modelos matemáticos complejos.
- **LUBRICACIÓN:** Proceso o técnica empleada para reducir el rozamiento entre dos superficies que se encuentran muy próximas y en movimiento una respecto de la otra, interponiendo para ello una sustancia entre ambas denominada lubricante que soporta o ayuda a soportar la carga (presión generada) entre las superficies enfrentadas.
- **LUMINANCIA:** En Fotometría, la luminancia se define como la densidad angular y superficial de flujo luminoso que incide, atraviesa o emerge de una superficie siguiendo una dirección determinada. Alternativamente, también se puede definir como la densidad superficial de intensidad luminosa en una dirección dada.
- **MANIOBRABILIDAD:** Facilidad de un vehículo para ser dirigido.
- **MAPEO:** Creación de mapas. Determinación e identificación precisa de puntos en un espacio 3D.
- **MATRIZ:** En matemáticas, una matriz es un arreglo bidimensional de números, y en su mayor generalidad de elementos de un anillo. Las matrices se usan generalmente para describir sistemas de ecuaciones lineales, sistemas de ecuaciones diferenciales o representar una aplicación lineal (dada una base).
- **MATRIZ ANTISIMÉTRICA:** Una matriz es antisimétrica cuando es una matriz cuadrada, y es igual a la opuesta de su traspuesta.
- **MATRIZ CUADRADA:** Una matriz de  $n$  por  $m$  elementos, es una matriz cuadrada si el número de filas es igual al número columnas, es decir,  $n = m$  y se dice, entonces que la matriz es de orden  $n$ .
- **MATRIZ JACOBIANA:** Es una matriz formada por las derivadas parciales de primer orden de una función. Una de las aplicaciones más interesantes de esta matriz es la posibilidad de aproximar linealmente a la función en un punto. En este sentido, el jacobiano representa la derivada de una función multivariable.

- **MATRIZ TRASPUESTA:** Se llama matriz traspuesta de una matriz  $A$  de dimensión  $m \times n$ , a la matriz que se obtiene al cambiar en  $A$  las filas por columnas o las columnas por filas. Se representa por  $A^t$  y su dimensión es  $n \times m$ .
- **MICRO-CONTROLADOR:** Es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica.
- **MICROELECTRÓNICA:** Es la aplicación de la ingeniería electrónica a componentes y circuitos de dimensiones muy pequeñas, microscópicas y hasta de nivel molecular para producir dispositivos y equipos electrónicos de dimensiones reducidas pero altamente funcionales
- **MORFOLOGÍA:** Es la disciplina que estudia la generación y las propiedades de la forma. Se aplica en casi todas las ramas del Diseño.
- **NANOFABRICACIÓN:** Es un término usado para describir ya sea la producción de materiales de nanoescala, que pueden ser polvos o líquidos, o para describir la fabricación de piezas a partir de materiales nanoescala o en pequeños pasos para una alta precisión, que se utiliza en varias tecnologías tales como la ablación por láser, ataque químico y otros.
- **NENÚFAR:** El término nenúfar se aplica, en general, a plantas acuáticas con flores que crecen en lagos, lagunas, charcas, pantanos o arroyos de corriente lenta, estando usualmente enraizadas en el fondo.
- **NEUTRALIDAD BOYANTE O FLOTABILIDAD NEUTRA:** La flotabilidad es la capacidad de un cuerpo para sostenerse dentro de un fluido. La flotabilidad es positiva cuando el cuerpo tienda a ascender dentro del fluido, es negativa cuando el cuerpo tiene a descender dentro del fluido, y es neutra cuando se mantiene en suspensión dentro del fluido.
- **NICHO ECOLÓGICO:** En ecología, un nicho es un término que describe la posición relacional de una especie o población en un ecosistema.
- **OCELO:** Órgano visual de los artrópodos y otros metazoarios simples. También, mancha redonda y multicolor que presentan las alas de algunos insectos y aves.
- **ODOMETRIA:** Es el estudio de la estimación de la posición de vehículos con ruedas durante la navegación. Para realizar esta estimación se usa información

sobre la rotación de las ruedas para estimar cambios en la posición a lo largo del tiempo.

- **OMNIDIRECCIONALIDAD:** Capacidad de moverse en cualquier momento y en cualquier dirección a lo largo de un plano  $(x, y)$  o  $(x, y, z)$ , independientemente de ciertas condiciones existentes.
- **ORNITÓPTERO:** Es un aerodino que obtiene el empuje necesario del movimiento batiente de sus alas de forma análoga a como lo hacen las aves y de ahí su nombre que en griego significa “con alas (en griego= pteros) de pájaro (en griego ornos,ornitos)”.
- **ORTOGONAL:** Que está en ángulo recto.
- **PARALELIZACIÓN:** Paralelización automática, también auto paralelización, autoparalelización, o paralelización, se refiere a convertir código secuencial en multi-hilo o vectorizado (o los dos) con objeto de utilizar múltiples procesadores simultáneamente en una máquina con multiprocesador (SMP) de memoria compartida.
- **PERCEPTO:** Es el resultado de haber logrado un grado de excelencia en el producto artístico por el cual éste se volvería perdurable y trascendente al artista e incluso a la totalidad de la obra.
- **PETROQUÍMICA:** Es la extracción de cualquier sustancia química a partir de combustibles fósiles.
- **PLANITUD:** Tolerancia geométrica tridimensional que controla cuánto una característica puede desviarse de un plano liso.
- **PLEXIGLÁS:** Es una resina sintética que tiene el aspecto de vidrio. Es también llamada como vidrio orgánico.
- **POLÍMERO:** Los polímeros son macromoléculas (generalmente orgánicas) formadas por la unión de moléculas más pequeñas llamadas monómeros.
- **POLIMORFO:** Que puede tener varias formas sin cambiar su naturaleza.
- **POLIQUETO:** Clase de gusanos anélidos, predominantemente marinos y unisexuales, de cuerpo cilíndrico, con branquias, anillos provistos de numerosas cerdas llamadas quetas y una región cefálica diferenciada con ojos y tentáculos.

- **POLIURETANO:** El poliuretano (PUR) es un polímero que se obtiene mediante condensación de di-bases hidroxílicas combinadas con disocianatos.
- **PROPULSOR:** Es un dispositivo mecánico usado enormemente en vehículos de locomoción para generar su impulso. La palabra se utiliza comúnmente en dispositivos marítimos, e implica un conjunto mecánico que es más complicado que una hélice.
- **PVC:** El policloruro de vinilo o PVC (del inglés poly(vinyl chloride)) es un polímero termoplástico. Se presenta como un material blanco que comienza a reblandecer alrededor de los  $80^{\circ}C$  y se descompone sobre  $140^{\circ}C$ .
- **RALENTIZAR:** Hacer más lento un proceso o una actividad.
- **REDES NEURONALES ARTIFICIALES:** Denominadas habitualmente como “RNA” o en inglés como “ANN” Artificial Neural Network son un paradigma de aprendizaje y procesamiento automático inspirado en la forma en que funciona el sistema nervioso de los animales. Se trata de un sistema de interconexión de neuronas en una red que colabora para producir un estímulo de salida.
- **ROBOT:** Máquina electrónica que puede ejecutar distinto tipo de operaciones y movimientos de forma automática.
- **ROBOT HÍBRIDO:** Corresponden a aquellos de difícil clasificación cuya estructura se sitúa en combinación de elementos morfológicos, bien sea por conjunción o por yuxtaposición. Por ejemplo un robot con ruedas y dotado de un brazo robótico para la toma de muestras.
- **ROBOT HOLONÓMICO:** Son capaces de cambiar de sitio en cualquier dirección y en direcciones aleatorias.
- **ROBOT MÓVIL:** Robot montado sobre una base móvil controlada automáticamente.
- **ROBOT NO HOLONÓMICO:** El movimiento se produce solo por dos desplazamientos y el vehículo posee solo dos grados de libertad.
- **ROVER:** Es un vehículo de exploración espacial diseñado para moverse a través de la superficie de un planeta u otro objeto astronómico.

- **SAGITAL:** Los planos sagitales, en anatomía, son aquellos planos perpendiculares al suelo y en ángulo recto con los planos frontales, que dividen al cuerpo en mitades izquierda y derecha. La línea sagital es una vertical de referencia que teóricamente cruza el cuerpo por la parte media y central, a modo de plomada imaginaria.
- **SEA CHEST:** caja estanca construida contra el casco del buque comunicándolo con el océano a través de una rejilla, para que las válvulas y tuberías unidos a este permitan que el agua entre para el lastre, refrigeración del motor, y extinción de incendios.
- **SENSORES:** Dispositivo capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas. Las variables de instrumentación pueden ser por ejemplo: temperatura, intensidad lumínica, distancia, aceleración, inclinación, desplazamiento, presión, fuerza, torsión, humedad, pH, etc.
- **SENSORES GEOFÍSICOS:** Los sensores geofísicos se utilizan a menudo para controlar la humedad del suelo de forma continua en aplicaciones agrícolas y científicas.
- **SERVOMOTOR:** También llamado servo, es un dispositivo similar a un motor de corriente continua que tiene la capacidad de ubicarse en cualquier posición dentro de su rango de operación, y mantenerse estable en dicha posición.
- **SIGMOIDE:** Muchos procesos naturales y curvas de aprendizaje de sistemas complejos muestran una progresión temporal desde unos niveles bajos al inicio, hasta acercarse a un clímax transcurrido un cierto tiempo; la transición se produce en una región caracterizada por una fuerte aceleración intermedia. La función sigmoide permite describir esta evolución. Su gráfica tiene una típica forma de “S”.
- **SILVICULTURA:** (del español silva, selva, bosque, y cultura, cultivo) es el cuidado de los bosques, cerros o montes y también, por extensión, la ciencia que trata de este cultivo.
- **SINCRÓNICO/SÍNCRONO:** Alude a un proceso o su efecto que se desarrolla en perfecta correspondencia temporal con otro proceso o causa.
- **SINUOSO:** Se dice del camino o terreno que forma curvas, ondas o recodos.

- **SITUATEDNESS:** Es un término, que entre otras cosas, describe a un robot inmerso dentro de un entorno, interaccionado con el mundo el cual influye de manera directa sobre su comportamiento.
- **SISTEMA BIOLÓGICO:** O Sistema orgánico es un conjunto de órganos y estructuras similares que trabajan en conjunto para cumplir alguna función fisiológica en un ser vivo relacionado con la biología
- **SISTEMA DE NAVEGACIÓN INERCIAL:** INS por sus siglas en inglés (Inertial Navigation System), es un sistema de ayuda a la navegación que usa un computador, sensores de movimiento y sensores de rotación para calcular continuamente mediante estima la posición, orientación, y velocidad de un objeto en movimiento sin necesidad de referencias externas. Es usado en vehículos como barcos, aeronaves, submarinos, misiles, y naves espaciales
- **SISTEMAS BIOINSPIRADOS:** Son sistemas construidos por medio de hardware configurables y sistemas electrónicos que emulan la forma de pensar, el modo de procesar información y resolución de problemas de los sistemas biológicos.
- **SISTEMAS EMBEBIDOS:** Es un sistema de computación diseñado para realizar una o algunas pocas funciones dedicadas frecuentemente en un sistema de computación en tiempo real.
- **SONDA PITOT:** Es utilizada para calcular la presión total, también llamada presión de estancamiento, presión remanente, se utiliza mucho para medir la velocidad del viento en aparatos aéreos y para medir las velocidades de aire y gases en aplicaciones industriales.
- **SUBOPTIMAL:** No comparable, menos que óptimo. Una solución subóptima rápidamente producida a un problema es a menudo más útil que una óptima que lleva mucho tiempo para producir.
- **SUPRAYACENTE:** Se dice de aquello que es colocado en o sobre otra cosa.
- **TEFLÓN:** El Teflón (PTFE) es un polímero similar al polietileno, en el que los átomos de hidrógeno han sido sustituidos por átomos flúor.
- **TELEOPERACIÓN:** Un sistema teleoperado es aquél que permite gobernar un robot *slave* (controlar su movimiento y la fuerza ejercida) ubicado en una zona remota (puede que el esclavo esté realmente muy lejos o puede que esté en un

entorno hostil del que haya que proteger al operador) a través del manejo de un robot master localizado en el punto de trabajo del operador.

- **TESELADO:** Un teselado o teselación es una regularidad o patrón de figuras que cubre o pavimenta completamente una superficie plana que cumple con dos requisitos, que no queden huecos y que no se superpongan las figuras.
- **TOPOLOGÍA:** Es la rama de las matemáticas dedicada al estudio de aquellas propiedades de los cuerpos geométricos que permanecen inalteradas por transformaciones continuas.
- **TRACTOR ORUGA:** Es un dispositivo de transporte utilizado principalmente en vehículos pesados, como tanques y tractores, u otro tipo de vehículos. Consiste en un conjunto de eslabones modulares que permiten un desplazamiento estable aun en terrenos irregulares.
- **TRASVASAR:** Pasar un contenido de un receptáculo (recipiente) a otro.
- **TÚNEL DE VIENTO:** En ingeniería, un túnel de viento o túnel aerodinámico es una herramienta de investigación desarrollada para ayudar en el estudio de los efectos del movimiento del aire alrededor de objetos sólidos. Con esta herramienta se simulan las condiciones que experimentará el objeto de la investigación en una situación real.
- **TURBULENCIA:** En términos de la dinámica de fluidos, turbulencia o flujo turbulento es un régimen de flujo caracterizado por baja difusión de momento, alta convección y cambios espacio-temporales rápidos de presión y velocidad.
- **UAV:** Por sus siglas en inglés “Unmanned Aerial Vehicle”, en español vehículo aéreo no tripulado o sistema aéreo no tripulado, UAS (Unmanned Aerial System), conocido en castellano por sus siglas como VANT, es una aeronave que vuela sin tripulación humana a bordo.
- **VACÍO:** Es la ausencia total de material en los elementos (materia) en un determinado espacio o lugar, o la falta de contenido en el interior de un recipiente. Por extensión, se denomina también vacío a la condición de una región donde la densidad de partículas es muy baja, como por ejemplo el espacio interestelar; o la de una cavidad cerrada donde la presión de aire u otros gases es menor que la atmosférica.

- **VÁLVULA:** Es un mecanismo que regula el flujo de la comunicación entre dos partes de una máquina o sistema.
- **VENTOSAS:** Las ventosas o válvulas de aeración son dispositivos que se instalan para controlar de forma automática la presencia de aire en los conductos.
- **WAYPOINTS:** Son coordenadas para ubicar puntos de referencia tridimensionales utilizados en la navegación fundamentada en GPS (Global Positioning System). La palabra viene compuesta del inglés way (camino) y point (punto), en realidad se emplean para trazar rutas mediante agregación secuencial de puntos.

## RESUMEN

**TÍTULO:**  
**SEMINARIO DE INVESTIGACIÓN EN ROBÓTICA MÓVIL AUTÓNOMA<sup>1</sup>**

**AUTORES<sup>2</sup>:** DANIEL RICARDO CARRILO DUARTE  
GARY ALEJANDRO SÁNCHEZ AMAYA  
JONATHAN FABIAN NIÑO CASTILLO

**PALABRAS CLAVE:** Robótica móvil autónoma, robot, cinemática, dinámica, percepción, localización, navegación.

**DESCRIPCIÓN:** La robótica es la rama de la tecnología que se dedica al diseño, construcción, operación, disposición estructural, manufactura y aplicación de los robots. En particular la robótica móvil autónoma es aquella que se encarga de estudiar y desarrollar robots capaces de desplazarse y actuar sin intervención humana, la cual ha evolucionado enormemente con el paso del tiempo permitiéndole al ser humano alcanzar lugares o hacer labores que son difíciles o imposibles para él, tales como la desactivación de bombas, la vigilancia aérea, la exploración marítima y la exploración espacial.

El estudio de la robótica móvil autónoma se ha convertido en una de las principales líneas de investigación en muchas universidades en el mundo y por eso se hace necesario que en la Universidad Industrial de Santander también se comience avanzar en este aspecto y el proyecto desarrollado se convierte en el primer paso que la escuela de ingeniería mecánica de la universidad da para ingresar a esta línea de investigación. El presente seminario de investigación en robótica móvil autónoma se realizó para profundizar en este tema, generando un documento que contiene las bases del estudio de la locomoción, cinemática, dinámica, percepción, localización y navegación de los robots móviles con miras a que en un futuro próximo, se creen más proyectos relacionados con el desarrollo de la robótica móvil autónoma y al mismo tiempo se establezcan las bases para una asignatura de pregrado con este contenido en la escuela de ingeniería mecánica.

---

<sup>1</sup>Trabajo de grado

<sup>2</sup>Facultad de Ingenierías Físico-Mecánicas, Escuela de Ingeniería Mecánica, Director: MSc Carlos Borrás Pinilla

## ABSTRACT

**TITLE:**  
**SEMINARY OF INVESTIGATION IN AUTONOMOUS MOBILE ROBOTICS<sup>3</sup>**

**AUTHORS<sup>4</sup>:** DANIEL RICARDO CARRILO DUARTE  
GARY ALEJANDRO SÁNCHEZ AMAYA  
JONATHAN FABIAN NIÑO CASTILLO

**KEY WORDS:** Autonomous mobile robotics, robot, kinematics, dynamics, perception, localization, navigation.

**DESCRIPTION:** Robotics is the branch of technology devoted to the design, construction, operation, structural disposition, manufacture and application of robots. In particular autonomous mobile robotics is that which is responsible for studying and developing robots able to move and act without human intervention, which has evolved tremendously over time allowing the human being to reach places or doing tasks that are difficult or impossible to him, such as bomb disposal, air surveillance, maritime exploration and space exploration.

The study of autonomous mobile robotics has become one of the main lines of research in many universities in the world and therefore it is necessary that the Universidad Industrial de Santander also begins developing this aspect and this project developed becomes in the first step that the mechanical engineering school of the university gives to enter this line of research. The seminary of autonomous mobile robotics research was conducted to investigate this subject, generating a document that provides the basis of the study of locomotion, kinematics, dynamics, perception, localization and navigation of mobile robots in order to bring in the near future more projects related to the development of autonomous mobile robotics while establishing the foundation for an undergraduate course with this content in the mechanical engineering school.

---

<sup>3</sup>Thesis

<sup>4</sup>Physical-Mechanical Engineering Faculty , Mechanical Engineering School, Director: MSc Carlos Borrás Pinilla

## INTRODUCCIÓN

Gracias a la invención de la robótica se ha logrado conseguir un avance gigantesco en la automatización de tareas industriales. La demanda de nuevos retos y labores no solo en la industria sino en otros campos para la robótica ha generado que dicha tecnología se vea en la necesidad de desarrollar una rama de la misma en la que dichos robots sean capaces de hacer lo que en muchos casos para los humanos no es viable o no pueden hacer, o en otro caso la realización de trabajos que representarían un riesgo enorme para su vida. La Robótica Móvil Autónoma es el área de la robótica que desarrolla sistemas complejos de robots capaces de desplazarse y actuar sin intervención humana. Para ello el robot debe percibir su entorno y actuar de forma adecuada, además de llevar a cabo su tarea. El papel de la robótica móvil autónoma en misiones espaciales, misiones militares tales como la desactivación de minas, sistemas de rastreo y vigilancia, en la construcción y muchos otros campos es realmente importante y de allí parte la necesidad del estudio de este tema.

La Universidad Industrial de Santander es una institución académica que tiene como propósito la formación de personas de alta calidad ética, política y profesional que lideren el desarrollo económico-social, y que además realicen proyectos educativos e investigativos, que contribuyan al logro de las metas del desarrollo del país. Su misión implica la construcción, aplicación y divulgación del conocimiento científico, por lo que la escuela de ingeniería mecánica en busca de aportar para el cumplimiento de dicha misión, está indagando y explorando en diversos campos con el fin de implementar nuevos programas que propendan por el continuo aumento de la calidad de la producción intelectual.

Con la puesta en marcha de los estudios de maestría de la escuela de ingeniería mecánica de la universidad industrial de Santander y junto con el crecimiento del grupo de investigación avalado y clasificado en COLCIENCIAS, Sistemas Dinámicos Multifísicos, Control y Robótica DICBOT se ha contribuido enormemente al desarrollo investigativo y docencia al pregrado de la escuela de ingeniería mecánica de la Universidad Industrial de Santander, aunque es claro que el área de la robótica no es tan fuerte como se desea y no existe un curso dedicado al estudio de sistemas robóticos dirigido a estudiantes de pregrado. Es por esto que con este proyecto de grado se buscó fortalecer los trabajos de investigación en el tema de la robótica móvil autónoma y propender por la creación de una asignatura referente al tema, de la cual se beneficien los estudiantes de ingeniería mecánica, brindando los fundamentos, las bases conceptuales, mas sin profundizar o enfatizar en cada uno de los tópicos debido a la complejidad del tema.

**Parte I**

**SEMINARIO DE INVESTIGACIÓN**

---

---

# CAPÍTULO 1

---

## GENERALIDADES

La Universidad Industrial de Santander permite realizar un Seminario de Investigación (Seminario Alemán), cuyo origen se dio a finales del siglo XVIII en la Universidad de Gottingen, como trabajo de grado, que consiste en estudiar, discutir e intercambiar experiencias acerca de un tema en particular, en un grupo, en el cual sus participantes, se intercomunican exponiendo dicho tema (Relatoría), complementándolo, evaluándolo (Correlatoría), aportando entre todos (Discusión), sacando conclusiones y planteando nuevos interrogantes permitiendo que todo ello quede en la memoria escrita (Protocolo). El Seminario de Investigación se programa por temas los cuales son seleccionados con la orientación del director del seminario, quien con su experiencia y conocimiento del tema central, guía la selección con la debida pertinencia, actualidad y ubicación en el contexto. Los temas son desarrollados en sesiones planificadas, en las cuales, los miembros del grupo deben asumir diferentes roles, de acuerdo con la descripción anterior.

El objetivo principal de esta actividad académica es formar a los participantes para la investigación científica mediante el desarrollo de habilidades específicas aplicadas al asumir los diferentes roles dentro del seminario. Dichas habilidades están orientadas a desarrollar la capacidad de lector crítico de resultados de investigación en cualquiera de las áreas del conocimiento, a fortalecer la capacidad de observar e identificar los problemas presentes en tópicos bajo análisis, a buscar respuestas a preguntas claves y sustentarlas teórica y metodológicamente en forma verbal y por escrito, y a identificar las relaciones del problema objeto de estudio con el contexto económico, político o social, a fin de enriquecer con una mirada de integralidad, el conocimiento para el grupo

de estudiantes.

El seminario de investigación presenta varias ventajas, como por ejemplo fortalecer el hábito de documentarse acerca del tema bajo estudio, permitir que los participantes desempeñen diferentes roles dentro del grupo, desarrollando habilidades comunicativas y de relaciones interpersonales, complementarias para la formación tanto personal como profesional, fomentar el aprendizaje como una experiencia grupal, permitir el uso de distintas herramientas didácticas de apoyo al desarrollo de las sesiones, así como un control sobre la planificación establecida de éstas y es una metodología integradora centrada en el estudiante. Sus ventajas la hacen aplicable a todas las áreas del conocimiento humano.

## **1.1. ORGANIZACIÓN DEL SEMINARIO DE INVESTIGACIÓN**

El Seminario de Investigación se compone de las siguientes actividades: la Relatoría, la Correlatoría, la Discusión y el Protocolo, las cuales deben girar en torno a un tema, del que se desprenden los subtemas a tratar durante las sesiones. Dichas actividades son responsabilidad de los integrantes del grupo, por lo cual a cada uno de ellos es asignado un rol, de carácter rotativo; es así como una persona que en una sesión asume el rol de correlator, en la siguiente puede ser el protocolante, es decir el responsable del protocolo. La organización del seminario también implica establecer el lugar del seminario, el número de sesiones y las fechas para realizarlas, así como la duración de cada una de las actividades (Relatoría, Correlatoría, Discusión y Protocolo).

En nuestra investigación, las sesiones se llevaron a cabo en las instalaciones de maestría de la Escuela de Ingeniería Mecánica, y el número de sesiones que abordan los temas y subtemas relacionados con el seminario de especifican más adelante. Las sesiones se programaron para una duración de 1 hora y media. En la siguiente tabla se especifican los participantes que desarrollaron esta investigación así como el director de seminario.

<b>Director</b>	<b>Ing. Carlos Borrás Pinilla</b>
<b>Participantes.</b>	Jonathan Fabian Niño Castillo Gary Alejandro Sánchez Amaya Daniel Ricardo Carrillo Duarte

## 1.2. TEMA DEL SEMINARIO DE INVESTIGACIÓN

La Robótica Móvil Autónoma es el área de la robótica que desarrolla sistemas complejos de robots capaces de desplazarse y actuar sin intervención humana. Para ello el robot debe percibir su entorno y actuar de forma adecuada, además de llevar a cabo su tarea. El papel de la robótica móvil autónoma en misiones espaciales, misiones militares tales como la desactivación de minas, sistemas de rastreo y vigilancia, en la construcción y muchos otros campos es realmente importante y de allí parte la necesidad del estudio de este tema. Este seminario se centra en el estudio de la locomoción, cinemática, dinámica, percepción, localización y navegación en los robots móviles.

## 1.3. DIRECCIÓN DEL SEMINARIO DE INVESTIGACIÓN

La dirección del Seminario de Investigación en Robótica Móvil Autónoma estuvo a cargo del ingeniero Carlos Borrás Pinilla que cuenta con la siguiente formación:

### Cuadro 1.1: Formación Ing. Carlos Borrás Pinilla

- **Doctorado** The University Of Oklahoma Aerospace And Mechanical Engineering  
Doctor Of Philosophy In Mechanical Engineering  
Agostode2001 - Enero de 2009  
Micro Peristaltic Pump in Single Chip
- **Maestría/Magister** University Of Oklahoma  
Master Of Science In Mechanical Engineering  
de1999 - de 2001  
Pattern Recognition of Hydraulic Backlash using Neural Network
- **Especialización** Universidad Industrial de Santander - UIS  
Especialización En Gerencia de Mantenimiento  
de1996 - de 1998  
Mantenimiento Total Productivo Una Aproximación Colombiana
- **Pregrado/Universitario** Universidad Industrial de Santander - UIS  
Ingeniería Mecánica  
de1981 - de 1989  
DISEÑO Y CONSTRUCCIÓN DE ELECTROVALVULAS HYDRAULICAS Y NEUMATICAS
- **Perfeccionamiento** Japan International Cooperative Agency  
Oil Hydraulic And It's Application Fluid Power  
de1994 - de 1995  
NO MONOGRAFIA. PROYECTOS DE ESTUDIOS EN FLUID POWER.

## 1.4. METODOLOGÍA DEL SEMINARIO DE INVESTIGACIÓN

Las diferentes etapas que se realizan dentro de la dinámica del seminario, se pueden agrupar en tres grandes grupos: Planeación, ejecución y finalización; las cuales se relacionan entre si, dependiendo cada una de la anterior. La figura 1.1 ejemplifica esta dinámica.

Figura 1.1: Metodología seminario de investigación



---

---

# CAPÍTULO 2

---

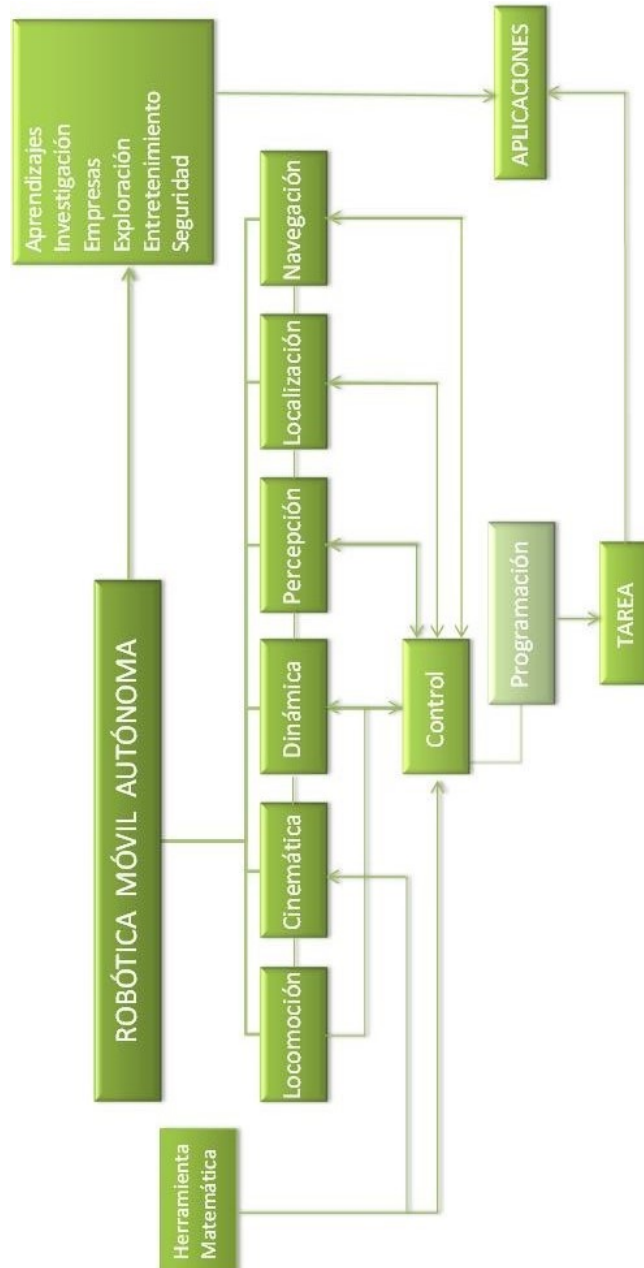
## PLANEACIÓN

En esta etapa se establecen los lineamientos bajos los cuales se desarrollará el seminario, se define el alcance y los resultados que se esperan obtener. El alcance del seminario depende del grado de profundidad del estudio, la formación de los participantes, y la orientación y perfil del director. En la figura 2.1 se puede observar un esquema general de la robótica móvil autónoma.

Los objetivos del seminario son:

- Realizar un Seminario de Investigación en Robótica Móvil Autónoma, mediante la discusión crítica, propositiva y argumentativa de acuerdo a los lineamientos establecidos por la vicerrectoría académica en el acuerdo 004 de 2007 artículo 4 para los seminarios de investigación.
- Elaborar un documento síntesis original en  $\text{\LaTeX}$  donde se recojan los resultados de la investigación y la discusión realizada en cada sesión a partir de los resúmenes, las relatorías recopiladas y los protocolos en el seminario de investigación de acuerdo con los siguientes temas:
  1. Introducción
  2. Estado del arte
  3. Locomoción
  4. Cinemática
  5. Dinámica

Figura 2.1: Esquema general de la Robótica Móvil Autónoma



6. Percepción
7. Localización
8. Planificación y Navegación
9. Futuro de la Robótica Móvil Autónoma

- Cumplir con los roles asignados a cada integrante del seminario de investigación.
- Elaborar presentaciones en PowerPoint y documentos síntesis de cada uno de los temas investigados, obtenidos de los las relatorías y los protocolos recopilados de cada sesión.
- Desarrollar programas en MATLAB y/o SIMULINK alusivos a los temas que se encuentran en el temario del seminario.
- Diseñar un robot móvil para realizar rutinas de prueba.

## 2.1. ESTUDIO BIBLIOGRÁFICO

Se usó la siguiente bibliografía antes de dar comienzo al seminario:

- CLAWAR 2008 (Climbing and Walking Robots and the Support Technologies for Mobile Machines), fue la onceava de una serie de conferencias donde los temas principales fueron los robots caminadores y trepadores. El procedimiento incluyó la investigación del estado del arte y los resultados presentados durante el desarrollo CLAWAR 2008 con sus 153 conferencias y presentaciones técnicas por autores de 32 países de los cinco continentes.
- BRÄUNL, Thomas. Embedded Robotics (2006). Este libro esta dividido en tres partes principales, la primera contiene información acerca de los “Embedded systems”, control de estos robots, sensores, operación a distancia y otros temas. La segunda parte tiene que ver con diseño de robots móviles como robots omnidireccionales, caminadores de seis piernas, aviones, vehículos submarinos (AUV), y todo lo que tiene que ver con su control. Por ultimo la tercera parte esta encargada de explicar las aplicaciones de los robots móviles en la actualidad.
- HOLLAND, John. Designing Autonomous Mobile Robots (2004). Este no es un libro que trate sobre como construir un robot, mas bien se trata de un libro sobre como organizar un diseño de un robot para que su construcción de realice de forma natural, sencilla y sin complicaciones: es un manual de construcción asequible a cualquier persona, basándose solo en los conocimientos generales que cada quien tiene y que ha adquirido en su formación básica.

- SIEGWART, Roland., NOURBAKHS, Illah R., Introduction to Autonomous Mobile Robots (2004). Este libro presenta una introducción a los fundamentos de la robótica móvil, que abarca el movimiento de estos robots, la cinemática y la dinámica de los mismos, visualización, percepción y navegación.
- XIE, Ming. Fundamentals Of Robotics, Linking Perception To Action, (2003). Este libro sirve como guía en un curso de robótica para principiantes, cubre los conocimientos prácticos en la comprensión, el desarrollo y el uso de robots como equipo versátil para automatizar una variedad de procesos industriales o tareas, Pero, también discutirá las posibilidades que podemos esperar cuando somos capaces de crear una visión guiada, aprendizaje automático.
- ÁNGELES, Jorge. Fundamentals Of Robotic Mechanical Systems Theory, Methods, and Algorithms, Third Edition, (2007). La robótica abarca un espectro bastante amplio de conocimientos, el alcance de este libro es sólo una banda estrecha de este espectro. El objetivo del libro es hablar de los sistemas mecánicos de los robots.
- BRAZ, José., ARAÚJO Helder, VIEIRA Alves And ENCARNAÇÃO, Bruno. Informatics In Control, Automation And Robotics I, (2006). Este libro contiene una selección de trabajos de (ICINCO 2004) la primera Conferencia Internacional sobre Informática en la automatización de Control y Robótica, en dicha conferencia se trataron las siguientes temáticas: “Sistemas Inteligentes de Control y Optimización”, “Robótica y Automatización” y “Modelado de Sistemas, Procesamiento de Señales y Control”.
- SANDINE, Paul E., Mechanisms and Mechanical Devices Illustrated (2003). Este libro está destinado a ser interesante, útil y educativa para los aficionados, estudiantes, educadores, e ingenieros de nivel medio o el estudio de diseño de robots móviles que hacen el verdadero trabajo. Se centra fundamentalmente en los mecanismos y dispositivos que se refieren a los vehículos que se mueven por sí mismos y, de hecho hacer las cosas de manera autónoma, es decir, un robot.
- BARRIENTOS, Antonio. Fundamentos de robótica, 2da edición. En este texto se ha procurado llegar a un adecuado equilibrio entre aquellos temas relacionados con el conocimiento profundo del funcionamiento de un robot (en sus aspectos mecánico, informático y de control) y aquellos en los que se proporcionan criterios para evaluar la conveniencia de utilizar un robot y el modo más adecuado de hacerlo.

- SANTOS, José., DURO, Richard J. Evolución artificial y robótica autónoma. El libro se centra en el estudio de temas como los algoritmos para simular la evolución, la evaluación de la calidad en robots, la coevolución para la obtención automática y simultánea de comportamientos y morfologías óptimas, los procesos de desarrollo, las estructuras que soportarán el control y las ventajas de utilización de computación neuronal en robótica.

Posteriormente y en base al análisis realizado en la bibliografía inicial se programan los temas y subtemas del seminario de investigación. Estos se muestran en la tabla 2.1.

Cuadro 2.1: Temas y subtemas del seminario de investigación

<b>Tema</b>	<b>Subtema</b>
<b>INTRODUCCIÓN</b>	Historia
	La palabra ROBOT
	Origen y desarrollo de la robótica
	La robótica en Colombia
	La robótica en la Universidad Industrial de Santander
	La robótica en otras universidades de Santander
	Definición de robótica
	Clasificación de los robots
<b>ESTADO DEL ARTE</b>	Metodología de diseño
	La percepción y conocimiento de la situación
	Control de robots móviles
	Robots móviles
	Compañías fabricantes de robots móviles en la actualidad
<b>LOCOMOCIÓN DE ROBOTS MÓVILES</b>	Introducción
	Generalidades de la locomoción
	Robots móviles con piernas
	Robots móviles con ruedas
	Robots trepadores
	Biorobots
<b>CINEMÁTICA DE LOS ROBOTS MÓVILES</b>	Generalidades de la cinemática
	Supuestos considerados
	Modelos y restricciones cinemáticas
Continúa en la página siguiente	

**Cuadro 2.1 – Continuación**

Tema	Subtema
	Maniobrabilidad del robot móvil
	Espacio de trabajo del robot móvil
	Mas allá de la cinemática básica
	Control de movimiento (control de cinemática)
	Cinemática de robots rodantes
	Cinemática de marcha
<b>DINÁMICA DE LOS ROBOTS MÓVILES</b>	Robots móviles tipo Hilare
	Dinámica de robots móviles no holonómicos
	Dinámica de robots rodantes
	Dinámica de robots bípedos
	Dinámica de buques de superficie
	Dinámica de helicópteros autónomos
<b>PERCEPCIÓN EN LOS ROBOTS MÓVILES</b>	Sensores para robots móviles
	Fundamentos de la visión por computador
	Fundamentos de procesamiento de imagen
	Extracción de características
	Extracción de características de imagen: detectores de punto de interés
	Reconocimiento de lugar
	Extracción de características basada en datos de alcance (láser, ultrasonidos)
<b>LOCALIZACIÓN EN LOS ROBOTS MÓVILES</b>	Generalidades
	El desafío de la localización: ruido y aliasing
	Localizar o no localizar: localización basada en navegación versus soluciones
	Representación de la creencia
	Representación del mapa
	Localización probabilística basada en mapas
	Construcción autónoma de mapas
<b>PLANIFICACIÓN Y NAVEGACIÓN EN LOS ROBOTS MÓVILES</b>	Generalidades
	Competencias para la navegación: planificación y reacción
	Planificación de ruta
	Evasión de obstáculos
Continúa en la pagina siguiente	

**Cuadro 2.1 – Continuación**

<b>Tema</b>	<b>Subtema</b>
	Arquitecturas de navegación
<b>FUTURO DE LA ROBÓTICA MÓVIL</b>	Tendencias actuales y futuras
	Evolución artificial
	¿Qué se puede evolucionar en robótica?
	¿Cuál puede ser el camino a seguir en la robótica móvil autónoma?

---

---

# CAPÍTULO 3

---

## EJECUCIÓN

Esta etapa se llevó a cabo lo establecido en la planeación. Se muestra el desarrollo de las sesiones respectivas y la manera en que los distintos roles en las mismas se distribuyeron. Estos roles se muestran en la tabla 3.1 donde R, C, P, D, y A indican Relator, Correlator, Protocolante, Discusión y Apertura respectivamente.

El protocolo para realizar una sesión es el siguiente:

- Realizar un acta, con la fecha, hora de inicio de la sesión, y un listado de los temas a tratar en la misma, con sus respectivas observaciones. Cada acta debe al final de la presentación debe quedar firmada por cada uno de los integrantes del seminario de investigación así como el director de proyecto.
- Entregar al director de proyecto un documento en  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  correspondiente al tema tratado en cada sesión.
- Almacenar un documento en PowerPoint de la presentación.

### 3.1. ACTAS

Un acta es un escrito que tiene por objeto dejar constancia de lo tratado, sucedido y acordado en una junta o reunión, en este caso, cada una de las sesiones desarrolladas en el seminario de investigación. En ella, se debe registrar la fecha y la hora en

Cuadro 3.1: Asignación de roles para cada una de las sesiones

Expositor	Carlos Borrás				Daniel Carrillo				Jonathan Niño				Gary Sánchez			
	A	R	C	P	D	R	C	P	D	R	C	P	D			
1°	X		X				X	X	X							
2°	X	X				X						X	X			
3°	X			X	X	X					X					
4°	X		X				X	X	X							
5°	X	X				X						X	X			
6°	X			X	X	X					X					
7°	X		X				X	X	X							
8°	X	X				X						X	X			
9°	X			X	X	X					X					
10°	X		X				X	X	X							
11°	X	X				X						X	X			
12°	X			X	X	X					X					
13°	X		X				X	X	X							
14°	X	X				X						X	X			
15°	X			X	X	X					X					
16°	X		X				X	X	X							



A: Apertura, R: Relator, C: Correlator, P: Protocolante, D: Discusión

que se desarrolla la sesión, el tema y cada uno de los subtemas a tratar de forma específica y concisa y debe estar firmada por el director de proyecto y cada uno de los realizadores de la investigación. En la figura 3.1 se puede observar un ejemplo de un acta realizada para una de las sesiones expuestas en el seminario de investigación en robótica móvil autónoma, la cual fue aprobada posteriormente por el director de proyecto.

### 3.2. TALLERES DE APLICACIÓN

Con el objetivo de llevar a la práctica todos los temas tratados en la investigación, se desarrollaron programas, talleres de aplicación usando MATLAB y/o SIMULINK. Los enunciados de dichos talleres, sus respectivos códigos y funcionamiento, se pueden

Figura 3.1: Ejemplo de un acta

 	
<b>SEMINARIO DE INVESTIGACIÓN EN ROBÓTICA MÓVIL AUTÓNOMA.</b> Acta Sesión 5.	
Fecha: 7 DE AGOSTO DE 2013	Hora: 9:00 a.m.
RELATOR: DANIEL RICARDO CARRILLO DUARTE. CORRELATOR: JONATHAN FABIAN NIÑO CASTILLO. PROTOCOLANTE: GARY ALEJANDRO SÁNCHEZ AMAYA. DISCUSIÓN: GARY ALEJANDRO SÁNCHEZ AMAYA.	
DIRECTOR: ING. CARLOS BORRÁS PINILLA	
<b>TEMA: CINEMÁTICA DE LOS ROBOTS MÓVILES (PRIMERA PARTE).</b>	
<b>ORDEN DEL DÍA:</b> <ul style="list-style-type: none"> <li>• Presentación de las modificaciones hechas a la tercera y cuarta sesión.</li> <li>• A continuación se dará inicio al tema de la quinta sesión, correspondiente a la cinemática de los robots móviles, primera parte.</li> <li>• Primero se hablará sobre las generalidades de la cinemática de la robótica móvil, teniendo en cuenta de antemano que solo se tratará en esta sesión lo concerniente a la cinemática de los robots móviles con ruedas.</li> <li>• Posteriormente se hablará brevemente sobre los diferentes tipos de ruedas existentes y sus configuraciones, algo de lo que se había hablado previamente en las anteriores sesiones, pero se hace necesario hacer hincapié en esto para comprender los modelos cinemáticos a tratar en el transcurso de la sesión.</li> <li>• También, se incluirán definiciones básicas como centro instantáneo de rotación (ICR, por sus siglas en inglés), tipos de restricciones y otras.</li> <li>• Seguidamente, se dará paso a los modelos y restricciones cinemáticas como tal teniendo en cuenta cada uno de los tipos de rueda y restricciones cinemáticas del robot.</li> </ul>	
• Habiendo expuesto todo lo anterior se proseguirá a hablar sobre la maniobrabilidad del robot y sus grados de movilidad, dando así fin al tema correspondiente a la primera parte de la cinemática de los robots móviles.	
<hr/> DANIEL RICARDO CARRILLO DUARTE. ESCUELA DE INGENIERÍA MECÁNICA. UNIVERSIDAD INDUSTRIAL DE SANTANDER.	
<hr/> JONATHAN FABIAN NIÑO CASTILLO. ESCUELA DE INGENIERÍA MECÁNICA. UNIVERSIDAD INDUSTRIAL DE SANTANDER.	
<hr/> GARY ALEJANDRO SÁNCHEZ AMAYA. ESCUELA DE INGENIERÍA MECÁNICA. UNIVERSIDAD INDUSTRIAL DE SANTANDER.	
<hr/> ING. CARLOS BORRÁS PINILLA. ESCUELA DE INGENIERÍA MECÁNICA. UNIVERSIDAD INDUSTRIAL DE SANTANDER.	

encontrar detalladamente en el *Anexo C*. Ya que el objetivo principal de esta investigación en robótica móvil autónoma es sentar las bases teóricas para la creación de una asignatura en la escuela de ingeniería mecánica, estos talleres se convierten en una herramienta útil de aprendizaje y son la base para futuros trabajos en la escuela.

En la tabla 3.2 se muestran los talleres realizados así como una breve descripción de los mismos y los temas a los que están vinculados. Cabe aclarar que para el taller teórico de la simulación cinemática, se hizo la comprobación práctica con nuestro robot LEVI. En las figuras 3.2 y 3.3 se pueden observar el entorno gráfico y el respectivo código en MATLAB de esta simulación matemática.

### 3.3. ROBOT LEVI

Se construyó también un robot diferencial de dos ruedas y un apoyo, para realizar algunas de las rutinas que en competencias nacionales de robótica móvil se implementan. Las rutinas que el robot hace son las siguientes:

- Seguimiento de línea
- Cruces a 90° a derecha e izquierda y giros de 180°

Cuadro 3.2: Talleres de aplicación: MATLAB y/o SIMULINK

TEMA	TALLER	DESCRIPCIÓN
CINEMÁTICA	Modelos y restricciones cinemáticas	Simulación del robot diferencial construido. Determinación de su vector de velocidad en base a parámetros iniciales.
DINÁMICA	Robots móviles "Hilare"	Modelado de las ecuaciones dinámicas del robot diferencial.
PERCEPCIÓN	Fundamentos de la visión por computador	<i>Estructura de estéreo</i> : Correlación de puntos en imágenes con cierta disparidad. (Programa recomendado). <i>Seguimiento de color</i> : Detección y seguimiento de colores primarios a través de una WebCam.
	Fundamentos de procesamiento de imagen / Extracción de características de imagen	Detección de bordes y filtros de suavizado en imágenes que el usuario puede escoger.
		Detección de esquinas y manchas en imágenes que el usuario puede escoger.
LOCALIZACIÓN	Construcción autónoma de mapas (EKF SLAM)	Detección de bordes en tiempo real a través de una WebCam. Programas en Simulink.
		Simulación de la localización de un robot mediante 4 puntos de referencia en el mapa. (Programa recomendado).
PLANIFICACIÓN Y NAVEGACIÓN	Planificación de ruta (Búsqueda gráfica)	<i>Construcción de gráfico</i> : Construcción de diagramas aleatorios de Voronoi en 2D y 3D. <i>Búsqueda de gráfico determinista</i> : Simulación del desplazamiento de un robot en un espacio dado desde un punto de inicio a un objetivo usando el algoritmo de "búsqueda primero en amplitud".
	Evasión de obstáculos	<i>Algoritmo de Bug</i> : Simulación que permite establecer el punto de partida y el objetivo, así como los obstáculos en el camino que un robot móvil debe circunnavegar. (Programa recomendado).

- Subida de rampa
- Seguimiento de pared
- Regulación de velocidad del robot en aproximación a objetos

Más información sobre las especificaciones del robot, su programación y sus rutinas se pueden encontrar en el *Anexo C*. En la figura 3.4 se puede observar al robot LEVI desarrollado.

Figura 3.2: Entorno gráfico de la simulación cinemática en MATLAB

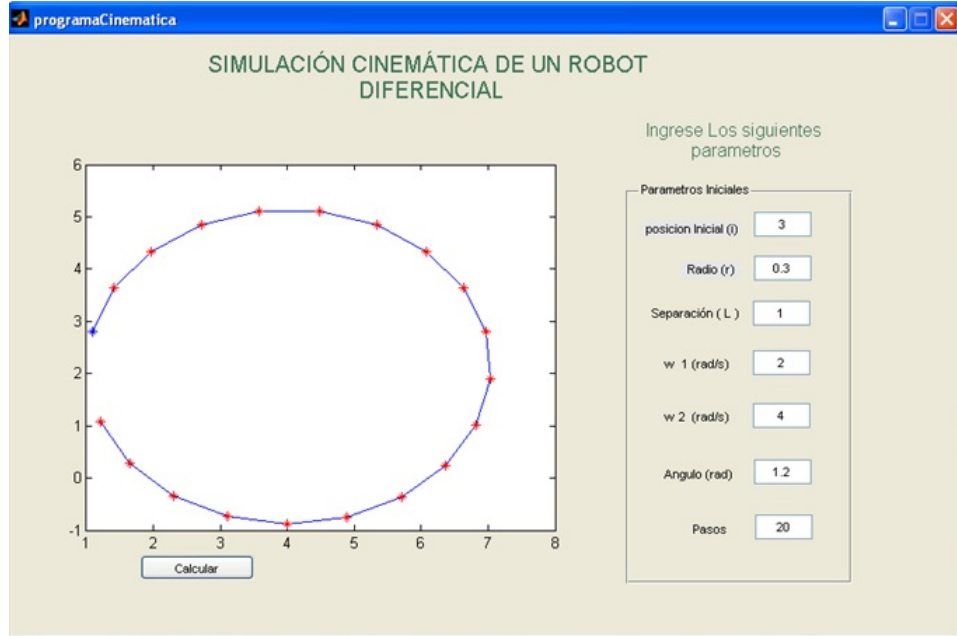


Figura 3.3: Código en MATLAB del taller de simulación cinemática

```

Editor - G:\programaCinematica.m
Stack: Base fx
191 % --- Executes on button press in pushbutton1.
192 function pushbutton1_Callback(hObject, eventdata, handles)
193 % hObject handle to pushbutton1 (see GCBO)
194 % eventdata reserved - to be defined in a future version of MATLAB
195 % handles structure with handles and user data (see GUIDATA)
196
197 clc
198
199 r=str2double(get(handles.edit1,'String'));
200 r1=str2double(get(handles.edit7,'String'));
201 l=str2double(get(handles.edit2,'String'));
202 w1=str2double(get(handles.edit3,'String'));
203 w2=str2double(get(handles.edit4,'String'));
204 teta=str2double(get(handles.edit5,'String'));
205 pasos=str2double(get(handles.edit6,'String'));
206
207 x=zeros(1,pasos);
208 y=zeros(1,pasos);
209
210 x(1,1)=r*cos(teta);
211 y(1,1)=r*sin(teta);
212
213 for ind=2:pasos
214
215     m1=[cos(teta), -sin(teta), 0; ...

```

Figura 3.4: Robot LEVI



---

---

# CAPÍTULO 4

---

## FINALIZACIÓN

Al finalizar todas las sesiones del seminario, se procedió a elaborar el informe final con todas las normas y técnicas establecidas; para ello se acudió a las memorias de cada una de las sesiones (protocolos). A continuación se describe la documentación obtenida del seminario de robótica móvil autónoma.

### 4.1. PRESENTACIONES

Las presentaciones son una guía para el expositor en cada sesión ya que le permiten desarrollar el tema con gran facilidad, permitiendo también que el público comprenda con claridad el tema tratado. Estas presentaciones consistieron en una serie de diapositivas realizadas en PowerPoint y presentan a los integrantes del seminario, institución universitaria, escuela de estudios, tema de estudio y su contenido correspondiente, y al final sus referencias bibliográficas. La figura 4.1 muestra un ejemplo de inicio de sesión y la figura 4.2 es un ejemplo de las diapositivas utilizadas.

### 4.2. DOCUMENTO

Se recopiló toda la información obtenida en este seminario en un documento. Este documento fue redactado en Textmaker bajo la norma ISO-8859-1. Existen diferentes editores  $\text{\LaTeX}$  pero la razón por la cual se usó Textmaker es sencilla: Es un editor con un entorno amigable para el usuario y por lo tanto fácil de usar, además es un editor

Figura 4.1: Ejemplo de diapositiva en el inicio de una sesión



Universidad Industrial de Santander

CONSTRUIMOS FUTURO

**DICBot**  
Centro de Investigación en Dinámica, Control y Robótica

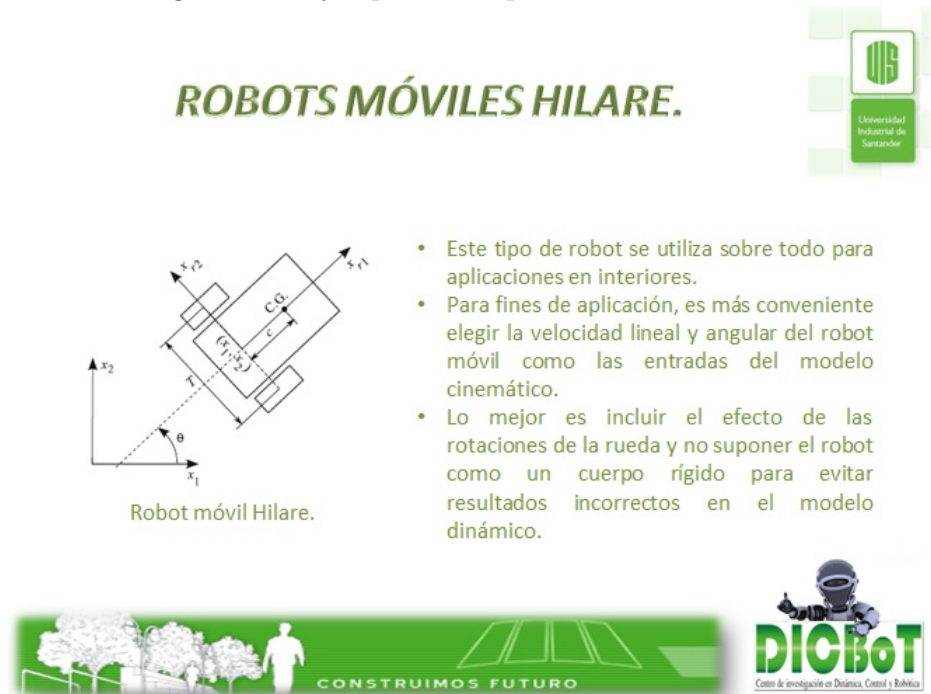
## SEMINARIO DE INVESTIGACIÓN EN ROBÓTICA MÓVIL AUTÓNOMA

Escuela De Ingeniería Mecánica

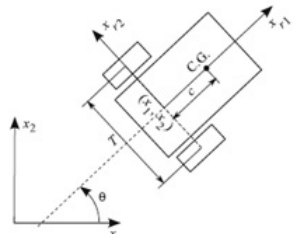
Jonathan Fabian Niño Castillo  
Gary Alejandro Sánchez Amaya  
Daniel Ricardo Carrillo Duarte

Sesiones 7 y 8

Figura 4.2: Ejemplo de diapositiva en las sesiones



### ROBOTS MÓVILES HILARE.



Robot móvil Hilare.

- Este tipo de robot se utiliza sobre todo para aplicaciones en interiores.
- Para fines de aplicación, es más conveniente elegir la velocidad lineal y angular del robot móvil como las entradas del modelo cinemático.
- Lo mejor es incluir el efecto de las rotaciones de la rueda y no suponer el robot como un cuerpo rígido para evitar resultados incorrectos en el modelo dinámico.

Universidad Industrial de Santander

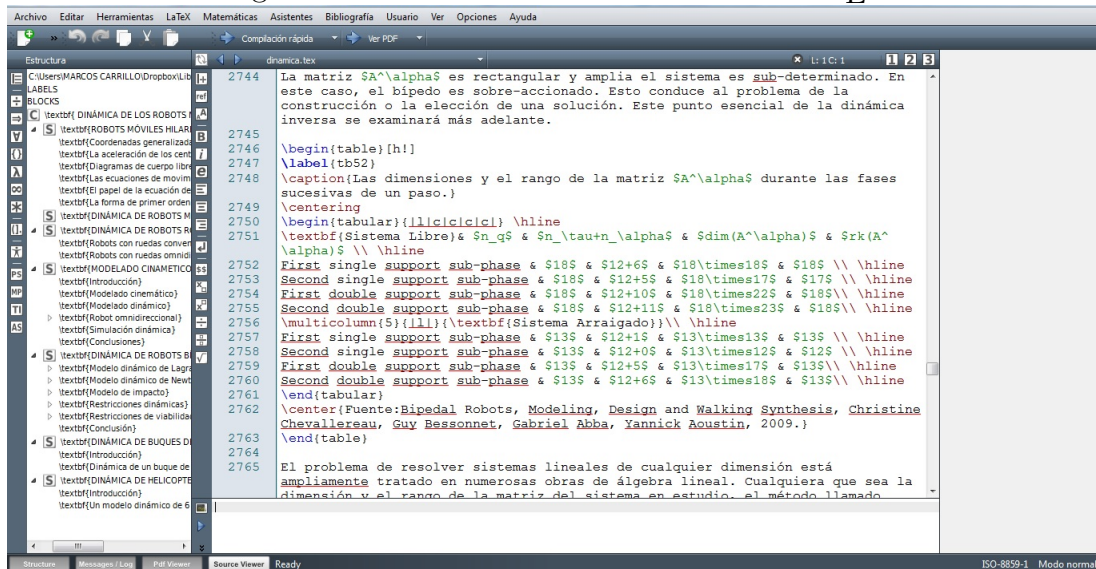
CONSTRUIMOS FUTURO

**DICBot**  
Centro de Investigación en Dinámica, Control y Robótica

gratuito. Su versión de 32 bit es compatible en sistemas Windows de 64 bit. En la figura 4.3 se puede observar el entorno de trabajo de este editor.

$\text{\LaTeX}$  es un sistema de composición de textos muy utilizado para la composición de artículos académicos, tesis y libros técnicos, dado que la calidad tipográfica de los documentos realizados es comparable a la de una editorial científica de primera línea. De ahí nuestra necesidad de usarlo para compilar los resultados de la investigación.

Figura 4.3: Textmaker: Editor de texto en  $\text{\LaTeX}$ .



## **Parte II**

# **MEMORIAS Y RESULTADOS**

---

---

# CAPÍTULO 5

---

## INTRODUCCIÓN A LOS ROBOTS MÓVILES

La humanidad a lo largo de su historia, se ha dado a la tarea de desarrollar mecanismos o aparatos que realicen acciones por que faciliten su día a día. Estos mecanismos fueron llamados inicialmente por los griegos autómatos, palabra que dio origen a la actual autómatas, que designa a maquinas o mecanismos que son capaces de imitar las acciones y gestos de los seres vivos. Lo que en épocas antiguas como en la edad media, se daba, creando autómatas solo por una motivación intelectual más que científica, hoy día se ha transformado en una increíble revolución en donde día a día se avanza a pasos agigantados, desarrollando estos mecanismos, desde principios del siglo XX denominados robots.

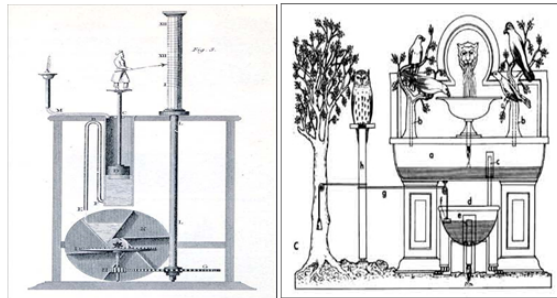
### 5.1. HISTORIA

A lo largo de toda la historia, el hombre se ha sentido fascinado por máquinas y dispositivos capaces de imitar las funciones y los movimientos de seres vivos. Los griegos tenían una palabra específica para denominar a estas máquinas: autómatos. De esta palabra deriva la actual autómatas: máquina que imita la figura y movimientos de un ser animado.

En el siguiente documento citaremos algunos ejemplos de los mecanismos automáticos más relevantes creados por el hombre a lo largo de la historia, principalmente occidental, y de los que se tienen noticias ciertas de su creación y aplicación. Los pri-

meros mecanismos de los cuales se tienen noticias son los de Ctesibius en el año 270 a.C fue un gran ingeniero griego de Alejandría y aplico sus conocimientos de neumática e hidráulica para hacer los primeros relojes de agua mejor conocidos en su época como Clepsidras, también fabrico órganos con figuras en movimiento. Herón de Alejandría (62 a.C), se guio por el trabajo de Ctesibius y creo los teatros automáticos, en los cuales las figuras funcionaban mediante mecanismo que se basaban en dispositivos hidráulicos, poleas y palancas con fines principalmente de aprendizaje. En la figura 5.1 se pueden observar a la izquierda el reloj hidráulico de Ctesibius y a la derecha las pilas para beber agua de Herón de Alejandría

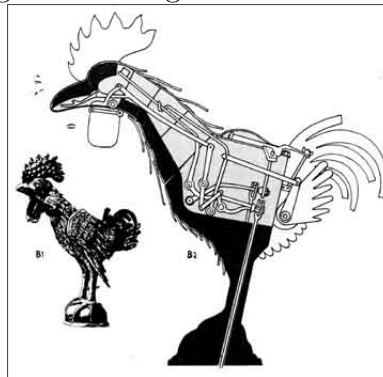
Figura 5.1: Reloj hidráulico de Ctesibius, y pilas para beber agua de Herón de Alejandría



Fuente: Fundamentos de robótica, Antonio Barrientos 2007

Se conoce que El Gallo de Estrasburgo de 1352 es el autómata más antiguo que aún se conserva actualmente, se encontraba en el reloj de la catedral de Estrasburgo y al dar la hora movía las alas el pico y cacareaba tres veces.

Figura 5.2: El gallo de Estrasburgo



Fuente: Fundamentos de robótica, Antonio Barrientos 2007

Durante los siglos XV y XVI s conocen invenciones como el del León mecánico que

Leonardo Da Vinci hizo para el Rey Luis XII de Francia, el autómata se abría el pecho con su garra y mostraba el escudo de armas del rey.

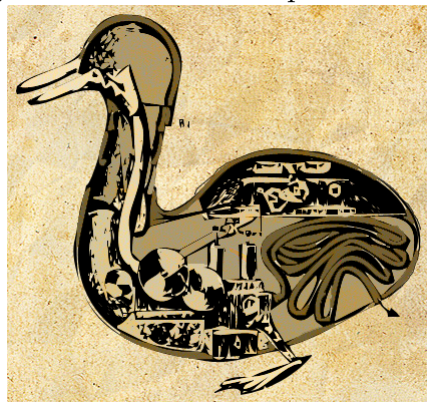
Figura 5.3: León mecánico de Da Vinci



Fuente: Fundamentos de robótica, Antonio Barrientos 2007

Para mencionar un ejemplo bastante popular en esa época y aun actualmente, tenemos el Pato de Jaques Vaucanson (1709-1782), este era un autómata hecho en cobre que representaba un pato que era capaz de realizar las funciones básicas de alimentación y de interacción con el exterior. De hecho, era capaz de deglutir comida, digerirla y defecar; Vaucanson también fue autor de otros logros como el primer telar mecánico, y varios muñecos tales como un flautista y un tamborilero capaces de ejecutar varias melodías programadas.

Figura 5.4: Pato de Jacques Vaucanson



Fuente: <http://www.actuallynotes.com/>

Años más tarde Pierre Jaques Droz (1721-1790) posiblemente el mejor y más conocido creador de autómatas de la historia, suizo nacido en 1721, es el responsable de

los tres autómatas más complejos y famosos del siglo XVIII. Sus tres obras maestras (La Pianista, El Dibujante y El Escritor) causaron asombro en la época llegando a ser contemplados por reyes y emperadores tanto de Europa como de China, India o Japón. El primero de ellos, “La Pianista” como se ve en la figura 5.5, es un autómata con forma de mujer que toca el órgano, con la particularidad de que es la propia figura la que interpreta las obras pulsando las teclas con sus dedos sin tener el sonido pregrabado o procedente de otro lugar.

Figura 5.5: La pianista y el escritor de Pierre Jaquet Droz



Fuente: <http://heliopolisblog.wordpress.com/>

Luego en Japón más exactamente en 1796, Yoriano (hanzo) Hosokawa emitió tres volúmenes de un manual titulado “Karakurizui” que se puede traducir “aparatos mecánicos” para producir sorpresa en las personas, estos volúmenes eran bocetos de autónomas en los cuales se describía la construcción y el funcionamiento de un muñeco automático que servía el té, inclinaba la cabeza y luego se iba al haber efectuaba las los acciones anteriores, se dice que se construyeron centenares de estos autómatas.

Este kakakuri o dispositivo mecánico es quizá el antecesor más directo de los actuales vehículos guiados autónomos (AGV por sus siglas anglosajonas), pero con la particularidad de que su aplicación venía sustentadas por una motivación exclusivamente artístico-intelectual, y en ningún caso con el objetivo de aliviar, descargar o ayudar en el trabajo al ser humano (figura 5.6).

El ingeniero español Leonardo Torres Quevedo invento en 1912 una máquina que era capaz de jugar un conjunto limitado de partidas de ajedrez y fue una de las primeras máquinas, no solo capaces de manejar la información, sino también utilizarla para tomar decisiones y el mecanismo se muestra en la figura 5.7. Leonardo Torres Quevedo fue además, uno de los precursores de los vehículos teledirigidos o teleoperados (el Telekino), y el inventor de diversas maquinas algebraicas, convirtiéndose en uno de los precursores

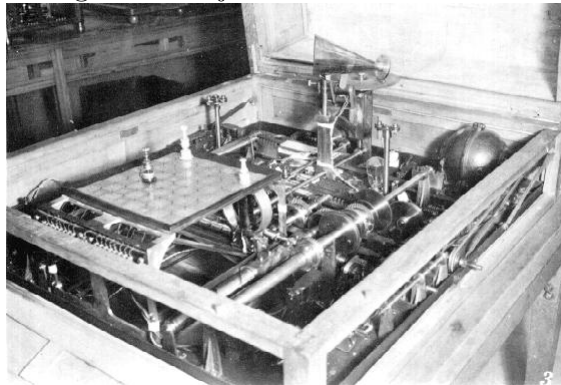
Figura 5.6: Karakuri



Fuente: Fundamentos de robótica, Antonio Barrientos 2007

y fundadores del concepto y disciplina de la actual automática.

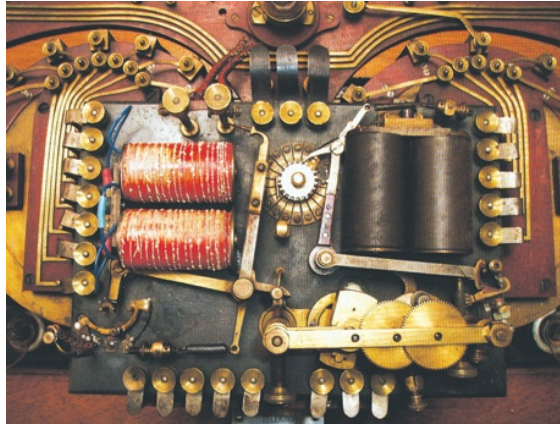
Figura 5.7: Jugador de ajedrez de Leonardo Torres Quevedo



Fuente: Fundamentos de robótica, Antonio Barrientos 2007

El Telekino, (figura 5.8) era un autómata que ejecutaba órdenes a distancia transmitidas mediante ondas Hertzianas. Fue concebido para controlar a distancia barcos y dirigibles. Está considerado como el primer aparato de radio dirección en el mundo, y fue demostrado al rey Alfonso XII en España en 1906 en el puerto de Bilbao dirigiendo un bote desde la orilla.

Figura 5.8: Telekino



Fuente: Fundamentos de robótica, Antonio Barrientos 2007

## 5.2. LA PALABRA ROBOT

La palabra robot fue usada por primera vez en el año 1921, cuando el escritor checo Karel Capek (1890-1938) estreno en el teatro nacional de Praga su obra Rossum's Universal Robot (R.U.R). Su origen es la palabra eslava robota, que se refiere al trabajo realizado de manera forzada. Los robots de R.U.R. eran maquinas androides fabricadas a partir de la formula obtenida por un brillante científico llamado Rossum. Estos robots servían a sus jefes humanos desarrollando los trabajos físicos, hasta que finalmente se rebelan contra sus dueños, destruyendo toda la vida humana, a excepción de uno de sus creadores, con la frustrada esperanza de que les enseñe a reproducirse. De no haber sido por los escritores de ciencia ficción, lo más probable que el termino robot hubiese caído en desuso.

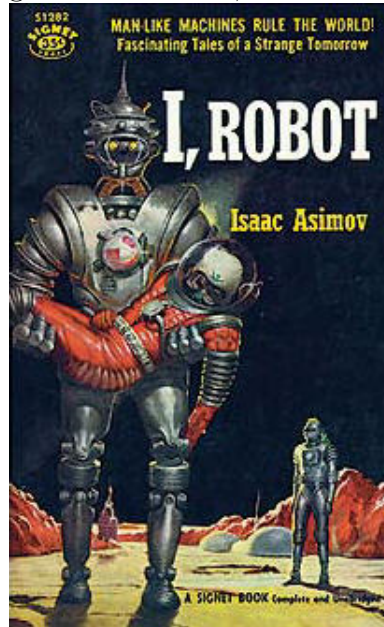
Pero fue el escritor americano de origen ruso Isaac Asimov (1920-1992), el principal impulsor de la palabra robot a quien también se le da crédito por haber creado el término Robotics (robótica) termino que usó en uno de sus pequeños relatos, por ello Asimov fue un personaje decisivo en la divulgación de este término tan común hoy en día. Dentro de las obras de Asimov la más conocida es The Caves of Steel una de sus historias publicada en la revista Galaxy Science Fiction publicada en 1942 en la que por primera vez enuncia las famosas tres leyes de la robótica:

1. Un robot no puede perjudicar a un ser humano, ni con su inacción permitir que un ser humano sufra daños.
2. Un robot ha de obedecer las órdenes recibidas de un ser humano, excepto si tales

órdenes entran en conflicto con la primera ley.

3. Un robot debe proteger su propia existencia mientras tal protección no entre en contacto con la primera o segunda ley.

Figura 5.9: I robot, Isaac Asimov



Fuente: Fundamentos de robótica, Antonio Barrientos 2007

### 5.3. ORIGEN Y DESARROLLO DE LA ROBOTICA

Los primeros dispositivos que responden a lo que hoy se conoce como robot, no adoptaron inicialmente esta denominación. Los progenitores de los robots fueron los manipuladores teleoperados. En 1948 R.C. Goertz del Argonne National Laboratory desarrollo, con el objetivo de manejar elementos radioactivos sin riesgo para el operador, el primer sistema de telemanipulación (figura 5.10).

Años más tarde, en 1954, Goertz hizo uso de la tecnología electrónica y del servo-control sustituyendo la transmisión mecánica por eléctrica y desarrollando así el primer telemanipulador con servocontrol bilateral. Otro de los pioneros de la tele manipulación fue Ralph Mosher, ingeniero de la General Electric que en 1958 desarrollo un dispositivo denominado Handy-Man. Junto a la industria nuclear, a lo largo de los años sesenta la

Figura 5.10: R. C Goertz del Argonne National Laboratory

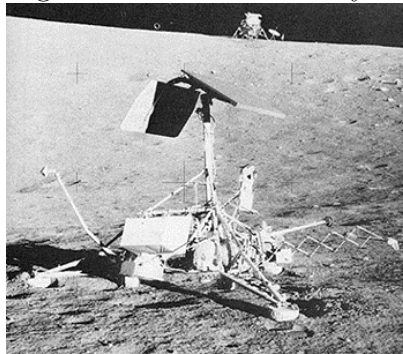


Fuente: Fundamentos de robótica, Antonio Barrientos 2007

industria submarina comenzó a interesarse por el uso de los telemanipuladores. A este interés se sumó la industria espacial en los años setenta.

La sustitución del operador por un programa de ordenador que controlase los movimientos del manipulador dio paso al concepto de robot, espoleado por las necesidades de automatización de las cadenas de fabricación. La primera patente de un dispositivo robótico fue solicitada en marzo de 1954 por el inventor británico C.W. Kenward. En 1954 Devol concibió la idea de un dispositivo de transferencia de artículos programada que se patentó en Estados Unidos en 1961.

Figura 5.11: Sonda Surveyor-3

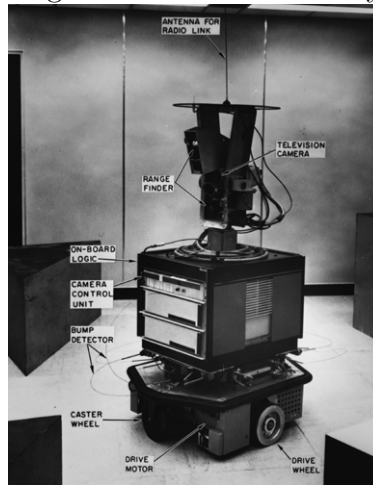


Fuente: <http://observatorio.info>

En 1956 Joseph F. Engelberger es director de ingeniería de la división aeroespacial de la empresa Manning Maxwell y Moore en Stanford, Connecticut. Juntos Devol y Engelberger comenzaron a trabajar en la utilización industrial de sus máquinas, fundando la Consolidated Controls Corporation, que más tarde se convierte en Unimation (Uni-

versal Automation), e instalaron su primera máquina Unimate (1960), en la fábrica de General Motors de Trenton, Nueva Jersey, en una aplicación de fundición por inyección. Otras grandes empresas como AMF, emprendieron la construcción de máquinas similares (Versatran 1960).

Figura 5.12: Robot Shakey



Fuente: <http://boiledbeans.net>

Más tarde, y por motivos comerciales, se empezó a denominar robots a estas máquinas destinadas a transferir piezas de manera versátil o universal, a pesar de que su aspecto no era el humanoide de los robots de la literatura o del cine. Este afortunado cambio de nombre, favoreció notablemente la difusión y aceptación de los robots por parte de la industria, proyectando una imagen de modernismo y avance tecnológico. En 1964 se crean los primeros laboratorios de inteligencia artificial en el MIT, Stanford y la Universidad de Edimburgo, y en 1966 Joseph Weinsenbaum, presenta su programa ELIZA, capaz de manejar un lenguaje rudimentario y mantener una conversación sencilla. La sonda Surveyor-3 se posa sobre la luna en 1967 y utiliza un brazo robótico para la toma de muestras (figura 5.11).

En 1968 J.F. Engelberger visitó Japón y poco más tarde se firmaron acuerdos con Kawasaki para la construcción de robots tipo Unimate. El crecimiento de la robótica en Japón aventaja en breve a los Estados Unidos gracias a Nissan, que formó la primera asociación robótica del mundo, la Asociación de Robótica industrial de Japón (JIRA) en 1972. Dos años más tarde se formó el Instituto de Robótica de América (RIA), que en 1984 cambió su nombre por el de Asociación de Industrias Robóticas, manteniendo las mismas siglas (RIA). En los años finales de la década de los sesenta y setenta se

Figura 5.13: Brazo de Stanford



Fuente: <http://todosobreroobotica.blogspot.com>

ponen las bases de la investigación en robótica en las universidades y se crean robots como el Shakey por el Stanford Research Institute.

También se ponen las bases de los diseños actuales de brazos manipuladores con el diseño del brazo de Stanford por Víctor Scheinman y se observa en la figura 5.13.

Este diseño llevo más tarde al diseño, fabricación y comercialización del robot PUMA, mostrado en la figura 5.14 uno de los robots industriales más famosos y utilizados tanto en ambientes industriales como de investigación.

Figura 5.14: Robot PUMA

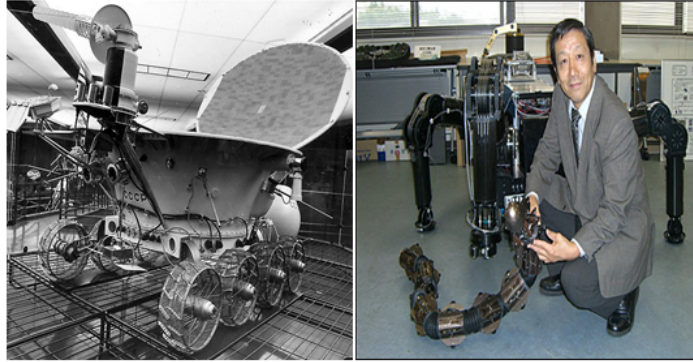


Fuente: <http://metropolis.sci.univr.it>

Mientras tanto en 1970 las URSS es capaz de teleoperar desde la tierra el robot

móvil Lunakhod, mejor conocido como el Tractor Ruso. Shigeo Hirose diseña un robot que imita el movimiento de las serpientes en el año de 1972 (figura 5.15).

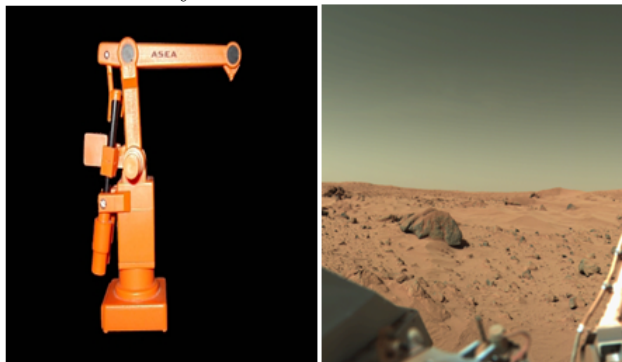
Figura 5.15: Robot móvil Lunakhod y robot que imita a las serpientes



Fuente: Fundamentos de robótica, Antonio Barrientos 2007

Por su parte Europa tuvo un despertar más tardío. En 1973 la firma sueca ASEA construyó el primer robot con accionamiento totalmente eléctrico, el robot IRb6. La empresa alemana KUKA, también fue pionera en el desarrollo de robots industriales en la década de los setenta, siendo una de las primeras en instalar una línea de soldadura equipada únicamente con robots. En 1976 los brazos de las sondas Viking-1 y Viking-2 son teleoperados desde la tierra para tomar muestras del planeta Marte. El brazo robótico se muestra en la figura 5.16

Figura 5.16: Robot IrB6 y brazo robótico a bordo de la sonda Viking-1



Fuente: Fundamentos de robótica, Antonio Barrientos 2007

En 1979, Sankyo e IBM comercializan el primer robot SCARA (Selective Compliance Assembly Robot Arm), desarrollado por el profesor Makino de la Universidad de Yamanashi de Japón.

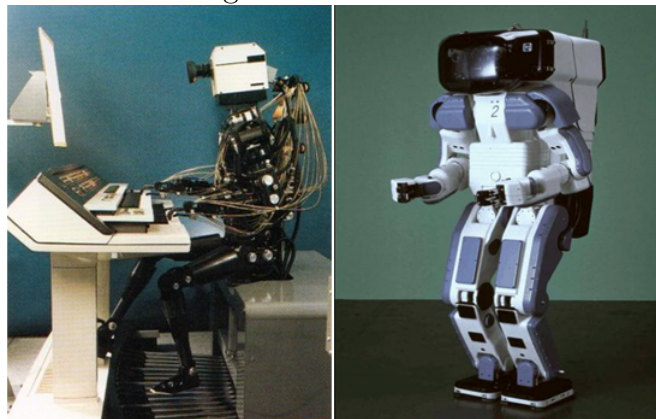
Figura 5.17: Robot Scara



Fuente: <http://roboticlem.blogspot.com>

En 1980 se fundó la Federación Internacional de Robótica con sede en Estocolmo, Suecia. En 1982 el profesor Makino de la Universidad Yamanashi de Japón, desarrolla un concepto de Robot SCARA (Selective Compliance Assembly Robot Arm) que busca un robot con un número reducido de grados de libertad (3 o 4), costo limitado y una configuración desarrollada para el ensamble de piezas como se ve en la figura 5.17. En 1985 se hace la presentación de uno de los primeros robots humanoides desarrollado por el profesor Ichiro Kato de la Universidad de Waseda, capaz de tocar el piano, el robot se llamó WASUBOT (figura 5.18).

Figura 5.18: Wasubot

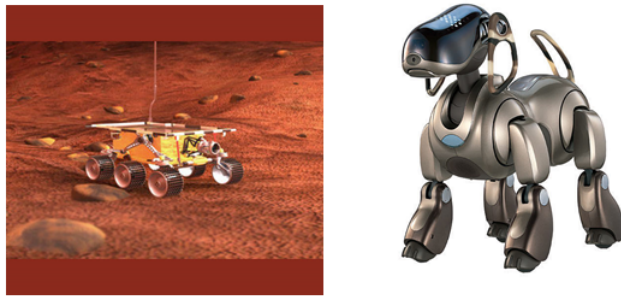


Fuente: <http://usefulnessity.blogspot.com>

Más recientemente en el año de 1995 se lleva a cabo la primera cirugía en un humano utilizando un robot. Durante esta década crece en Japón un interés impactante hacia

el desarrollo de nuevos Robots que puedan imitar más movimientos a los que podemos ejecutar los humanos es por eso que en 1996 Honda presente el robot humanoide P-2, capaz de caminar por sí solo y realizar algunos movimientos básicos como subir escaleras, un año más tarde es lanzada la versión P-3 de este robot con mejoras en su peso y autonomía, ese mismo año el robot móvil Sojourner recorrió más de 100 metros sobre la superficie de Marte siendo teleoperado desde la tierra. Para final de siglo el pionero de los robots para entretenimiento basado en formas animales fue Sony, con su robot mascota AIBO, que fue mostrado al público en junio de 1999 (figura 5.19).

Figura 5.19: a) Robot móvil Sojourner. b) Robot mascota AIBO de Sony



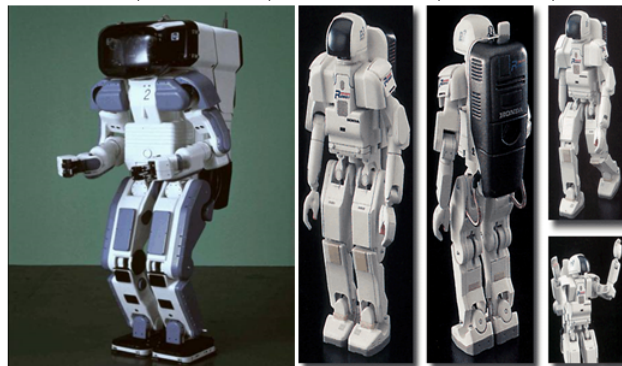
Fuente: Fundamentos de robótica, Antonio Barrientos 2007

Durante la nueva década se han creado una serie de robots con habilidades de mucha mayor complejidad no solo en cuanto a su movimiento sino también al punto de poder expresar sentimientos comunes tales como la ira o la tristeza. En el 2002 Honda presenta a ASIMO, una versión evolucionada de los robots humanoides P-2 y P-3, y en ese mismo año uno de los hitos de la historia de la robótica es su introducción masiva a los hogares en el 2002 a través de la aspiradora autónoma denominada ROOMBA y desarrollada por la empresa iRobot. (figuras 5.20 y 5.21).

En 2005 la compañía Nipona Sohgo Security Services Co, presentó su nuevo robot de seguridad llamado Guardrobo D1, con 109 centímetros de altura, equipado con una cámara fotográfica y sensores. Guardrobo D1 es capaz de patrullar oficinas, centros comerciales y bancos. En caso de presentarse alguna emergencia, se pondrá en contacto vía radio y enviara el video de los hechos (figura 5.22).

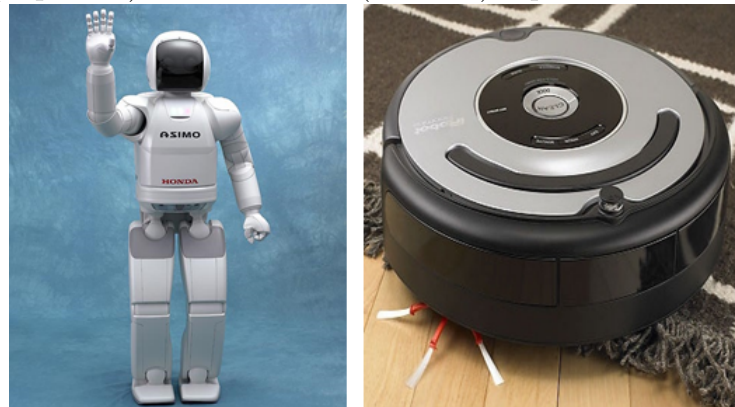
También en 2005 es lanzado Nuvo, Mide 39 cm de alto y pesa 2.5 kilogramos (figura 5.23). Fue sido diseñado para que interactúe en el hogar y, por tanto, se le han añadido sistemas de reconocimiento de voz, que le permitirán ejecutar acciones dictadas verbalmente, y funciones curiosas como la capacidad de bailar o emitir luces de colores. Nuvo incorpora una cámara de video en su cabeza que, a través de una conexión telefónica,

Figura 5.20: (Izquierda) robot P2. (Derecha) robot P3



Fuente: <http://www.hondanews.com>

Figura 5.21: (Izquierda) robot ASIMO. (Derecha) aspiradora autónoma ROOMBA



Fuente: <http://www.hondanews.com>

que permite contemplar lo que observen “sus ojos” en la pantalla de nuestros teléfonos móviles, pudiendo ver en todo momento lo que ocurre en nuestra casa.

Como si un robot androide fuera poco en 2007 se da a conocer el Toyota’s “Violin Playing Robot”, en una exposición en Diciembre en Tokio, Japón. La división de robótica de Toyota presento un robot bípedo de poco más de un metro y medio de altura, que es capaz de interpretar piezas musicales en violín. El robot músico es capaz incluso de imitar el gesto necesario para producir un “Vibrato”. Toyota afirma que este robot tiene también un gran futuro como ayudante en el hogar. Lo que más me sorprende es la movilidad, soltura, precisión y naturalidad de esta creación de Toyota (figura 5.24).

A la par con los robots de uso doméstico y de seguridad, los robots de entretenimiento están tomando gran auge, en 2007 una empresa japonesa creó un robot capaz de dirigir

Figura 5.22: Guardrobo D1



Fuente: <http://informativoinnovacion.blogspot.com>

Figura 5.23: Robot NUVO



Fuente: <http://informativoinnovacion.blogspot.com>

orquestas. Con nombre Vinegar, mostrado en la figura 5.25 dispone de 24 partes móviles en la parte superior de su cuerpo. Todavía no se conoce su finalidad, pero se estima que simplemente sea de entretenimiento.

Para 2009 los avances de la robótica van más allá de la habilidad motora y se muchos quieren enfocarse más en la expresividad de los mismos por esta razón robots como Kobian podrían ser más comunes de lo que pensamos para los siguientes años, los robots en si no tienen sentimientos, pero Kobian es el mejor robot que se hizo hasta ahora encargado de simular algunos sentimientos humanos. Kobian (2009) es capaz de

Figura 5.24: Robot violinista



Fuente: <http://www.neoteo.com/>

Figura 5.25: Robot Vinegar

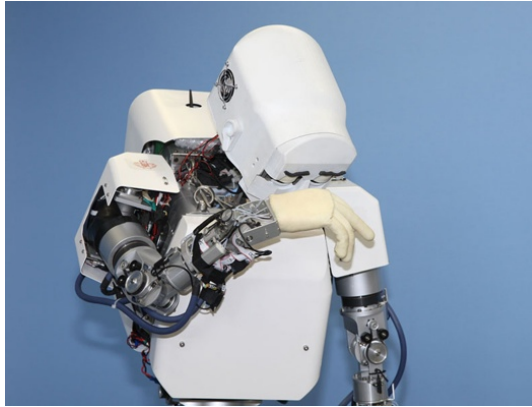


Fuente: <http://www.topblogposts.com>

mostrar hasta siete emociones, con la alegría, enojo, felicidad e incluso confusión entre ellas (figura 5.26).

Siguiendo en 2009 es lanzado HRP-4C, lejos de ser un robot de ayuda o competir con Asimo, es un moda-bot desarrollado para la industria del entretenimiento (por eso su estilo anime) que se presentó en Instituto Nacional de Tecnología y Ciencia Industrial Avanzada de Tokio. Con aspecto de una niña este humanoide cuenta con 42 motores de movimiento programado para imitar los movimientos de los modelos de carne y hueso. Tiene 1.58 metros de altura (la altura media de las mujeres japonesas de 19 a 29 años) y pesa 43 kilogramos (figura 5.27).

Figura 5.26: Robot Kobian



Fuente: <http://www.topblogposts.com>

Figura 5.27: HRP-4C



Fuente: <http://www.xatakaciencia.com/robotica>

Mientras en occidente todavía nos preguntamos si los robots tendrán lugar en las casas, los japoneses parecen divertirse haciendo que sus Bots hagan prácticamente lo que sea. El pequeño Manoi Baila Break Dance y se muestra en la figura 5.28.

En los últimos años, Hiroshi Ishiguro se ha convertido en el gran padre de los robots con apariencia humana. Primero presentó un clon de sí mismo, el primero de la serie Geminoid (figura 5.29). Después, a otra de sus creaciones, Geminoid F, con el aspecto de una mujer real, que incluso ha co-protagonizado una obra teatral en Japón. Ahora ha llegado el turno de Geminoid-DK, el clon robótico más realista jamás construido. Desarrollado en base a la apariencia física del profesor danés Henrik Scharfe, este androide tiene tal nivel de detalle en sus gestos faciales, que será objeto de estudio en lo relacionado a la interacción humana con robots.

Figura 5.28: Robot Manoi



Fuente: <http://www.newlaunches.com/archives>

Figura 5.29: Diseñador Hiroshi Ishiguro con su clon robótico Geminoid



Fuente: <http://spectrum.ieee.org/robotics/humanoid>

En la actualidad, Hanson Robotics desarrolla y fabrica robots humanoides y un software inteligente que llevar a nuestros robots a la vida. Robots Hanson incluye el primer robot bípedo expresivo, Albert Hubo, anunciada por WIRED como “El genio”, y el pequeño robot Zeno, un robot infantil de bajo costo que se está convirtiendo en un miembro de la familia (figura 5.31).

## 5.4. LA ROBÓTICA EN COLOMBIA

Aunque es verdad que el país está lejos de una utilización intensa de la robótica, como ocurre por ejemplo en Japón o en Estados Unidos, hay investigación en el tema

Figura 5.30: GEMINOID DK



Fuente: <http://geminoid.dk/>

Figura 5.31: Robots de la empresa Hanson Robotics



Fuente: <http://www.hansonrobotics.com/>

y una incipiente producción de este tipo de máquinas en varios centros académicos e industriales colombianos.

La compañía A1A Visa, con sede en Bogotá, fabrica uno muy llamativo, especializado en vigilancia y seguridad. Los ingenieros lo bautizaron simplemente Robot vigilante, mostrado en la figura 5.32. Este robot puede recorrer un piso de oficinas y percatarse de la presencia de humanos en el área, detectar conatos de incendio u otras emergencias y comunicarlo vía inalámbrica a una central. Está dotado de una cámara de video con zoom 10x y giro horizontal y vertical, y dispone de sensores de temperatura, humo, gases e inundaciones.

Figura 5.32: Robot vigilante



Fuente: <http://www.elspectador.com/tecnologia>

La Escuela Colombiana de Ingenieros Julio Garavito, con sede en Bogotá es un importante epicentro de investigación en robótica, especialmente orientado hacia el área de la salud. Saltaron a la fama hace siete años, cuando implantaron la primera mano artificial creada en Colombia, a la famosa paciente Virgelina Contreras, una niña de 15 años. La mano ha sido mejorada sin pausa desde entonces y recibió elogios en Barcelona, en donde fue implantada recientemente en pacientes españoles de los hospitales San Juan de Dios y Valdehebrón (figura 5.33).

Figura 5.33: Mano artificial



Fuente: <http://www.minci.gob.ve/reportajes>

La robótica móvil es uno de los campos más difíciles de resolver, a juicio de los expertos. La facultad de ingeniería electrónica de la Universidad Javeriana en Bogotá trabaja en este tipo de problemas, con resultados sin duda importantes en el contexto nacional. El robot Limbo, creado por los ingenieros Carlos Felipe Santacruz y Camila Pontón,

es una máquina bípeda capaz de desplazarse como lo hace un ser humano. Madidas es un robot 'borracho' que puede moverse como un individuo ebrio, dando tumbos sin caerse jamás. Amorfo, un robot basado en diseño modular, es capaz de tomar diferentes formas. Cada año, estudiantes de esta facultad se gradúan con artefactos fabulosos que hacen avanzar, poco a poco, la maravillosa ciencia de la robótica colombiana.

### 5.4.1. Universidad Nacional

El Grupo de Plataformas Robóticas se creó en el año 2001 como iniciativa de un grupo de profesores y estudiantes interesados en el tema de la Robótica. Inicialmente se desarrollaron proyectos pequeños y seminarios extra clase sobre diferentes temas concernientes al campo. En septiembre de 2003 el grupo obtiene la primera financiación de sus proyectos de desarrollo, bajo el nombre de “Semillero de Diseño Mecánico para Robots Móviles”. En 2005 transforma su nombre y adquiere así el nombre actual de “Grupo de Plataformas Robóticas”, dedicado al desarrollo de proyectos tanto en robótica móvil como en robótica industrial. A continuación se ofrece una pequeña reseña de robots que este grupo ha desarrollado:

- **UN-PIER.** (Plataforma de Investigación Experimental en Robótica): uno de los primeros robots del grupo, desarrollado por Máximo Roa y Claudia Garzón en II-2001. Se trata simplemente de una plataforma modular diseñada para implementar y probar diferentes tareas encaminadas a dotar de inteligencia nuestros robots.

Figura 5.34: Robot UN-PIER



Fuente: <http://www.unrobot.unal.edu.co>

- **AGV.** (Vehículo Guiado Automáticamente). Dispositivo desarrollado como parte de la tesis de Maestría en Automatización Industrial del codirector del grupo Ricardo Ramírez Heredia: “Desarrollo de un AGV y Evaluación como Medio de Transporte en una Celda de Manufactura Flexible” sustentada y aprobada en junio de 2005. El vehículo se muestra en la figura 5.35

Figura 5.35: Robot AGV



Fuente: <http://www.unrobot.unal.edu.co>

- **UN-ROVER.** Desarrollado por Diego Becerra, Edwin Cruz y Edgar Bautista Cotte en II-2003. El robot se desarrolló como una plataforma robusta para la exploración de terrenos abruptos; su concepción de diseño se basa en gran parte en el Mars Sojourner. Ha sido durante todo este tiempo el robot insignia del grupo (figura 5.36). Actualmente se está buscando el desarrollo de un sistema de navegación robusto para el robot.

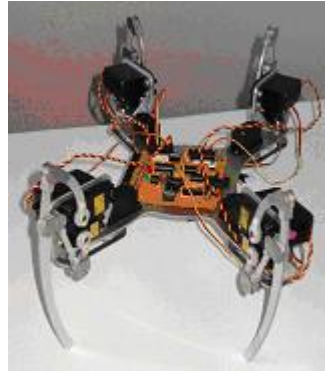
Figura 5.36: UN-ROVER



Fuente: <http://www.unrobot.unal.edu.co>

- **UN-BOT.** Desarrollado por Camilo Villegas, Emerson Rodríguez y Freddy Romano en I-2004. Se trata de un caminador cuadrúpedo que desarrolla una marcha regular, simétrica y estable, aunque un poco lenta (figura 5.37). Actualmente Emerson Rodríguez está trabajando en el rediseño mecánico del robot y en la implementación de una caminata dinámica.
- **UNROCA-III.** Desarrollado por Carlos Ramos y Carlos Parra a lo largo del 2004. Lograron así el primer caminador bípedo del grupo que no requería de una plataforma de apoyo. Pese a que sus pies son anchos, lo cual hace pensar que

Figura 5.37: UN-BOT



Fuente: <http://www.unrobot.unal.edu.co>

la caminata se logra solamente por la gran base de sustentación, en realidad el contacto pie-suelo se da a lo largo de una línea y no de una superficie. El robot se muestra en la figura 5.38

Figura 5.38: UNROCA-III



Fuente: <http://www.unrobot.unal.edu.co>

#### **5.4.2. Universidad del Valle**

El Grupo de investigación en Control Industrial-GICI, es un grupo de la Escuela de Ingeniería Eléctrica y Electrónica de la Universidad del Valle, escalafonado en la categoría B de Colciencias en el 2010. Cabe destacar unos de los proyectos realizados

por este grupo el cual es un robot SCARA de características industriales de cuatro grados de libertad con una herramienta tipo electroimán, con su unidad de potencia, también posee un robot educativo Hero, mesa de dos DOF (grados de libertad) y un dispositivo cooperativo para un robot.

Figura 5.39: Robot SCARA 4DOF



Fuente: <http://gici.univalle.edu.co/Proyectos.html>

### 5.4.3. Pontificia Universidad Javeriana

El grupo de investigación SIRP (Systems, Intelligent, Robotics and Perception) tiene como objetivo Implementar sistemas inteligentes tales como robots y sistemas de percepción, para cumplir las demandas de la industria y defensa (con fines humanitarios) y uno de sus proyectos más representativos es el robot Arcadio, un robot teleoperado móvil basado en un sistema de tracción diferencial que le permite desplazarse en un terreno desigual. La movilidad del robot se complementa con la capacidad de un manipulador de seis GDL, cuenta con seis cámaras, algunas de ellas son PTZ, lo que le permite al operador ver las características del terreno alrededor del robot remotamente. El objetivo del robot es manipular objetos sospechosos con el fin de evitar la pérdida de vidas humanas.

Figura 5.40: Robot Arcadio



Fuente: <http://puj-portal.javeriana.edu.co>

## **5.5. LA ROBÓTICA EN LA UNIVERSIDAD INDUSTRIAL DE SANTANDER**

### **5.5.1. Diseño y construcción de una plataforma robótica cooperativa basada en un sistema embebido para la gestión y manejo de inventarios**

Esta idea nace del grupo de Investigación de Electrónica y Robótica Aplicada ERA, al sentir la necesidad de estudiar más a fondo el uso de sistemas cooperativos automatizados, dirigida a entornos empresariales en donde se manejen sistemas de gestión de mercancías o inventarios, como por ejemplo: hospitales, cadenas de restaurantes, bibliotecas, bodegas, etc.

### **5.5.2. Diseño y construcción de un prototipo de robot móvil autónomo de exploración, 2009**

El proyecto planteó el diseño y construcción de una plataforma mecánica, hardware y software de control, para el desarrollo y crecimiento teórico y práctico en el área de la robótica móvil en estudiantes de pregrado como un incentivo para seguir desarrollando y generando más ideas que impulsen el desarrollo tecnológico en Colombia. Como resultado se construyó un robot móvil autónomo que puede desplazarse por terrenos irregulares y texturas cambiantes para ambientes exteriores poco agrestes.

Figura 5.41: Versión final del robot



Fuente: Autores del proyecto mencionado

### **5.5.3. Robot cartesiano neumático para el laboratorio de sistemas mecatrónicos de la escuela a de ingeniería mecánica. diseño y construcción, 2010**

Figura 5.42: Robot cartesiano



Fuente: Autores del proyecto

Este proyecto surge con la necesidad de fortalecer la formación de los estudiantes de ingeniería mecánica en el área de automatización industrial, dando la posibilidad de llevar a la práctica los conocimientos adquiridos por los estudiantes en asignaturas a lo largo de su carrera. Por ellos se diseñó y construyó un sistema automático de control Robot/manipulador cartesiano con tres grados de libertad y alimentación neumática.

#### 5.5.4. Diseño y construcción de un helibot 3 dof como herramienta de investigación y desarrollo para el laboratorio de sistemas dinámicos e ingeniería del control

Figura 5.43: Planta dinámica de pruebas



Fuente: Autores del proyecto

El HELIBOT 3DOF es una plataforma de investigación y desarrollo que permite analizar y validar a futuro las diferentes leyes de control moderno (dominio del tiempo y la frecuencia).

#### 5.5.5. Diseño y construcción de una plataforma de entrenamiento para helicóptero a escala 60

Desarrollada para un helicóptero a escala 60. La plataforma cuenta con cuatro grados de libertad de movimiento (tres de rotación y uno de traslación en el eje z) para monitorear mediante lazos de control abierto el comportamiento de la plataforma y del helicóptero a escala 60 con el uso de sensores de posición angular y desplazamiento. La plataforma tiene la capacidad de medir y almacenar gráficamente variables de interés en el campo de la automatización y la robótica (ángulos de alabeo y cabeceo, el giro en el eje z, y la elevación).

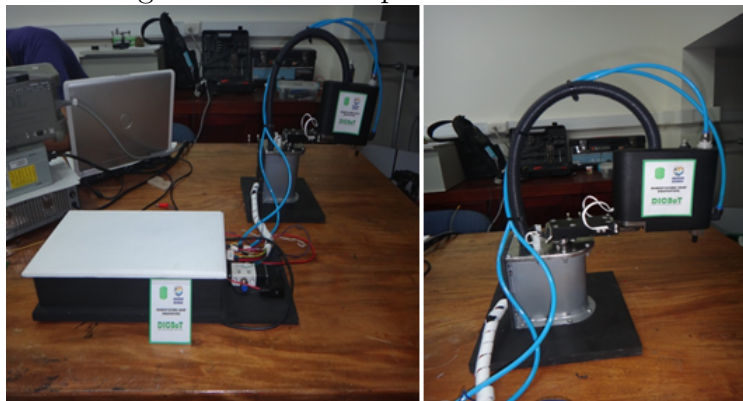
Figura 5.44: Estructura mecánica HeliT60



Fuente: Autores del proyecto

### 5.5.6. Diseño y construcción de un prototipo de robot scara con tres grados de libertad

Figura 5.45: Prototipo de robot SCARA



Fuente: Autores del proyecto

Con la finalidad de contribuir con la misión de la Universidad Industrial de Santander, de generar conocimiento y aportar soluciones útiles a la comunidad mediante la Investigación, concretamente en el área de la robótica se diseñó y construyó un prototipo de brazo robótico SCARA de 3 DOF (grados de libertad) para propósitos experimenta-

les, de accionamiento servoasistido como primera fase para realizar pruebas relacionadas con la cinemática directa e inversa, con la siguientes características:

- Dimensiones generales máximas: 30x25x25 cm.
- Alcance horizontal entre 16 y 20 cm.
- Desplazamiento vertical entre 6 y 10.
- Accionamiento eléctrico utilizando 3 servomotores DC.
- Potencia eléctrica consumida entre 100 y 300 W.

### 5.5.7. Seminario de investigación en robótica industrial

Figura 5.46: Robot AL5B Lynxmotion

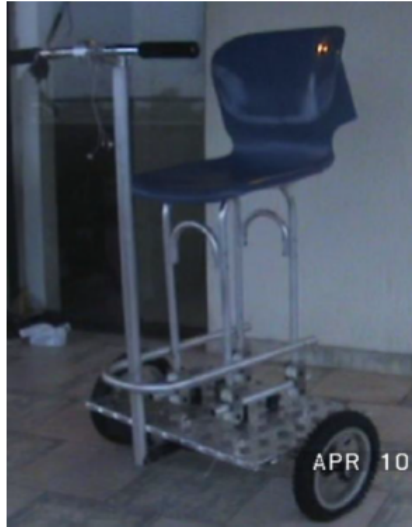


Fuente: Autores del proyecto

Con este proyecto de grado se busca generar una base metodológica común para el seminario de investigación en robótica industrial, de forma que permita desarrollar esta línea de investigación en la Universidad Industrial de Santander y crear a futuro, una asignatura dirigida a los estudiantes de pregrado de la escuela de Ingeniería. El robot que se puede apreciar en la figura 5.46, cuenta con 5 grados de libertad es accionado por servomotores y gobernado por la tarjeta controladora SSC-32 de la misma empresa (Lynxmotion), al cual se le añadió un sexto grado de libertad al brindarle la rotación de la muñeca. El robot AL5B, fue caso de estudio a lo largo del seminario, aplicando varios de los conceptos discutidos en las sesiones.

### 5.5.8. Diseño y construcción de un prototipo experimental de un vehículo eléctrico autobalanceado unipersonal veau

Figura 5.47: Fotografía del modelo final VEAU



Fuente: Autores del proyecto

El vehículo eléctrico autobalanceado unipersonal fue una iniciativa de estudiantes del centro de investigaciones DICBOT que tiene como característica principal el uso de dos ruedas concéntricas (paralelas), una a cada lado, generando en el sistema un estado de equilibrio inestable, el cual es controlado por la implementación de un filtro PID, el sistema automático acciona los motores acoplados a cada una de las ruedas, de acuerdo a la inclinación que el usuario produzca sobre el vehículo.

## 5.6. DEFINICIÓN DE ROBÓTICA

- **Robótica.** La robótica es una ciencia o rama de la tecnología, que estudia el diseño y construcción de máquinas capaces de desempeñar tareas realizadas por el ser humano o que requieren del uso de inteligencia. Las ciencias y tecnologías de las que deriva podrían ser: el álgebra, los autómatas programables, las máquinas de estados, la mecánica o la informática.
- **Robot.** Un Robot es un dispositivo generalmente mecánico, que desempeña tareas automáticamente, ya sea de acuerdo a supervisión humana directa, a través de un programa predefinido o siguiendo un conjunto de reglas generales, utilizando

técnicas de inteligencia artificial. Generalmente estas tareas reemplazan, asemejan o extienden el trabajo humano, como ensamble en líneas de manufactura, manipulación de objetos pesados o peligrosos, trabajo en el espacio, etc.

- **Robótica autónoma.** La Robótica Autónoma es el área de la Robótica que desarrolla robots capaces de desplazarse y actuar sin intervención humana. Para ello el robot debe percibir su entorno y actuar de forma adecuada, además de llevar a cabo su tarea.

## 5.7. CLASIFICACION DE LOS ROBOTS

Existen varios criterios para clasificar los robots, los cuales son presentados a continuación.

### 5.7.1. Según su cronología

La que a continuación se presenta es la clasificación más común.

- **Primera Generación:** Manipuladores. Son sistemas mecánicos multifuncionales con un sencillo sistema de control, bien manual, de secuencia fija o de secuencia variable. Un ejemplo se muestra en la figura 5.48.

Figura 5.48: Robot de primera generación



Fuente: <http://www.robotodosis.com>

- **Segunda Generación:** Robots de aprendizaje. Repiten una secuencia de movimientos que ha sido ejecutada previamente por un operador humano. El modo de

hacerlo es a través de un dispositivo mecánico. El operador realiza los movimientos requeridos mientras el robot le sigue y los memoriza.

Figura 5.49: Robot de segunda generación



Fuente: <http://www.robotodosis.com>

- **Tercera Generación:** Robots con control sensorizado. El controlador es una computadora que ejecuta las órdenes de un programa y las envía al manipulador para que realice los movimientos necesarios.

Figura 5.50: Robots de tercera generación



Fuente: <http://www.robotodosis.com>

- **Cuarta Generación:** Robots inteligentes. Son similares a los anteriores, pero además poseen sensores que envían información a la computadora de control sobre el estado del proceso. Esto permite una toma inteligente de decisiones y el control del proceso en tiempo real.
- **Quinta Generación:** La siguiente generación será una nueva tecnología que incorporara 100% inteligencia artificial y utilizara métodos como modelos de conducta y una nueva arquitectura de subsunción, además de otras tecnologías actualmente en desarrollo como la nanotecnología.

Figura 5.51: Robots de cuarta generación



Fuente: <http://www.robotodosis.com>

Figura 5.52: Robots de quinta generación



Fuente: <http://www.robotodosis.com>

### 5.7.2. Según su morfología

La subdivisión de los Robots, con base en su arquitectura, se hace en los siguientes grupos: Poliarticulados, Móviles, Androides, Zoomórficos e Híbridos.

- **Poliarticulados:** En este grupo están los Robots de muy diversa forma y configuración cuya característica común es la de ser básicamente sedentarios y estar estructurados para mover sus elementos terminales. En este grupo se encuentran los manipuladores, los Robots industriales, los Robots cartesianos y se emplean cuando es preciso abarcar una zona de trabajo relativamente amplia o alargada, actuar sobre objetos con un plano de simetría vertical o reducir el espacio ocupado en el suelo.
- **Móviles:** Son Robots con gran capacidad de desplazamiento, basada en carros o plataformas y dotada de un sistema locomotor de tipo rodante. Siguen su camino por telemando o guiándose por la información recibida de su entorno a través de sus sensores. Algunos ejemplos se aprecian en la figura 5.54.
- **Androides:** Son Robots que intentan reproducir total o parcialmente la forma y el comportamiento cinemática del ser humano. El principal problema es controlar

Figura 5.53: Ejemplos de robots poliarticulados



Fuente: <http://robotiica.blogspot.com>

Figura 5.54: Ejemplos de robots móviles

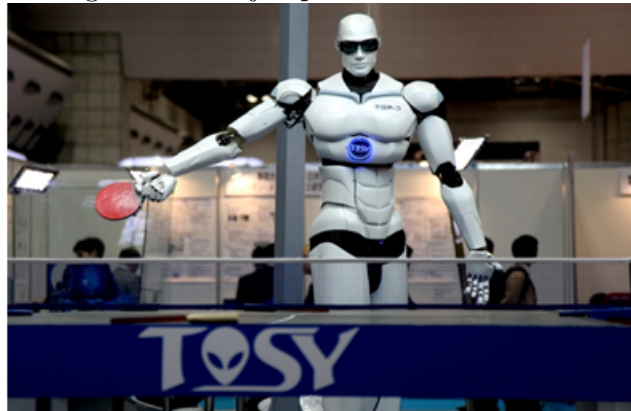


Fuente: <http://robotiica.blogspot.com>

dinámica y coordinadamente en el tiempo real el proceso y mantener simultáneamente el equilibrio del Robot. Un ejemplo se observa en la figura 5.55.

- **Zoomórficos:** Los Robots zoomórficos, que considerados en sentido no restrictivo

Figura 5.55: Ejemplos de robot androide



Fuente: <http://robotiica.blogspot.com>

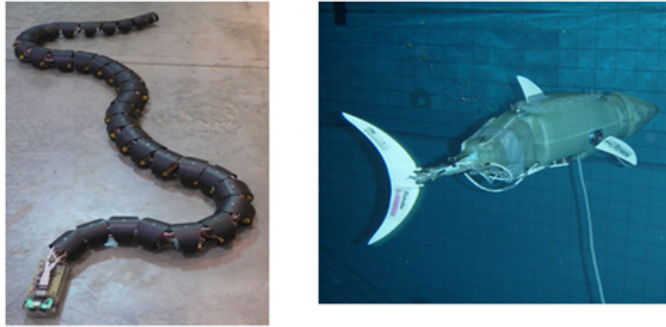
podrían incluir también a los androides, constituyen una clase caracterizada principalmente por sus sistemas de locomoción que imitan a los diversos seres vivos. Las aplicaciones de estos Robots serán interesantes en el campo de la exploración espacial y en el estudio de los volcanes. Algunos ejemplos se aprecian en la figura 5.56.

- **Híbridos:** Corresponden a aquellos de difícil clasificación cuya estructura se sitúa en combinación con alguna de las anteriores ya expuestas, bien sea por conjunción o por yuxtaposición. Por ejemplo, un dispositivo segmentado articulado y con ruedas, es al mismo tiempo uno de los atributos de los Robots móviles y de los Robots zoomórficos. Dos robots de este tipo se ven en la figura 5.57.

### 5.7.3. Según su grado de inteligencia

- **Control manual o Tele-operados:** Son los robots que requieren de un operador humano para realizar la tarea para la cual fueron diseñados. Carecen por si solos de conocimientos de su entorno y sus movimientos dependen exclusivamente de las órdenes dadas por el operador. La comunicación con la estación de control puede hacerse por diferentes tipos de enlace inalámbrico e incluso haciendo uso de redes de telefonía móvil o por medio de un cable como en algunos robots submarinos.
- **Programables:** Carecen de conocimientos de su entorno, tienen un programa que repiten cíclicamente para realizar su función. La mayoría de los robots industriales Poliarticulados y cartesianos son de este tipo. Las trayectorias de movimientos

Figura 5.56: Ejemplos de robots zoomórficos



Fuente: <http://robotiica.blogspot.com>

Figura 5.57: Ejemplos de robots híbridos



Fuente: <http://robotiica.blogspot.com>

son predefinidas y programadas. No responden satisfactoriamente ante cambios inesperados de su entorno.

- **Autónomos:** Poseen los elementos básicos y funciones de un sistema de control retroalimentado con la capacidad de adquirir datos de su entorno por medio de

Figura 5.58: Clasificación de los robots según su grado de inteligencia



Fuente: Autores del seminario de investigación en robótica móvil autónoma

sensores, hacer la adaptación y el procesamiento de las señales provenientes de dichos sensores y emitir señales a los actuadores. El procesador tiene grabado un programa que le permite al robot tomar decisiones en respuesta a las variaciones que le impone su entorno y readaptar sus funciones. No interviene la mano de un operador.

La pirámide en la figura 5.58 esquematiza el desarrollo que ha tenido la robótica.

#### 5.7.4. Según su capacidad de movimiento

- **Estáticos:** La mayoría de robots industriales Poliarticulados y cartesianos son de este tipo. Trabajan en un espacio limitado en el que son instalados, están anclados al suelo y no poseen un mecanismo de locomoción.
- **Móviles:** Estos poseen un mecanismo de locomoción que les permite desplazarse en un espacio determinado para realizar su tarea o misión. Pueden estar diseñados para trabajar en tierra, aire o agua. Pueden ser autónomos, programados o teleoperados.

#### 5.7.5. Según su aplicación

- **Industriales:** Los robots industriales son artilugios mecatrónicos reprogramables destinados a realizar de forma automática determinados procesos de fabricación o manipulación. Son en la actualidad los de mayor uso y producción.
- **Exploración:** Estos robots son utilizados para realizar labores de exploración o

misiones en lugares de difícil acceso para el hombre o que representan un riesgo para su integridad, tales como zonas volcánicas, desiertos, zonas contaminadas, naufragios, exploración planetaria, zonas de desastre (robots ápodos), detección de explosivos, etc. Para poder llevar a cabo su misión deben tener las siguientes características:

- Un tipo de locomoción acorde al medio donde se va a desempeñar.
- Manipuladores y/o herramientas especializadas en las tareas requeridas según la misión. Esto hace que algunos de estos robots puedan ser considerados de morfología híbrida.

En caso de ser autónomos además requieren:

- Sistemas de percepción del entorno, posicionamiento y orientación para realizar su desplazamiento de manera controlada.
- Dispositivos de comunicación que permitan intercambiar información con centrales de supervisión.

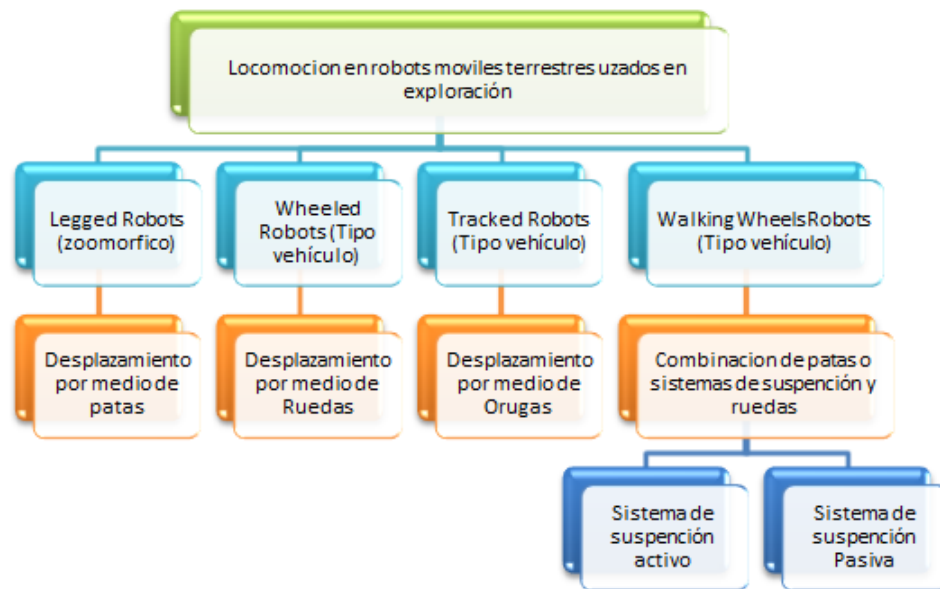
Figura 5.59: Robots de exploración



Fuente: <http://robotiica.blogspot.com>

Se pueden clasificar en “outdoors” si son diseñados para desplazarse en ambientes externos o “indoors” si son para ambientes internos. En el caso de exploración terrestre existen varias opciones en cuanto a la locomoción.

Figura 5.60: Locomoción en robots móviles de exploración terrestre



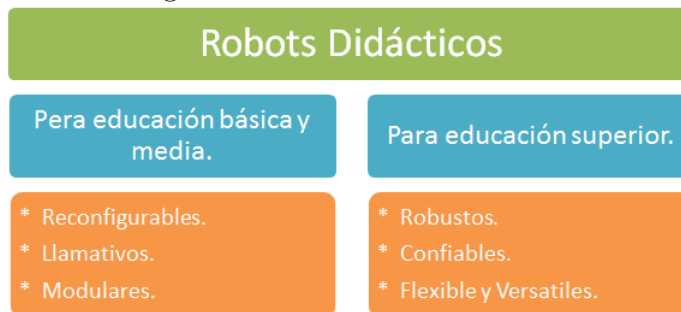
Fuente: Autores del seminario de investigación en robótica móvil autónoma

Los robots con piernas son utilizados en terrenos escarpados, escabrosos, rocosos y con numerosas irregularidades (outdoors). Los robots de oruga son especiales para terrenos arenosos dado que tienen una gran superficie de contacto disminuyendo la presión ejercida sobre el suelo y por consiguiente el hundimiento. Sin embargo, tienen un mayor consumo de potencia debido al deslizamiento que se presenta cada vez que gira.

- **Didácticos:** Son robots utilizados para enseñar la robótica a nivel de educación básica, media y superior. Correspondiendo con estos niveles es posible distinguir dos categorías de robots didácticos y estos se muestran en la figura 5.61.

*Robots didácticos para educación básica y media:* El objetivo de estos robots es introducir a los niños en el mundo de la robótica, de una forma atractiva y divertida, motivarlos para que desde edades tempranas se interesen en este tema. Los robots utilizados son estructuras simples, generalmente móviles, modulares, desarmables y reconfigurables que resultan atractivos por su forma y sus colores y en la mayoría de los casos parecen carros de juguete pero son realmente robots que los niños pueden programar de una forma fácil y didáctica. Existen varias marcas que ofrecen kits de robótica dirigidos a la enseñanza a nivel de educación básica y media. Entre éstas, la más conocida es LEGO® , que tiene infinidad de

Figura 5.61: Categorías de los robots utilizados en la enseñanza



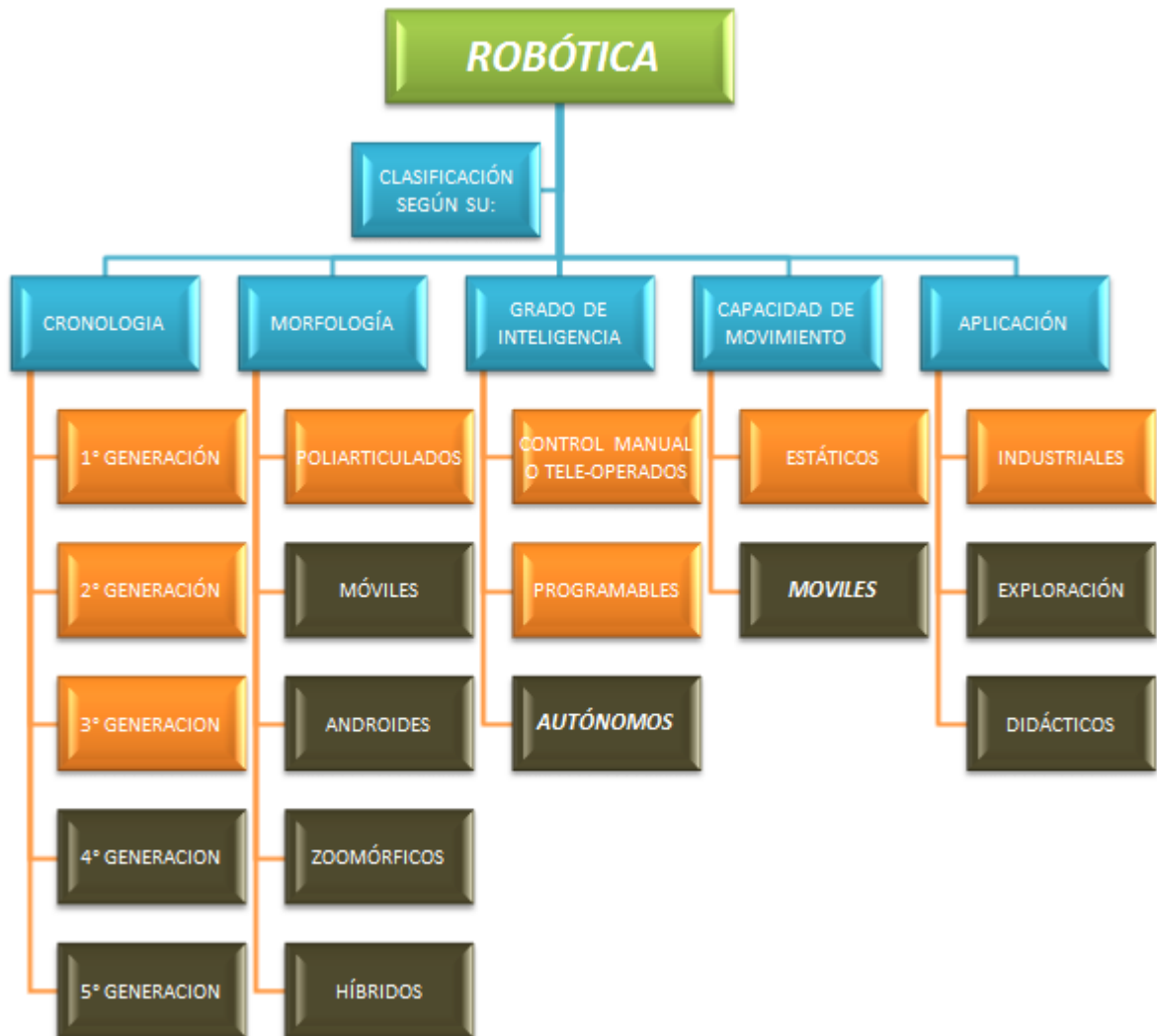
Fuente: Autores del seminario de investigación en robótica móvil autónoma

productos que buscan estimular la creatividad, la resolución de problemas y el trabajo en equipo, así como habilidades y destrezas en los niños.

En resumen, se podría clasificar la robótica de la siguiente manera:

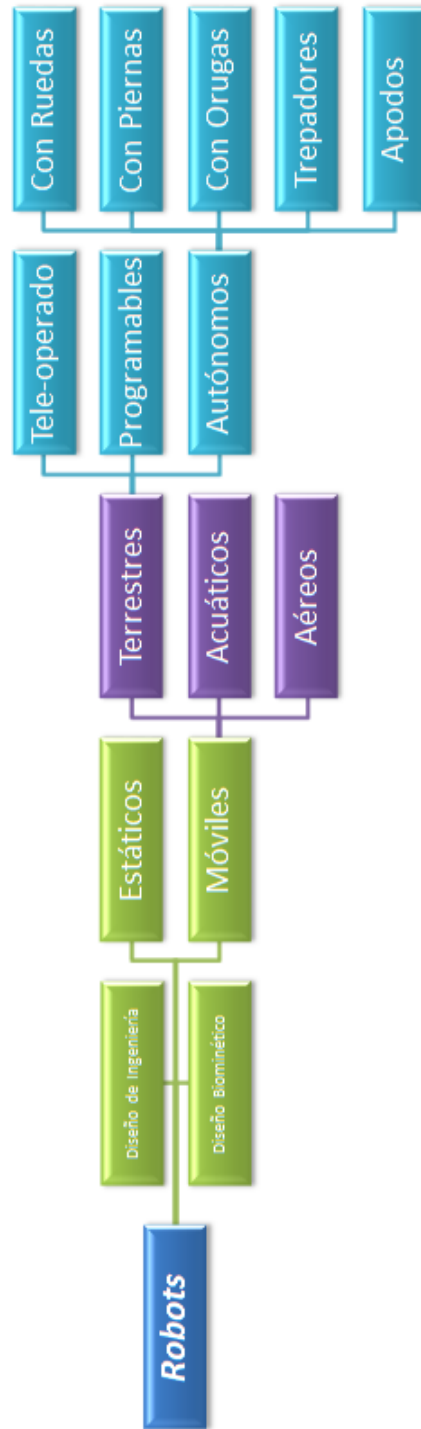
La clasificación de la robótica móvil autónoma, se puede mostrar a través del siguiente mapa conceptual:

Figura 5.62: Mapa conceptual de la robótica



Fuente: Autores del seminario de investigación en robótica móvil autónoma

Figura 5.63: Mapa conceptual de los robots móviles autónomos



Fuente: Autores del seminario de investigación en robótica móvil autónoma

---

---

## CAPÍTULO 6

---

# ESTADO DEL ARTE DE LOS ROBOTS MÓVILES

### 6.1. METODOLOGÍA DE DISEÑO

El desarrollo de cualquier tipo de robot móvil es un desafío y requiere mucho tiempo y consideraciones. Sería posible mejorar la productividad del trabajo de desarrollo mediante la optimización de los métodos y herramientas de diseño. El proceso de diseño de un robot móvil se puede dividir en tres partes lógicas basadas en la arquitectura: software, hardware y mecánicos. El software se divide jerárquicamente en dos partes:

- **Software de alto nivel** que permite que el robot móvil para funcionar de forma autónoma y cumplir con su misión asignada.
- **Software de bajo nivel** incluye las funciones motoras básicas, tales como el algoritmo de manejo, así como otras funciones como las rutinas de comunicación o programas de prevención de colisiones.

### 6.2. LA PERCEPCIÓN Y CONOCIMIENTO DE LA SITUACIÓN

Los términos tales como la percepción y el conocimiento (conciencia) han sido sustituidos por el término más general “cognición”. En la robótica, la cognición incluye

la creación de información de alto nivel de la combinación de piezas de bajo nivel de información y memoria. Existen varios posibles estados de la cognición. Este modelo incluye siete diferentes estados mentales (percepción y adquisición, razonamiento, atención, reconocimiento, aprendizaje, planificación, acción y coordinación), así como dos tipos diferentes de memoria (a corto y a largo plazo). La cognición es este modelo realizado mediante tres ciclos. Estos tres ciclos son el “ciclo de adquisición”, el “ciclo de la percepción” y el “ciclo de coordinación y aprendizaje”. Mientras que la visión depende sobre todo de hardware, el reconocimiento se basa fundamentalmente en técnicas computacionales. Aquí está una lista de técnicas de reconocimiento:

- Igualación de plantilla (TEMPLATE MATCHING): esta técnica se basa en el procesamiento de imágenes, donde la imagen es analizada para encontrar una igualdad con la imagen de la plantilla.
- Modelo basado en características: en lugar de buscar las plantillas, se analiza la imagen para encontrar algunas de las características o patrones. Estas características suelen ser formas geométricas o colores específicos.

## **6.3. CONTROL DE ROBOTS MÓVILES**

### **6.3.1. Control de formación**

Un gran número de diferentes estrategias para controlar la formación de un grupo de robots móviles se pueden encontrar en los libros. Varios marcos destacan por el número de estrategias que han sido desarrollados, incluyendo los esquemas de líder-seguidor, los métodos basados en el comportamiento y las técnicas de estructura virtual. Entre estas tres técnicas, el enfoque de líder-seguidor es el más reconocido: uno o más robots móviles han sido designados como líderes y están a cargo de dirigir la formación, los otros robots no tienen información sobre sus títulos y simplemente siguen al/a los líder(es): llamados seguidores.

En los métodos basados en el comportamiento, los robots tratan de mantener una distancia determinada gracias a la formación de dos procesos complementarios. El método de la estructura virtual utiliza la idea de puntos en el espacio que mantienen una relación geométrica fija y cuyo comportamiento puede ser observado de la misma manera que puntos de un cuerpo rígido que se mueven por el espacio. Cuando los robots se comportan de esta manera, se están moviendo dentro de una estructura virtual.

### **6.3.2. Control de robots terrestres no Holonómicos**

La dinámica de robots No Holonómicos se caracterizan por ecuaciones que involucran las derivadas temporales de las variables del sistema y limitaciones. Estas ecuaciones dinámicas no son integrables. La No-holonomía normalmente se encuentra cuando el sistema tiene menos entradas de control de las variables controladas. Como ejemplo, consideremos un robot con ruedas móviles que tiene dos controles (velocidad lineal y angular), mientras que el dominio en el que se desarrolla es en tres dimensiones. Por lo tanto, cada señal factible de control no corresponde necesariamente a un camino factible para el sistema. Esta es la razón por la que las técnicas basadas en la geometría desarrollada en la planificación de movimiento de los sistemas holonómicos no pueden aplicarse directamente a los no-holonomicos.

### **6.3.3. Control de vehículos aéreos no tripulados.**

Los vehículos aéreos no tripulados suelen ser controlados por un piloto automático, es decir, un sistema que le permita al UAV (Unmanned Aerial Vehicle) volar de manera autónoma. Hoy en día, los sistemas de piloto automático se implementan automáticamente en los aviones modernos. El propósito de los sistemas de piloto automático UAV es dirigir el UAV a fin de seguir una ruta predefinida o volar entre puntos de referencia (Waypoints). Los sistemas avanzados de piloto automático de los UAV son capaces de hacer volar el UAV durante todas las fases de vuelo, tales como el despegue, ascenso, descenso, seguimiento de la trayectoria, y aterrizaje.

## **6.4. ROBOTS MÓVILES**

Estos robots se pueden mover por tierra, aire, bajo el agua o incluso en el espacio exterior, y se mueven por sí mismos, utilizando su propia energía. Gracias a sus sensores y su programación, son capaces de guiarse por el entorno autónomamente, pero sin embargo, no es de extrañar que su funcionamiento esté monitorizado por seres humanos o incluso teledirigido. En este último caso, el robot mantiene una comunicación constante con el operador, ya sea por cable o inalámbricamente. Su morfología es variable dependiendo del entorno y de la tarea a realizar.

### **6.4.1. Aplicaciones de robots móviles**

La capacidad de que una máquina se pueda mover de forma autónoma, es una capacidad que abre enormemente el rango de aplicaciones de la robótica. Los robots

móviles son importantes por su capacidad de desplazarse donde las personas no pueden, o donde las condiciones son inadecuadas. Para hablar de las aplicaciones de los robots móviles tendremos que referirnos y tratar de definir que son los robots de servicio, cuál es su principal función, que tareas realizan y donde radica su principal uso.

## **Robots de servicio**

Así como ocurre con el término “Robot” no existe una definición absoluta que nos pueda describir que es un robot de servicio. El Fraunhofer IPA: Fraunhofer-Institut für Produktionstechnik und Automatisierung (Instituto para la Ingeniería de Manufacturas y Automatización) establece que un robot de servicio es:

**“Un dispositivo móvil programable, que desarrolla servicios de manera total o parcialmente automática; entendiendo por servicios a aquellas tareas que no sirven directamente a la fabricación de bienes, si no a la realización de servicios a las personas o equipos”**

En la práctica las, actuales y potenciales aplicaciones no industriales de los robots son tan variadas y diferentes, que es difícil encontrar una definición concreta de lo que realmente es un robot de servicio, aun mas con la inmensa cantidad de aplicaciones, que en muchos casos son únicas y hacen difícil su catalogación y conocimiento. La IFR también ha propuesto una clasificación para tratar de establecer una primera división de estas aplicaciones no industriales de los robots:

- Aplicaciones de servicios a humanos (personal, protección, entretenimiento).
- Aplicaciones de servicio a equipos (mantenimiento, reparación, limpieza).
- Otras funciones autónomas (vigilancia, transporte, adquisición de datos, inspección, etc).
  - Vigilancia y seguridad.
    - Sector militar.
    - Inspección y desactivación de bombas.
  - Inspección de grandes estructuras.
  - Sector Minero.
  - Sector Nuclear.
  - Submarinos.
- Espacio y exploración planetaria.

- Medicina y Salud
  - Ayuda a discapacitados.
  - Sector quirúrgico.
  - Prótesis.
- Ocio y entretenimiento
- Aprendizaje.

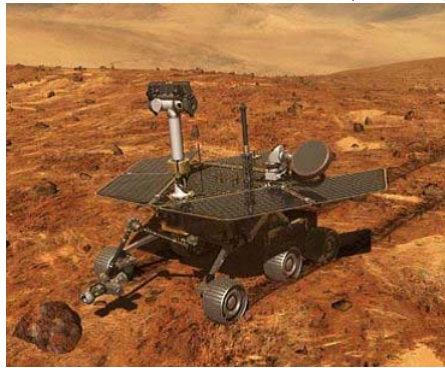
En el área científica diversos dispositivos son empleados para hacer investigaciones en campos en los cuales el hombre no puede ir o le es muy difícil ir, por tratarse de medios hostiles o demasiado peligrosos (submarino, espacial, irradiado por centrales nucleares, etc). El interés investigativo de este seminario es el de conocer, prototipos reales de robots móviles con alguna de las siguientes aplicaciones industriales:

- Operación en ambientes industriales hostiles, por ejemplo en refinerías de petróleos, plantas químicas o nucleares, o procesos que impliquen
- Realización de rutinas de mantenimiento y seguridad en caso de accidentes
- Inspección de tuberías terrestres y submarinas para detectar posibles filtraciones o fisuras
- Detección, recolección y clasificación de frutos o flores delicadas en el sector agrario
- Exploración submarina
- Reconocimiento de ambientes donde el hombre no puede llegar
- Tecnificación de la pesca
- Realización de labores de construcción que impliquen gran esfuerzo físico, o actividades que requieran constante concentración
- Vigilancia y seguridad de plantas industriales
- Inspección y reparación de tubos de evacuación que demanden controles y sensores robustos para trabajar en entornos muy contaminados
- Transporte de materiales en actividades de producción industrial

- Sillas de ruedas eléctricas dotadas de sistemas de sensores que permitan a los usuarios discapacitados evitar obstáculos y realizar desplazamientos seguros
- Implantación de sistemas de telerobots para el mantenimiento de líneas de distribución de energía eléctrica
- Extinción de pequeños incendios caseros
- Fabricación de juguetes sofisticados y educativos.

La *National Aeronautics and Space Administration (NASA)* utiliza de forma asidua estos robots, como por ejemplo los robots Spirit y Opportunity. Fueron lanzados a mediados de 2003 y llegaron a Marte en enero de 2004 en diferentes posiciones del planeta. Están provistos de avanzados sistemas para análisis geológicos y atmosféricos, entre otros.

Figura 6.1: Robot Spirit (NASA)



(Fuente: Robótica. Estado del arte, Jonathan Ruiz de Garibay Pascual.)

Los oceanógrafos están desarrollando avanzadas tecnologías para monitorizar los océanos, desde pequeños organismos biológicos hasta el estudio de las corrientes submarinas. El instituto francés Ifremer ha desarrollado diferentes dispositivos para la realización de tareas submarinas, como por ejemplo el robot *SAR (Système Acoustique Remorqué - Towed Acoustic System)*. Contiene un conjunto de sensores geofísicos de alta resolución para el estudio de la estructura de los fondos marinos, y es capaz de sumergirse a 6.000 metros de profundidad.

### **Manipulación de residuos radioactivos**

Para manipular remotamente estos residuos se hace uso tanto de manipuladores con unión mecánica y seguimiento directo del proceso por parte del operador a través de

Figura 6.2: Submarino SAR, de la compañía Ifremer



(Fuente: Robótica. Estado del arte, Jonathan Ruiz de Garibay Pascual.)

un cristal (en caso de baja contaminación), como con sistemas con mando remoto por radio o cable en el caso de contaminación elevada. Estos manipuladores permiten la flexibilidad necesaria para manipular elementos de peso variable y forma no definida.

### **Robots de seguridad y defensa**

En materia de defensa la robotización permite aumentar el área controlada, es el caso de robots voladores que supervisan áreas fronterizas o equipos terrestres que bordean el perímetro en busca de intrusos. Frente al equipo humano los robots no solo tienen la ventaja de no manifestar cansancio o reducir su capacidad en función de variables ajenas al propio trabajo; sino que además podemos añadir la posibilidad de incluir en los robots capacidades de detección muy superiores a las que puede manejar un ser humano, como visión en espectros diferentes del visible o capacidad para detectar sonidos apenas perceptibles por el oído humano.

### **Robotización de la agricultura**

Cabe por ello hacer un análisis del estado actual, ventajas y posibilidades de robotización de las tareas agrícolas. Los objetivos que se plantean son:

- Permitir la sustitución de operarios en tareas peligrosas para la salud, como la pulverización de productos fitosanitarios.
- Abordar la realización de tareas repetitivas y tediosas, como la recolección de frutos
- Realizar tareas en horas nocturnas, lo cual permite el ahorro de tiempo, por ejemplo, en la recolección.

- Mejorar la precisión en algunas de las tareas agrícolas, como las relacionadas con la biotecnología, y en concreto la multiplicación de plantas a partir de tejido vegetal.
- Optimizar la eficiencia y calidad de algunas de las tareas como la uniformidad en la realización de huecos para el trasplante.
- Lograr la disminución de riesgos ambientales como la reducción de la cantidad de producto fitosanitario que se emite al aire.
- Reducir costes, ya que se disminuye la cantidad de combustible y de productos utilizados en algunas tareas.
- Elevar la calidad de los productos como por ejemplo, la utilización de menos pesticidas.

Una segunda clase de aplicaciones de los vehículos robóticos, son la realización de tareas rutinarias en entornos, donde la eficiencia y la movilidad de estas máquinas reemplazan directamente la presencia humana. Por ejemplo, la agricultura a gran escala requiere de máquinas que puedan trabajar de forma autónoma.

Figura 6.3: Robot móvil para escardar



(Fuente: Robótica. Estado del arte, Jonathan Ruiz de Garibay Pascual.)

### **Robots domésticos**

Otra de las tantas aplicaciones de los robots de servicio se ven reflejadas en los hogares, cabe anotar que hoy en día esto solo se ve en países del primer mundo y que la tasa de demanda de este tipo de Robots sigue siendo muy baja, aun así han experimentado un grado de su aceptación y proliferación creciente en la última década

Figura 6.4: Silla de ruedas avanzada: IBOT



(Fuente: Robótica. Estado del arte, Jonathan Ruiz de Garibay Pascual.)

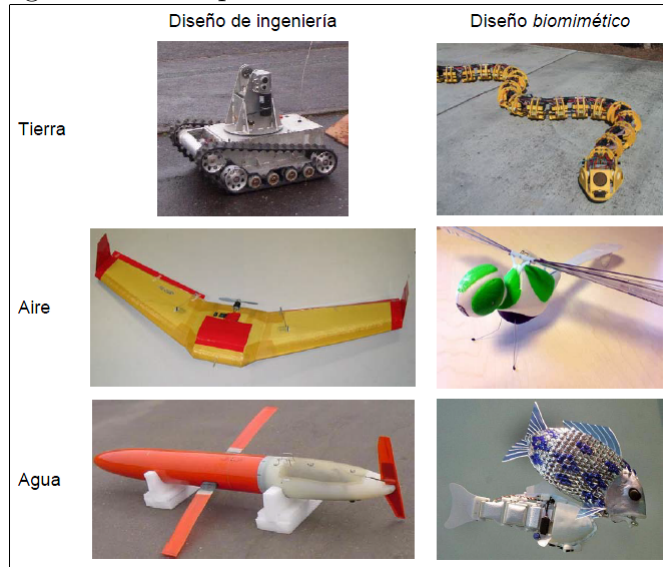
y prometen ser una parte importante no solo de las viviendas sino también de las oficinas del futuro. Una silla de ruedas robótica provee movilidad a las personas, y gracias a sus sensores, su inteligencia computacional y su fuente de alimentación, proporciona unas capacidades adicionales a estos dispositivos (figura 6.4).

#### 6.4.2. Tecnologías utilizadas en los sistemas motrices

Dependiendo del medio, se pueden proponer diferentes soluciones tecnológicas:

- **Estrategias de ruedas o propulsores (wheels and props):** Se basan en la evolución de las tecnologías que el ser humano ha ido desarrollando a través de siglos para el transporte. Los robots terrestres utilizan ruedas o cadenas, como los automóviles, los robots aéreos hélices y propulsores, al igual que los robots acuáticos.
- **Estrategias de marcha y aleteo (running and flapping):** Inicialmente los robots nacieron como una imitación del ser humano, por lo que hay una creciente tendencia a desarrollar robots que se inspiren en sistemas biológicos para el desarrollo de nuevas soluciones tecnológicas. A este nuevo conjunto de tecnologías se las conoce como biomimética.

Figura 6.5: Comparativa de modelos de locomoción



(Fuente: Robótica. Estado del arte, Jonathan Ruiz de Garibay Pascual.)

### Áreas de investigación

Se podría hablar de 4 áreas dominantes en la investigación de los robots móviles:

- **Mecanismos de movilidad:** Como ya se ha descrito, existen dos tendencias para el diseño de robots móviles, los diseños de ingeniería y los diseños biomiméticos. Las propiedades de los materiales y el diseño, son un punto de interés principal en los nuevos mecanismos. El uso de materiales ligeros y a su vez de una gran resistencia e incluso deformables dinámicamente, es un área muy importante de investigación.
- **Aprovechamiento de la energía:** Un mayor tiempo de autonomía de los vehículos implica una eficiencia energética del movimiento mayor. Por un lado se encuentran las investigaciones centradas en el desarrollo de novedosas tecnología para baterías. Un segundo punto de interés es, como se ha comentado anteriormente es, la eficiencia del movimiento, de forma que se limite el uso de energía lo máximo posible. Por último, diseñar robots capaces de autoabastecerse de energía, ya sea energía solar o aprovechando las corrientes submarinas de los grandes océanos.
- **Computación y control:** La introducción de la microinformática ha permitido el uso de los sistemas integrados más pequeños, ligeros y eficientes que los anteriores sistemas de control.

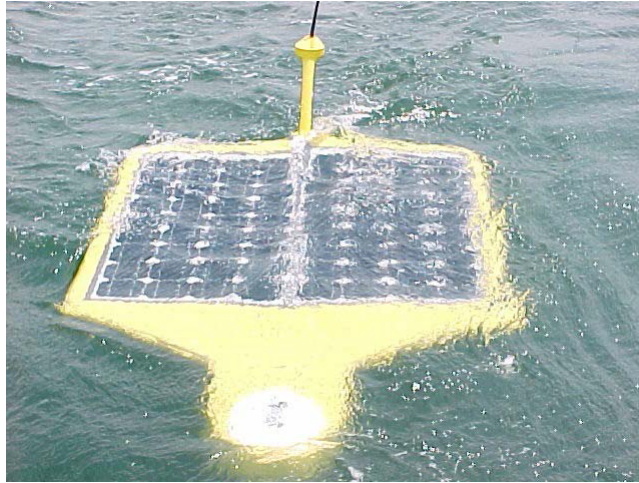


Figura 6.6: Autonomous Underwater Vehicles (AUVs)

(Fuente: Robótica. Estado del arte, Jonathan Ruiz de Garibay Pascual.)

- **Sensores y navegación:** Los sensores tiene dos propósitos principales en los vehículos robóticos:
  1. Los sensores monitorizan el entorno y son usados para el control de tareas interactivas.
  2. Los sensores son fundamentales para permitir la navegación autónoma de un robot móvil.

La navegación y la localización es un aspecto muy importante y en el que se ha investigado mucho en los últimos 10 años. El problema de los sistemas de localización y mapeo simultáneo (*Simultaneous Localization and Mapping, SLAM*), es su dificultad para la implantación en aplicaciones reales. Estos sistemas intentan resolver de manera concurrente dos problemas:

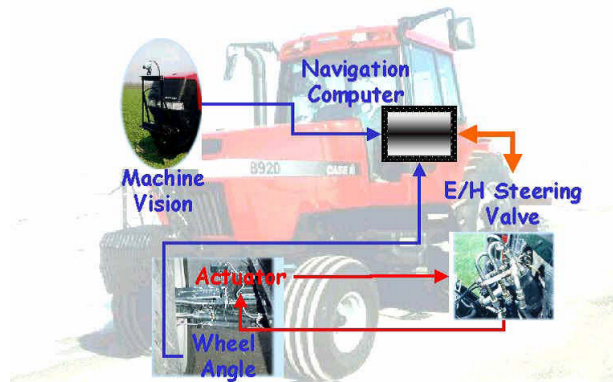
- Construir un mapa del entorno, identificando las zonas libres y las ocupadas.
- Localizarse en el mapa que se está construyendo.

### 6.4.3. Robots terrestres

#### Robots agrícolas

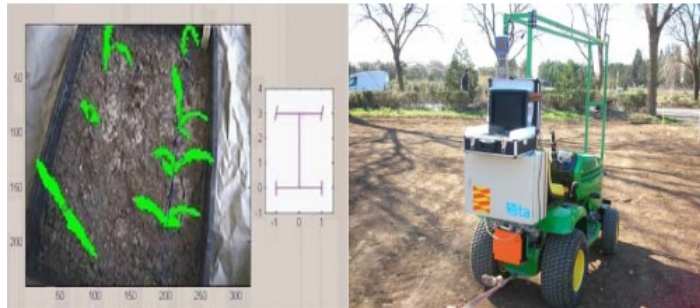
Automatizar muchas de las tareas a realizar en la agricultura es un ahorro enorme en los costes de producción de este tipo de alimentos. La Universidad de Illinois, trabaja en

Figura 6.7: Tractor autónomo para agricultura



(Fuente: Robótica. Estado del arte, Jonathan Ruiz de Garibay Pascual.)

Figura 6.8: Robot SAAPIN, guiado por un sistema de visión.



(Fuente: Robótica. Estado del arte, Jonathan Ruiz de Garibay Pascual.)

el desarrollo de novedosas máquinas para el entorno agrícola. Por ejemplo, en el tractor de la figura 6.7 se han incluido todos los dispositivos necesarios para que éste sea capaz de seguir las filas de una plantación de forma automática.

- **Robot SAAPIN**: El robot SAAPIN (Sistema Autónomo para Agricultura de Precisión e Integrada) ha sido desarrollado en el Centro de investigación y tecnología agroalimentaria de Aragón. Su objetivo es crear un vehículo agrícola capaz de desplazarse de forma autónoma por un campo, arrastrando un apero que pueda tomar los datos necesarios para realizar un mapa de salinidad.
- **ROBOCROP**: Es un tractor robotizado con visión por computador basado en sistema de orientación para de control de químicos en malezas, desarrollado por la empresa Tillet and Hague Technology Ltd., de UK. Este robot usa navegación

Figura 6.9: Robocrop, tractor robotizado de la empresa Tillet and Hague Technology Ltd



(Fuente: Los Robots en el Sector Agrícola. J. A. García V, L. A. Vázquez A. Universidad Politécnica de Madrid, Departamento de Automática, Ingeniería Electrónica e Informática Industrial.)

por guiado a través de las líneas de cultivo y al detectar una mala hierba por medio de visión artificial este la elimina a través de un corte realizado en forma mecánica. Figura 6.9.

### Robots de seguridad

- **Robots MOSRO y OFRO:** Durante la Copa Mundial de Fútbol en Alemania en el año 2006, los robots OFRO y MOSRO (figura 6.10) de RoboWatch Technologies fueron puestos a prueba en tareas de vigilancia exterior e interior respectivamente. Este tipo de robots permite detectar agentes químicos o biológicos, sustancias explosivas y movimientos sospechosos en un radio de 30 metros. Estos sistemas se pueden utilizar para patrullar en barcos y puertos, plantas industriales, zonas de acceso restringido o aeropuertos.
- **mSecurity:** El mSecurit de MoviRobotics es un robot de vigilancia con capacidad para realizar rondas en exteriores de forma semi-autónoma, detectar intrusos y dar la alarma al centro de control (figura 6.11).
- **RIOTBOT:** El RiotBot está pensado, diseñado y desarrollado para una amplia gama de operaciones policiales, militares y de seguridad en general, principalmente aquellas en las que los miembros de las unidades que han de intervenir no

Figura 6.10: Robots MOSRO (interior) y OFRO (exterior)



(Fuente: Robots de Seguridad Civil, Jose Breñosa, Patricia García. UPM.)

Figura 6.11: Robot mSecurit de MoviRobotics.



(Fuente: Robots de Seguridad Civil, Jose Breñosa, Patricia García. UPM.)

tienen plenamente garantizada su seguridad personal o ésta puede correr peligro. Es por ello que el RiotBot pretende ser un aliado indispensable en este tipo de acciones, consiguiendo mediante su uso la salvaguarda de la integridad física de los operadores y equipos que lo dirigen (figura 6.12)

- **AUNAV:** La familia de vehículos robotizados AUNAV y SUPER AUNAV, ha sido creada específicamente para tareas de desactivación de explosivos, es actualmente el estándar reglamentario utilizado desde el año 2003 por todos los Grupos Policiales Españoles (figura 6.13).
- **TALON:** El robot terrestre Talon se enmarca dentro de los robots portátiles, con sus 57kg de peso, destinados a desactivación de explosivos. Una de las principales características de estos robots consiste en su gran resistencia. Entre los ejemplos



Figura 6.12: Robot ofensivo Riotbot de TECHNOROBOT.

(Fuente: Robots de Seguridad Civil, Jose Breñosa, Patricia García. UPM.)



Figura 6.13: Robot AUNAV.

(Fuente: Robots de Seguridad Civil, Jose Breñosa, Patricia García. UPM.)

citados por el fabricante se encuentran un caso en el que el mismo robot ha llegado a detonar explosivos hasta en tres ocasiones durante su desactivación o el caso de uno que resistió la caída desde un puente tras ser alcanzado el transporte en el que se trasladaba (figura 6.14).

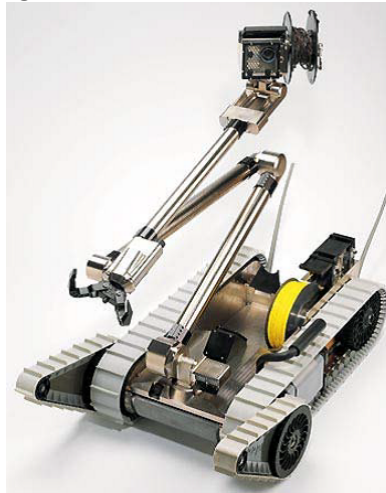
- **PACKBOT**; El Packbot está equipado con cámaras y sistemas de comunicación y puede llevar diferentes brazos robotizados por lo que es idóneo para labores de reconocimiento en extensiones de corto alcance o desactivar explosivos (figura 6.15).

Figura 6.14: Robot militar TALON de FORSTER MILLER en configuraciones Sword)



(Robots de Seguridad y Defensa. Pablo García-Robledo, Jesús Torrijos. Universidad Politécnica de Madrid.)

Figura 6.15: Robot PACKBOT



(Robots de Seguridad y Defensa. Pablo García-Robledo, Jesús Torrijos. Universidad Politécnica de Madrid.)

- **BigDog:** BigDog es un robot andador, cuadrúpedo, dinámicamente estable, para uso militar. Fue creado en 2005 conjuntamente por las compañías Boston Dynamics y Foster-Miller, el Laboratorio de Propulsión a Chorro de la Nasa y la Concord Field Station de la Universidad de Harvard. Es capaz de atravesar terrenos complicados a una velocidad de  $6,4 \text{ km/h}$  cargando hasta  $150 \text{ kg}$  de peso y de subir pendientes de  $35^\circ$  (figura 6.16).

Figura 6.16: BigDog de la Boston Dynamics.

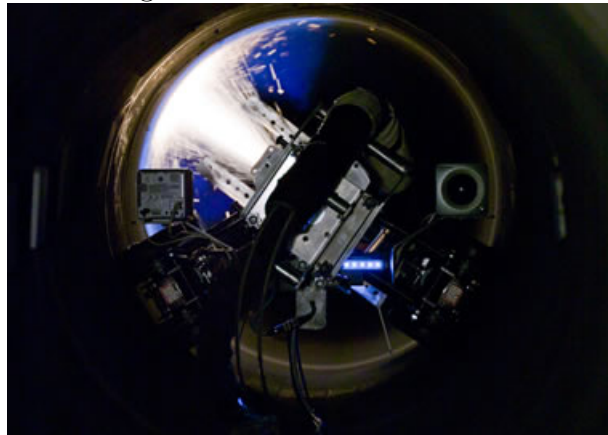


(Fuente: <http://www.yalosabes.com>)

### Robots de mantenimiento e inspección

- **Tirant III:** El Tirant III es un robot móvil capaz de operar en el interior de tuberías y de realizar tareas de mantenimiento e inspección. Tirant III, que ha sido desarrollado en colaboración con Grupo Dominguis, está patentado a nivel mundial. La función del robot es metalizar las tuberías de centrales nucleares por las que circula vapor, un trabajo era realizado anteriormente por operarios en condiciones realmente adversas (figura 6.17).

Figura 6.17: Robot Tirant III



(Fuente: <http://www.robotnik.es>)

## Cordinación de robots

Existen numerosos trabajos de investigación que trabajan en la coordinación de robots, con resultados bastante interesantes.

- **CROMAT:** El objetivo del Proyecto CROMAT es generar nuevos métodos y técnicas que permitan la actuación coordinada de robots móviles aéreos y terrestres (figura 6.18).

Figura 6.18: Robots autónomos del proyecto CROMAT



(Fuente: Robótica. Estado del arte, Jonathan Ruiz de Garibay Pascual.)

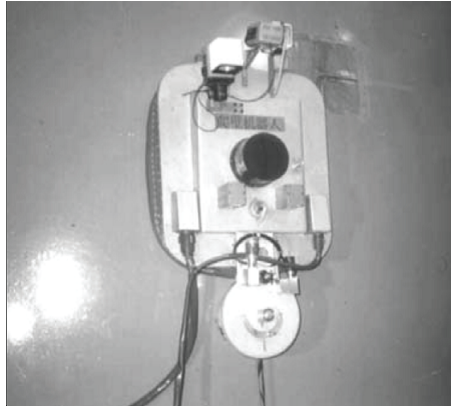
- **Otros:** La creación de sillas de ruedas que eviten los obstáculos de forma autónoma puede ser una aplicación muy importante y útil (figura 6.19).

Figura 6.19: Silla de ruedas autónoma desarrollada por Simuzhu Corporation.



(Fuente: Robótica. Estado del arte, Jonathan Ruiz de Garibay Pascual.)

Figura 6.20: Un robot escalador de muro de una sola ventosa.



(Fuente: Mobile Robots - State of the Art in Land, Sea, Air, and Collaborative Missions, XiaoQi Chen, Y.Q. Chen and J.G. Chase.)

#### 6.4.4. Robots biomiméticos

Investigadores de robótica móvil se han inspirado en el desarrollo transdisciplinar de la biónica, también conocido como biomimética, o biomimetismo. La biónica aplica métodos y sistemas biológicos encontrados en la naturaleza con el estudio y diseño de sistemas de ingeniería y tecnología moderna. Las excepcionales habilidades naturales en muchos animales e insectos han llamado mucho la atención de los biorrobotistas. Un enfoque común es construir robots con características similares a las de los animales y dichos robots se conocen como robots biomiméticos o simplemente biorobots.

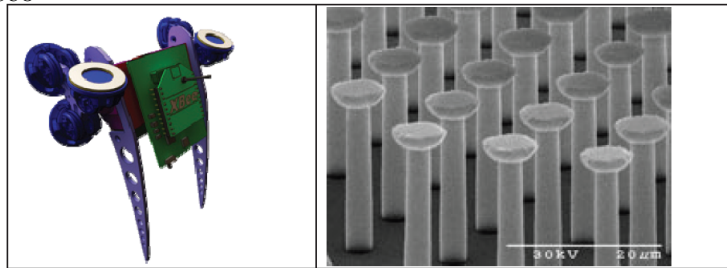
En términos generales, se utilizan tres tipos principales de mecanismos de fijación: de succión, y mecanismos de adhesión seca y magnética. El método de succión crea un vacío dentro de las ventosas a través de una bomba de vacío, las ventosas son presionadas contra la pared o el techo para que la fuerza de adherencia se genere entre las ellas y la superficie. Este efecto depende de una superficie impermeable suave para crear una fuerza suficiente para sostener el robot. Se ha hecho estudios con un robot escalador de muros con una sola ventosa. Este se compone de tres partes: una bomba de vacío, un mecanismo de sellado con una cámara de aire y muelles reguladores, y un mecanismo de accionamiento. Se tomaron dos ejemplos de aplicación:

- Inspección ultrasónica de Tanques cilíndricos de almacenamiento nuclear de acero inoxidable.
- Limpieza de edificios de gran altura.

Se ha implementado la adherencia magnética en robots escaladores de muros para aplicaciones específicas, tales como las instalaciones nucleares o inspección de tanques de gas y aceite. En casos específicos en la superficie lo permite, la fijación magnética puede ser muy conveniente para su confiabilidad inherente. Recientemente, los investigadores han desarrollado y aplicado adhesivos fibrilares sintéticos para emular la adherencia en seco bio-inspirada que se encuentran en los pie de la Salamandra. Un ejemplo es Waalbot utilizando adhesivos secos sintéticos, desarrollado por la Carnegie Mellon University, se muestra en la figura 6.21.

También basado en los principios de adherencia en seco del robot bioinspirado "Stickybot". Se afirma que el robot sube sin problemas superficies verticales tales como el vidrio (mostrado en la figura 6.22, plástico y baldosas de cerámica a  $4\text{cm/s}$ . La parte inferior de los dedos del pie del Stickybot están cubiertos con paneles de pequeños tallos de polímeros en ángulo.

Figura 6.21: Tri-leg Waalbot. (a) Modelo CAD (b) Fibras con tallos para lograr una adherencia en seco



(Fuente: Mobile Robots - State of the Art in Land, Sea, Air, and Collaborative Missions, XiaoQi Chen, Y.Q. Chen and J.G. Chase.)

Se cree que por primera vez las ventosas de Bernoulli se han desarrollado con éxito como una sujeción fiable para los robots escaladores de muro, como se muestra en la figura 6.23.

#### 6.4.5. Robots espaciales

Las condiciones extremas y los requerimientos que impone el espacio exterior hacen que el diseño de este tipo de robots deba cumplir ciertas restricciones:

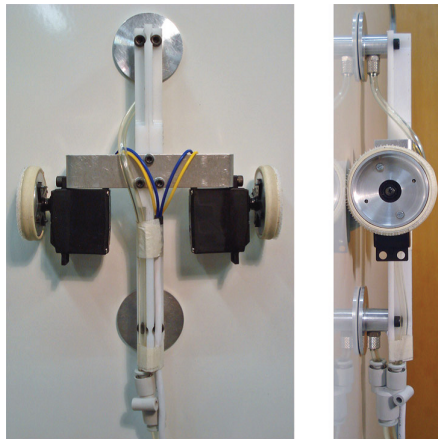
- El robot debe ser capaz de soportar las aceleraciones producidas en su lanzamiento desde la Tierra.

Figura 6.22: Stickbot



(Fuente: Mobile Robots - State of the Art in Land, Sea, Air, and Collaborative Missions, XiaoQi Chen, Y.Q. Chen and J.G. Chase.)

Figura 6.23: Un innovador robot escalador de muro basado en el efecto Bernoulli.



(Fuente: Mobile Robots - State of the Art in Land, Sea, Air, and Collaborative Missions, XiaoQi Chen, Y.Q. Chen and J.G. Chase.)

- Los aterrizajes en otros planetas no suelen ser realizados con mucha delicadeza, y puede provocar la rotura del robot si éste no es suficientemente robusto.
- Funcionar en condiciones ambientales difíciles: Las características de la atmósfera (presión, temperatura, etc.) deben ser tenidas en cuenta.
- La ingravidez es un aspecto muy importante, debido a que las pruebas realizadas en la Tierra pueden ser inválidas.

Además de estos aspectos, también se deben tener otras consideraciones relativas a las características de las tareas que deberá realizar:

- El robot debe tener un alto grado de autonomía, ya no sólo en cuanto a su fuente de energía, sino también a ser automantenible; debido a que ningún ser humano interactuará directamente con él.
- La redundancia es un aspecto muy importante en caso de fallos o roturas.
- Suelen ser robots de altísima complejidad debido a que suelen integrar varios subsistemas de diferentes fabricantes.
- Las comunicaciones están limitadas a determinadas ventanas temporales, y además existen grandes retardos en la comunicación con la Tierra.

De igual manera general, dentro de las ventajas que aporta el uso de los robots en el espacio se pueden mencionar las siguientes:

- Incremento de la seguridad (ambiente hostil).
- Incremento de la productividad (reducción del tiempo).
- Reducción de los costos (aumento de productividad y reducción de costes de recursos para los astronautas: entrenamiento, soporte, etc).
- Incremento de la fiabilidad.
- Posibilidad de realizar tareas imposibles para el ser humano (exploración de otros planetas, asteroides cometas, etc).
- Incremento de la flexibilidad (pueden ser aplicados a varias funciones).

**La NASA clasifica las misiones de los robots en el espacio en tres grupos:**

- **Robots extravehiculares (EVR):** Son robots que deben realizar misiones en el exterior de naves en órbita terrestre, como la puesta en órbita y recogida de satélites, ensamblado y mantenimiento de la estación internacional espacial (ISS) y apoyo a los astronautas en operaciones extravehiculares.
- **Robots exploradores:** se trata de robots móviles, con frecuencia dotados de capacidades de manipulación que han sido y son utilizados para la exploración de superficies de otros planetas.

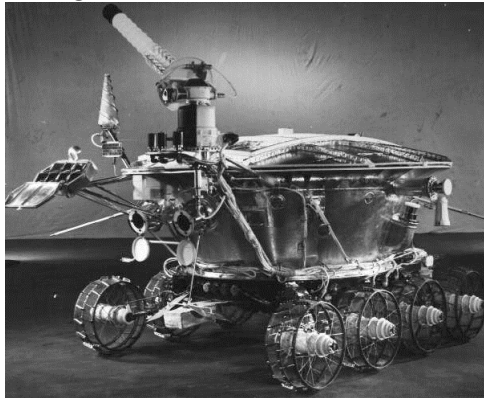
- **Robots para tareas intravehiculares:** se trata de robots, de características no muy diferentes a las de robots industriales incorporados a la nave con el objetivo de realizar experimentos científicos en entornos controlados.

### Ejemplos de robots espaciales

Las investigaciones realizadas con este tipo de robots suelen ser de alto secreto, por lo que es muy difícil acceder a las más novedosas tecnologías utilizadas por las diferentes agencias espaciales. Por ello, a continuación solo se muestran un par de ejemplos de lo que podemos encontrar.

- **Robots Espaciales Exploradores.** Estos Robots tienen como fin la exploración de la superficie de planetas, y cuya idea básica consiste en diseñarlos, construirlos y programarlos para que se desenvuelvan de manera autónoma en las superficies planetarias tales como las de la Luna o Marte. El primer robot que viajó en la superficie del cuerpo extraterrestre fue Lunakhod. (1970) Figura 6.24, desarrollado por la antigua Unión Soviética. Fue operado de forma remota desde la Tierra y atravesó más de 10,5 km en la luna.

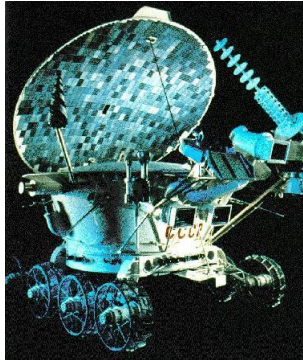
Figura 6.24: Robot Lunakhod.



(Fuente: <http://apod.nasa.gov/apod/ap100606.html>.)

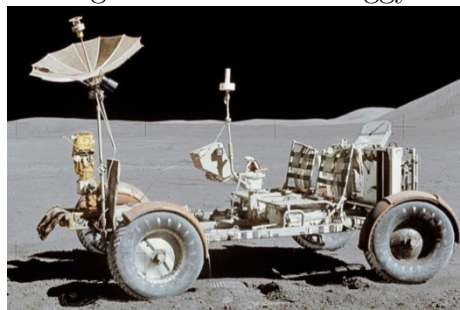
Por otro lado, el vehículo explorador lunar (LRV) o “Moon Buggy” (Figura 6.26) fue utilizado en el programa Apolo de la NASA (Apolo 15, 16 y 17, durante 1971-1972). Este vehículo lunar fue un carro de cuatro ruedas de accionamiento eléctrico que puede llevar a dos astronautas y puede ser accionado manualmente, como un carrito de golf. Era útil para expandir el área de la expedición de las personas en los lugares de desembarque.

Figura 6.25: Robot Lunakhod-2



(Fuente: <http://nssdc.gsfc.nasa.gov/nmc/masterCatalog.do?sc=1973-001A>)

Figura 6.26: Moon Buggy



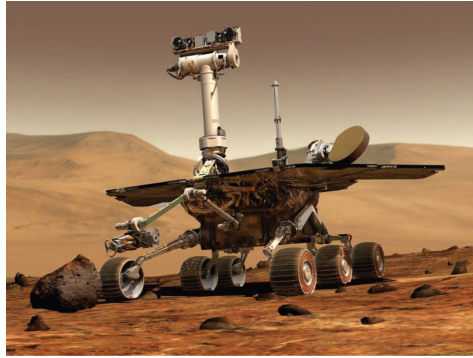
(Fuente: <http://www.hq.nasa.gov/office/pao/History/alsj/a15/images15.html>)

En cuanto a la exploración de Marte, los primeros módulos de aterrizaje (Landers) con éxito son Viking 1 y 2 de la NASA (aterrizó en 1976). A pesar de que eran Landers estáticos, tienen un brazo robótico para recoger muestras del suelo y realizar análisis in situ. En cuanto a los robots móviles (rovers) en Marte, el rover Sojourner de la misión Mars Pathfinder (1997), misión cuyo objetivo fue el de demostrar tecnológicamente cómo enviar un módulo de amortizaje con instrumental y un rover robótico autónomo a la superficie del planeta rojo, que fuera capaz de devolver una cantidad enorme de datos sobre Marte; y los gemelos Spirit y Opportunity de fabricación estadounidense en la misión Mars Exploration Rover (MER) (2004-2009, ver Figura 6.27) han tenido un éxito notable.

Los robots Spirit Opportunity (figura 6.27) fueron enviados a Marte a finales de 2003 y primeros de 2004, respectivamente. Fueron construidos en el Jet Propulsion Laboratory de la NASA entre 2002 y 2003. Tras su amortizaje, cada uno de ellos

en lados opuestos del planeta, han desempeñado diferentes tareas de investigación.

Figura 6.27: Robots Spirit y Opportunity, de la NASA



(Fuente: Achievements in space robotics.)

El robot Opportunity quedó atrapado en una duna de arena de la que salió de forma airosa. Para conseguirlo, el equipo de la Tierra recreó el escenario en el que se encontraba el robot y simuló con un robot gemelo del Opportunity, la forma de salir para después enviar los comandos precisos al robot atrapado en Marte.

#### 6.4.6. Robots acuáticos

Un robot submarino fundamentalmente debe tener la capacidad de desplazarse y maniobrar bajo el agua. Aplicaciones que se benefician de los robots acuáticos son:

- Investigación oceánica

Figura 6.28: Robot Opportunity saliendo de una duna.



(Fuente: Robótica. Estado del arte, Jonathan Ruiz de Garibay Pascual.)

- Las aplicaciones de tipo militar centradas en la vigilancia, localización y neutralización de minas o armamento hundido.
- Localización de barcos y aviones hundidos
- Inspección y reparación de buques o de construcción con estructura sumergida (represas, puentes, etc)

Los robots acuáticos son un tipo concreto de robots móviles que, en este caso, usan el agua como medio de transporte. Esto requiere algunas consideraciones importantes a tener en cuenta en este tipo de robots.

- Un robot acuático tiene 6 grados de libertad porque se puede mover en 3 dimensiones diferentes, al igual que los robots aéreos.
- La atracción gravitacional es un factor que pierde relevancia debido a la densidad del agua y al flujo de agua desplazado por el cuerpo del robot.
- El ensamblaje del robot debe ser estanco y resistente a la corrosión. Además, dependiendo de la profundidad a la que trabaje el robot, se deberá tener en cuenta la presión que debe ser capaz de soportar.

Estos robots pueden ser totalmente autónomos y no requerir de una comunicación continua con la estación, pero en caso contrario, puede haber problemas derivados de la comunicación. En algunos casos, el robot se comunica de forma alámbrica (lo que proporciona una forma rápida de recuperar el robot en caso de fallo), mientras que en otros, se puede utilizar el sonido por medio de boyas introducidas en el agua, debido a que el sonido se propaga mejor en este medio.

### **Sistemas de desplazamiento**

En cuanto a los sistemas mecánicos utilizados, existe una gran variedad y como en casos anteriores, se pueden usar tantos sistemas de ingeniería como sistemas biomiméticos.

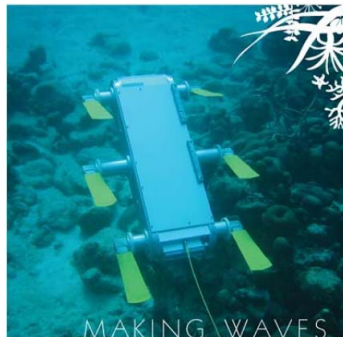
- **Turbinas/paletas de dirección:** En este caso, se utiliza una turbina que proporciona la propulsión correspondiente y una paleta de dirección con posibilidad de giro, para permitir la maniobrabilidad del robot (Figura 6.29).
- **Aletas:** Imitando el movimiento de algunos animales, se puede utilizar un conjunto de aletas para desplazarse por medios líquidos (Figura 6.30).

Figura 6.29: Robot submarino con turbinas.



(Fuente: Robótica. Estado del arte, Jonathan Ruiz de Garibay Pascual.)

Figura 6.30: Robot acuático con aletas.



(Fuente: Robótica. Estado del arte, Jonathan Ruiz de Garibay Pascual.)

- **Movimiento ondulante:** Similar a serpientes o caracoles pero usados generalmente para el desplazamiento sobre la superficie (Figura 6.31).
- **Caminantes:** Tanto con patas como con ruedas pueden desplazarse por el fondo y algunos incluso, caminar sobre la superficie del agua.(Figura 6.32)

### Ejemplos de robots submarinos

Existen un gran número de líneas de investigación específicas de los robots acuáticos. La mayoría de ellas, se centran en la exploración submarina, perolas hay también dedicadas al estudio de animales marinos, rescates submarinos, etc.

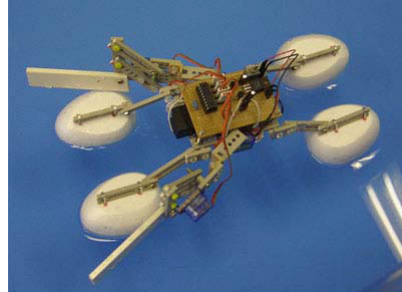
- **Exploración submarina Institut Français de Recherche pour L'exploitation de la Mer (IFremer).** La compañía Ifremer lleva trabajando varios años en ro-

Figura 6.31: Robot marino con movimiento ondulante.



(Fuente: Robótica. Estado del arte, Jonathan Ruiz de Garibay Pascual.)

Figura 6.32: Robot capaz de caminar sobre el agua.



(Fuente: Robótica. Estado del arte, Jonathan Ruiz de Garibay Pascual.)

bots submarinos para el estudio geográfico de los fondos marinos, así como para el estudio del hábitat marino, y ha desarrollado varios modelos de robots. Como ejemplo, el robot *SAR* (*Système Acoustique Remorqué - towed acoustic system*) contiene un conjunto de sensores geofísicos de alta resolución para el estudio de la estructura de los fondos marinos, y es capaz de sumergirse a 6.000 metros de profundidad. Esta compañía también estuvo presente en el desastre del Prestige (19 de noviembre de 2002) realizando tareas de reconocimiento con el vehículo submarino Nautille.

- **AUV Lab del MIT Sea Grant.** Desarrollaron su primer vehículo a primeros de los años 90, llamado Odyssey I, y actualmente disponen de más de una decena de vehículos, algunos de ellos en diferentes versiones.
- **AN/BLQ-11.** Se trata de un sistema submarino (UUV) desarrollado por la empresa Boeing, y clasificado como sistema de reconocimiento de minas a largo plazo

Figura 6.33: Submarino RAS.



(Fuente: Robótica. Estado del arte, Jonathan Ruiz de Garibay Pascual.)



Figura 6.34: Robot Nautilie.

(Fuente: Robótica. Estado del arte, Jonathan Ruiz de Garibay Pascual.)

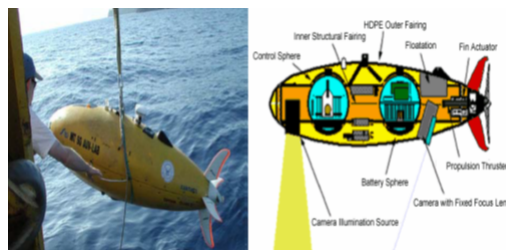


Figura 6.35: Robot Xanthos, AUV.

(Fuente: Robótica. Estado del arte, Jonathan Ruiz de Garibay Pascual.)

(“long-term mine reconnaissance system”, LMRS). Proporciona un rápido y fiable medio de detección de campos de minas submarinas, es de gran ayuda en operaciones navales anfibias y otros tipos de operaciones en el transcurso de la batalla y, además, permite la navegación segura alrededor de aguas minadas.

Figura 6.36: Robot AN/BLQ-11.



(Fuente: Robots de Seguridad y Defensa. Pablo García-Robledo, Jesús Torrijos. Universidad Politécnica de Madrid.)

- **ROV Penguin B3.** Minisubmarinos robot como este ROV Penguin B3 son usados en los modernos cazaminas para detectar y destruir minas navales. Para aumentar su flexibilidad, ciertos tipos de cazaminas son al mismo tiempo buques capaces de sembrarlas.

Figura 6.37: Robot Penguin B3.



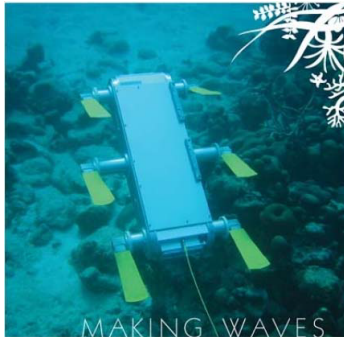
(Fuente: Robots de Seguridad Civil, José Breñosa, Patricia García. UPM)

- **Proyecto AQUA.** El proyecto AQUA es una colaboración entre varias universidades. Su objetivo es el diseño de un robot anfibio totalmente autónomo, que sea capaz de explorar los fondos marinos y recoger información, minimizando el impacto en la vida marina.

#### Estudios sobre animales marinos.

- **Robot Madeleine.** En el laboratorio Bioengineering and Biorobotics at Vassar College, con la colaboración de Nekton Research, LLC, han desarrollado un robot

Figura 6.38: Robot AQUA.



(Fuente: Robótica. Estado del arte, Jonathan Ruiz de Garibay Pascual.)

tetrápodo marino para el estudio de diferentes teorías de locomoción en animales marinos con 4 aletas (pingüinos, tortugas marinas, etc.)

Figura 6.39: Robot Madeleine.



(Fuente: Robótica. Estado del arte, Jonathan Ruiz de Garibay Pascual.)

- **Robot Salamandra.** Dentro del Biologically Inspired Robotics Group (BIRG) del School of Computer and Communication Sciences en el EPFL, tienen varios ejemplos de robots acuáticos y entre ellos, destaca el proyecto Salamandra.

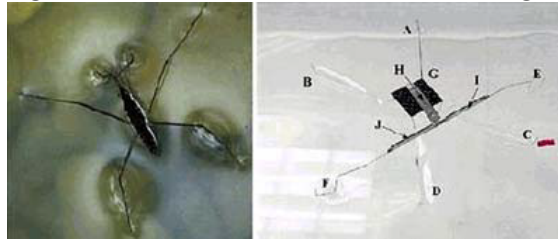
**Otros.** Otros ejemplos de robots acuáticos, son aquellos que son capaces de andar sobre el agua. Por ejemplo, se muestran los principios biológicos de los animales capaces de andar sobre el agua y su aplicación en robótica. Este tipo de robots, conocidos como robots STRIDE (Surface Tension Robotic Insect Dynamic Explorer), son capaces de andar sumergiendo sus patas entre 3 y 4 milímetros y poseen una gran maniobrabilidad. Sus fines son meramente educativos en estos momentos, aunque se podría utilizar para el control medioambiental.

Figura 6.40: Robot Salamadra.



(Fuente: Robótica. Estado del arte, Jonathan Ruiz de Garibay Pascual.)

Figura 6.41: Robot andador sobre el agua.



(Fuente: Robótica. Estado del arte, Jonathan Ruiz de Garibay Pascual.)

### 6.4.7. Robots Aereos.

La robótica aérea es un campo relativamente nuevo con grandes posibilidades para la investigación y el desarrollo de este tipo de robots. Se han conseguido grandes avances en sus sistemas motrices, pero sus capacidades de procesamiento de información y su aplicabilidad a tareas reales sigue siendo limitada en la mayoría de los casos. El problema principal radica en la realización autónoma de las maniobras adecuadas y la reactividad necesaria para adaptarse a los cambios del entorno.

#### Sistemas de desplazamiento.

Al igual que en el resto de las áreas de la robótica, se pueden encontrar sistemas de desplazamiento desarrollados desde la ingeniería o por métodos biomiméticos:

- **Hélices o turbinas:** A modo de los tradicionales aviones o helicópteros, las configuraciones pueden ser varias, pero el principio en el que se basan todas es el desplazamiento del aire.

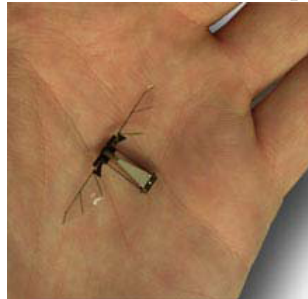
Figura 6.42: Robot de tipo helicóptero.



(Fuente: Robótica. Estado del arte, Jonathan Ruiz de Garibay Pascual.)

- **Alas:** Imitando a los animales con capacidad de volar, especialmente a los insectos, se desarrollan novedosos robots capaces de volar.

Figura 6.43: Insecto robótico capaz de volar.



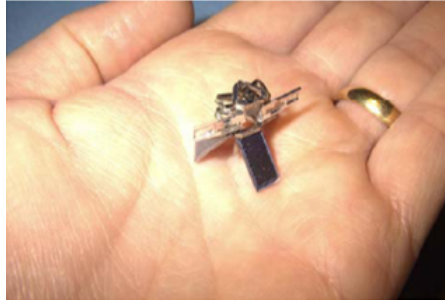
(Fuente: Robótica. Estado del arte, Jonathan Ruiz de Garibay Pascual.)

### Ejemplos de robots aéreos.

Al ser un tema relativamente nuevo en el área de la robótica, los mayores esfuerzos de investigación se están realizando aún en el diseño y mejora de los sistemas de propulsión. En la mayoría de los casos, estos robots son teledirigidos por un operador situado en tierra, aunque en los últimos años ya se trabaja en hacer que estos robots sean autónomos.

- **Prototipos de robots voladores Proyecto MFI (Micromechanical Flying Insect).** En la universidad de Berkeley, en colaboración con el Center for Information Technology Research in the Interest of Society (CITRIS), se ha llevado a cabo un proyecto cuyo objetivo fue crear un pequeño robot de 25 mm.
- **DragonFly.** La compañía WowWee ha puesto a la venta recientemente el primer robot teledirigido que vuela al estilo de los insectos.

Figura 6.44: Robot del proyecto MFI.



(Fuente: Robótica. Estado del arte, Jonathan Ruiz de Garibay Pascual.)

Figura 6.45: DragonFly de WowWee.



(Fuente: Robótica. Estado del arte, Jonathan Ruiz de Garibay Pascual.)

**Colonias de robots voladores Defense Science and Technology Organization, DSTO.** Como parte del Departamento de Defensa del gobierno de Australia, este grupo ha trabajado en el desarrollo de pequeños robots voladores capaces de realizar una misión de reconocimiento de forma autónoma y coordinada entre ellos mismos.

- **Aerospatial Controls Laboratory.** Los robots son autónomos, cooperan e intercambian información entre ellos y son capaces de reorganizarse cuando uno de ellos falla.

#### **Otros Ejemplos de robots aéreos.**

- **PREDATOR.** El UAV Predator constituye el ejemplo de vehículo aéreo no tripulado utilizado en gran diversidad de aplicaciones. Con sus más de 8 metros de longitud y cerca de 15 *m* de envergadura, el Predator es un robot resistente y capaz de volar a media altura, típicamente 25.000 pies.
- **UAV AirRobot.** El AirRobot es la solución UAV de AirRobotGMBH&Co. que

Figura 6.46: Robots voladores.



(Fuente: Robótica. Estado del arte, Jonathan Ruiz de Garibay Pascual.)

Figura 6.47: UAV PREDATOR.



(Fuente: Robots de Seguridad y Defensa. Pablo García-Robledo, Jesús Torrijos. Universidad Politécnica de Madrid.)

es un cuadricóptero que ha alcanzado gran protagonismo debido a las detenciones que se han realizado con su ayuda en el Reino Unido. Gracias a su sistema de propulsión eléctrica silenciosa y a su reducido tamaño es prácticamente indetectable. Resulta ideal para vigilancia y observación, filmación y fotografía aérea, y se puede emplear como plataforma para cargas inferiores a 330 gramos.

- **EADS TALARION.** El EADS Talarion es un vehículo aéreo no tripulado, diseñado por el consorcio aeronáutico europeo EADS para satisfacer las futuras necesidades militares europeas de reconocimiento, inteligencia, y vigilancia.
- **X-47B.** Northrop Grumman está desarrollando una nueva generación de aeronaves no tripuladas que prescindirían del operador que las controla desde tierra. El proyecto, con nombre en clave X-47B, está trabajando en un avión experimental que volaría de forma autónoma, es decir, sin necesidad de un piloto.

Por ahora, el X-47B es un proyecto experimental y aún está lejos de convertirse

Figura 6.48: UAV AirRobot



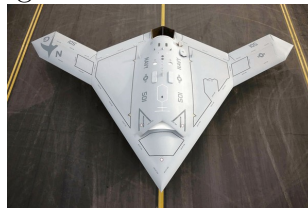
(Fuente: Robots de Seguridad Civil, Jose Breñosa, Patricia García. UPM.)

Figura 6.49: UAV “Talarion” de EADS-CASA.



(Fuente: Robots de Seguridad Civil, Jose Breñosa, Patricia García. UPM.)

Figura 6.50: UAV X-47B.



(Fuente: <http://alt1040.com>)

en realidad pero, viendo algunas de sus características.

## 6.5. COMPAÑÍAS FABRICANTES DE ROBOTS MÓVILES EN LA ACTUALIDAD

En la actualidad se pueden enunciar muchas empresas encargadas del diseño y construcción de robots móviles de todo tipo y a continuación se muestran solo algunas de

ellas.

### 6.5.1. Robots de inspección, exploración, rescate y defensa

#### Terrestres

- *Aquiles Robotics Systems (España)*. Robots de seguridad Aquiles I y Aquiles Protector. Dispone de brazo que le permite realizar operaciones sobre objetos.
- *Automax Robots (Japón)*. Robots manipuladores para trabajos peligrosos (Cables de alta tensión, sustancias peligrosas).
- *Bluebotics (Suiza)*. Plataforma móvil Shrimp III para terrenos abruptos (unión de ruedas y patas).
- *Foster-Miller (EEUU)*. Robots TALON.
- *Iberobotics (España)*. Robots modelo iVision para videoinspección de conductos, falsos techos, derrumbes y zonas de difícil acceso en general.
- *iRobot (EEUU)*. Diversos robots terrestres de inspección y militares, como Packbot, SUGV, 110 Fisrtlook y el 710 Warrior.
- *Robotnik Automation SLL (España)*. Robots RESCUER, GUARDIAN y Summit.
- *Robowatch Technologies (Alemania)*. Vehículos autónomos o teleoperados como CHRYSOR.
- *Vecna (EEUU)*. Robot BEAR (Battlefield Extraction and Retrieval Robot) para el rescate de soldados heridos en el campo de batalla. También el robot Porter para transporte y logística militar y apoyo en misiones.
- *Yujin Robot (Corea)*. Robots móvil con orugas ROBHAZ-DT3 para aplicaciones civiles y militares.

#### Tanques y tuberías

- *Cobra Technologies (EEUU)*. Fabricante de robots para la inspección con video cámara de tuberías y conductos de distintos diámetros.
- *Danduct Clean (Dinamarca)*. Robots para la inspección y limpieza de conductos de aire en edificios.

- *Hirose Robotics Lab (Japón)*. Robot Thes-III.
- *Iberobotics (España)*. Robots modelo iVision para videoinspección de conductos, falsos techos, derrumbes y zonas de difícil acceso en general.
- *Osaka Gas (Japón)*. Robot de inspección de tanques de almacenamiento de gas esféricos. Este robot con ruedas magnéticas inspecciona las soldaduras en el exterior del tanque para detectar filtraciones de gas.

### **Marinos y subacuáticos, pueden ser autónomos (AUVs) o de control remoto (ROVs)**

- *Boeing Company (EEUU)*. Long-term Mine Reconnaissance System (LMRS), robot submarino
- *Cybernetix (Francia)*. Amplia gama de robots subacuáticos (AUVs: OCTOPUS, SPIDER, SWIMMER, ALIVE, ICARE...)
- *Hydro-Quebec Research Institute (Canadá)*. Robot submarino de inspección ROV-3.
- *Ifremer (Francia)*. Robots submarinos para investigación (Victor 6000, Nautilo, SAR, Scampi, MIMOSA, ...)
- *Meerestechnik Bremen (Alemania)*. Robots submarinos sobre el fondo marino con orugas (crawlers).
- *Mitsui Engineering & Shipbuilding (MES) (Japón)*. Robots subacuáticos para distintas aplicaciones.
- *RovBuilder (Rusia)*. Empresa que fabrica pequeños robots submarinos para inspección.

### **Aéreos, también llamados UAVs**

- *AEROVISIÓN (España)*. Mini avión autónomo (UAV) FULMAR, capaz de transmitir imágenes de video e infrarrojo en tiempo real
- *Boeing Company (EEUU)*. X-45, avión de combate no tripulado.
- *Cybaero (Suecia)*. Helicópteros UAV APID y VANTAGE. En España el modelo APID55 es distribuido por Roboser.

- *Droidworx (Australia)*. Helicópteros de 3, 4, 6 y 8 hélices orientados a la realización de fotografía y videgrabación aérea.
- *Kawada-AIST (Japón)*. Helicóptero sin piloto RoboCopter.
- *Northrop-Grumman (EEUU)*. MQ-8B Fire Scout, Global Hawk, Navy UCAS X-47, Hunter.

### 6.5.2. Robots de entretenimiento y aprendizaje

Desde robots mascotas y juguetes para los más pequeños de la casa, pasando por robots programables con distintas capacidades e inteligencia, hasta kits de robots móviles o humanoides para montar y programar. Muchos de estos robots programables pueden ser utilizados en concursos de robótica y competiciones, como las de sumo o de fútbol.

#### Juguetes, mascotas y robots para el hogar de fácil programación

- *Cye (EEUU)*. Robot móvil al que se pueden acoplar distintos elementos (aspirador, mesa, cámara) y realizar distintas funciones.
- *SEGA Toys (Japón)*. Pequeños robots de juguete C-Bot y W-Bot, y también el perrito idog (YT) y el pez ifish (YT).
- *Speecys Corp (Japón)*. Robots humanoides para entretenimiento (YT).
- *Vstone (Japón)*. Robots humanoides como Robovie X, Robovie-R2 y Robovie-R3 entre otros.

#### Kits de montaje y robots programables a bajo nivel

- *Dajin Robot (Corea)*. Robots humanoides Lucy III, cuadrúpedos y hexápodos.
- *DFRobot (China)*. Kits de robots móviles basados principalmente en Arduino.
- *Hanson RoboKind (EEUU)*. Desarrolla cabezas robóticas con capacidad de mostrar expresiones para su incorporación en robots humanoides.
- *K-Team (Suiza)*. Kephera, Hemisson y Koala. Robots orientados a la investigación.

- *LEGO Mindstorms (Dinamarca)*. Robotic Invention System 2.0, para montar robot mediante piezas Lego y con toda la electrónica y entorno de programación necesarios.
- *Lynxmotion (EEUU)*. Robots bípedos, cuadrúpedos, hexápodos, móviles. Un amplio catálogo de kits y componentes.
- *Roboblock (Corea)*. Dispone de kits de robots con patas (humanoides y hexápodos) basados en servos, y también robots móviles tipo seguidor de líneas.
- *Robosoft (Francia)*. Plataformas móviles de alto nivel orientadas a la investigación.

### 6.5.3. Robots de vigilancia y seguridad

Robots para vigilancia y seguridad tanto en el hogar como a nivel profesional en oficinas, naves y exteriores.

#### Vigilancia en el hogar

- *Cye (EEUU)*. Spy-cye.
- *Dasatech (Corea)*. Robot de seguridad en el hogar EGIS-SR.

#### Vigilancia profesional

- *Cybermotion (EEUU)*. Sistema y Robots de seguridad Cyberguard.
- *Mostitech (Corea)*. Su plataforma robot universal puede ser utilizada como robot de vigilancia.
- *MoviRobotics (España)*. Robots móviles para vigilancia e inspección.

### 6.5.4. Robots en sectores de construcción y minería

Robots profesionales utilizados principalmente para tareas de demolición/minería, carga de materiales en la construcción, e incluso la construcción de edificios.

#### Robots en la construcción

- *Brokk AB (Suecia)*. Robots de demolición teleoperados.
- *Fujita Corporation (EEUU)*. Sistema de construcción autónomo.

- *Shiraishi Corporation (Japón)*. Unmanned Deep Shaft Construction System

### 6.5.5. Robots en sectores ganadero, agrícola y forestal

Robots utilizados en determinadas tareas de estos tres sectores, destacando por el momento el abono y fumigación en agricultura, y el ordeño automático en ganadería.

#### Robots en sectores agrícola y forestal

- *AGROBOT (España)*. Fabricante de cosechadoras de fresas con múltiples brazos robóticos guiados con visión artificial.
- *Harvest Automation (EEUU)*. Fabricante de robots colaborativos para la manipulación de objetos en tareas agrícolas, como el transporte de tiestos en invernaderos y viveros de plantas.

#### Robots en el sector ganadero

- *DCEngineering (Reino Unido)*. Sistema de ordeño robotizado GM Liberty.
- *DeLaval (Suecia)*. Sistema de ordeño robotizado VMS.
- *Fullwood (Reino Unido)*. Sistema de ordeño robotizado Merlin.
- *Lely Industries (Holanda)*. Sistema de ordeño robotizado Astronaut.

### 6.5.6. Robots para limpieza y mantenimiento

Robots tanto profesionales como para el hogar, utilizados para la limpieza de suelos (aspiradores) y piscinas, y para el mantenimiento de jardines como cortacésped y la recogida de pelotas por ejemplo de golf. También se incluyen robots para limpieza profesional de sistemas de aireación, buques, fachadas, depósitos, etc.

#### Limpieza del hogar

- *ISG Micro (Corea)*. Robot limpiacristales Windoro.
- *Kaily (China)*. Robots aspiradores series KL-robot2 y KL-robot5.
- *Kärcher (Alemania)*. Robocleaner RC3000, robot aspirador.
- *Palson (España)*. Robots aspiradores Diábolo y Diábolo Plus.

### **Limpieza profesional (suelos, fachadas, tanques, conductos de aire)**

- *Eventronics (España)*. Robots para la limpieza de conductos de aireación EV-Clean.
- *Intellibot (EEUU)*. Robots para la limpieza profesional de suelos de distintas formas: productos Hydrobot, Aerobot y Duobot.
- *RoboSoft (Francia)*. Robots para limpieza industrial de suelos y cristales.
- *Trenz-Cleaner (España)*. Robots de inspección y limpieza integral de conductos de aire acondicionado y ventilación.

### **Cortacésped**

- *Friendly Robotics (Australia)*. Robots cortacésped Robomower RL850 y RL1000.
- *Husqvarna AB (Electrolux) (Suecia)*. Cortacéspedes automáticos (un modelo es solar) Automower.
- *Precise Path (EEUU)*. Robots cortacésped especializados en el mantenimiento de campos de golf

## **6.5.7. Robots para cuidados y ayuda a personas**

Robots para el cuidado de niños y personas mayores, y robots para ayudar a personas mayores, discapacitados, minusválidos o enfermos para mejorar con ello su calidad de vida.

### **Robots sociales para la ayuda en el cuidado de personas**

- *Fraunhofer IPA (Alemania)*. Care-O-Bot II, Robawork, Secur-O-Bot.
- *Gecko Systems (EEUU)*. CareBot, robot personal útil para el cuidado de niños y personas mayores.

### **Sistemas robotizados para mejora de la movilidad, accesibilidad y rehabilitación**

- *Cyberdyne (Japón)*. Empresa que desarrolla y comercializa el exoesqueleto Robot Suit HAL.
- *Univ. de Bremen (Alemania)*. Silla de ruedas robotizada Rolland.

---

---

# CAPÍTULO 7

---

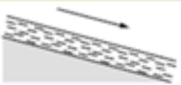




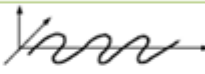






## LOCOMOCIÓN DE LOS ROBOTS MÓVILES

### 7.1. INTRODUCCIÓN

Un robot móvil necesita de mecanismos de locomoción que le permitan moverse sin límites a través de su medio ambiente. La mayoría de estos mecanismos de locomoción se han inspirado en sus homólogos biológicos (figura 7.1). Hay, sin embargo, una excepción: la rueda es una invención humana que logra un rendimiento extremadamente alto en terreno plano. Nuestro sistema de caminado bípedo puede ser aproximado por un polígono móvil, con lados iguales de longitud  $d$  de medida (figura 7.2). Sin embargo, replicar la naturaleza de los sistemas biológicos es extremadamente difícil por varias razones. Para empezar, la complejidad mecánica se consigue fácilmente en los sistemas biológicos a través de la replicación estructural. La división celular, en combinación con la especialización, puede producir fácilmente un milpiés con varios cientos de piernas y varias decenas de miles de cilios individualmente sensados. En las estructuras hechas por el hombre, cada parte debe ser fabricado de forma individual. Con un tamaño muy pequeño y el peso, los insectos alcanzan un nivel de solidez que no hemos sido capaces de igualar con las técnicas de fabricación humana.

Debido a estas limitaciones, los robots móviles en general, se mueven, ya sea usando mecanismos de ruedas (figura 7.2). En general, la locomoción de piernas requiere mayores grados de libertad y por lo tanto una mayor complejidad mecánica que la locomoción de ruedas. Las ruedas, además de ser sencillas, son extremadamente adecuadas

Figura 7.1: Mecanismos de locomoción utilizados en los sistemas biológicos

Tipo de movimiento	Resistencia al movimiento	Cinemática básica del movimiento
Flujo en un canal 	Fuerza Hidrodinámica	Remolinos 
Rastreo 	Fuerza de fricción	Vibraciones longitudinales 
Ondulante 	Fuerza de fricción	Vibraciones transversales 
Corriendo 	Perdida de energía cinética	Movimiento oscilatorio de un péndulo multibrazo 
Saltando 	Perdida de energía cinética	Movimiento oscilatorio de un péndulo multibrazo 
Caminando 	Fuerza gravitacional	Polígono de rodadura 

Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition, 2004

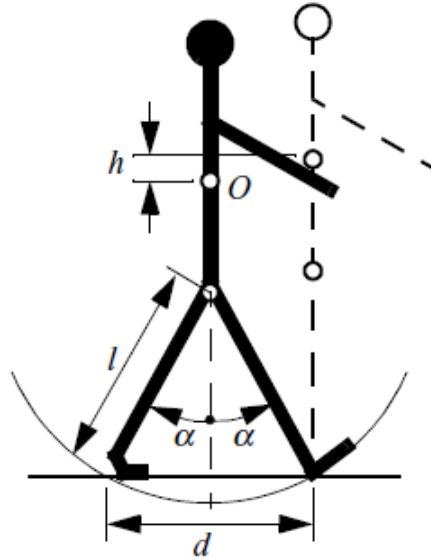
para terreno plano. Como muestra la (figura 7.3), sobre superficies planas la locomoción de ruedas es de uno a dos órdenes de magnitud más eficiente que la locomoción de patas.

## 7.2. GENERALIDADES DE LA LOCOMOCIÓN

### 7.2.1. Niveles en la locomoción

Hay dos aspectos importantes a tener en cuenta: el *control* y la *voluntariedad*. Para que el movimiento sea considerado locomoción el individuo tiene que querer realizarlo y además controlarlo. El estudio de la locomoción se divide en dos niveles. El *nivel inferior* es el encargado del control y la coordinación de los músculos (o actuadores en el caso de los robots) para que el individuo pueda desplazarse. El *nivel superior* se encarga de la planificación de trayectorias, navegación y otras tareas de mayor nivel. Está relacionado con la voluntariedad.

Figura 7.2: Un sistema de caminado bípedo puede ser aproximado por un polígono de rodadura, con pasos iguales de longitud  $d$ .



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition, 2004

### 7.2.2. Tipos de locomoción

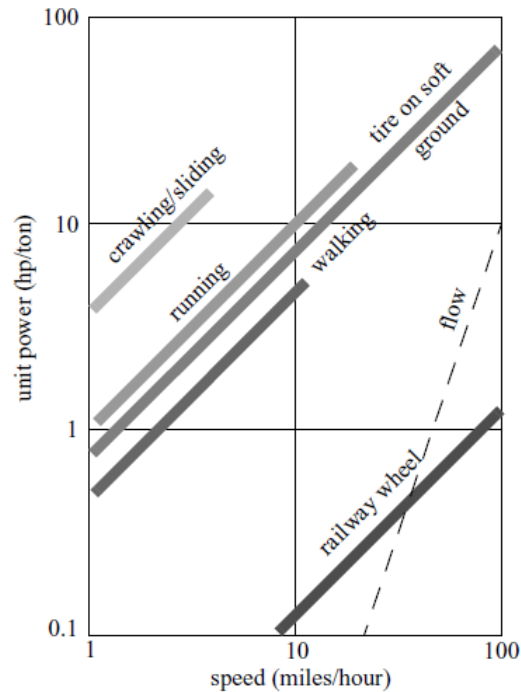
En la naturaleza, la locomoción de los animales se ha adaptado al medio en el que viven. Se puede realizar una primera clasificación según el medio en el que se desplazan. Así, la locomoción puede ser: aérea, acuática o terrestre. La locomoción terrestre se puede dividir a su vez en dos categorías, según los órganos empleados para realizar el desplazamiento: locomoción mediante patas (mamíferos, insectos) o mediante movimientos corporales (serpientes, orugas, lombrices).

### 7.2.3. Diseño y Locomoción de robots móviles

La locomoción de los individuos está especialmente adaptada al entorno en el que normalmente se desenvuelven, para el diseño de un robot móvil es esencial conocer el terreno por el que se va a desplazar. El proceso de diseño se podría resumir en los siguientes pasos:

1. Estudiar el entorno en el que se va a desplazar el robot.

Figura 7.3: La potencia específica en comparación con la velocidad alcanzada de varios mecanismos de locomoción



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition, 2004

2. Seleccionar los efectores
3. Implementar los modos de caminar.

#### 7.2.4. Las cuestiones clave para la locomoción

Uno de los grandes retos es el de desarrollar un robot que sea capaz de moverse por cualquier tipo de entorno por muy escarpado y complicado que sea. La locomoción es el complemento de la manipulación. En la manipulación, el brazo del robot está fijo pero mueve objetos en el espacio de trabajo, impartiendo la fuerza para ellos. En la locomoción, el medio ambiente es fijo y el robot se mueve impartiendo la fuerza al medio ambiente. La locomoción y manipulación por lo tanto comparten los mismos temas centrales de la estabilidad, las características de contacto, y el tipo del medio ambiente:

- Estabilidad

Figura 7.4: Ejemplos de robots con diferentes efectores para la locomoción terrestre



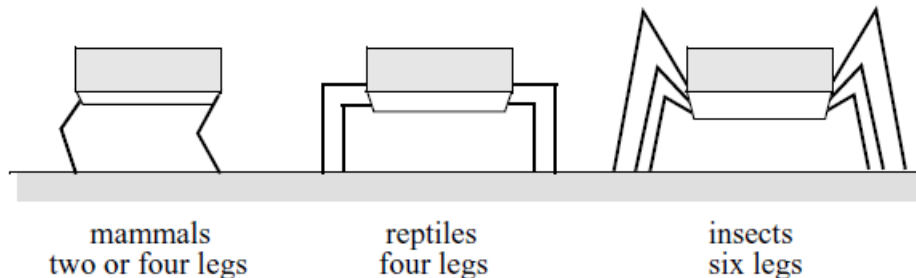
Fuente: Diseño de robots ápodos, Juan González Gómez

- Número y la geometría de puntos de contacto
- El centro de gravedad
- Estabilidad estática / dinámica
- La inclinación del terreno
- Características de contacto
  - Punto de contacto / tamaño y forma de la trayectoria
  - Ángulo de contacto
  - La fricción
- Tipo de ambiente
  - Estructura
  - Medio, (por ejemplo, agua, aire, suelo blando o duro)

## 7.3. ROBOTS MÓVILES CON PIERNAS

La locomoción de los robots con piernas se caracteriza por una serie de contactos puntuales entre el robot y el suelo. Las principales ventajas incluyen la capacidad de adaptación y maniobrabilidad en terrenos difíciles. Las principales desventajas de la locomoción con piernas incluyen la energía y la complejidad mecánica. La pierna, que puede incluir varios grados de libertad, debe ser capaz de sostener parte del peso total del robot, y, en muchos robots deben ser capaces de levantar y bajar el robot. Adicionalmente, una alta maniobrabilidad sólo se logrará si las piernas tienen un número suficiente de grados de libertad para impartir fuerzas en un número de direcciones diferentes.

Figura 7.5: Disposición de las piernas de varios animales



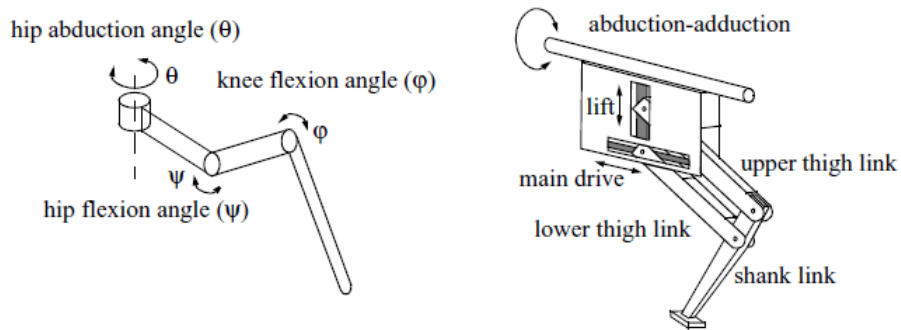
Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition, 2004

### 7.3.1. Configuraciones de las piernas y la estabilidad

Un número de configuraciones diferentes de las piernas han tenido éxito en una variedad de organismos (figura 7.5). Los animales grandes, como los mamíferos y los reptiles, tienen cuatro patas, mientras que los insectos tienen seis o más patas. En algunos mamíferos, la capacidad de caminar sobre dos piernas se ha perfeccionado. Especialmente en el caso de los humanos, el equilibrio ha evolucionado hasta el punto que incluso podemos saltar con una pierna. Esta maniobrabilidad excepcional tiene un precio: un control activo mucho más complejo para mantener el equilibrio a fin de lograr caminar pasivamente, un robot debe tener al menos seis patas. En tal configuración, es posible diseñar un modo de andar en el que un trípede estáticamente estable de las piernas esté en contacto con el suelo en todo momento (figura 7.8).

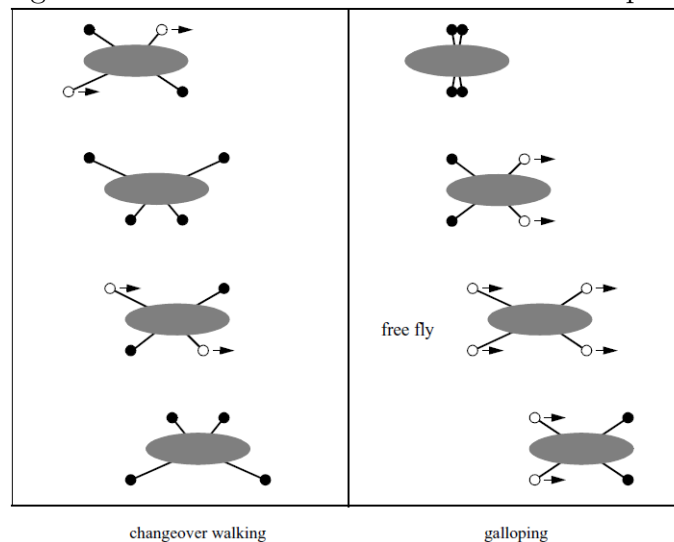
En el caso robots móviles de patas, se necesita un mínimo de dos grados de libertad generalmente para mover una pierna hacia adelante levantándola y balanceándola hacia adelante. Más común es la adición de un tercer grado de libertad para maniobras más complejas, resultando en piernas como las que se muestra en la figura 7.6. Los recientes éxitos en la creación de robots bípedos se han añadido un cuarto grado de libertad en la articulación del tobillo.

Figura 7.6: Dos ejemplos de piernas con tres grados de libertad



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition, 2004

Figura 7.7: Dos modos de marcha con cuatro patas.



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition, 2004

Las desventajas principales de articulaciones y actuadores adicionales son que los actuadores adicionales requieren de energía y control, y también añaden masa a la pata, aumentando aún más la energía y los requisitos de carga en los actuadores existentes. Para un robot móvil con las piernas  $k$ , el número total de eventos posibles  $N$  para una máquina caminante es:

$$N = 2k - 1! \quad (7.1)$$

Para un bípedo caminante  $k = 2$  piernas, el número de eventos posibles  $N$  es:

$$N = 2k - 1! = 3! = 3 \times 2 \times 1 \times = 6 \quad (7.2)$$

Los seis eventos diferentes:

1. Levantar la pierna derecha.
2. Levantar la pierna izquierda.
3. Soltar la pierna derecha.
4. Soltar la pierna izquierda.
5. Levantar las dos piernas juntas.
6. Soltar las dos piernas juntas.

Un robot con seis patas tiene mucho más modos teóricamente:

$$N = 11! = 39916800 \quad (7.3)$$

Las figuras 7.6 y 7.7 muestran varios modos de marche de cuatro patas y el modo de marcha trípode estático de seis patas.

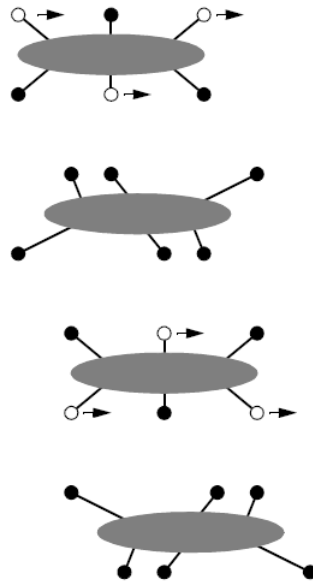
### 7.3.2. Ejemplos de locomoción de robots de piernas

Aunque no existen aplicaciones industriales de gran volumen hasta la fecha, la locomoción de piernas es un área importante de investigación a largo plazo. Varios diseños interesantes se presentan a continuación, comenzando con el robot con una sola pierna y terminando con robots de seis patas.

## Una pierna

Minimizar el número de piernas es beneficioso por varias razones. La masa corporal es especialmente importante para las máquinas andantes y una sola pierna minimiza la masa acumulada. La coordinación de la pierna es necesaria cuando un robot tiene varias patas, pero con una sola pierna tal coordinación no es necesaria. Un robot de una sola pierna requiere sólo una secuencia de los contactos individuales, lo que lo vuelve susceptible a los terrenos más difíciles. Para un robot con una pierna, no sólo la caminata estática es imposible, sino que la estabilidad estática cuando está quieto también es imposible.

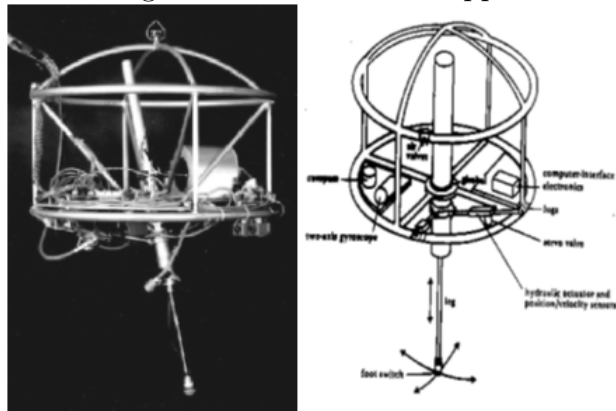
Figura 7.8: Caminar estático con seis patas. Un trípede formado por tres patas siempre existe



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition, 2004

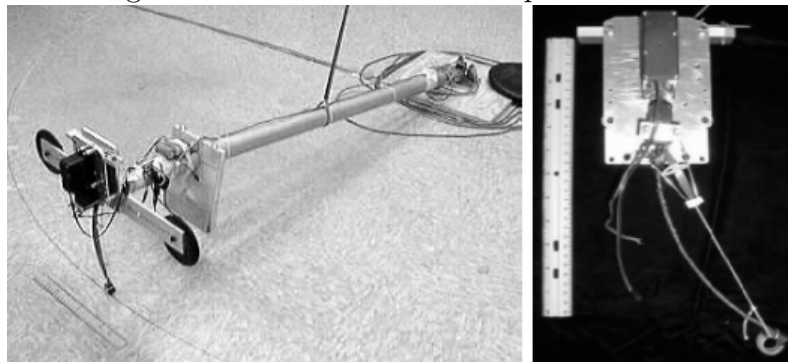
La figura 7.9 muestra al Raibert Hopper, uno de los más conocidos robots creados de una sola pierna. Este robot hace correcciones continuas a la postura del cuerpo y la velocidad del robot ajustando el ángulo de la pierna con respecto al cuerpo. El accionamiento es hidráulico, incluyendo la extensión longitudinal de alta potencia de la pierna durante el apoyo para saltar en el aire. Aunque poderosos, estos actuadores requieren una bomba hidráulica grande, para ser conectado al robot en cada momento. La figura 7.10 muestra un diseño más eficiente en energía desarrollada más recientemente.

Figura 7.9: El Raibert Hopper



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition,2004

Figura 7.10: El saltamontes de pierna de arco



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition,2004

## Dos piernas (Bípedos)

Los robots bípedos han demostrado correr, saltar, subir y bajar escaleras, e incluso hacer trucos aéreos, tales como saltos mortales. En el sector comercial, Honda y Sony han hecho avances significativos en la última década y han hecho posibles robots bípedos altamente capacitados. Ambas compañías diseñaron pequeñas y poderosas articulaciones, que alcanzan el rendimiento poder-peso sin precedentes en servomotores disponibles en el mercado.

El robot de Sony Dream, el modelo SDR-4X II, se muestra en la figura 7.11. Este robot con treinta y ocho grados de libertad tiene siete micrófonos para la localización

Figura 7.11: El Sony SDR-4X II



Especificaciones:

Peso: 7 kg

Altura: 58 cm

# DOF del cuello: 4

# DOF del cuerpo: 2

# DOF del brazo: 2 x 5

# DOF de las Piernas: 2 x 6

Manos de cinco dedos.

Fuente: <http://www.sony.net/>

Figura 7.12: Robot humanoide P2 de Honda, Japan



Especificaciones:

Velocidad máxima: 2 km/h

Autonomía: 15 min

Peso: 210 kg

Altura: 1,82 m

# DOF de la pierna: 2 x 6

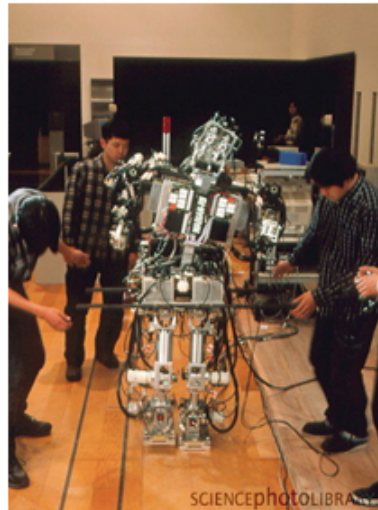
# DOF del brazo: 2 x 7

Fuente: <http://www.comunistrobot.com/>

de sonido, reconocimiento de personas basada en imagen, reconstrucción de un mapa a profundidad estéreo en miniatura a bordo, y reconocimiento de voz limitada. El proyecto humanoide de Honda tiene una historia importante, pero, de nuevo, han abordado el mismo reto de ingeniería muy importante de la actuación. La figura 7.12 muestra el modelo P2, que es un predecesor inmediato al más reciente modelo de Asimo. El mayor tamaño permite la movilidad práctica en el mundo humano de las escaleras y salientes,

manteniendo un tamaño y postura no amenazante.

Figura 7.13: El robot humanoide WABIAN-RIII en la Universidad Waseda en Japón



Especificaciones:

Peso: 131 kg  
Altura: 1,88 m  
DOF en total: 43  
Extremidades inferiores: 2 x 6  
Tronco: 3  
Brazos: 2 x 10  
Cuello: 4  
Ojos: 2 x 2

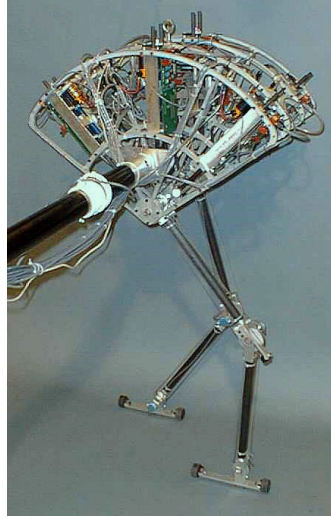
Fuente: [www.sciencephoto.com](http://www.sciencephoto.com)

WABIAN está diseñado para emular el movimiento humano, e incluso está diseñado para bailar como uno. Los robots bípedos sólo pueden ser estáticamente estables dentro de ciertos límites, y por lo tanto los robots como P2 y WABIAN (figura 7.13) en general deben llevar a cabo el balance correctivo con servos continuamente incluso cuando están inmóviles. Un elegante diseño de un robot bípedo es el Spring Flamingo del MIT (figura 7.14). Este robot inserta resortes en serie con los actuadores de las piernas para lograr un modo de marcha más elástico.

### **Cuatro patas (cuadrúpedos)**

Sony invirtió varios millones de dólares para desarrollar un robot de cuatro patas llamado AIBO (figura 7.15). Para crear este robot, Sony produjo tanto un nuevo sistema operativo de robot casi en tiempo real de servomotores orientados y que tienen un par motor lo suficientemente alto para soportar el robot. Además de desarrollar los motores y software personalizados, Sony incorporó un sistema de visión a color que permite a los AIBO perseguir una pelota de colores vivos. Sin embargo, el número de motores y la inversión en tecnología detrás de este perro robot dio lugar a un precio muy alto de aproximadamente \$1500. Los robots de cuatro patas tienen el potencial de servir como artefactos eficaces para la investigación en la interacción humano-robot (Figura 7.16).

Figura 7.14: El Spring Flamingo desarrollado en el MIT



Fuente: [www.ai.mit.edu](http://www.ai.mit.edu)

Figura 7.15: AIBO, el perro artificial de Sony, en Japón

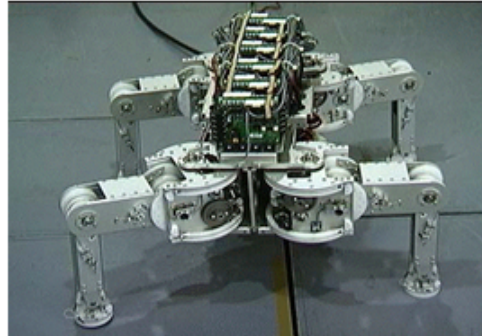


Fuente: <http://www.sony.net>

### **Seis patas (hexápodos)**

Las configuraciones de seis patas han sido muy populares en la robótica móvil, debido a su estabilidad estática durante la marcha, lo que reduce la complejidad del control (figura 7.17). En la mayoría de los casos, cada pata tiene tres grados de libertad, incluyendo la flexión de la cadera, flexión de la rodilla y abducción de la cadera (ver figura 7.6). Genghis es un robot de “Hobby” comercialmente disponible que tiene seis

Figura 7.16: Titan VIII, un robot cuadrúpedo desarrollado en el Tokyo Institute of Technology



Especificaciones:

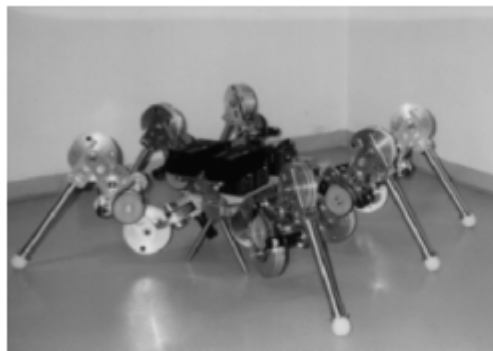
Peso: 1 kg 9  
Altura: 0,25 m  
DOF: 4 x 3

Fuente: <http://www-robot.mes.titech.ac.jp/>

patas, cada una de las cuales tiene dos grados de libertad proporcionados por los servos hobby (Figura 7.18). Este robot, que consta sólo de flexión de la cadera y la abducción de la cadera, tiene menos capacidad de maniobra en el terreno áspero, pero se desempeña muy bien en terreno plano.

Los insectos, que son posiblemente las criaturas más exitosas de locomoción en la tierra, sobresalen en recorrer toda clase de terrenos con seis patas. En la actualidad, la brecha entre las capacidades de los insectos de seis patas y los robots artificiales de seis patas es aún muy grande.

Figura 7.17: Lauron II, una plataforma hexápoda desarrollada en la Universidad de Karlsruhe, Alemania

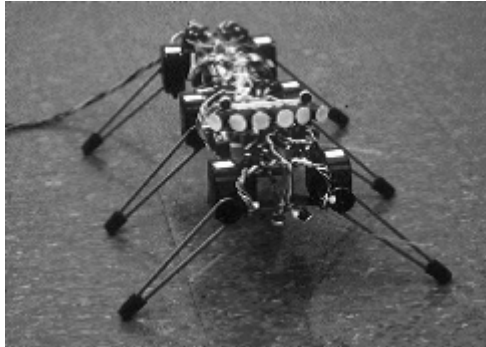


Especificaciones:

Velocidad máxima: 0,5 m / s  
Peso: 1 kg 6  
Altura: 0,3 m  
Longitud: 0,7 m  
N ° de piernas: 6  
DOF en total: 6 x 3  
Consumo de energía: 10 W

Fuente: <http://www.kit.edu/>

Figura 7.18: Genghis Khan, uno de los robots caminantes más famosos del MIT, utiliza servomotores comunes como sus actuadores



Fuente: <http://www.ai.mit.edu/projects/genghis>

## 7.4. ROBOTS MÓVILES CON RUEDAS

Se puede lograr una eficiencia muy buena, como se muestra en la figura 7.3, y lo hace con una ejecución mecánica relativamente simple. Además, el equilibrio no es generalmente un problema de investigación en los diseños de robot con ruedas, porque éstos están casi siempre diseñados de manera que todas las ruedas están en contacto con el suelo en todo momento. Por lo tanto, tres ruedas son suficientes para garantizar el equilibrio estable.

La investigación en robots con ruedas tiende a centrarse en los problemas de tracción y estabilidad, maniobrabilidad y el control: es decir que puedan las ruedas del robot proporcionar tracción y estabilidad suficientes para que el robot cubra todo el terreno deseado, y que la configuración de las ruedas del robot permita un control suficiente sobre la velocidad del robot.

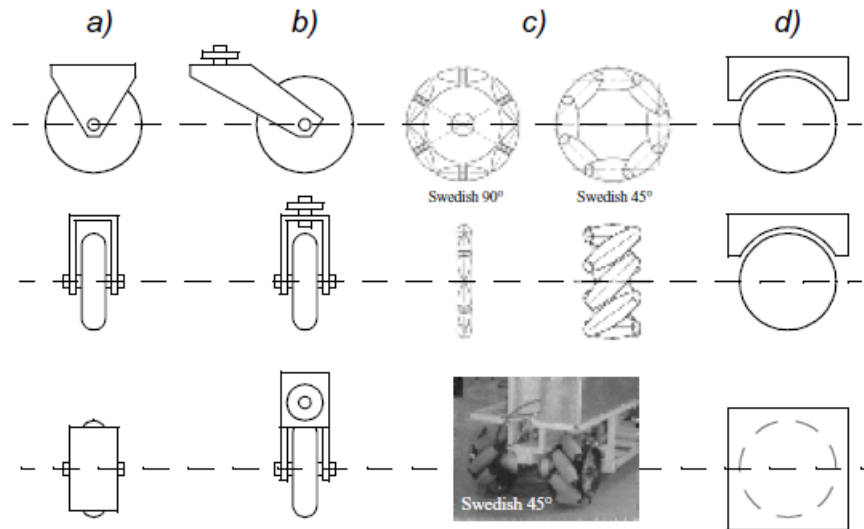
### 7.4.1. Locomoción de ruedas: el espacio de diseño

Hay un espacio muy grande de posibles configuraciones de la rueda cuando se tienen en cuenta las técnicas posibles para la locomoción del robot móvil. Es necesario hablar inicialmente de la rueda en detalle, ya que hay una serie de tipos de ruedas diferentes, con fortalezas y debilidades específicas.

#### Diseño de la rueda

Hay cuatro clases principales de la rueda, como se muestra en la figura 7.19. Ellas difieren ampliamente en sus cinemáticas, y por lo tanto la elección del tipo de rueda

Figura 7.19: Los cuatro tipos básicos de ruedas.



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition, 2004

tiene un gran efecto sobre la cinemática general del robot móvil. La rueda estándar y la rueda loca tienen un eje primario de la rotación y por lo tanto son muy direccionales. La diferencia fundamental entre estas dos ruedas es que la rueda estándar puede lograr este movimiento de dirección sin efectos secundarios, porque el centro de rotación pasa a través de la zona de contacto con el suelo, mientras que la rueda loca gira alrededor de un eje de desplazamiento, causando una fuerza para ser impartida al chasis del robot durante su conducción.

La rueda sueca y la rueda esférica son los dos diseños que no están tan limitados por la direccionalidad de la rueda convencional estándar. La rueda sueca funciona como una rueda normal, pero ofrece una baja resistencia en otra dirección también, a veces perpendicular a la dirección convencional, como en la sueca 90 y, a veces en un ángulo intermedio, como en la sueca 45. Los pequeños rodillos unidos alrededor de la circunferencia de la rueda son pasivos y el eje primario de la rueda sirve como la única articulación activamente accionada. La rueda esférica es una rueda verdaderamente omnidireccional, a menudo diseñada de modo que puede ser accionada para girar activamente a lo largo de cualquier dirección.

Figura 7.20: Navlab I, el primer vehículo de carretera autónomo que dirige y controla el acelerador usando sensores de visión y radar. Desarrollado en la CMU



Fuente: [www.cs.cmu.edu](http://www.cs.cmu.edu)



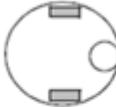




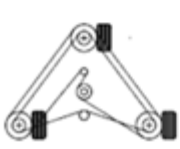
### **Geometría de las ruedas**

¿Por qué el tipo y la geometría de la rueda importan?. Tres características fundamentales de un robot se rigen por las siguientes opciones: la maniobrabilidad, la capacidad de control y estabilidad. Pocos robots utilizan la configuración de ruedas Ackerman de los automóviles debido a su maniobrabilidad pobre, con la excepción de los robots móviles diseñados para el sistema vial (figura 7.20). Las tablas 7.1, 7.2 y 7.3 ofrecen un resumen de las configuraciones de ruedas ordenados por el número de ruedas. A continuación, se identifican algunas de las principales ventajas y desventajas en términos de los tres temas que se han identificado anteriormente: estabilidad, maniobrabilidad y capacidad de control.

### **Estabilidad**

Un robot de accionamiento diferencial de dos ruedas puede lograr la estabilidad estática si el centro de masa está por debajo del eje de la rueda. Cye es un robot móvil comercial que utiliza ésta configuración (Figura 7.21). Sin embargo, en circunstancias normales esta solución requiere que los diámetros de las ruedas sean impracticablemente grandes. La dinámica también puede causar que un robot de dos ruedas golpee el suelo con un tercer punto de contacto, por ejemplo, con pares de motor lo suficientemente altos de parada. Convencionalmente, la estabilidad estática requiere un mínimo de tres ruedas.

Cuadro 7.1: Configuraciones de ruedas para vehiculos rodantes (2 y 3 ruedas)

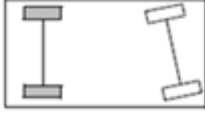
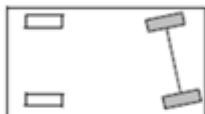
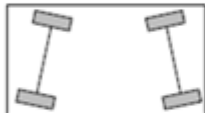

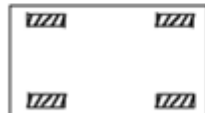


N° de Ruedas	Arreglo	Descripción	Ejemplos típicos
2		Una rueda orientable en la parte delantera, una rueda de tracción en la parte trasera	Bicicleta, motocicleta
		Conducción diferencial de dos ruedas motrices con el centro de masa (COM) por debajo del eje	Cye robot personal
3		Conducción diferencial de dos ruedas centradas con un tercer punto de contacto	Nomad Scout, samartRob EPFL
		Dos ruedas motrices de forma independiente en la parte trasera/delantera, una de las ruedas omnidireccional y sin motor en la parte delantera/trasera	Muchos robots en interiores, incluyendo el EPFL pygmalion robots y Alice
		Dos ruedas de tracción conectadas (diferencial) en la parte trasera, 1 rueda orientable libre en el frente	Mini-camiones Piaggio
		Dos ruedas libres en la parte trasera, una rueda de tracción orientable al frente	Neptuno (de la Universidad Carnegie Mellon), Hero-1
		Tres ruedas suecas o esféricas motrices dispuestas en un triángulo, el movimiento omnidireccional es posible.	Stanford rueda de Tribolo EPFL, Palm Pilot Robot Kit (CMU)
		Tres ruedas motrices y orientables de forma sincrónica, la orientación no es controlable.	"Unidad de sincronización" Denning MR2, Georgia Institute of Technology, I-Robot B24, Nomad 200

Fuente: Autores del proyecto

### Capacidad de control

En general, existe una correlación inversa entre la capacidad de control y la maniobrabilidad. El control de un robot omnidireccional para una dirección específica de recorrido es aún más difícil y, a menudo menos preciso en comparación con los dise-

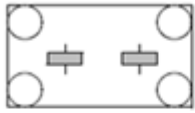
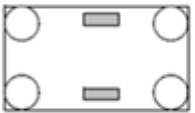







Cuadro 7.2: Configuraciones de ruedas para vehiculos rodantes (4 ruedas)

N° de Ruedas	Arreglo	Descripción	Ejemplos típicos
4		Dos ruedas motrices en la parte trasera, 2 ruedas orientables en el frente, la dirección tiene que ser diferente para las 2 ruedas para evitar el deslizamiento/derrape.	Coche con tracción trasera
		Dos ruedas motrices y orientables en la parte delantera, 2 ruedas libres en la parte trasera; la dirección tiene que ser diferente para las 2 ruedas para evitar el deslizamiento/derrape	Coche con tracción delantera
		Cuatro ruedas motrices y orientables	En las cuatro ruedas, cuatro ruedas de dirección de Hyperion (CUM)
		Dos ruedas de tracción (diferencial) en la parte trasera/delantera, 2 ruedas omnidireccionales en la parte delantera/trasera	Charlie (DMT-EPFL)
		Cuatro ruedas omnidireccionales	Carnegie Mellon Uranus
		Dos ruedas motrices con diferencial de 2 puntos de contacto adicionales.	EPFL Khepera, Hyperbot Chip
		Cuatro ruedas locas motrices y dirigibles.	Nomad XR4000

Fuente: Autores del proyecto

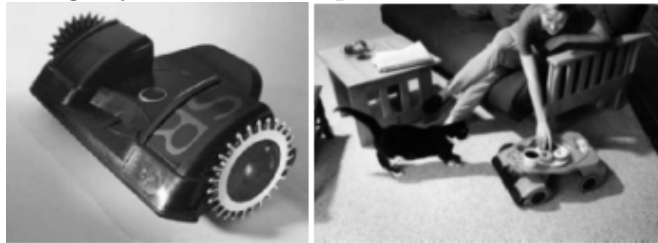
ños más maniobrables. No hay un “ideal” de configuración de conducción que a la vez maximice la estabilidad, maniobrabilidad y capacidad de control. Cada aplicación de robot móvil pone restricciones únicas sobre el problema de diseño del robot, y la tarea del diseñador es elegir la configuración más adecuada posible a partir de este espacio de compromisos.

Cuadro 7.3: Configuraciones de ruedas para vehiculos rodantes (6 ruedas)

Nº de Ruedas	Arreglo	Descripción	Ejemplos típicos
6		Dos ruedas motrices y dirigidas alineadas en el centro, una rueda omnidireccional en cada esquina	First
		Dos ruedas de tracción (diferencial) en el centro, una rueda omnidireccional en cada esquina	Terregator (Universidad Carnegie Mellon)
Iconos para cada uno de los tipos de ruedas son los siguientes:			
	Rueda omnidireccional sin motor (esférica, castor, swedish)		
	Rueda Motriz Sueca (Stanford rueda)		
	Rueda estándar sin alimentación		
	Rueda estándar motriz		
	Rueda castor dirigible y motriz		
	Rueda estándar orientable		
	Rueda conectada		

Fuente: Autores del proyecto

Figura 7.21: Cye, un robot doméstico disponible comercialmente que puede aspirar y hacer entregas en el hogar y es construido por Aethon Inc



Fuente: <http://www.aethon.com>

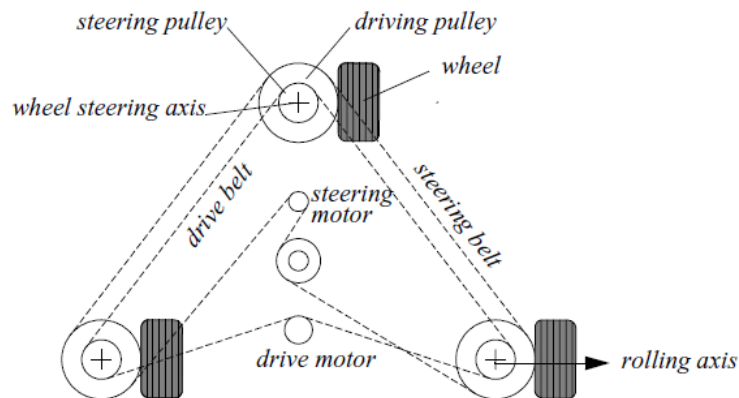
#### 7.4.2. Locomoción de ruedas: estudios del caso

A continuación se describen cuatro configuraciones específicas de ruedas.

## Configuración Síncrona

La configuración síncrona (figura 7.22) es un mecanismo popular de las ruedas en las aplicaciones de robots móviles en interiores. Aunque hay tres ruedas motrices y directrices, sólo dos motores se utilizan en total. El motor de traslación ajusta la velocidad de las tres ruedas juntas, y el motor de dirección hace girar todas las ruedas juntas sobre cada uno de sus ejes de dirección vertical individuales. La configuración síncrona es particularmente ventajosa en los casos en que se requiere omnidireccionalidad.

Figura 7.22: Configuración Síncrona



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition, 2004

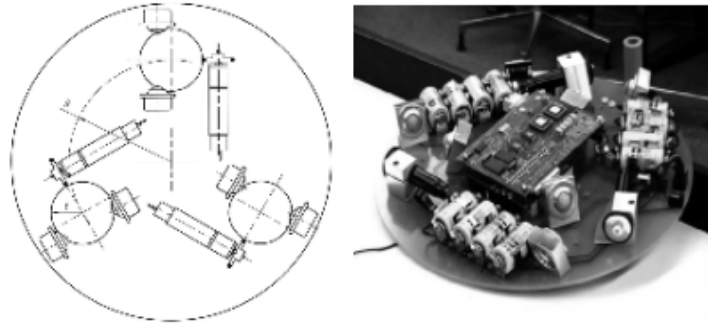
## Locomoción omnidireccional

Los robots omnidireccionales que son capaces de moverse en cualquier dirección  $(x, y, \theta)$  en cualquier momento, también son holonómicos. Se pueden realizar ya sea usando ruedas esféricas, ruedas locas o ruedas suecas. Tres ejemplos de tales robots holonómicos se presentan a continuación.

- **Locomoción omnidireccional con tres ruedas esféricas.** El robot omnidireccional representado en la figura 7.23 se basa en tres ruedas esféricas, cada una accionada por un motor. En este diseño, las ruedas esféricas están suspendidas por tres puntos de contacto, dados por dos cojinetes esféricos y una por una rueda conectada al eje del motor. Este concepto proporciona una excelente maniobrabilidad y es de diseño simple. Sin embargo, se limita a las superficies planas y

cargas pequeñas, y es muy difícil encontrar ruedas redondas con coeficientes de alta fricción.

Figura 7.23: El Tribolo diseñado en el EPFL (Instituto Federal Suizo de Tecnología de Lausana, Suiza)



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition, 2004

- **Locomoción omnidireccional con cuatro ruedas suecas.** La disposición omnidireccional que se muestra en la figura 7.24 se ha utilizado con éxito en varios robots de investigación, incluyendo el Carnegie Mellon Uranus. Esta configuración se compone de cuatro ruedas suecas de  $45^\circ$ , cada una accionada por un motor independiente. Al variar la dirección de rotación y velocidades relativas de las cuatro ruedas, el robot puede moverse a lo largo de cualquier trayectoria en el plano y, aún más impresionante, al mismo tiempo puede girar alrededor de su eje vertical.

Figura 7.24: El robot Carnegie Mellon Uranus, un robot omnidireccional con cuatro ruedas suecas motorizadas de  $45^\circ$



Fuente: <http://www.cs.cmu.edu/>

- **Locomoción omnidireccional con cuatro ruedas locas y ocho motores.** Otra solución para omnidireccionalidad es el uso de ruedas locas. Esto se hace

para el Nomad XR4000 de Nómadic Technologies (figura 7.25), dándole una excelente maniobrabilidad. Por desgracia, Nomadic ha cesado la producción de robots móviles.

Figura 7.25: Los Nomad XR4000 de Nomadic Technologies tenían un arreglo de cuatro ruedas locas para el movimiento holonómico.



Fuente: <http://cbcis.ttu.edu/>

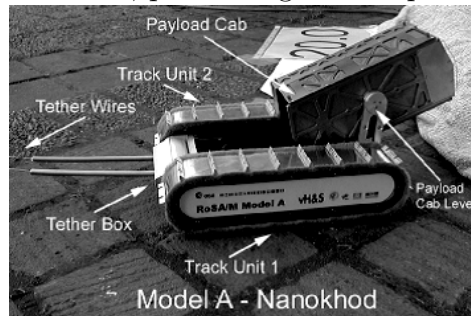
### **Locomoción de Deslizamiento/derrape tipo oruga**

Una forma alternativa de dirección, denominada deslizamiento/derrape, se puede utilizar para reorientar el robot haciendo girar las ruedas que se enfrentan a la misma dirección a diferentes velocidades o en direcciones opuestas. Un tanque de guerra opera de esta manera, y la Nanokhod (figura 7.26) es un ejemplo de un robot móvil basado en el mismo concepto. Los robots que hacen uso de la banda de rodadura tienen zonas de contacto mucho más grandes con la tierra, y esto puede mejorar significativamente su capacidad de maniobra. Sin embargo, debido a que esta zona de contacto con el suelo es grande, cambiar la orientación del robot por lo general requiere un giro de arrastre, en la que una gran parte de la pista debe deslizarse contra el terreno.

### **Ruedas caminates**

Los robots caminantes pueden ofrecer la mejor maniobrabilidad en terrenos difíciles. Sin embargo, son ineficaces en terreno plana y necesitan de control sofisticado. Las soluciones híbridas, que combina la adaptabilidad de las piernas con la eficiencia de las ruedas, ofrecen un interesante arreglo. Un robot móvil diseñado para aplicaciones similares fue recientemente producido por la EPFL (figura 7.27). Este robot, llamado

Figura 7.26: El microrover Nanokhod, desarrollado por von Hoerner y Sulger GmbH y el Instituto Max Planck, de Mainz, para la Agencia Espacial Europea (ESA)



Fuente: <http://www.esa.int/TEC/Robotics/>

Shrimp, tiene seis ruedas motrices y es capaz de escalar objetos de hasta dos veces el diámetro de su rueda. Esto le permite subir las escaleras regulares sin problema.

Figura 7.27: Shrimp, un robot todo terreno con excelentes habilidades pasivas de escalada (EPFL)



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition, 2004

En contraste, se muestra un Rover personal con un cambio activo en el COM para escalar obstáculos cuya medida es varias veces el diámetro de sus ruedas, como se puede ver en la figura 7.28. En este caso, el Rover personal logra un control de lazo cerrado por inferir terreno basado en mediciones de flujo de corriente a cada una de las ruedas manipuladas independientemente.

## 7.5. ROBOTS TREPADORES

### 7.5.1. Introducción

Son sistemas robóticos diseñados para moverse en entornos 2D o 3D complejos, tales como paredes, techos, cubiertas y estructuras geométricas para la realización de

Figura 7.28: El Rover personal, demostrando escalada de obstáculos usando un cambio continuo de COM



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition, 2011

tareas diversas. Por lo tanto, los robots trepadores pueden ser grandes alternativas que incrementen la eficiencia operativa y protejan la salud y la seguridad humana en las tareas peligrosas, tales como los siguientes: limpieza e inspección de edificios de gran altura, la evaluación y el diagnóstico de los tanques de almacenamiento en las centrales nucleares e instalaciones petroquímicas, mantenimiento y soldadura de los cascos de buques, entre otras. Los siguientes requisitos se deben observar para diseñar robots trepadores.

- Ligereza de peso, que es seguido por un bajo consumo de energía, para aumentar la autonomía y la carga útil del equipo auxiliar.
- La alta movilidad, que permite a los robots trepadores moverse sobre diferentes entornos, con diferentes geometrías y materiales tales como ladrillos, vidrio, cemento, acero, etc.
- Un mecanismo confiable para poder escalar en varios tipos de superficie.

Con respecto al mecanismo de locomoción, los robots trepadores pueden ser divididos en seis categorías:

- De patas.
- Impulsado por ruedas.
- Tipo oruga.
- De traslación.

- Impulsado por cable.
- Combinado.

Por otro lado, desde el punto de vista de adhesión, los robots escaladores se pueden clasificar en cinco categorías:

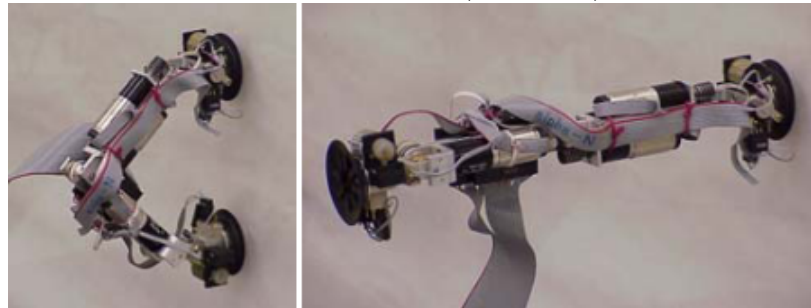
- De aspiración.
- De tipo magnético.
- Tipo de agarre.
- Guiados por carril.
- De tipo biomimético.

## 7.5.2. Mecanismos de locomoción de los robots trepadores

### Locomoción con piernas

Los robots escaladores que emplean locomoción de patas tienen desde dos hasta ocho patas que están equipadas con ventosas de vacío, pinzas de agarre, o dispositivos magnéticos en el extremo de los pies. Estos dispositivos de fijación permiten una adhesión fuerte y estable a la superficie. Dado que por lo general tienen una gran cantidad de grados de libertad, pueden moverse sobre superficies escabrosas y grietas, y son capaces de evitar eficazmente obstáculos. Sin embargo, requieren complicados sistemas de control.

Figura 7.29: RAMR 1 caminando (trepando) en una superficie

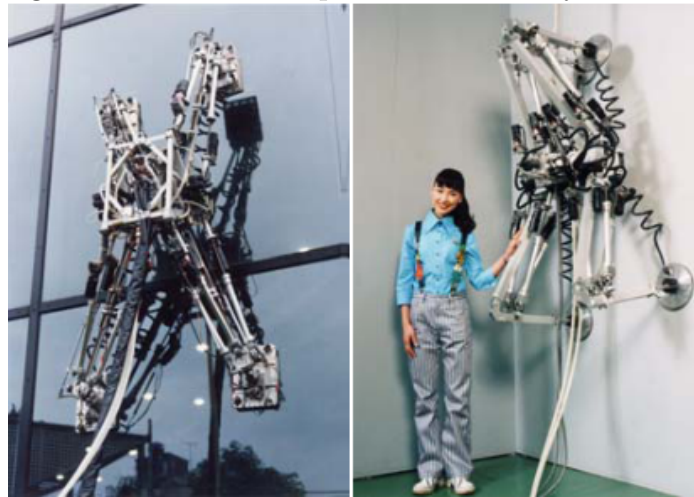


Fuente: A Survey of Climbing Robots: Locomotion and Adhesion, Baeksuk Chu, Kyungmo Jung, Chang-Soo Han and Daehie Hong, 2010

RAMR 1 es un robot bípedo con cuatro articulaciones, cinco conexiones, y un mecanismo de adhesión de succión en los extremos de sus patas (Figura 7.29). Cuando se requiere una mayor seguridad o mayor capacidad de carga útil, se adoptan robots cuadrúpedos trepadores. La figura 7.30 muestra los robots cuadrúpedos trepadores, Ninja-I y-II, que el Instituto de Tecnología de Tokio ha desarrollado con el propósito de la inspección de la fachada y mantenimiento de edificios altos y puentes. Estos se componen de:

- Un mecanismo de conexión paralela 3D capaz de producir la fuerza motriz para moverse en una superficie.
- Un mecanismo de paralelogramo conducido por cable para ajustar la postura de los tobillos.
- Una válvula de ventosa múltiple regulada por válvula capaz de generar fuerza, incluso en superficies irregulares.

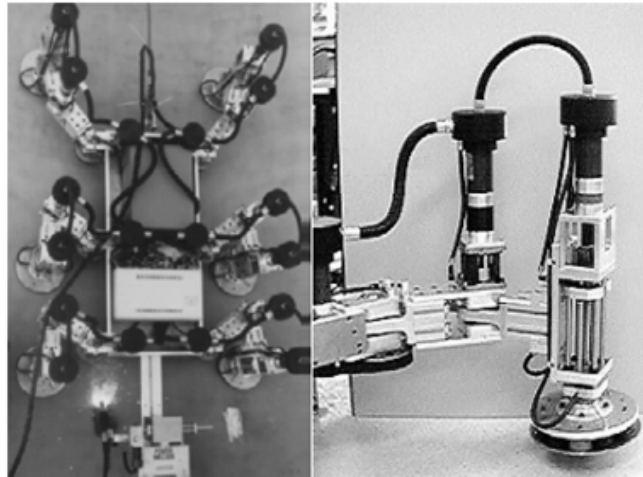
Figura 7.30: Robots trepadores NINJA I y NINJA II



Fuente: A Survey of Climbing Robots: Locomotion and Adhesion, Baeksuk Chu, Kyungmo Jung, Chang-Soo Han and Daehie Hong, 2010

El robot trepador REST I, que tiene seis patas tipo reptil con 3 DOFs cada una, fue originalmente propuesto para realizar la inspección, limpieza y tareas de soldadura del casco de un barco (figura 7.31). Las desventajas de una mayor estabilidad y capacidad de carga son el aumento de tamaño, complejidad y peso.

Figura 7.31: Robot REST I soldando un casco de barco



a) REST 1

b) Pinza electromanética.

Fuente: A Survey of Climbing Robots: Locomotion and Adhesion, Baeksuk Chu, Kyungmo Jung, Chang-Soo Han and Daehie Hong, 2010

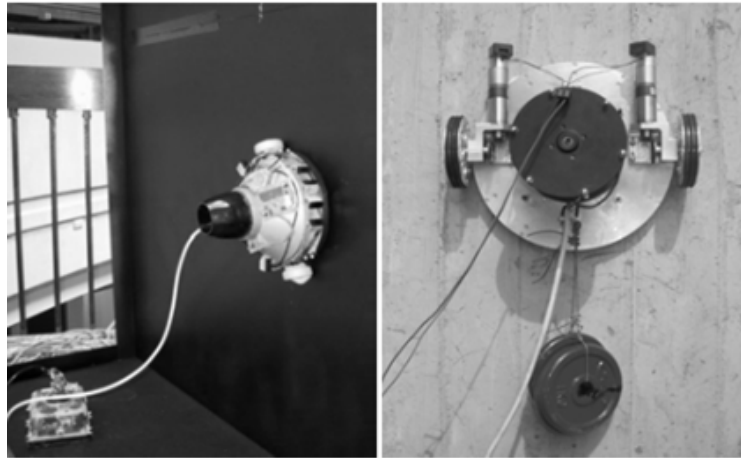
### Locomoción por ruedas

Los robots trepadores conducidos por ruedas suben planos verticales y techos mediante la combinación de ruedas para la traslación y la rotación; y bombas de vacío o imanes para la sujeción a la superficie. Por lo tanto, se mueven de forma continua y por ende, su velocidad puede ser mejorada considerablemente. Estos robots no pueden maniobrar sobre grandes obstáculos y su capacidad de carga es pequeña.

El objetivo de Alicia 1 y 2 es inspeccionar paredes verticales no porosas, tales como aquellas por encima del suelo en los tanques petroquímicos. Para mantener el funcionamiento del sistema independiente del material superficie, se utilizan dos ruedas motrices, ventosas, y un generador de vacío (figura 7.32). Están diseñados para pasar por encima de pequeños obstáculos (alrededor de 1 cm) a una velocidad normal y obstáculos más grandes a una menor velocidad. Mientras tanto, un robot trepador versátil de peso ligero ha sido diseñado como un vehículo de configuración síncrona de tres ruedas (figura 7.33). Es capaz de moverse en el casco de los buques para tareas como la inspección de soldaduras. El mecanismo de dirección sincronizada permite el movimiento en cualquier dirección sin cambio en la orientación.

Para un robot trepador que utiliza dos ruedas de dirección independientes y una copa de succión, el WallWalker, ha adoptado un sistema de vacío basado adhesión en húmedo (figura 7.34). Con la habilidad de cambiar continuamente su velocidad de viaje,

Figura 7.32: Prototipos de ALICIA 1 y 2

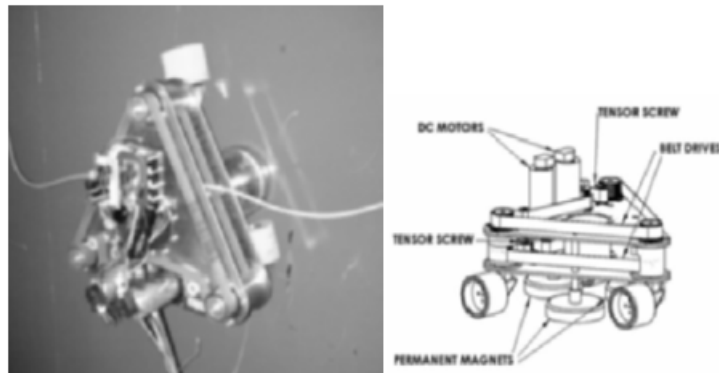


a) ALICIA 1

b) ALICIA 2

Fuente: A Survey of Climbing Robots: Locomotion and Adhesion, Baeksuk Chu, Kyungmo Jung, Chang-Soo Han and Daehie Hong, 2010

Figura 7.33: Vehículo de tres ruedas sincronizadas



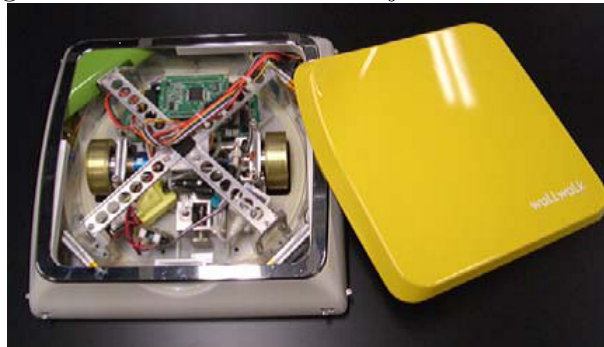
a) Robot en una superficie vertical

b) modelo CAD

Fuente: A Survey of Climbing Robots: Locomotion and Adhesion, Baeksuk Chu, Kyungmo Jung, Chang-Soo Han and Daehie Hong, 2010

el robot puede limpiar toda la suciedad en el cristal de la ventana usando el equipo de limpieza que está instalado en su cuerpo. Especialmente, este utiliza la acción de sellado y lubricación de un líquido para ser adherible y deslizable sobre la superficie.

Figura 7.34: WallWalker trabajando en la ventana

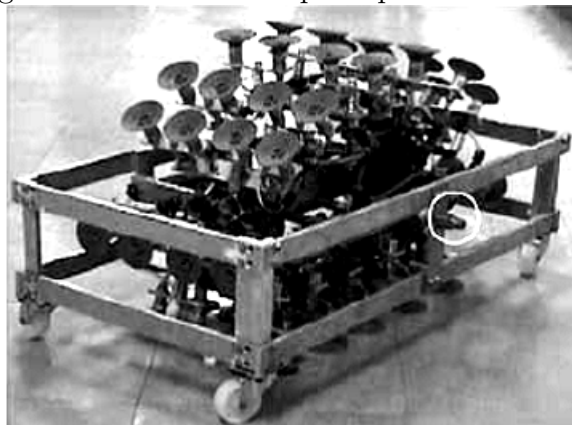


Fuente: A Survey of Climbing Robots: Locomotion and Adhesion, Baeksuk Chu, Kyungmo Jung, Chang-Soo Han and Daehie Hong, 2010

### **Locomoción de Oruga**

Los robots escaladores de oruga tienen la similitud con los robots trepadores de ruedas en que ambos se mueven con un mecanismo de rotación. Sin embargo, utilizando una cadena de oruga como mecanismo de locomoción, este tipo de robots trepadores son mejores a la hora de evitar los obstáculos y adherirse a la superficie.

Figura 7.35: Estructura principal de Cleanbot II

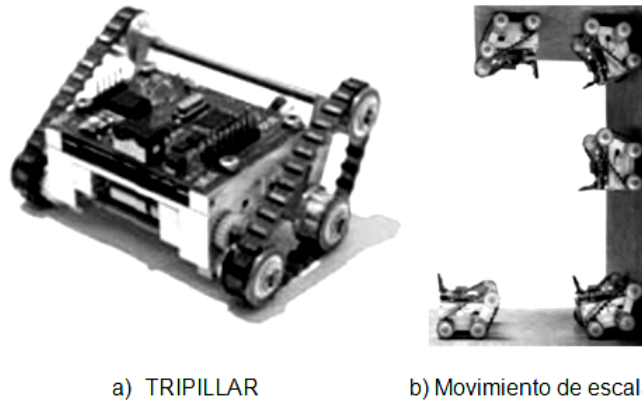


Fuente: A Survey of Climbing Robots: Locomotion and Adhesion, Baeksuk Chu, Kyungmo Jung, Chang-Soo Han and Daehie Hong, 2010

Cleanbot II, que emplea una cadena de oruga en la que las están instaladas 52 ventosas, virtualmente pueden lograr un movimiento continuo (figura 7.35). Las ventosas de vacío son controladas por válvulas de solenoide y suministran fuerza adhesiva para

hacer que el robot se pegue a la superficie del vidrio. Un robot trepador miniatura llamado TRIPILLAR utiliza orugas magnéticas en una forma triangular (figura 7.36). La fuerza adhesiva es proporcionada por una combinación de imanes pequeños que están en las orugas e imanes que están fijados al bastidor del robot.

Figura 7.36: Movimiento del TRIPILLAR desde el piso hasta el techo



Fuente: A Survey of Climbing Robots: Locomotion and Adhesion, Baeksuk Chu, Kyungmo Jung, Chang-Soo Han and Daehie Hong, 2010

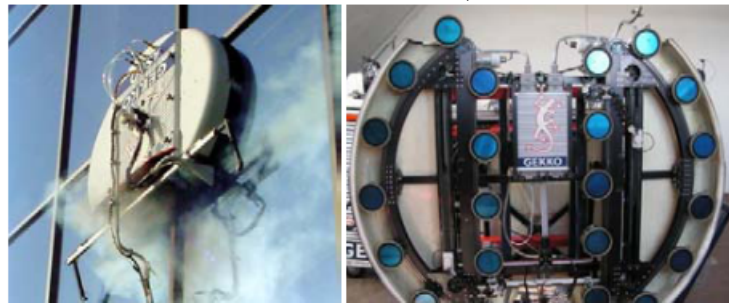
Recientemente, un robot trepador comercializado, GEKKO III, ha sido desarrollado por una empresa alemana, ARGECO. Utiliza dos pistas planas como mecanismo de locomoción y una serie de copas de succión para la unión a las paredes de cristal, como se muestra en la figura 7.37. Se pueden limpiar superficies de vidrio y marcos sin dejar manchas de agua y trabajar para marcos de hasta 4 cm de espesor.

### Locomoción de traslación

La estrategia de control y el proceso para operar robots trepadores de traslación no son complicados debido al fácil movimiento que comprende este movimiento por adherencia. Sin embargo, hay algunas desventajas tales como el gran tamaño que les impide ser utilizado en un espacio estrecho. Los Sky Cleaner 1, 2, y 3 son robots trepadores representativos de traslación. Estos robots que se desarrollan para la limpieza de paredes de vidrio son accionados por cilindros neumáticos para la traslación, ellos son absorbidos contra la pared de cristal por medio de garras de vacío (figura 7.38).

Recientemente, se ha desarrollado un robot trepador para realizar la operación de “Sand Blasting” en los astilleros. El robot adopta una plataforma deslizante doble que utiliza imanes permanentes para la adherencia (Figura 7.39). El sistema se basa en dos

Figura 7.37: Robot trepador GECKO III (comercializado ampliamente)

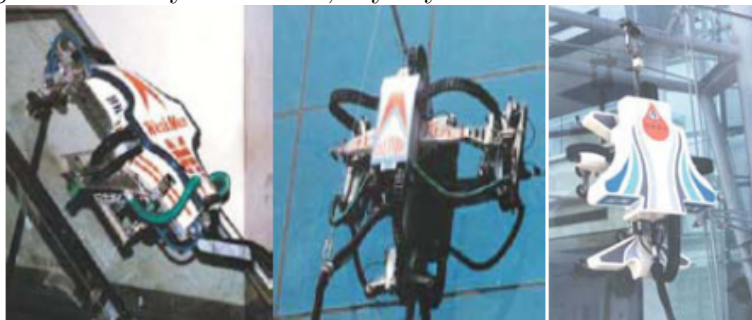


a) GECKO III

b) Trayectoria plana

Fuente: A Survey of Climbing Robots: Locomotion and Adhesion, Baeksuk Chu, Kyungmo Jung, Chang-Soo Han and Daehie Hong, 2010

Figura 7.38: Sky Cleaner 1, 2 y 3 y el mecanismo de traslación



a) Sky cleaner 1

b) Sky cleaner 2

c) Sky cleaner 3



d) Mecanismo de traslación de los robots Sky cleaner 1, 2 y 3.

Fuente: A Survey of Climbing Robots: Locomotion and Adhesion, Baeksuk Chu, Kyungmo Jung, Chang-Soo Han and Daehie Hong, 2010

módulos que se pueden mover relativamente entre sí y puede moverse hacia arriba y a lo largo del costado del barco con cualquier inclinación durante el granallado.

Figura 7.39: Grit blasting robot

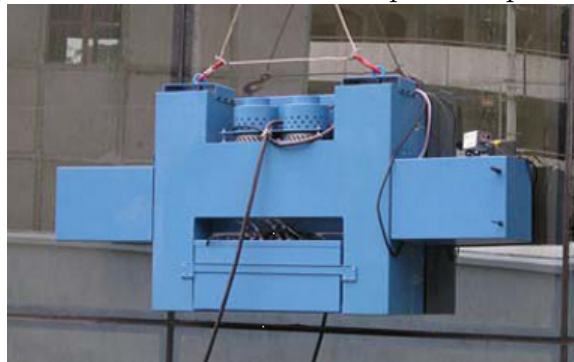


Fuente: A Survey of Climbing Robots: Locomotion and Adhesion, Baeksuk Chu, Kyungmo Jung, Chang-Soo Han and Daehie Hong, 2010

### **Locomoción por medio de cables**

Un sistema de cable está equipado con un carro en el techo de un edificio o estructura, puede realizar movimientos verticales y horizontales y sostener el cuerpo del robot, al mismo tiempo. Se diseñó y realizó un robot trepador soportado por cable con el propósito de limpieza de paneles de vidrio. El robot no tiene su propio mecanismo de conducción, pero puede moverse sobre superficies de cristal liso al depender de la fuerza de gravedad y de elevación de la grúa en el techo, mientras que se adhiere a las superficies utilizando dos ventosas de vacío (Figura 7.40).

Figura 7.40: Robot escalador soportado por cables



Fuente: A Survey of Climbing Robots: Locomotion and Adhesion, Baeksuk Chu, Kyungmo Jung, Chang-Soo Han and Daehie Hong, 2010

Figura 7.41: Robots trepadores por cables



a) Taisei Corporation      b) Kajima Construction.

Fuente: A Survey of Climbing Robots: Locomotion and Adhesion, Baeksuk Chu, Kyungmo Jung, Chang-Soo Han and Daehie Hong, 2010

La figura 7.41 muestra dos ejemplos de robots trepadores para pintura e inspección de muros, desarrollados por Taisei Corporation y Kajima Construction, respectivamente. Los robots son de tipo rectangular y consisten en un carro vertical que está coronado por un efector final. El carro se mueve hacia arriba y hacia abajo, cubriendo las zonas verticales.

### **Locomoción combinada**

La última posibilidad para la aplicación de sistemas de locomoción es la combinación de los mecanismos de la locomoción antes mencionados para mejorar la capacidad para escalar. Un robot automático de limpieza de fachada, SIRIUSc, ha sido desarrollado para su uso en edificios de gran altura. El robot se mantiene por los cables de una grúa en la cima de una fachada, el cual baja por el lado (Figura 7.42). El sistema de la grúa con cables desempeña el papel de mover al SIRIUSc verticalmente y lateralmente desde un panel de la fachada al siguiente, el cual es el primer mecanismo de locomoción.

Alicia 3 es una nueva versión de Alicia 1 y 2 mencionados anteriormente para mejorar el rendimiento al pasar por encima de obstáculos más grandes. Alicia 3 conecta los tres módulos de adhesión por medio de dos varillas y una articulación de rotación especial, como se muestra en la figura 7.43. Como resultado, este mecanismo combina tanto la locomoción de ruedas como locomoción de patas y permite que el robot se mueva hacia arriba y abajo para superar obstáculos de 10 - 12 cm de altura.

Figura 7.42: El robot de limpieza de fachada SYRIUSc

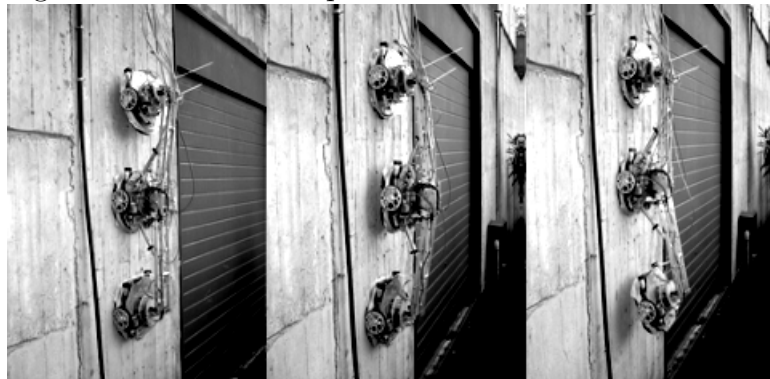


a) SYRIUSc

b) Sistema de Grúa

Fuente: A Survey of Climbing Robots: Locomotion and Adhesion, Baeksuk Chu, Kyungmo Jung, Chang-Soo Han and Daehie Hong, 2010

Figura 7.43: ALICIA 3 pasando a través de los obstáculos



Fuente: A Survey of Climbing Robots: Locomotion and Adhesion, Baeksuk Chu, Kyungmo Jung, Chang-Soo Han and Daehie Hong, 2010

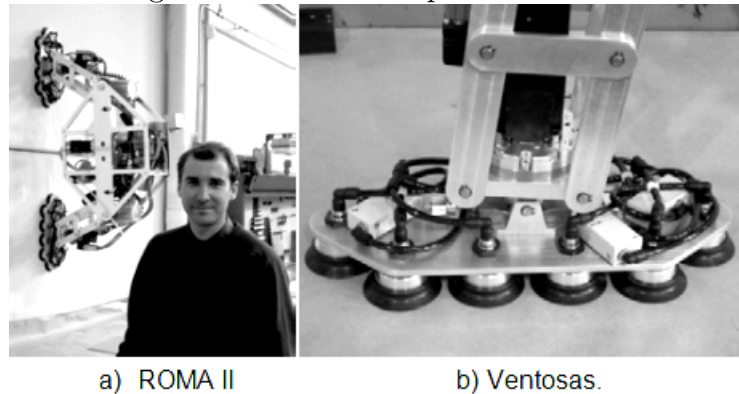
### 7.5.3. Mecanismos de adhesión de robots trepadores

#### Adhesión por succión y propulsión

La succión de vacío, el cual es el método de adherencia más comúnmente usado, puede ser ampliamente adoptada para superficies menos rugosas, ya que permite fuerte adhesión a cualquier superficie. La principal desventaja está relacionada con el mecanismo de adhesión y es que cualquier ranura en el sellado o la pared pueden causar la caída del robot. El uso de más de una ventosa, como en el mecanismo de locomoción de orugas, puede ser una solución para evitar la pérdida de presión y fuerza adhesiva

debido a las irregulares en la superficie. El vacío puede generarse a través del principio Ventri o una bomba de vacío que esté ya sea a bordo o externa al robot. ROMA II está diseñado para inspeccionar entornos complejos en 3D con una cinemática de 4 grados de libertad (Figura 7.44). A fin de lograr la movilidad para visitar todas las caras de las estructuras metálicas, ROMA II tiene dos piernas y un mecanismo de agarre manipulado neumáticamente.

Figura 7.44: Robot trepador ROMA II



Fuente: A Survey of Climbing Robots: Locomotion and Adhesion, Baeksuk Chu, Kyungmo Jung, Chang-Soo Han and Daehie Hong, 2010

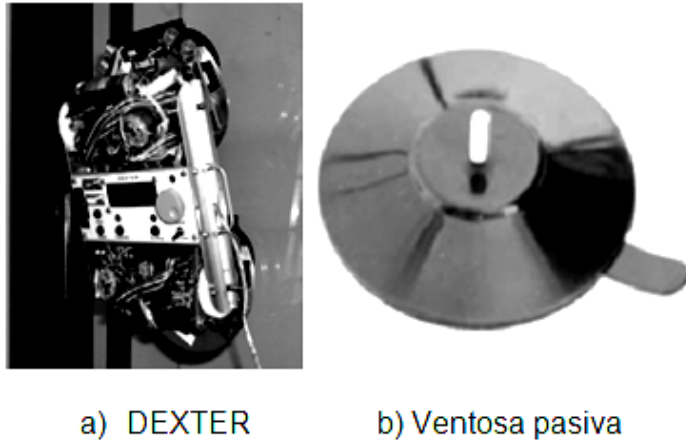
Escalar sin una bomba de vacío, el robot escalador DEXTER utiliza “Succión pasiva” para adherirse a la superficie (figura 7.45). En general, las ventosas de succión son vaciadas continuamente por al menos una bomba de vacío que está montada sobre el robot. Esto se llama “Succión activa”. Sin embargo, Dexter utiliza Ventosas de succión pasivas que están hechas de material elástico y son vaciadas simplemente presionándolas contra la superficie.

Para mantener el robot de limpieza de muros basado en propulsión, TITO, en posición vertical, se usa una grúa, que puede moverse a lo largo del borde exterior del techo de los edificios (figura 7.46). Un robot que cuelga de un cable es un sistema clásico de oscilación (péndulo). A fin de superar su falta de estabilidad, el ya mencionado robot escalador, SIRIUSc, ha empleado ventosas de succión junto con un mecanismo de marco deslizante.

### **Adhesión magnética**

Las soluciones de adhesión magnéticas, incluyendo magnetización permanente o magnetización eléctrica, son adecuadas para la fijación a sólo superficies ferromagné-

Figura 7.45: DEXTER escalando una ventana



Fuente: A Survey of Climbing Robots: Locomotion and Adhesion, Baeksuk Chu, Kyungmo Jung, Chang-Soo Han and Daehie Hong, 2010

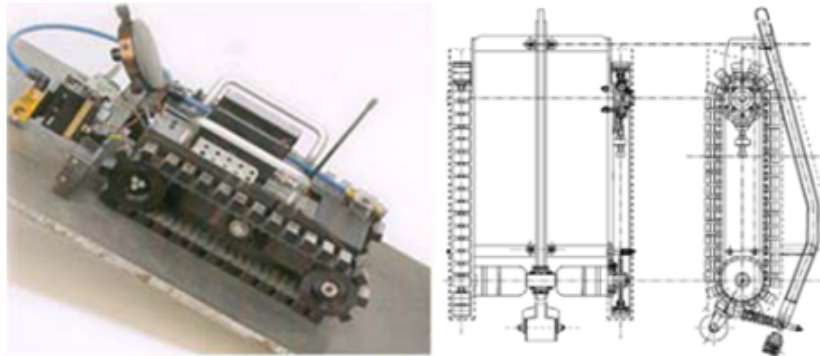
Figura 7.46: Robot limpiador TITO



Fuente: A Survey of Climbing Robots: Locomotion and Adhesion, Baeksuk Chu, Kyungmo Jung, Chang-Soo Han and Daehie Hong, 2010

ticas, pero son altamente deseables debido a su confiabilidad inherente. Sin embargo, dado que un mecanismo de adhesión magnética no requiere tiempo para generar una fuerza adhesiva suficiente, a diferencia del mecanismo de adherencia basado en succión, esto permite una rápida locomoción.

Figura 7.47: Robot escalador de pared para la inspección de tanques de aceite



a) Cuerpo entero.

b) Cadenas cinemáticas.

Fuente: A Survey of Climbing Robots: Locomotion and Adhesion, Baeksuk Chu, Kyungmo Jung, Chang-Soo Han and Daehie Hong, 2010

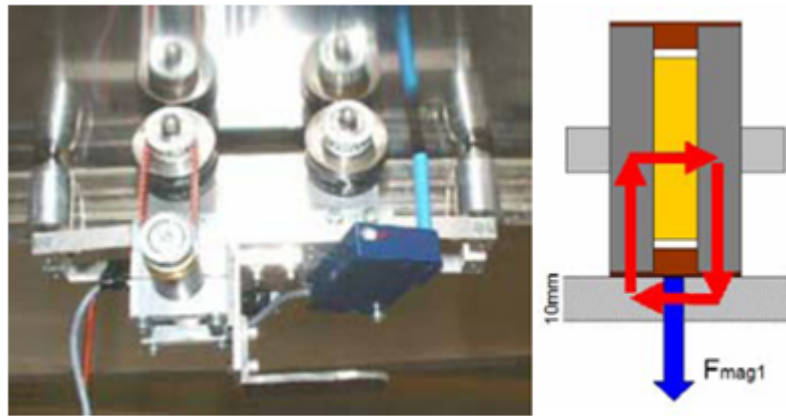
Se muestra un robot escalador de muro. Cada pista se compone de una cadena de rodillos, dos ruedas dentadas, y algunas unidades magnéticas permanentes dispuestas uniformemente, como se muestra en la figura 7.47. Cuando el robot se mueve, siempre hay un cierto número de unidades en buen contacto con la superficie, lo que permite al robot para permanecer de forma fiable en ella, mientras lleva una carga de más de 30 kg.

El Instituto Federal Suizo de Tecnología de Zurich ha desarrollado un robot trepador móvil sobre ruedas magnéticas para la inspección de las superficies interiores de los tanques de gasolina que están hechos de láminas metálicas delgadas como se muestra en la figura 7.48. Las ruedas magnéticas se componen de un imán cilíndrico en el medio, dos placas de acero magnético y de diámetro ligeramente más grande en cada lado, y una fina capa de caucho alrededor de las placas de acero para aumentar la fricción a lo largo del suelo.

### **Adhesión usando equipamiento de agarre**

Se han sugerido mecanismos de agarre para la adhesión para permitir a los robots escaladores viajar a lo largo de entornos complejos 3D, mientras que otros mecanismos de adhesión son generalmente aplicables para escalar en paredes y techos planos. Estos mecanismos de agarre, están conectados generalmente a las partes de las estructuras como vigas, columnas, tuberías, conductos, e incluso entornos naturales, a través de un cuidadoso control de las fuerzas de agarre.

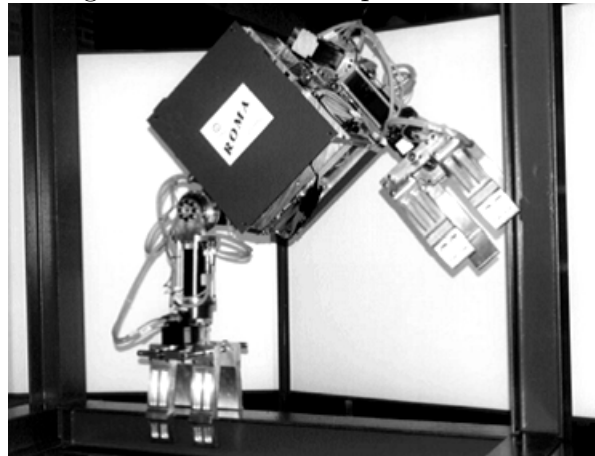
Figura 7.48: Prototipo de robot trepador móvil



a) Robot trepador con ruedas magnéticas. b) Estructura de la rueda.

Fuente: A Survey of Climbing Robots: Locomotion and Adhesion, Baeksuk Chu, Kyungmo Jung, Chang-Soo Han and Daehie Hong, 2010

Figura 7.49: Robot trepador ROMA I



Fuente: A Survey of Climbing Robots: Locomotion and Adhesion, Baeksuk Chu, Kyungmo Jung, Chang-Soo Han and Daehie Hong, 2010

ROMA I es un robot trepador multifuncional autónomo y autosuficiente que puede viajar en un ambiente complejo de metal en 3D que forma parte de las vigas y columnas. ROMA I tiene dos pinzas para realizar las tareas de agarre. Las pinzas rodean la viga para crear un alto grado de fuerza de fricción para evitar el deslizamiento y apoyar el sistema de locomoción para el movimiento 3D, como se muestra en la figura 7.49. El

robot ESTERAS se puede mover de una habitación a otra o desde un entorno estático (paredes, tablas, etc) hasta sillas de ruedas, o viceversa, para escalar como se muestra en la figura 7.50. Tiene un sistema simétrico con 5-DOF y un mecanismo especial de agarre compuesto de “estaciones de acoplamiento” (DSs), que se colocan en el entorno y al final del robot escalador.

Figura 7.50: Robot MATS y su estación de acoplamiento (DS)



a) El robot MATS

b) (DS)

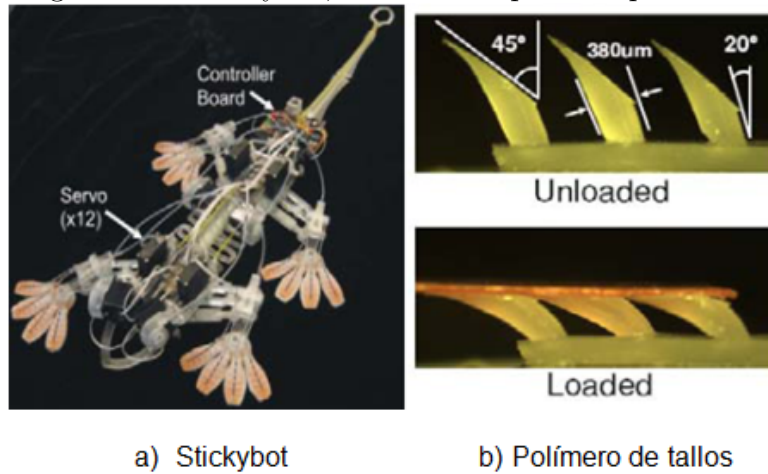
Fuente: A Survey of Climbing Robots: Locomotion and Adhesion, Baeksuk Chu, Kyungmo Jung, Chang-Soo Han and Daehie Hong, 2010

### **Adhesión biomimética**

Recientemente, un gran número de investigadores se han centrado en la capacidad de adherencia de los geckos. Se adhieren a las superficies utilizando los parches de pelos microscópicos que proporcionan un mecanismo para la adherencia en seco a través de fuerzas van der Waals. Dado que la adhesión en seco se debe principalmente a fuerzas moleculares, los geckos tienen la capacidad de adherirse a casi cualquier superficie, ya sea seca o húmeda, lisa o rugosa. Sin embargo, tienen algunas desventajas, tales como una carga útil muy baja y sensibilidad a las condiciones de superficie que implican el polvo.

Stickybot es un robot inspirado en la biología que trepa superficies lisas y verticales, tales como vidrio, plástico y cerámica a 4 cm/s. Los dedos del Stickybot están cubiertos con matrices de polímero en forma de tallos pequeños inclinados, que inducen fuerzas adhesivas con respecto a la superficie (figura 7.51). Al igual que con las estructuras adhesivas direccionales utilizadas por los geckos, los dedos se adhieren fácilmente cuando

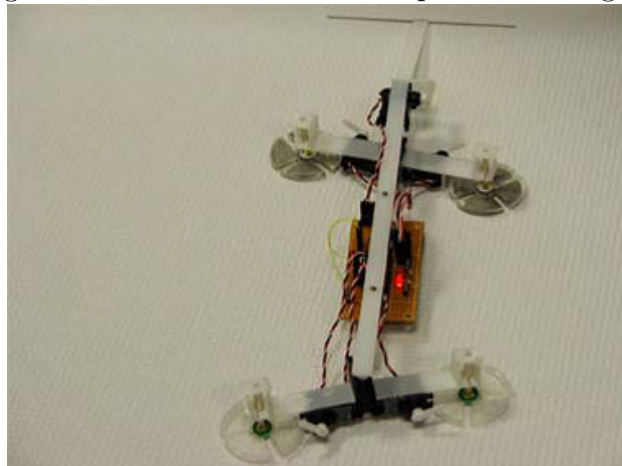
Figura 7.51: Stickybot, un robot trepador experimental



Fuente: A Survey of Climbing Robots: Locomotion and Adhesion, Baeksuk Chu, Kyungmo Jung, Chang-Soo Han and Daehie Hong, 2010

se hala tangencialmente desde las puntas hacia los tobillos y se liberan cuando se hala en la dirección opuesta.

Figura 7.52: Robot escalador inspirado en un gecko

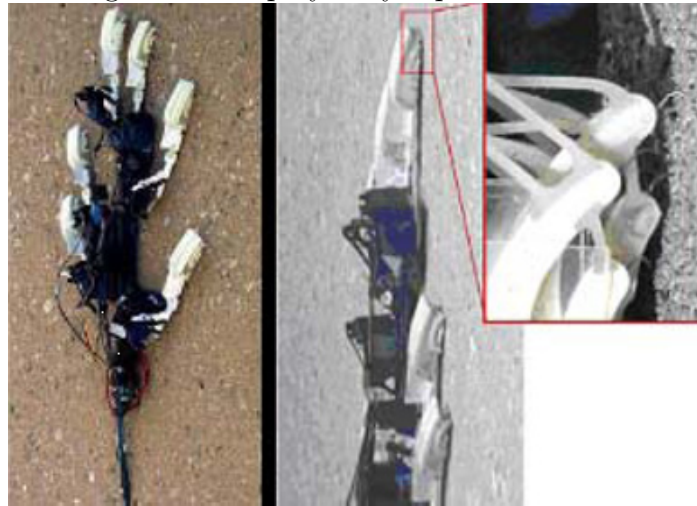


Fuente: A Survey of Climbing Robots: Locomotion and Adhesion, Baeksuk Chu, Kyungmo Jung, Chang-Soo Han and Daehie Hong, 2010

Otro robot escalador inspirado en un gecko, es el Geckobot, que utiliza un mecanismo sintético de adhesión en seco, y se centra más en la cinemática de las patas que son similares a la marcha de escalada de un gecko (Figura 7.52). Un robot escalador de 400

gramos, el Spinybot, fácilmente puede trepar por superficies duras como ladrillos de concreto y estuco y muros de piedra arenisca. Emplea una serie de espinas miniatura que se aferran en las asperezas de la superficie (Figura 7.53).

Figura 7.53: Spinybot y espinas miniatura



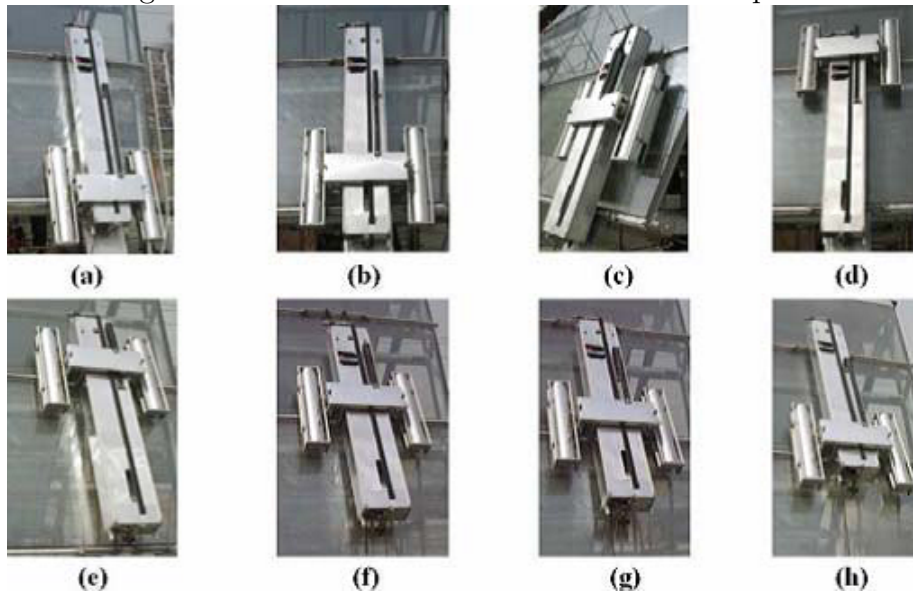
Fuente: A Survey of Climbing Robots: Locomotion and Adhesion, Baeksuk Chu, Kyungmo Jung, Chang-Soo Han and Daehie Hong, 2010

### **Adhesión guiada por carril**

En la construcción de un edificio, la base para el diseño de un mecanismo de adhesión para los robots escaladores puede ser establecida de antemano. Uno de los métodos representativos es instalar un carril para el deslizamiento del robot. El carril puede ser utilizado como agarre para evitar ser retirado de la superficie, así como para una guía a lo largo de la cual el robot puede deslizarse. O, por el contrario, para sujetarse ellos mismos, los robots escaladores pueden utilizar activamente algún tipo de estructura de carril como la que ya está instalada en la pared.

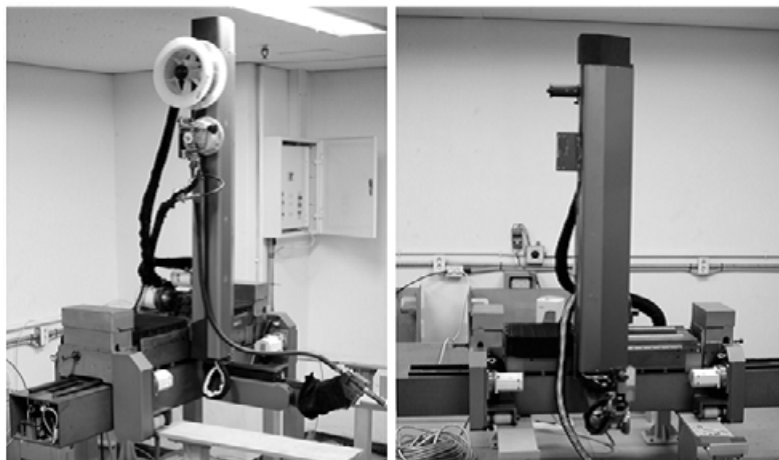
Un nuevo tipo de robot trepador de pared exterior se propone para la limpieza de las superficies curvas externas y complejas del Gran Teatro Nacional de China (Figura 7.54). El robot puede subir automáticamente en la dirección vertical y limpiar en la dirección lateral. El “Rail Runner (RRX3)” un robot móvil de soldadura, es capaz de moverse libremente en ambas direcciones, transversal y longitudinal, y realiza tareas de soldadura en las estructuras de doble casco. El mecanismo de movimiento del RRX, que es capaz de moverse libremente en estructuras de doble casco, tiene dos funciones: moverse en una dirección longitudinal a través de ruedas motrices que utilizan las

Figura 7.54: Pruebas de un robot escalador de pared



Fuente: A Survey of Climbing Robots: Locomotion and Adhesion, Baeksuk Chu, Kyungmo Jung, Chang-Soo Han and Daehie Hong, 2010

Figura 7.55: Mecanismo de movimiento de RRX en dirección longitudinal



Fuente: A Survey of Climbing Robots: Locomotion and Adhesion, Baeksuk Chu, Kyungmo Jung, Chang-Soo Han and Daehie Hong, 2010

caras longitudinales como “carriles” y se mueva en una dirección transversal por deslizamiento a lo largo de las caras longitudinales de apoyo utilizando brazos de extensión (Figura 7.55).

## Aplicaciones de los robots escaladores

Los sistemas robóticos pueden realizar tareas diversas, incluso en entornos muy peligrosos que están a menudo acompañados de grandes alturas, altas temperaturas, altas presiones, radiación, etc. Campos industriales donde los robots trepadores pueden resultar útiles se clasifican de la siguiente manera: construcción, infraestructura civil, plantas petroquímicas, centrales nucleares, construcción naval, inspección de aeronaves, sectores de servicios, y así sucesivamente. La Tabla 7.4 presenta una lista completa de robots trepadores de acuerdo a las clases antes mencionadas con las tareas de aplicación.

Cuadro 7.4: Lista de los robots escaladores de acuerdo a los campos industriales, con tareas de la aplicación

Field	Robot	Manufacturer	Country	Application	Locomotion	Adhesion
Construction industry	ROMA I, II	University Carlos III of Madrid	Spain	Inspection	Legged	Gripping
	ROBIN	Vanderbilt University	US	Inspection	Legged	Suction
	CROMSCI	University of Kaiserslautern	Germany	Inspection	Wheel-driven	Suction
	Alicia 1, 2	Universita' degli Studi di Catania	Italy	Inspection	Wheel-driven	Suction
	WallWalker	Kawagwa University	Japan	Wall cleaning	Wheel-driven	Suction
	SIRIUSc	FIFOA	Germany	Wall cleaning	Combined	Suction
	Sky Cleaner 1, 2, 3	University of Hamburg	Germany	Wall cleaning	Translation	Suction
	TITO	IAI/CSIC	Spain	Wall cleaning	Cable-driven	Propulsion
	CAFÉ	DISAM	Spain	Wall cleaning	Cable-driven	N/A
	NINJA-I, II	Tokyo Institute of Technology	Japan	Inspection	Legged	Suction
GEKKO III	ARECO	Germany	Wall cleaning	Tracked	Suction	
Exterior Wall Painting Robot	Taisei	Japan	Painting	Cable-driven	Rail-guided	
Civil infrastructure	Bigfoot	Portech	Germany	Diagnosis	Wheel-driven	Suction
	SM2	Carnegie Mellon University	US	Inspection	Legged	Gripping
	RAMR I	Michigan State University	US	Reconnaissance	Legged	Suction
	Roboclimber	University of Genova	Italy	Consolidation of rocky walls	Combined	N/A
Petrochemical plant	MRWALLSPECT II	Sungkyunkwan University	Korea	Inspection	Translation	Suction
	MRWALLSPECT III	Sungkyunkwan University	Korea	Inspection	Legged	Suction
	ROBICEN I, II, III	University of Navarra	Spain	Inspection	Translation	Suction
	WCR	Shanghai Jiao Tong University	China	Maintenance	Tracked	Magnetic
	WCR	Dalhousie University	Canada	Inspection	Tracked	Magnetic
	SURFY	Università degli Studi di Catania	Italy	Diagnosis	Translation	Suction
	TRIPILLAR	EPFL-LSRO	France	Inspection	Wheel-driven	Magnetic
Nuclear plant	Robug IIs	City University of Hong Kong	China	Maintenance	Legged	Suction
Shipbuilding	RRX	Seoul National University	Korea	Welding	Translation	Rail-guided
	REST 1, 2	CSIC	Spain	Welding	Legged	Magnetic
	Climbing Robot for Grit Blasting	University of Coruña	Spain	Cleaning	Translation	Magnetic
	NDT robot	London South Bank University	UK	Inspection	Wheel-driven	Magnetic
	Modular Climbing Robot	Universidad de Vigo	Spain	Inspection	Wheel-driven	Magnetic
Aircraft	MACS	California Institute of Technology	USA	Inspection	Translation	Suction
Service industry	MATS	Universidad Carlos III de Madrid	Spain	Human care	Legged	Gripping
	Magnebot	MIT	US	Material handling	Wheel-driven	Magnetic
	Hand-Bot	EPFL-LSRO	France	Object manipulation	Legged	Gripping

A Survey of Climbing Robots: Locomotion and Adhesion, Baeksuk Chu, Kyungmo Jung, Chang-Soo Han and Daehie Hong, 2010

## 7.6. BIOROBOTS

### 7.6.1. Locomoción de arrastre

Debido a que son capaces de moverse en ambientes inaccesibles para los seres humanos, tales como tuberías o edificios derrumbados, numerosos robots como las serpientes han sido desarrollados para tareas de exploración e inspección, así como para la participación en misiones de búsqueda y rescate. La Salamandra Robótica (figura 7.56), desarrollada en la Escuela Politécnica Federal de Laussane (EPFL) en Suiza extiende estos enfoques, ya que es el primer robot que combina los tres modos de locomoción, como arrastrarse, nadar y caminar de la serpiente en un solo robot. Con la inspiración de los poliquetos, o gusanos de paletas, que utilizan paletas diminutas en los segmentos de su cuerpo para empujarse través de arena, barro o agua, se abordó el tema de suministro de las formas tradicionales de locomoción robótica de que no iba a funcionar en el ambiente peculiar de los intestinos (figura 7.56).

Figura 7.56: (a) Salamandra Robótica, un robot que se arrastra, nada, y camina (b) Un robot inspirado en un gusano submarino diseñado para rastrear a través de los intestinos



Fuente: Biologically Inspired Robots, Jean-Arcady Meyer, Agnès Guillot

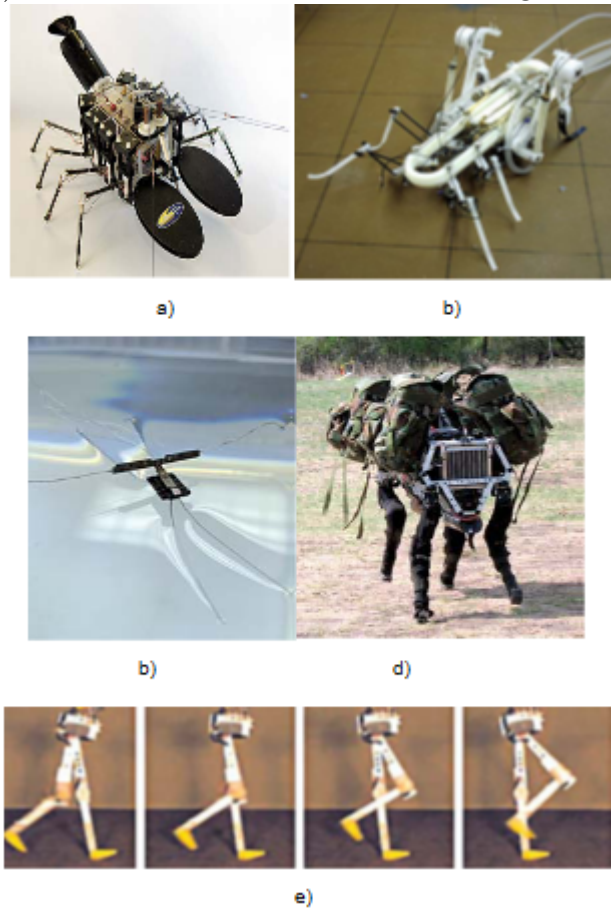
### 7.6.2. Locomoción de piernas (Caminadores)

#### Ocho patas

Joseph Ayers ha desarrollado un robot biomimético basado en la langosta americana en el Marine Science Center of North Eastern University (Figura 7.57). Aprovechando los últimos avances en microcontroladores, materiales inteligentes y dispositivos micro-

electrónicos, este robot de ocho patas está destinado a las detección autónoma de minas, operaciones en los ríos, puertos, y zonas litorales del fondo del océano.

Figura 7.57: (a) El robot langosta de la Northeastern University (b) El robot de cricket de la Case Western Reserve University (c) El robot zancudo de agua. (d) BigDog de Boston Dynamics (e) RunBot de la Universidad de Stirling



Fuente: Biologically Inspired Robots, Jean-Arcady Meyer, Agnès Guillot

### Seis patas

La fabricación de pequeñas formas ha permitido a los ingenieros concertar las piernas de la familia SPRAWL, robots hexápodos de mano inspirados en la cucaracha y que son muy rápidos, robustos, y auto-estabilizables a las perturbaciones sin ningún tipo de detección activa. Un robot se muestra en la figura 7.58. Aprovechando trabajo previo, un robot inspirado en el cricket, aproximadamente de 8 cm de largo, diseñado tanto

para caminar y saltar, se está desarrollando en la Case Western Reserve University, y se muestra en la figura 7.57e. Músculos artificiales Mckibben accionaran las piernas, el aire comprimido es generado por una planta de energía a bordo, y una red neuronal recurrente de tiempo continuo se utilizará para el control. Además, las patas delanteras le permiten trepar sobre grandes obstáculos, y también se utiliza para controlar la inclinación del cuerpo antes de un salto y, por tanto, tiene como objetivo el salto de la distancia o la altura.

Figura 7.58: Robot Sprawlette, a) Una simple antena montada en el robot Sprawlette, b) Un dispositivo táctico más avanzado



Fuente: Biologically Inspired Robots, Jean-Arcady Meyer, Agnès Guillot

Los robots STRIDER (Surface Tension Robotic Insect Dynamic Explorer) son capaces de caminar sobre el agua, ya que, en lugar de utilizar la flotabilidad, como los cuerpos de macro escala, estas criaturas muy pequeñas y ligeras equilibran su peso con fuerzas repulsivas de tensión superficial producida por los micro pelos hidrófobos que cubren sus piernas. En el Laboratorio de Nano de Robótica de la Universidad Carnegie Mellon, el Water Strider, un microrobot, camina sobre el agua con patas hechas de teflón hidrofóbico recubiertas con alambres y un cuerpo de fibra de carbono con un peso mínimo (Figura 7.57c).

### **Cuatro patas**

Los ingenieros de Boston Dynamics afirman haber desarrollado el robot cuadrúpedo más avanzado en la Tierra para el Ejército de los EE.UU. Se llama BigDog, camina, corre, y sube en terrenos difíciles, y lleva cargas pesadas. Siendo el tamaño de un perro grande, midiendo 1 metro de largo y 0.7 m de altura, y con un peso de 75 kg, BigDog ha trotado a 5 km/h, ha subido a una pendiente de 35° y ha llevado una carga de

50 kg hasta el momento. BigDog es impulsado por un motor de gasolina que tiene un sistema de accionamiento hidráulico. Sus piernas son articuladas como las de un animal, y cuentan con elementos compatibles con que absorben los golpes y la energía, reciclándola para el paso siguiente (Figura 7.57d).

### **Dos patas**

Desarrollado en la Universidad de Stirling, el RunBot es probablemente el robot bípedo más rápido del mundo para su tamaño. Con sus 30 cm de altura, puede caminar a una velocidad de 3.5 longitudes de piernas por segundo, lo cual es comparable a la velocidad más rápida con respecto de la marcha humana (Figura 7.57e). Este robot tiene algunas características mecánicas especiales, por ejemplo, pequeños pies curvos que permiten la acción de rodadura y un centro de masa bien posicionado, que facilitan una marcha rápida a través de la explotación de su dinámica natural.

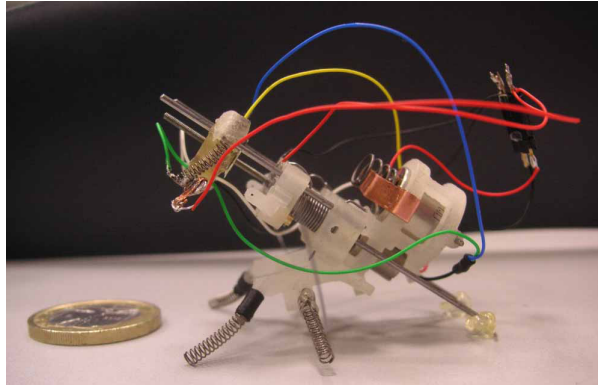
### **7.6.3. Locomoción mediante saltos**

En la perspectiva de exploración y vigilancia del ambiente, un estudio realizado en la escuela de ingeniería de biorobotica del MIT llamado: “Un concepto bioinspirado para locomoción de alta eficiencia en microrobots: El robot saltador GRILLO” un ligero microrobot que demuestra que el salto puede ser más eficiente energéticamente que simplemente caminar o escalar, y se puede utilizar para superar los obstáculos y terrenos irregulares. Durante la fase de vuelo, la energía de un micromotor eléctrico se recoge en los resortes del robot, y es liberado por un mecanismo de click durante el despegue. De esta manera la potencia instantánea entregada por las patas traseras es mucho mayor que la proporcionada por el motor.

### **7.6.4. Locomoción en el agua (nadadores)**

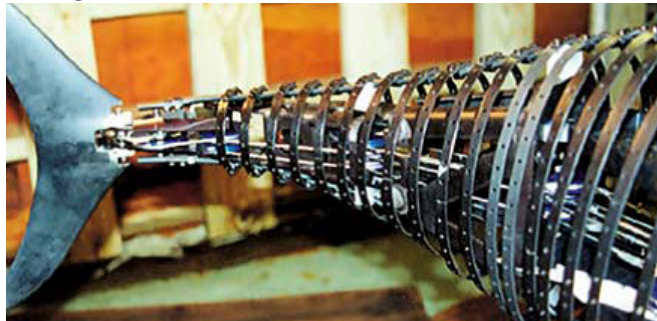
Varios robots biomiméticos se están produciendo para emular los sistemas de propulsión de los peces, delfines, focas, y explotar la compleja mecánica de fluidos que estos animales utilizan para impulsarse. El objetivo principal de estos proyectos es la construcción de máquinas capaces de maniobrar mediante el aprovechamiento de los flujos y posiciones del cuerpo, dando lugar a enormes ahorros de energía, y aumentar sustancialmente la cantidad de tiempo de autonomía. Por ejemplo, el grupo del MIT Towing Tank diseño dos peces robóticos, RoboTuna y RoboPike, que utilizan servomotores y espinas de resorte de los elementos (Figura 7.60), y sirven para demostrar las ventajas de la propulsión de aleteo.

Figura 7.59: El robot saltador GRILLO



Fuente: A bioinspired concept for high efficiency locomotion in micro robots: The jumping Robot Grillo, U. Scarfogliero, C. Stefanini, P. Dario (2006).

Figura 7.60: La cola del RoboTuna del MIT



Fuente: Biologically Inspired Robots, Jean-Arcady Meyer, Agnès Guillot

El robot Madeleine del Vassar College imita el diseño de una tortuga. Con dimensiones de 80 x 30 cm y peso de 24 kg, tiene una potencia de salida comparable, y sus aletas de poliuretano tiene la misma rigidez que las de una tortuga real, pero estas últimas están accionadas por motores eléctricos conectados a un ordenador a bordo (Figura 7.61).

### 7.6.5. Locomoción de vuelo

Batir las alas ofrece varias ventajas al compararse con las alas fijas de aviones de reconocimiento no tripulados actuales, como volar a velocidades bajas, flotar, haciendo giros bruscos, e incluso volar hacia atrás.

El objetivo del “Micromechanical Flying Insect (MFI) Project”, Proyecto del insecto

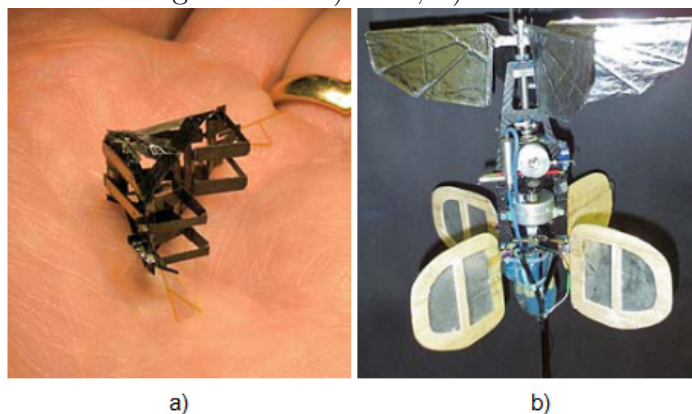
Figura 7.61: El robot Madeleine



Fuente: Biologically Inspired Robots, Jean-Arcady Meyer, Agnès Guillot

volador micromecánico en Berkeley es el desarrollo de un robot de 25 mm con capacidad de sustentar vuelo autónomo, lo que podría ser utilizado en la búsqueda, rescate, vigilancia y reconocimiento. Tales robots diminutos se basarán en principios biomiméticos que capturan algunos de los desempeños excepcionales de vuelo alcanzados por las moscas reales, es decir, grandes fuerzas generadas por la aerodinámica de estado no estacionario, un sistema motor de alta relación potencia-peso y un sistema de control de alta velocidad con sensores visuales y de inercia estrechamente integrados. Una primera plataforma de MFI (Micromechanical Flying Insects), que bate sus dos alas y es del tamaño adecuado, ya se ha producido (Figura 7.62).

Figura 7.62: a) MFI, b) Mentor



Fuente: Biologically Inspired Robots, Jean-Arcady Meyer, Agnès Guillot

El ornitóptero cuatro alas: Mentor (Figura 7.62), que está siendo desarrollado en

la Universidad de Toronto, como parte de un esfuerzo de investigación general dirigida al vuelo de aleteo, se dice que es el primer dispositivo artificial en agitar las alas que flotó con éxito, haciéndolo con la agilidad de un colibrí. En particular, se exhibe el comportamiento de vuelo que el animal utiliza para mejorar su elevación aplaudiendo con sus alas, y posteriormente alejándolas a altas velocidades.

### 7.6.6. Robots ápodos

Estos robots tienen unas características que los hacen únicos, al igual que sus homólogos las serpientes y gusanos. Por un lado está su capacidad para cambiar su forma. Frente a las rígidas estructuras del resto de robots, los ápodos pueden doblarse y adoptar la forma del terreno por el que se desplazan. Por otro lado, su sección es muy pequeña en comparación con su tamaño, lo que les permite introducirse por tubos u orificios pequeños y llegar a lugares donde otros robots no pueden.

Figura 7.63: Evolución de los robots serpiente de la familia ACM (Active Cord Mechanism). Hirose-Fukushima Robotics Lab



Fuente: Biologically Inspired Robots, Jean-Arcady Meyer, Agnès Guillot

### Institutos encargados del desarrollo de robots apodos

#### Instituto de tecnología de Tokyo: Familia ACM.

El pionero en el estudio de la biomecánica de las serpientes para su aplicación a la robótica fue Hirose, del Instituto de tecnología de Tokyo, que implementó en 1976 el primer robot de tipo serpiente denominado ACM-III (Active Cord Mechanims). Los

resultados de sus investigaciones han sido recopilados y publicados en 1987 en el libro de referencia “Biologically Inspired Robots”. Una de las mayores aportaciones a la ciencia del profesor Hirose fue el descubrimiento y formulación de la curva serpentina, que es la forma que adoptan las serpientes durante su desplazamiento. Existen diferentes prototipos desarrollados hasta el momento (ACM).

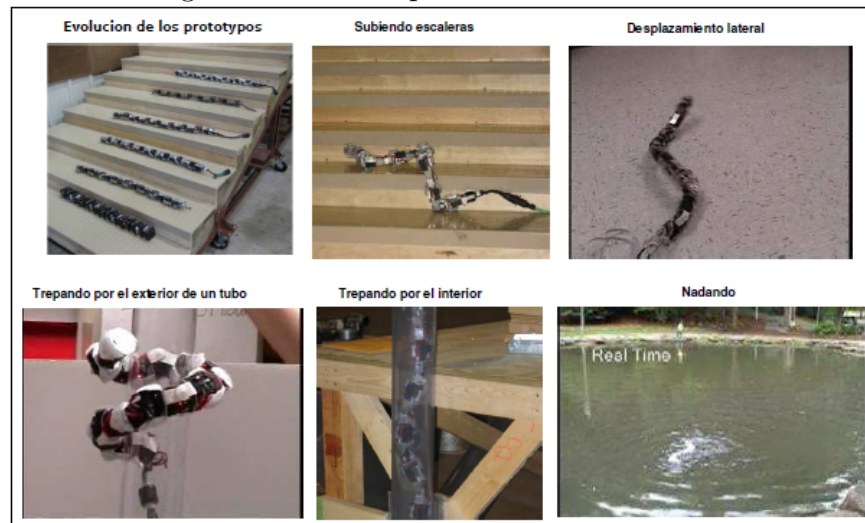
### Instituto de automática de ShenYan

Los trabajos de Hirose han servido de inspiración para otros investigadores. Uno de ellos es Shugen Ma que repitió y amplió los trabajos de Hirose sobre la propulsión deslizante y desarrollaron una versión simplificada del ACM-R1 de 12 módulos mecánicamente menos complejos y con un sistema de control mejorado. Además, desarrollaron un software para simular el movimiento real del robot en diferentes superficies.

### Instituto de Robótica de Carnegie Mellon University

En el nivel superior sus investigaciones se centran en la planificación de los movimientos, desarrollando algoritmos de locomoción y posicionamiento en lo que denominan robots hiper redundantes. En el área de la locomoción a bajo nivel están obteniendo unos resultados muy interesantes. Los prototipos diseñados (Figura 7.64) están basados en los módulos de Mark Yim.

Figura 7.64: La serpiente robótica del CMU

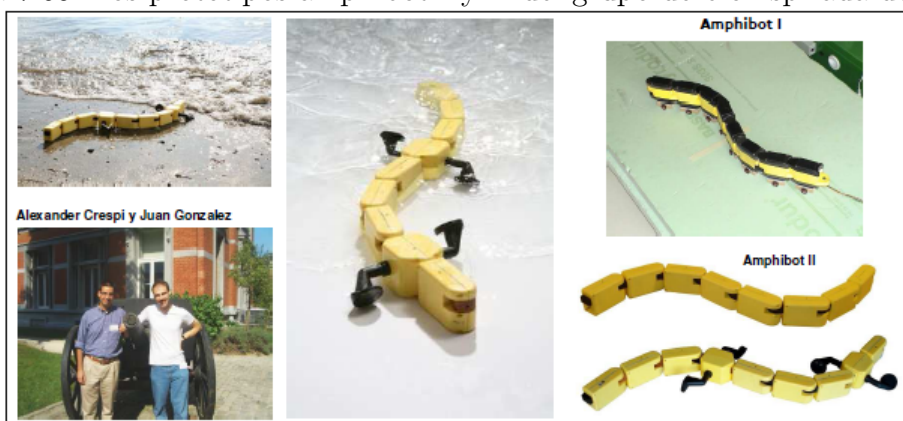


Fuente: K. Dowling. Limbless Locomotion: Learning to Crawl with a Snake Robot. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA

### Grupo de robótica bioinspirada del EPFL: Amphibot

El grupo de robótica bioinspirada del EPFL (*Ecole Polytechnique Fédérale de Lausanne*) ha desarrollado el robot anfibio Amphibot, que es capaz de moverse por agua y tierra. Está compuesto de 8 módulos que se mueve paralelamente al suelo y utiliza modelos de control bioinspirados para la locomoción, basados en los modelos de CPGs (*Central Pattern Generators*) de la lamprea, desarrollados por Ijspeert.

Figura 7.65: Los prototipos amphibot I y II del grupo de bioinspirada del EPFL



Fuente: A. Crespi and A. J. Ijspeert. Amphibot II: an amphibious snake robot that crawls and swims using a central pattern generator. In Proceedings of the 9th International Conference on Climbing and Walking Robots

En la parte inferior izquierda de la figura 7.65, el autor de esta tesis está junto a Alexander Crespi, autor de Amphibot, durante la asistencia al Clawar 2006 en Bruselas.

### Otros

El WormBot de Conrad et al, desarrollado en el Instituto de Neuroinformática de la Universidad de Zurich, es un prototipo de serpiente robótica que se mueve mediante ondulaciones de su cuerpo y que se basa en un modelo bioinspirado de CPGs. Han implementado el CPGs de la Lamprea. Con este principio consiguen movimientos muy rápidos y eficientes. La primera versión SES-1 está formada por circuitos analógicos exclusivamente.

### Robots ápodos propulsados

Frente a los robots ápodos que realizan la locomoción a partir de movimientos corporales están los robots ápodos autopropulsados en los que las diferentes partes que forman

Figura 7.66: Wombot (izquierda), diseñado por Conrad. S5, Miller (derecha)



Fuente: A review on modelling, implementation, and control of snake robots

el robot tienen ruedas u orugas para conseguir el desplazamiento. Aunque tienen la forma de una serpiente, no se trata de robots bioinspirados. Este tipo de locomoción no se encuentra en la naturaleza. Sin embargo se incluyen en este estudio por ser la mayoría de ellos robots modulares, constituidos mediante la unión de cadena de módulos similares.

### Hirose Fukushima Robotics Lab (Titech)

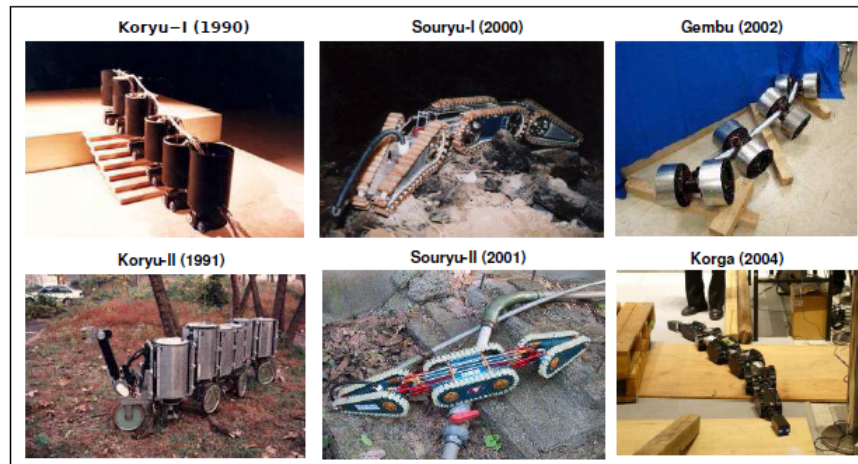
El profesor Hirose fue también pionero en este tipo de robots. A partir del ACM III, desarrolló estructuras encadenadas con módulos autopropulsados, que denominó cuerpos articulados (*articulated bodies*). Entre las ventajas de este tipo de robots destacan su fácil transporte: los módulos se separan unos de otros y luego se vuelve a unir, pueden llevar carga distribuida por todo el robot, pueden moverse por caminos estrechos y sinuosos y el sistema es redundante, si un módulo falla se sustituye por otro.

Para explorar las capacidades locomotivas de sus cuerpos articulados desarrollaron prototipos como **KORYU I**, **KORYU-II**, **Souryu-I**, **Souryu-II** y los robots **Genbu** (I, II y III) mostrados en la figura 7.67.

### Centro nacional Alemán de investigación (GMD)

En el GMD han desarrollado dos prototipos de robot ápodos autopropulsados. Uno es el **GMD SNAKE** (prototipos 1 y 2). Está constituido por 12 ruedas motrices en cada módulo. Tiene 6 módulos más uno en la cabeza. La aplicación principal para la que fue diseñado es la inspección de tuberías, aunque en se estudia su uso en aplicaciones de inspección de edificios. El otro es el robot **Makro** para la inspección de alcantarillas entre 30 y 60cm de diámetro. Está formado por 6 módulos y las uniones entre ellos

Figura 7.67: Robots auto-propulsados (serpentine robots) en el Hirose-Fukushima Robotics Lab



Fuente: A review on modelling, implementation, and control of snake robots

tienen 3 grados de libertad. Cada módulo dispone de dos ruedas para propulsarse.

**Laboratorio de robótica móvil de la Universidad de Michigan: OmniTread**

Uno de los robots ápodos autopropulsados más es **OmniTread** desarrollado por Granosik et al. en el laboratorio de robótica móvil de la Universidad de Michigan, para aplicaciones de inspección industrial y vigilancia. Es un robot es muy robusto y flexible. Utiliza articulaciones neumáticas lo que le da mucha fuerza. La versión inicial, omitread **OT-8** está formado por 5 módulos hexaédricos. (Figura 7.68).

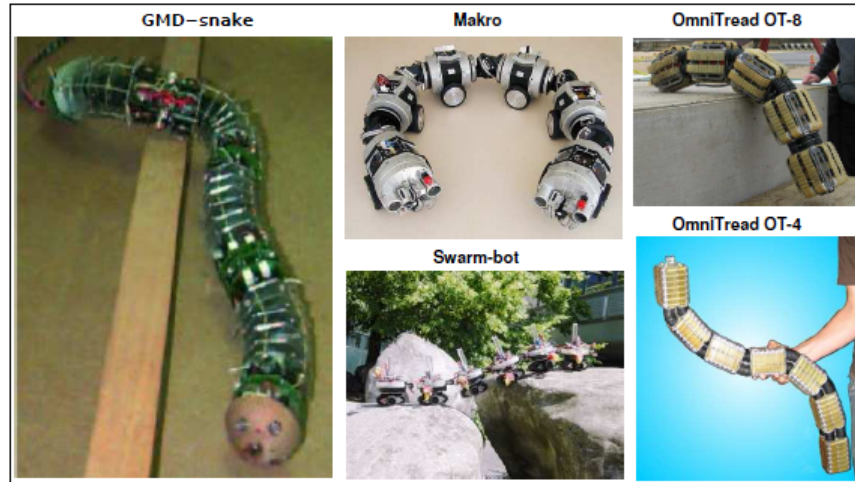
**Laboratorio de sistemas inteligentes del EPFL: Swarm-bot**

El robot Swarm-bot9 se está empleando en el Laboratorio de sistemas inteligentes del EPFL para el estudio de la “inteligencia colmena”: colonias que son capaces de auto-organizarse. El prototipo desarrollado está formado por pequeños robots móviles que tienen la capacidad de autoensamblarse para formar estructuras mayores y así realizar otras tareas. Por ejemplo, si tienen que cruzar por una grieta, se pueden organizar en una cadena. Cada uno de los módulos se llama s-bot y son totalmente autónomos. Para desplazarse utilizan orugas y están dotados de sensores, (Figura 7.68).

**Grupo de robótica de la Universidad de Beihang (BUAA): JL-I**

El grupo de robótica de la Universidad de Beihang comenzó con el diseño de este tipo de robots en 1999, con el diseño de un prototipo de dos módulos. Cada uno con dos orugas y

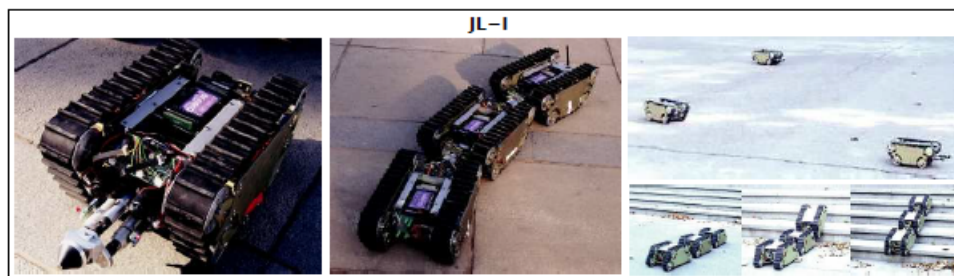
Figura 7.68: Diferentes prototipos de robots ápodos con propulsión: GMD-snake, Makro, Swarm-bot, OmniThread OT-8 y OT-4



Fuente: Diseño de robots ápodos, Juan González Gómez

una articulación de 2 grados de libertad. Además, las articulaciones son extensibles, permitiendo que se pueda aumentar o disminuir la longitud del robot. Basado en este prototipo inicial, Houxiang et al. Diseñaron el robot **JL-I**. Actualmente está formado por 3 módulos idénticos. Las articulaciones son de 3 grados de libertad lo que le dota de una gran capacidad de movimiento. No sólo puede cruzar obstáculos sino también subir escaleras o recuperarse ante vuelcos entre otras características.

Figura 7.69: El robot JL-I



Fuente: Y. Zhang, K.D. Roufas, and M. Yim. Software architecture for modular self-reconfigurable robots. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, volume 4

## **Robots modulares y locomoción**

### **Un nuevo enfoque al problema de la locomoción**

Mark Yim se puede considerar el padre de la robótica modular reconfigurable. Sus trabajos han inspirado a cientos de investigadores y algunos de sus artículos han sido citados más de 250 veces. Mark Yim propuso en su tesis doctoral en 1995 un nuevo enfoque al problema de la locomoción. Lo que propuso Yim fue utilizar robots formados a partir de módulos con la capacidad para ensamblarse unos con respecto a otros. Es decir, que toman lo mejor de los robots ápodos y de los robots con patas.

### **Polypod**

El primer robot que desarrolló, en su tesis, fue Polypod. Aunque lo que se proponía era el nacimiento de los robots modulares autoconfigurables, Polypod era manualmente reconfigurable, pero lo utilizó para implementar distintas configuraciones y demostrar la viabilidad de sus ideas. Los módulos de Polypod era mecánicamente complejos y están dotados de dos grados de libertad. Todos los detalles técnicos están incluidos en su tesis.

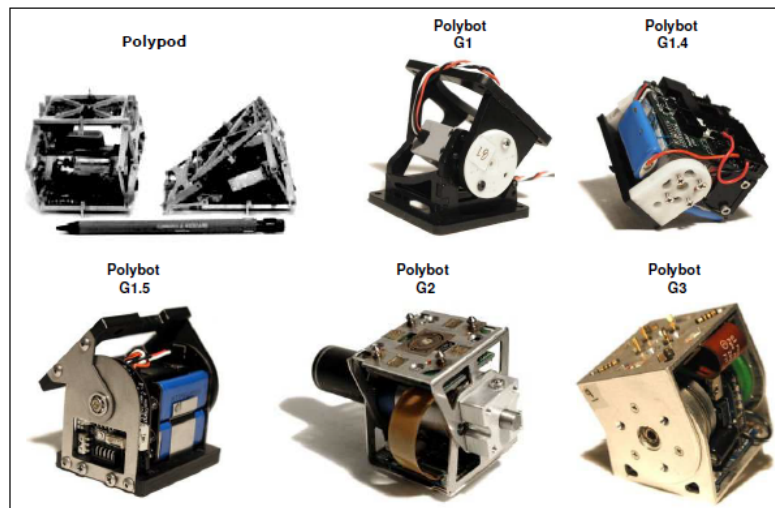
### **Polybot**

La versatilidad de estos módulos es debida a que estos robots pueden cambiar su forma y desplazarse por terrenos muy diversos. Hasta el momento han creado cinco tipos diferentes de módulos, agrupados en tres generaciones: G1, G2 y G3 (Figura 7.70). Uno de los objetivos de diseño de todos ellos es la simplicidad. Por ello, todos están dotados de un único grado de libertad. La Generación G1 no es auto-reconfigurable, ya que los módulos no tienen la capacidad de acoplarse automáticamente unos con otros. (Figura 7.71). Probada la viabilidad de los módulos, en la versión G1v4 se introdujeron sensores para realizar aplicaciones en bucle cerrado. Realizaron experimentos de gusanos trepadores. Las generaciones G2 y G3 tienen la capacidad de poder acoplarse y separarse, lo que permite construir con ellos auténticos robots auto configurables. Los módulos G1v5 son los últimos desarrollados. No son reconfigurables sin embargo están diseñados a partir de las lecciones aprendidas con todos los módulos anteriores. En la figura 7.72 se muestran diferentes configuraciones de las generaciones G1v4, G1v5, G2 y G3 de Polybot.

### **CkBot**

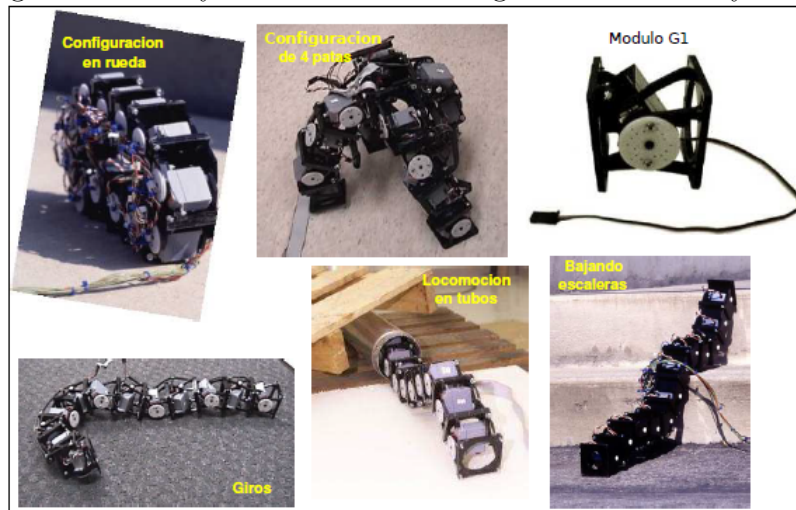
En la figura 7.73 se muestran los módulos y algunas de las configuraciones probadas. Para probar la viabilidad del sistema, han creado una configuración en forma de robot humanoide, constituida por 3 grupos de módulos (los denominan clusters). Cada

Figura 7.70: Los módulos de Polypod y Polybot



Fuente: M. Yim, D.G. Duff, and K. D. Roufas. PolyBot: a modular reconfigurable robot. In Proceedings of the IEEE International Conference on Robotics and Automation, April 2000

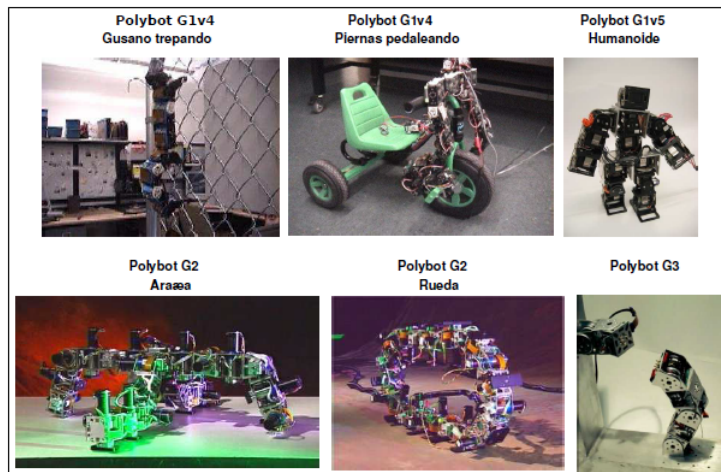
Figura 7.71: Polybot. Diferentes configuraciones de Polybot G1



Fuente: Diseño de robots ápodos, Juan González Gómez

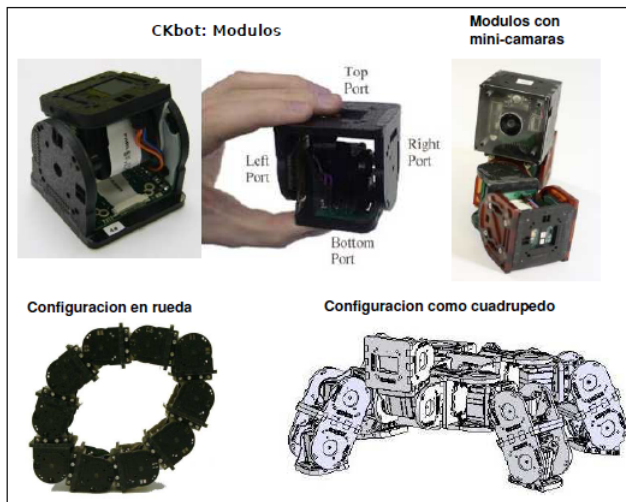
grupo está formado por 3 módulos Ckbot y un módulo con una mini-cámara. La unión mecánica entre los 3 clusters se hace mediante imanes permanentes, mientras que los módulos en el interior están unidos mediante tornillos.

Figura 7.72: Diferentes configuraciones de Polybot



Fuente: Diseño de robots ápodos, Juan González Gómez

Figura 7.73: Módulos y distintas configuraciones de CkBot

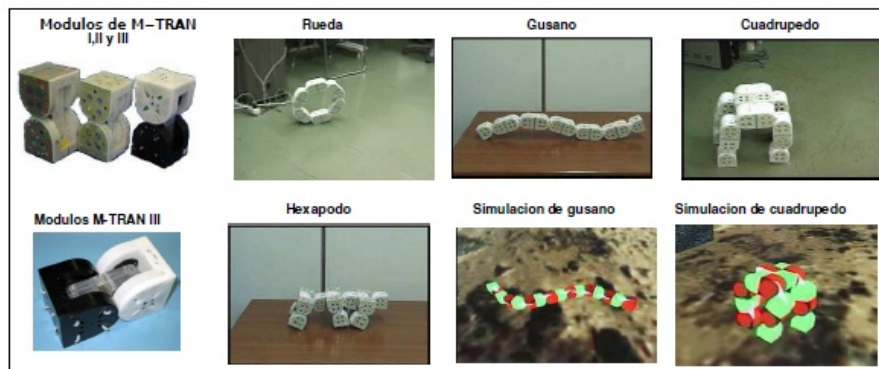


Fuente: Diseño de robots ápodos, Juan González Gómez

### M-TRAN

Uno de los robots modulares más avanzados que existen actualmente es el M-TRAN (Modular TRANformer) desarrollado en el Instituto Nacional de Ciencias y Tecnologías Industriales avanzadas de Japón (AIST). En la figura 7.74 se muestran los módulos y

Figura 7.74: Módulos y diferentes configuraiones del robot M-TRAN



Fuente: Diseño de robots ápodos, Juan González Gómez

diferentes configuraciones del robot.

### CONRO

Los módulos **CONRO** fueron desarrollados por Castano et al. en el ISI (*Information Science Institute*) de la Universidad de California Sur para la realización de lo que denominan sistemas metamórficos: robots que pueden cambiar su forma. Lo que Yim denominó reconfigurables (término que ha prevalecido). Estos módulos tiene dos grados de libertad y la capacidad de auto-acoplarse unos a otros. En la figura 7.75 se muestra el aspecto del módulo CONRO así como diferentes configuraciones y uno de los experimentos de reconfiguración llevados a cabo.

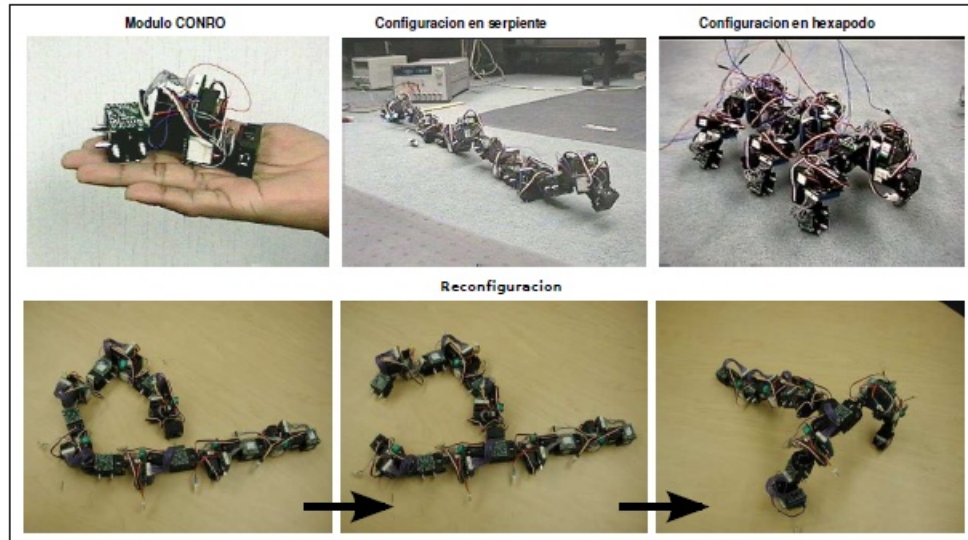
### SuperBot

SuperBot es un robot modular creado en el laboratorio de robótica polimórfica del ISI (*Information Science Institute*) en la Universidad de California sur. El módulo diseñado es de los más modernos (2005) y está inspirado en todos los anteriores: Conro, Polybot, MTRAN y ATRON. Es un proyecto financiado por la NASA y el DARPA. Entre 8 y 10 módulos se reconfiguran para formar la plataforma necesaria: como rueda (para una locomoción eficiente), araña, serpientes, torres de comunicaciones, etc.

### Yamor

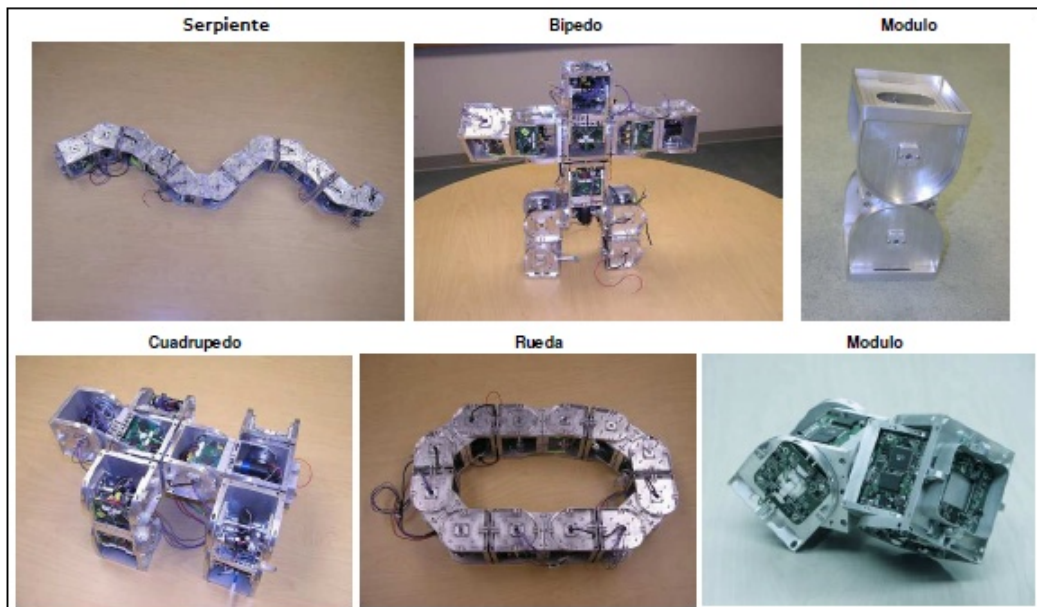
Es el robot modular desarrollado en el laboratorio de robótica bioinspirada del EPFL para estudiar la locomoción adaptativa. El módulo desarrollado tiene un único grado de libertad actuado por un servo y la comunicación entre módulos y módulos-PC se realiza a través de Bluetooth lo que permite que no existan cables. En la figura 7.77 se muestran

Figura 7.75: Módulo Conro y diferentes configuraciones



Fuente: A. Castano, R. Chokkalingam, and P. Will. Autonomous and self-sufficient CONRO modules for reconfigurable robots

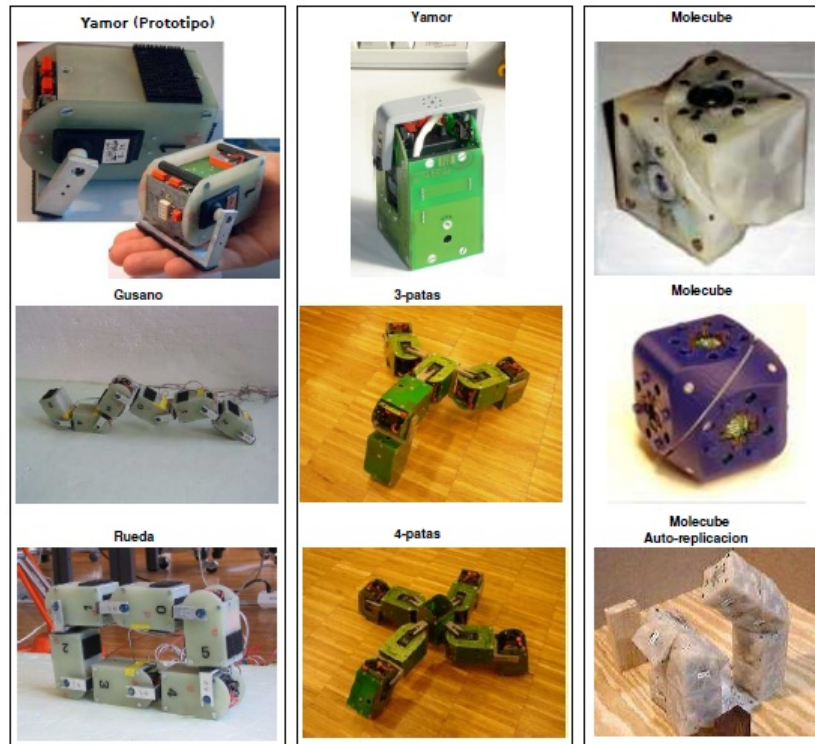
Figura 7.76: El módulo de Superbot y diferentes configuraciones



Fuente: [www.neoteo.com](http://www.neoteo.com)

las dos versiones realizadas de Yamor y las diferentes configuraciones probadas.

Figura 7.77: Los robots modulares Yamor y Molecube



Fuente: <http://biorob.epfl.ch/yamor>

### Molecube

En el CSL (*Computational Synthesis Laboratory*) de la Cornell University se han desarrollado los módulos que denominan Molecubes. Tienen un único grado de libertad y forma de cubo. Se diferencian del resto en que la rotación se realiza alrededor del eje diagonal que une dos puntos opuestos del cubo. Al realizar una rotación de  $90^\circ$  alrededor de este eje se obtiene otro cubo. Estos módulos no están adaptados para resolver el problema de la locomoción, aunque sí se pueden crear robots con capacidad locomotiva. Su propósito original fue el de construir el primer sistema modular capaz de auto-replicarse a sí mismo.

---

---

# CAPÍTULO 8

---

## CINEMÁTICA DE LOS ROBOTS MÓVILES

### 8.1. GENERALIDADES DE LA CINEMÁTICA.

La cinemática es el estudio más básico de cómo se comportan los sistemas mecánicos. En robótica móvil, tenemos que entender el comportamiento mecánico del robot, tanto para el diseño de robots móviles adecuados para las tareas y para entender cómo crear software de control para el caso del hardware de un robot móvil. En la siguiente sección, se introduce una notación que permite expresar el movimiento del robot en un marco de referencia global, así como en un marco de referencia local del robot. Luego, con esta notación, se demuestra la construcción de simples modelos cinemáticos de movimiento de avance, describiendo cómo el robot se mueve como un conjunto en función de su geometría y el comportamiento individual de sus ruedas.

### 8.2. SUPUESTOS CONSIDERADOS

#### 1. Supuestos de diseño:

- Los vehículos no poseen en su estructura partes flexibles, es decir: toda la estructura es rígida.
- Por cada rueda puede haber una o ninguna articulación de dirección.

- Todos los ejes de dirección existentes son perpendiculares a la superficie por la que se desplaza el vehículo.
- Las ruedas pueden estar directamente en contacto con el suelo o a través de unos rodillos.
- Las ruedas giran sin problemas de deslizamiento.

2. **Supuesto operacional:** La superficie de desplazamiento (suelo) es plana.

El primer supuesto de diseño permite aplicar la formulación de sólido rígido, mientras que los otros tres limitan los tipos de rueda a los que va dirigido esta metodología. En particular, los tipos de rueda que se consideran son:

- **Rueda fija:** No posee articulación de dirección, por lo que su posición respecto a la estructura es fija.
- **Rueda orientable centrada (orientable):** Tiene articulación de dirección, es decir es orientables respecto a la estructura del vehículo, pasando su eje de dirección por el centro de rotación de la rueda.
- **Rueda orientable descentrada (Rueda Castor o Loca):** Posee articulación de dirección, es decir es orientable respecto a la estructura del vehículo, su eje de dirección no pasa por el centro de rotación de la rueda.
- **Rueda fija con rodillos (sueca, universal, Mecanum ó Ilon):** Es fija respecto a la estructura del vehículo y posee rodillos entre la rueda y el suelo con una determinada orientación fija respecto a la rueda.

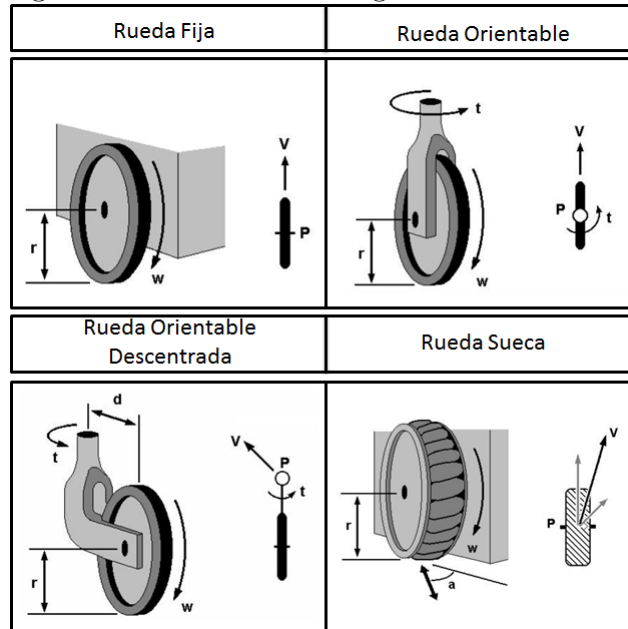
En ocasiones se utilizan dos hileras de rodillos para garantizar una mejor continuidad del punto de contacto con el suelo. Los tipos de rueda fija, orientable y castor se engloban dentro del tipo convencional, y también el tipo castor y sueca dentro del omnidireccional. Las ruedas omnidireccionales no restringen el movimiento del vehículo mientras que las otras sí.

### 8.2.1. Configuraciones de ruedas

Las formas más conocidas de disponer las ruedas en estos robots móviles son:

- Diferencial
- Síncrono
- Tipo triciclo

Figura 8.1: Diferentes configuraciones de ruedas



(Fuente: Sistemas de locomoción de robots móviles, Automatización y Robótica Móvil.)

- Ackerman
- Omnidireccional

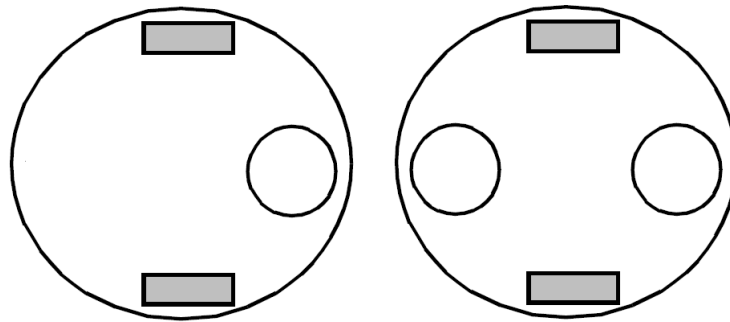
### Diferencial.

Este es uno de los esquemas más sencillos, básicamente consiste de dos ruedas en un eje común, donde cada rueda se controla independientemente cuyos movimientos son:

- Línea Recta
- En arco
- Vuelta sobre su propio eje

Este esquema utiliza una o dos ruedas adicionales (caster) para mantener el balance, estas formas tienen diferentes nombres dependiendo de las ruedas por ejemplo: Con 3 ruedas se denominan triángulo, este esquema puede presentar problemas de estabilidad y con 4 ruedas se conoce como diamante la pérdida de contacto de las ruedas de tracción hace requerir de un sistema de suspensión. Para que el movimiento sea recto se requiere que las ruedas vayan a la misma velocidad.

Figura 8.2: Configuración diferencial.



(Fuente: Sistemas de locomoción de robots móviles, Automatización y Robótica Móvil.)

### **Síncrono**

Las ruedas se mueven en forma síncrona, es decir, al mismo instante. El movimiento síncrono es un caso particular del diferencial, donde cada eje se mueve en forma dependiente para dar vuelta y avanzar. Las ruedas están ligadas de tal forma que siempre apuntan en la misma dirección y para dar vuelta se giran las ruedas sobre un eje vertical, por lo que la dirección de la estructura se mantiene y por tanto requiere de un mecanismo adicional para mantener el frente del chasis (estructura del robot) en la dirección de las ruedas (torreta).

En la figura 8.3 se aprecia este movimiento sobre el eje de rotación de las dos ruedas delanteras. Algunas ventajas de los sistemas Síncronos es que se evitan los problemas de inestabilidad, pérdida de contacto del diferencial y complejidad mecánica (eje de giro).

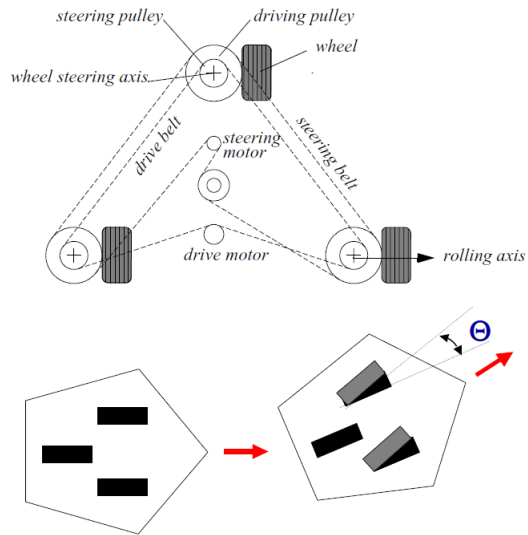
### **Tipo triciclo.**

Los triciclos tienen dos ruedas fijas que les dan tracción, además cuentan de una rueda para la dirección que normalmente no tiene tracción. Estos sistemas tienen buena estabilidad y simplicidad mecánica, tiene facilidad para ir recto y su cinemática es más compleja.

### **Configuración Ackerman.**

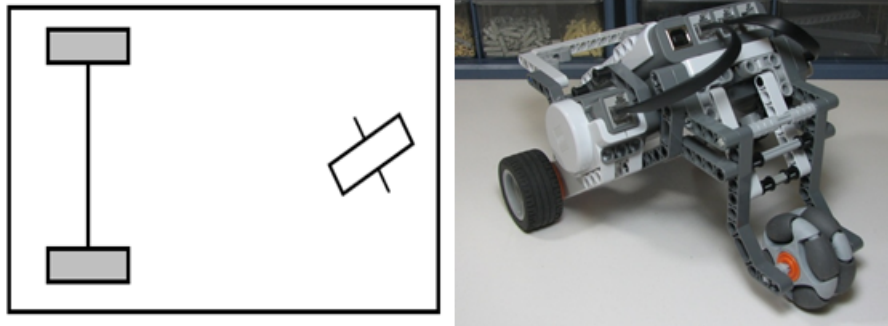
Los sistemas ackerman son similares al triciclo solo que cuentan con dos ruedas de tracción y dos ruedas para dirección. Tienen una mayor complejidad mecánica que el triciclo por el acoplamiento entre las 2 ruedas de dirección. Sus principales ventajas son

Figura 8.3: Configuración síncrona.



(Fuente: Sistemas de locomoción de robots móviles, Automatización y Robótica Móvil.)

Figura 8.4: Configuración tipo triciclo.



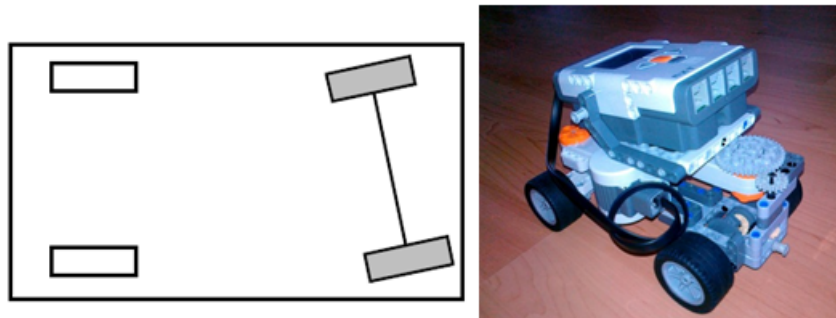
(Fuente: <http://www.hitechnic.com/>)

buena estabilidad y facilidad de moverse en línea recta. La desventaja es su complejidad cinemática.

### Locomoción Omnidireccional.

Los sistemas Omnidireccionales cuentan con 3 ruedas colocadas a  $120^\circ$  tal como se muestra en la figura. Donde cada una de las ruedas tiene la capacidad de girar en

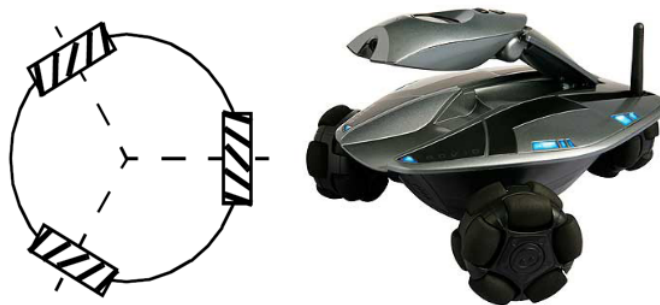
Figura 8.5: Configuración Ackerman.



(Fuente: <http://letsmakerobots.com/node/25589>)

ambos lados y se logra un control lineal más simplificado que en el caso del robot diferencial. Actualmente los campeones mundiales en fútbol de robots F-180 utilizan ruedas omnidireccionales.

Figura 8.6: Configuración omnidireccional.



(Fuente: Robots de Seguridad Civil, José Breñosa, Patricia García. UPM.)

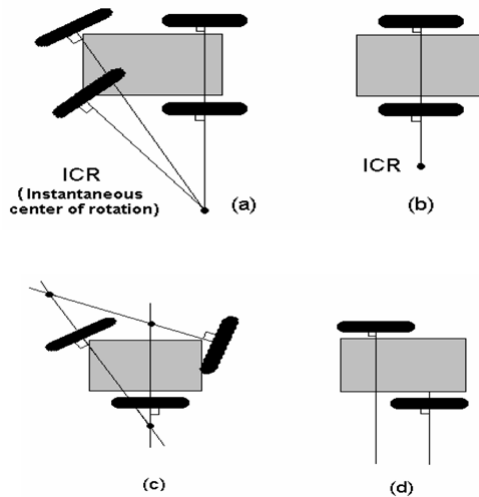
Los diferentes tipos de ruedas (tracción y dirección) tienen diferentes propiedades cinemáticas, por ejemplo un robot móvil normalmente tiene 3 grados de libertad respecto a una referencia: posición en el plano  $(X, Y)$  y orientación  $(\theta)$ . Idealmente, independientemente de donde inicie, el robot debe poder moverse a cualquier posición y orientación  $(X, Y, \theta)$ .

## 8.2.2. Algunos conceptos previos

### Centro Instantáneo De Rotación.

El centro instantáneo de rotación, referido al movimiento plano de un cuerpo, se define como el punto del cuerpo o de su prolongación en el que la velocidad instantánea del cuerpo es nula, es el punto en torno al cual gira un cuerpo en un instante determinado. Se define como la intersección de las perpendiculares a las trayectorias que recorren los puntos del cuerpo en movimiento. Durante el movimiento rectilíneo de un cuerpo, el centro instantáneo de rotación se halla en el infinito en dirección perpendicular al movimiento.

Figura 8.7: Configuración omnidireccional.



(Fuente: Sistemas de locomoción de robots móviles, Automatización y Robótica Móvil.)

### Restricciones cinemáticas.

Son limitaciones en el movimiento de los componentes de un sistema mecánico. Las restricciones cinemáticas se pueden considerar que tienen dos formas básicas:

- Las restricciones que surgen de las bisagras, deslizadores y las articulaciones de leva que definen la construcción del sistema, denominado restricciones holonómicas.
- Las restricciones impuestas en la velocidad del sistema, tales como la restricción del filo de la navaja de los patines de hielo sobre una superficie plana, o de rodadura

sin deslizamiento de un disco o una esfera en contacto con un plano, que se llaman restricciones no holonómicas.

Las restricciones también pueden surgir de otras interacciones tales como la rodadura sin deslizamiento. Es cualquier condición con relación a propiedades de un sistema dinámico que deben ser verdad en todo momento. A continuación se presentan algunos ejemplos comunes:

### **Rodadura sin deslizamiento.**

Un objeto que rueda sobre una superficie sin deslizamiento obedece a la condición de que la velocidad de su centro de masa es igual al producto cruz de su velocidad angular con un vector desde el punto de contacto hasta el centro de masa.

$$V_g(t) = \Omega \times r_{G/O}$$

Para el caso de un objeto que no se ladee o gire, esto reduce a  $V = \omega * R$

### **Cable inextensible.**

Este es el caso donde los cuerpos están conectados por un cable ideal que permanece en tensión y no puede cambiar su longitud. La restricción es que la suma de las longitudes de todos los segmentos de la cuerda es la longitud total, y en consecuencia la derivada respecto al tiempo (velocidad) de esta suma es cero. Un problema dinámico de este tipo es el péndulo. Otro ejemplo es un tambor que gira debido al peso de un cuerpo que cae, unido a su borde por dicho cable inextensible.

### **Tipos de restricciones.**

Se tienen dos tipos de restricciones que tiene que ver con el acoplamiento de los robot las cuales se conocen como:

- Holonómicos: los diferentes grados de libertad están desacoplados. Los robots diferenciales y síncronos: se puede desacoplar la posición de orientación (rotando sobre su eje).
- No-holonómicos: los grados de libertad están acoplados, por ejemplo los triciclos y carros: para dar vuelta debe moverse hacia el frente o hacia atrás, es más complejo llegar a la posición final deseada.

## 8.3. MODELOS Y RESTRICCIONES CINEMÁTICAS

Derivar un modelo para el movimiento total del robot es un proceso de abajo hacia arriba. Cada rueda individual contribuye al movimiento del robot y, al mismo tiempo, impone restricciones sobre el movimiento del robot. Las ruedas están unidas entre sí sobre la base de la geometría del chasis del robot, y por lo tanto sus limitaciones se combinan para formar restricciones sobre el movimiento general del chasis del robot. Pero las fuerzas y restricciones de cada rueda deben ser expresadas con respecto a un marco de referencia claro y coherente. Esto es particularmente importante en la robótica móvil debido a su naturaleza portable y móvil, por tanto es obligatoria una asignación clara entre los marcos globales y locales de referencia. Empezamos por la definición de estos marcos de referencia formalmente, luego, utilizamos el formalismo resultante para anotar la cinemática de las ruedas de forma individual y de los robots enteros.

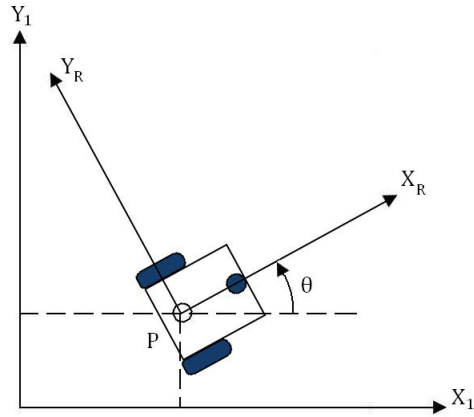
### 8.3.1. Representando la posición del robot

A lo largo de este análisis se debe modelar el robot como un cuerpo rígido sobre ruedas, que opera en un plano horizontal. La dimensionalidad total de este cuerpo del robot en el plano es de tres, dos para la posición en el plano y una para la orientación a lo largo del eje vertical, que es ortogonal al plano. Por supuesto, hay grados de libertad y flexibilidad gracias a los ejes de las ruedas, las articulaciones de dirección de las ruedas, y las articulaciones de las ruedas locas. Sin embargo por el chasis del robot nos referimos sólo al cuerpo rígido del robot, haciendo caso omiso de las articulaciones y los grados de libertad internos del robot y sus ruedas.

Con el fin de especificar la posición del robot en el plano se establece una relación entre el marco de referencia global del plano y el marco de referencia local del robot, como en la figura 8.8.

Los ejes  $X_1$  y  $Y_1$  definen una base inercial arbitraria en el plano como marco de referencia global desde un origen  $O : (X_1, Y_1)$ . Para especificar la posición del robot, se elige un punto  $P$  en el cuerpo del robot como punto de referencia de posición. La base  $(X_R, Y_R)$  define dos ejes con respecto a  $P$  en el chasis del robot y es por lo tanto el marco de referencia local del robot. La posición de  $P$  en el marco de referencia global se ha especificado por las coordenadas  $x$  y  $y$ , y la diferencia angular entre el marco de referencia global y local está dada por  $\theta$ . Podemos describir la postura del robot como un vector con estos tres elementos. Observe el uso del subíndice  $I$  para aclarar la base

Figura 8.8: El marco de referencia global y el marco de referencia local del robot.



(Fuente: Autores del proyecto.)

de esta postura como marco de referencia global:

$$\xi_I = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad (8.1)$$

Para describir el movimiento del robot en términos de movimiento de componentes, será necesario mapear movimiento a lo largo de los ejes del sistema de referencia global a movimiento a lo largo de los ejes del marco de referencia local del robot. Por supuesto, el mapeo es una función de la posición actual del robot. Este mapeo se realiza mediante la matriz de rotación ortogonal:

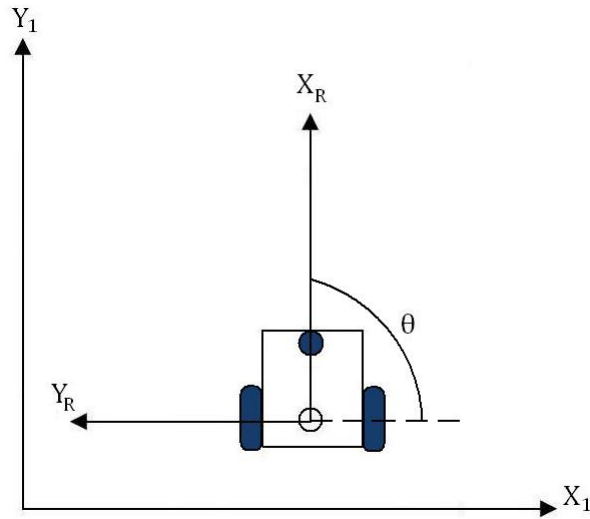
$$R(\theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (8.2)$$

Esta matriz puede ser utilizada para mapear el movimiento en el marco de referencia global \$(X\_1, Y\_1)\$ a movimiento en términos de marco de referencia local \$(X\_R, Y\_R)\$. Esta operación se denota por \$R(\theta)\xi\_I\$ ya que el cálculo de esta operación depende del valor de \$\theta\$:

$$\dot{\xi}_R = R(\theta)\dot{\xi}_I \quad (8.3)$$

$$\dot{\xi}_R = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}$$

Figura 8.9: El robot móvil alineado con un eje global.



(Fuente: Autores del proyecto.)

Por ejemplo, considere el robot en la figura 8.9. Para este robot, como  $\theta = \pi/2$  podemos fácilmente calcular la matriz de rotación instantánea R:

$$\begin{aligned} \dot{\xi}_R &= R\left(\frac{\pi}{2}\right)\dot{\xi}_I \\ R\left(\frac{\pi}{2}\right) &= \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (8.4)$$

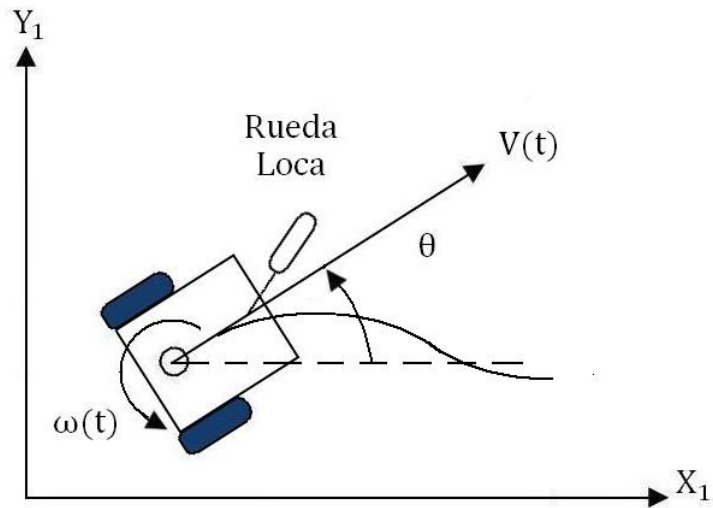
Dada una cierta velocidad  $(\dot{x}, \dot{y}, \dot{\theta})$ , en el marco de referencia global se pueden calcular las componentes de movimiento a lo largo de los ejes locales  $X_R$  y  $Y_R$  de este robot. En este caso, debido al ángulo específico del robot, el movimiento a lo largo  $X_R$  es igual a  $\dot{y}$  y el movimiento a lo largo de  $Y_R$  es  $-\dot{x}$ .

$$\dot{\xi}_R = R\left(\frac{\pi}{2}\right) \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{y} \\ -\dot{x} \\ \dot{\theta} \end{bmatrix} \quad (8.5)$$

### 8.3.2. Modelos cinemáticos de avance

En los casos más simples, el mapeo descrito por la ecuación 8.3 es suficiente para generar una fórmula que capture la cinemática directa del robot móvil: ¿cómo se mueve

Figura 8.10: Un robot de conducción diferencial en su marco de referencia global.



(Fuente: Autores del proyecto.)

el robot, dada su geometría y las velocidades de sus ruedas? Más formalmente, considere el ejemplo mostrado en la figura 8.10.

Este robot de tracción diferencial tiene dos ruedas, cada una con un diámetro  $r$ . Dado un punto  $P$  centrado entre las dos ruedas, cada rueda a una distancia  $l$  de  $P$ . Dados  $r, l, \theta$ , y la velocidad de giro de cada rueda,  $\dot{\varphi}_1$  y  $\dot{\varphi}_2$  un modelo cinemático podría predecir la velocidad global del robot en el marco de referencia global:

$$\dot{\xi}_I = f(l, r, \theta, \dot{\varphi}_1, \dot{\varphi}_2) \quad (8.6)$$

De la ecuación 8.3 sabemos que podemos calcular el movimiento del robot en el marco de referencia global del movimiento en su marco de referencia local

$$\dot{\xi}_I = R(\theta^{-1})\dot{\xi}_R$$

Por lo tanto, la estrategia será primero calcular la contribución de cada una de las dos ruedas en el marco de referencia local,  $\dot{\xi}_R$ . Para este ejemplo de un chasis de tracción diferencial, este problema es particularmente sencillo. Supongamos que el marco de referencia local del robot está alineado de tal manera que el robot se mueve hacia adelante a lo largo de  $+X_R$  como se muestra en la figura 8.8. Consideremos en primer lugar la contribución de la velocidad de giro de cada rueda a la velocidad de traslación en  $P$  en la dirección de  $+X_R$ . Si una rueda gira, mientras que la otra rueda no contribuye en nada y esta parada, dado que  $P$  está en medio de las dos ruedas, este se moverá de forma instantánea con la mitad de la velocidad:

$$\dot{x}_{r1} = \frac{1}{2}r\dot{\varphi}_1 \text{ y } \dot{x}_{r2} = \frac{1}{2}r\dot{\varphi}_2$$

En un robot diferencial, estas dos contribuciones se pueden añadir simplemente para calcular la componente  $X_R$  de  $\dot{\xi}_R$ . Considerando, por ejemplo, un robot diferencial en el que cada rueda gira con una velocidad igual pero en direcciones opuestas. El resultado es un robot estacionario que gira. Como se espera  $\dot{x}_R$  será cero en este caso. El valor de  $\dot{y}_R$  es incluso más simple de calcular. Ninguna rueda puede contribuir a movimiento lateral en el marco de referencia del robot, y así  $\dot{y}_R$  es siempre cero.

Finalmente se puede calcular la componente rotacional  $\dot{\theta}_R$  de  $\dot{\xi}_R$ . Una vez más, las contribuciones de cada rueda se pueden calcular de manera independiente y solo sumarlas. Considere la rueda derecha (vamos a llamar a esta rueda 1). El giro hacia adelante de esta rueda, resulta en rotación en sentido anti horario en el punto  $P$ . Recordemos que si la rueda gira sola, el robot pivota alrededor de la rueda 2. La velocidad de rotación  $\omega_1$  en  $P$  puede ser calculada porque la rueda se está moviendo instantáneamente a lo largo del arco de un círculo de radio  $2l$ :

$$\omega_1 = \frac{r\dot{\varphi}_1}{2l} \quad (8.7)$$

El mismo cálculo se aplica a la rueda izquierda, con la excepción de que la rotación hacia adelante resulta en una rotación en sentido de las manecillas del reloj en el punto  $P$ :

$$\omega_2 = \frac{-r\dot{\varphi}_2}{2l} \quad (8.8)$$

Combinando estas fórmulas individuales se obtiene un modelo cinemático para el robot diferencial del ejemplo:

$$\dot{\xi}_I = R(\theta)^{-1} \begin{bmatrix} \frac{r\dot{\varphi}_1}{2} + \frac{r\dot{\varphi}_2}{2} \\ 0 \\ \frac{r\dot{\varphi}_1}{2l} + \frac{-r\dot{\varphi}_2}{2l} \end{bmatrix} \quad (8.9)$$

Ahora podemos usar este modelo cinemático en un ejemplo. Sin embargo, primero tenemos que calcular  $R(\theta)^{-1}$ . En general, el cálculo de la inversa de una matriz puede ser un reto. En este caso, sin embargo, es fácil debido a que es simplemente una transformación desde  $\dot{\xi}_R$  a  $\dot{\xi}_I$  y no al revés:

$$R(\theta)^{-1} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (8.10)$$

Supongamos que el robot se coloca de tal forma que  $\theta = \pi/2$ ,  $r = 1, l = 1$ . Si el robot emplea sus ruedas de manera desigual, con velocidades  $\dot{\varphi}_1 = 4$  y  $\dot{\varphi}_2 = 2$  podemos

calcular su velocidad en el marco de referencia global:

$$\dot{\xi}_I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 1 \end{bmatrix} \quad (8.11)$$

Así que este robot se moverá de forma instantánea a lo largo del eje  $y$  del marco de referencia global a una velocidad 3, mientras que gira con velocidad 1. Este enfoque de modelado cinemático puede proporcionar información sobre el movimiento de un robot dadas sus componentes de velocidad de las ruedas en los casos sencillos. Sin embargo, se desea determinar el espacio de los posibles movimientos para cada diseño de chasis del robot. Para ello, debemos ir más allá, al describir formalmente las restricciones impuestas por cada rueda sobre el movimiento del robot. **El taller de simulación cinemática se puede encontrar en el Anexo C página 609.**

### 8.3.3. Restricciones cinemáticas de la ruedas

El primer paso para un modelo cinemático del robot consiste en expresar las limitaciones en los movimientos de las ruedas individualmente. Tal como mostramos anteriormente, los movimientos de las ruedas individuales más tarde se pueden combinar para calcular el movimiento del robot como un todo. También como se describió en la sección 8.2, hay cuatro tipos básicos de ruedas con una amplia variedad propiedades cinemáticas.

Por lo tanto, comenzamos por la presentación de los conjuntos de restricciones específicas de cada tipo de rueda. Se considerarán 5 tipos básicos de ruedas:

- Rueda Estándar Fija
- Rueda Estándar Orientable
- Rueda Loca (Castor)
- Rueda Sueca
- Rueda Esférica

Para esto, haremos unos supuestos importantes para simplificar esta presentación. Se supone que el plano de la rueda siempre permanece vertical y que hay en todos los casos un solo punto de contacto entre la rueda y el plano del suelo. Además, se supone que no hay deslizamiento en este punto de contacto único. Es decir, la rueda

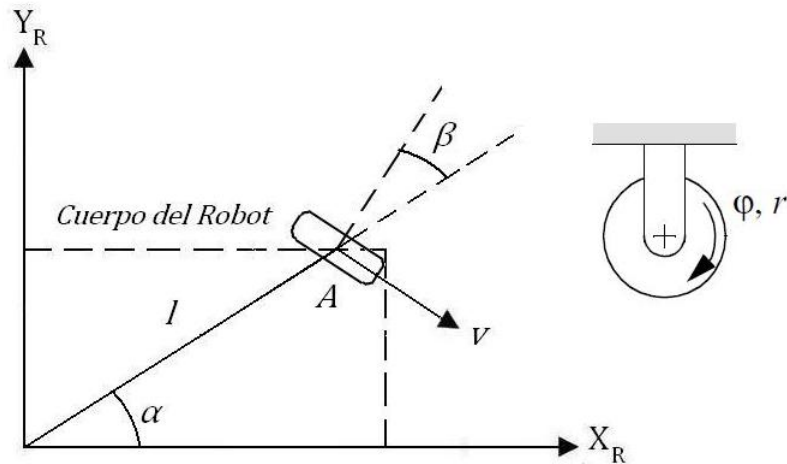
experimenta movimiento sólo bajo condiciones de rodadura pura y rotación alrededor del eje vertical que pasa por el punto de contacto.

Bajo estos supuestos, durante el movimiento, se presentan dos restricciones para cada tipo de rueda. La primera restricción aplica el concepto de contacto de rodadura - que la rueda debe rodar cuando el movimiento tiene lugar en la dirección apropiada. La segunda restricción aplica el concepto de no deslizamiento lateral - que la rueda no debe deslizarse ortogonal al plano de la rueda.

### Rueda Estándar Fija.

La rueda estándar fija no tiene eje vertical de rotación para dirección. Su ángulo con respecto al chasis está así fijo, y se limita a un movimiento hacia adelante y atrás a lo largo del plano de la rueda y la rotación alrededor de su punto de contacto con el plano del suelo. La figura 8.11 muestra una rueda estándar fija e indica su posición en relación con el marco de referencia local del robot ( $X_R, Y_R$ ). La posición de  $A$  se expresa en coordenadas polares por la distancia  $l$  y el ángulo  $\alpha$ . El ángulo del plano de la rueda con respecto al chasis se denota por  $\beta$ , el cual es fijo dado que la rueda estándar fija no es orientable. La rueda, que tiene un radio  $r$ , puede girar en el tiempo, y por tanto su posición rotacional alrededor de su eje horizontal es una función del tiempo  $t$ :  $\varphi(t)$ .

Figura 8.11: Una rueda fija estándar y sus parámetros.



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition.)

La restricción de la rodadura para esta rueda obliga a que todo el movimiento

a lo largo de la dirección del plano de la rueda deba ser acompañado por la cantidad apropiada de giro de la rueda de modo que haya rodadura pura en el punto de contacto:

$$\begin{bmatrix} \sin(\alpha + \beta) & -\cos \alpha + \beta & (-l) \cos \beta \end{bmatrix} \dot{\xi}_R - r\dot{\varphi} = 0$$

$$\begin{bmatrix} \sin(\alpha + \beta) & -\cos \alpha + \beta & (-l) \cos \beta \end{bmatrix} \begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta}_R \end{bmatrix} - r\dot{\varphi} = 0$$

Sabiendo que  $\dot{\xi}_R = r(\theta)\dot{\xi}_I$  tenemos

$$\begin{bmatrix} \sin(\alpha + \beta) & -\cos \alpha + \beta & (-l) \cos \beta \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta}_R \end{bmatrix} - r\dot{\varphi} = 0$$

Simplificando

$$\begin{bmatrix} \sin(\alpha + \beta) & -\cos \alpha + \beta & (-l) \cos \beta \end{bmatrix} R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0 \quad (8.12)$$

El primer término de la suma representa el movimiento total a lo largo del plano de la rueda. Los tres elementos del vector de la izquierda representan los mapeos de cada uno de  $\dot{x}, \dot{y}, \dot{\theta}$  a sus contribuciones para el movimiento a lo largo del plano de la rueda. Note que el término  $R(\theta)\dot{\xi}_I$  es usado para transformar los parámetros de movimiento  $\dot{\xi}_I$  que están en el marco de referencia global ( $X_1, Y_1$ ) en parámetros de movimiento en el marco de referencia local ( $X_R, Y_R$ ) como se muestra en la ecuación ejemplo 8.5. Esto es necesario porque todos los demás parámetros en la ecuación  $\alpha, \beta, l$ , están en términos de el marco de referencia local del robot. Este movimiento a lo largo del plano de la rueda debe ser igual, de acuerdo con esta restricción, al movimiento realizado al girar la rueda,  $r\dot{\varphi}$ . Ahora, la restricción de deslizamiento para esta rueda obliga a que la componente de movimiento de la rueda ortogonal al plano de la misma deba ser cero:

$$\begin{bmatrix} \cos(\alpha + \beta) & \sin(\alpha + \beta) & l \sin \beta \end{bmatrix} R(\theta)\dot{\xi}_I = 0 \quad (8.13)$$

$$\begin{bmatrix} \cos(\alpha + \beta) & \sin(\alpha + \beta) & l \sin \beta \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = 0$$

Por ejemplo suponiendo que la rueda  $A$  esta en una posición tal que ( $\alpha = 0$  y  $\beta = 0$ ). Esto colocaría el punto de contacto de la rueda sobre  $X_1$  con el plano de la

rueda orientado paralelamente a  $Y_1$ . Si  $\theta = 0$  entonces la constante de deslizamiento se reduce a:

$$\begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = 0 \quad (8.14)$$

Esto restringe la componente de movimiento a lo largo de  $X_I$  a cero y dado que  $X_I$  y  $X_R$  son paralelas en este ejemplo, la rueda no se desliza lateralmente, como se esperaba. Cumpliendo con la restricción de deslizamiento.

### Rueda Estándar Orientable.

La rueda orientable estándar difiere de la rueda fija estándar sólo en que hay un grado de libertad adicional: la rueda puede rotar alrededor de un eje vertical que pasa a través del centro de la rueda y el punto de contacto con el suelo. Las ecuaciones de posición para la rueda orientable estándar (Figura 8.12) son idénticas a la de la rueda fija estándar se muestra en la figura 8.11, con una excepción. La orientación de la rueda al chasis del robot ya no es un valor fijo único,  $\beta$ , sino que varia como una función del tiempo  $\beta(t)$ .

Las restricciones de rodadura y deslizamiento son:

$$\begin{bmatrix} \sin(\alpha + \beta) & -\cos(\alpha + \beta) & (-l) \sin \beta \end{bmatrix} R(\theta) \dot{\xi}_I - r \dot{\varphi} = 0 \quad (8.15)$$

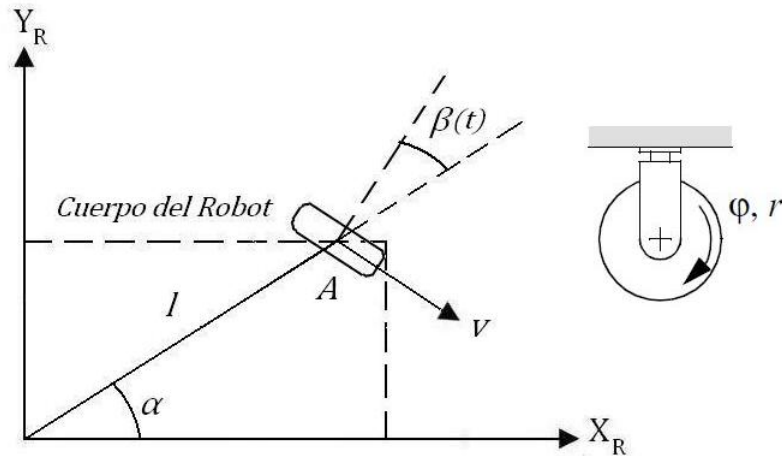
$$\begin{bmatrix} \cos(\alpha + \beta) & \sin(\alpha + \beta) & l \sin \beta \end{bmatrix} R(\theta) \dot{\xi}_I = 0 \quad (8.16)$$

Estas restricciones son idénticas a las de la rueda estándar fija porque, la única diferencia es que  $\dot{\varphi}$ ,  $\dot{\beta}$ , no tienen un impacto directo sobre las restricciones de movimiento instantáneo de un robot (en el caso de la rueda estándar orientable). Es sólo mediante la integración en el tiempo que los cambios en el ángulo de dirección pueden afectar a la movilidad de un vehículo. Esto puede parecer sutil, pero es una distinción muy importante entre el cambio en la posición de dirección,  $\dot{\beta}$ , y el cambio en el giro de la rueda,  $\dot{\varphi}$ .

### Rueda Loca (Castor).

Las ruedas locas son capaces de orientarse alrededor de un eje vertical. Sin embargo, a diferencia de la rueda estándar orientable, el eje vertical de rotación en una rueda loca no pasa por el punto de contacto con el suelo. La figura 8.13 representa una rueda loca, lo que demuestra que la especificación formal de la posición de la rueda loca requiere

Figura 8.12: Una rueda orientable estándar y sus parámetros.



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition.)

un parámetro adicional. El punto de contacto de la rueda se encuentra ahora en una posición  $B$ , la cual está conectada por una barra rígida  $AB$  de longitud fija  $d$  hasta el punto  $A$  que está fijo a la posición del eje vertical sobre el cual gira  $B$ , y este punto  $A$  tiene una posición definida en el marco de referencia del robot como se muestra en la figura 8.13.

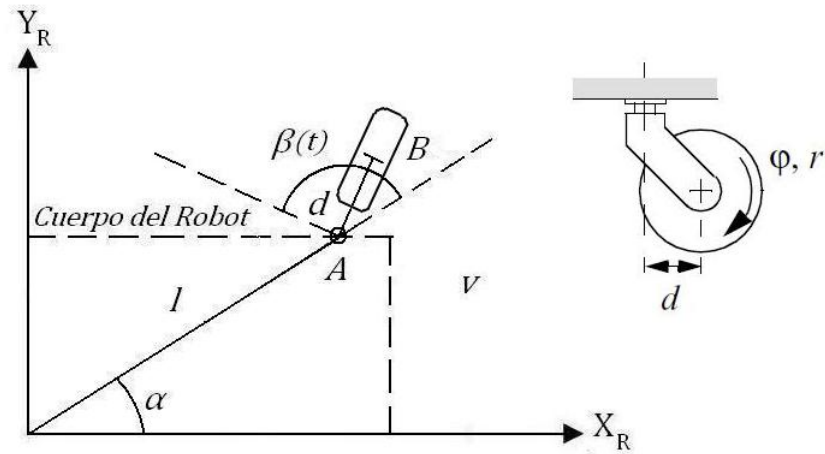
Se supone que el plano de la rueda está alineado con  $AB$  en todo momento. Similar a la rueda estándar orientable, la rueda loca tiene dos parámetros que varían como una función del tiempo.  $\varphi(t)$  representa el giro de la rueda en función del tiempo como se mencionó anteriormente y  $\beta(t)$  denota el ángulo de dirección y orientación de  $AB$  en el tiempo.

Para la rueda loca, la restricción de rodadura es idéntica a la ecuación 8.15. Es decir la de la rueda estándar orientable, ya que la distancia del eje no desempeña ningún papel durante el movimiento que está alineado con el plano de la rueda:

$$\begin{bmatrix} \sin(\alpha + \beta) & -\cos(\alpha + \beta) & (-l) \sin \beta \end{bmatrix} R(\theta) \dot{\xi}_I - r \dot{\varphi} = 0 \quad (8.17)$$

La geometría de esta rueda, sin embargo, tiene un impacto significativo sobre la restricción de deslizamiento. La cuestión fundamental es que la fuerza lateral en la rueda se produce en el punto  $A$  porque este es el punto de fijación de la rueda al chasis. Debido a la distancia del punto de contacto con el suelo con relación a  $A$ , la restricción de que no haya movimiento lateral sería un error. En cambio, la restricción es muy similar a una restricción de rodadura, en el que debe llevarse a cabo la rotación

Figura 8.13: Una rueda loca y sus parámetros.



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition.)

apropiada del eje vertical:

$$\begin{bmatrix} \cos(\alpha + \beta) & \sin(\alpha + \beta) & d + l \sin \beta \end{bmatrix} R(\theta) \dot{\xi}_I + d \dot{\beta} = 0 \quad (8.18)$$

En la ecuación 8.18, cualquier movimiento ortogonal al plano de la rueda debe ser equilibrado por una cantidad equivalente y opuesta al movimiento de dirección de la castor. Este resultado es crítico para el éxito de las ruedas locas porque al establecer el valor de  $\dot{\beta}$  cualquier movimiento lateral arbitrario puede ser aceptable.

Figura 8.14: Ruedas castor de una Silla de oficina.



(Fuente: Sistemas de locomoción de robots móviles, Automatización y Robótica Móvil.)

Más concisamente, puede suponerse a partir de las ecuaciones 8.17 y 8.18 que, dado cualquier movimiento del chasis del robot  $\dot{\xi}_I$  existe un cierto valor de velocidad de giro  $\dot{\varphi}$  y velocidad de dirección  $\dot{\beta}$  de tal manera que las restricciones se cumplan. Por lo tanto,

un robot con sólo ruedas locas puede moverse con cualquier velocidad en el espacio de posibles movimientos del robot. Denominamos a tales sistemas “omnidireccionales”.

Un ejemplo del mundo real de este sistema es la silla de oficina de cinco ruedas locas que se muestra en la figura 8.14. Suponiendo que todas las uniones son capaces de moverse libremente, usted puede seleccionar cualquier vector de movimiento en el plano de la silla y empujarla con la mano. Sus ruedas locas girarán y se orientarán cuando sea necesario para lograr ese movimiento sin punto de contacto deslizante. Por la misma razón, si cada una de las ruedas locas de la silla alojase dos motores, uno para el giro y uno para la dirección, entonces, un sistema de control sería capaz de mover la silla a lo largo de cualquier trayectoria en el plano. Por lo tanto, a pesar de que la cinemática de las ruedas locas es algo compleja, tales ruedas no imponen restricciones reales sobre la cinemática del chasis de un robot.

### **Rueda Sueca.**

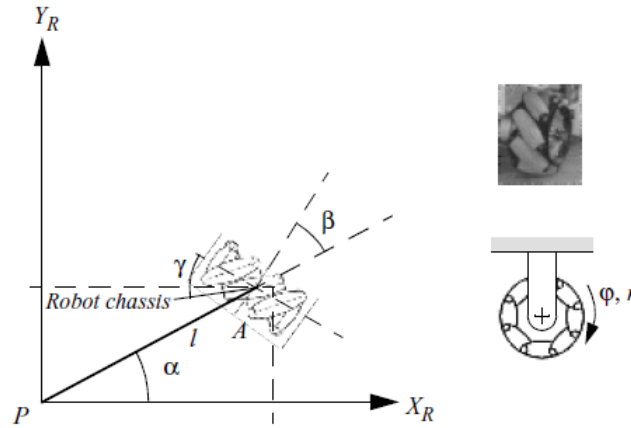
Las ruedas suecas no tienen eje vertical de rotación, y sin embargo son capaces de moverse omnidireccionalmente como las ruedas locas. Esto es posible mediante la adición de un grado de libertad a la rueda estándar fija. Las ruedas suecas consisten en una rueda estándar fija con rodillos unidos al perímetro de la rueda con ejes que son antiparalelos al eje principal de la componente de la rueda fija. El ángulo exacto  $\gamma$  entre los ejes de los rodillos y el eje principal puede variar, como se muestra en la figura 8.15.

Por ejemplo, dada una rueda sueca de 45 grados, los vectores de movimiento del eje principal y los ejes de rodillos se pueden dibujar como en la figura 8.15. Dado que cada eje puede girar en sentido horario o antihorario, se puede combinar cualquier vector a lo largo de un eje con cualquier vector a lo largo del otro eje. Estos dos ejes no son necesariamente independientes (excepto en el caso de la rueda sueca de 90 grados), sin embargo, es visualmente evidente que cualquier dirección deseada de movimiento es alcanzable mediante la elección de dos vectores apropiados.

La posición de una rueda sueca se expresa exactamente como en una rueda estándar fija, con la adición de un término,  $\gamma$ , que representa el ángulo entre el plano principal de la rueda y el eje de rotación de los pequeños rodillos circunferenciales. Esto se ilustra en la figura 8.15 dentro del marco de referencia del robot.

La formulación de la restricción para una rueda sueca requiere un poco de sutileza. La restricción instantánea es debido a la orientación específica de los pequeños rodillos. El eje alrededor del cual estos rodillos giran es una componente cero de velocidad en el punto de contacto. Es decir, moverse en esa dirección sin girar el eje principal no es posible sin deslizamiento. La restricción de movimiento que se deriva parece idéntica a

Figura 8.15: Rueda Sueca y sus parámetros.



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition.)

la restricción de rodadura de la rueda estándar fija en la ecuación 8.12, excepto que la fórmula se modifica mediante la adición de  $\gamma$  tal que la dirección efectiva en la cual la restricción de rodadura se mantiene es a lo largo de esta componente cero en lugar de a lo largo del plano de la rueda:

$$\begin{bmatrix} \sin(\alpha + \beta + \gamma) & -\cos(\alpha + \beta + \gamma) & (-l) \cos(\beta + \gamma) \end{bmatrix} R(\theta) \dot{\xi}_I - r \dot{\varphi} \cos \gamma = 0 \quad (8.19)$$

Ortogonal a esta dirección el movimiento no está restringido debido a la rotación libre  $\dot{\varphi}_{SW}$  de los pequeños rodillos.

$$\begin{bmatrix} \cos(\alpha + \beta + \gamma) & \sin(\alpha + \beta + \gamma) & l \sin(\beta + \gamma) \end{bmatrix} R(\theta) \dot{\xi}_I - r \dot{\varphi} \sin \gamma - r_{SW} \dot{\varphi}_{SW} = 0 \quad (8.20)$$

El comportamiento de esta restricción y por tanto en la rueda sueca cambia dramáticamente a medida que el valor de  $\gamma$  varía. Considerando  $\gamma = 0$  representa la rueda sueca de 90 grados. En este caso, la componente cero de velocidad está en línea con el plano de la rueda y entonces la ecuación 8.19 se reduce exactamente a la ecuación 8.12, que es la restricción de rodadura de la rueda estándar fija. Pero debido a los rodillos, no hay ninguna restricción de deslizamiento ortogonal al plano de la rueda (ecuación 8.20).

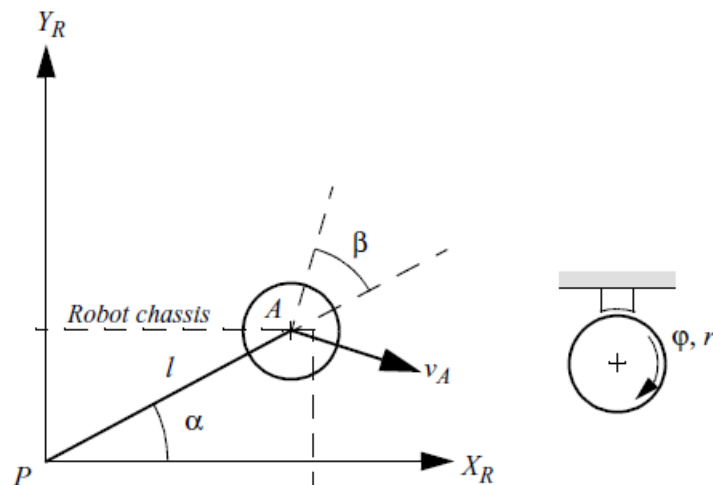
Al variar el valor de  $\varphi$ , se puede hacer cualquier vector de movimiento deseado para satisfacer la ecuación 8.19 y por lo tanto la rueda es omnidireccional. De hecho, este caso especial de diseño sueco resulta en movimiento completamente desacoplado, en que los rodillos y la rueda principal proporcionan direcciones ortogonales de movimiento.

En el otro extremo, consideramos  $\gamma = \pi/2$ . En este caso, los rodillos tienen ejes de rotación que son paralelos al eje principal de rotación de la rueda. Curiosamente, si este valor es sustituido por  $\gamma$  en la ecuación 8.19, el resultado es la restricción de deslizamiento de la rueda estándar fija, ecuación 8.13. En este caso la rueda principal nunca tiene que girar y por lo tanto, la restricción de rodadura desaparece. Esta es una forma “degenerada” de la rueda Sueca y por lo tanto se asume en el resto de este capítulo que  $\gamma \neq \pi/2$ .

### Rueda esférica.

El último tipo de rueda, una bola o rueda esférica, no impone restricciones directas sobre el movimiento (Figura 8.16). Dicho mecanismo no tiene eje principal de la rotación, y por lo tanto, no existen restricciones adecuadas de rodadura o deslizamiento. Al igual que con las ruedas locas y las ruedas suecas, la rueda esférica es claramente omnidireccional y no impone limitaciones sobre la cinemática del chasis del robot.

Figura 8.16: Una rueda esférica y sus parámetros.



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition.)

Por lo tanto la ecuación 8.21 describe simplemente la velocidad de rodadura de la esfera en la dirección del movimiento  $v_A$  del punto  $A$  del robot.

$$\begin{bmatrix} \sin(\alpha + \beta) & -\cos(\alpha + \beta) & (-l) \cos(\beta) \end{bmatrix} R(\theta) \dot{\xi}_I - r \dot{\varphi} = 0 \quad (8.21)$$

Por definición, la rotación de la rueda, ortogonal a esta dirección es cero.

$$\begin{bmatrix} \cos(\alpha + \beta) & \sin(\alpha + \beta) & l \sin \beta \end{bmatrix} R(\theta) \dot{\xi}_I = 0 \quad (8.22)$$

Como puede verse, las ecuaciones para la rueda esférica son exactamente las mismas que para la rueda estándar fija. Sin embargo, la interpretación de la ecuación 8.22 es diferente. La rueda esférica omnidireccional puede tener cualquier dirección arbitraria de movimiento, donde la dirección de movimiento dada por  $\beta$  es una variable libre deducida de la ecuación 8.22. Consideremos el caso en que el robot está en traslación pura en la dirección de  $Y_R$ . Entonces la ecuación 8.22 se reduce a  $\sin(\alpha + \beta) = 0$  entonces ( $\beta = -\alpha$ ) lo cual tiene sentido para este caso especial.

### 8.3.4. Restricciones Cinemáticas del Robot

Dado un robot móvil con  $M$  ruedas ahora podemos calcular las limitaciones cinemáticas del chasis del robot. La idea clave es que cada rueda impone cero o más restricciones en el movimiento del robot, por lo que el proceso es simplemente una combinación adecuada de todas las restricciones cinemáticas derivadas de todas las ruedas, basada en la posición de esas ruedas en el chasis del robot.

Hemos clasificado todas las ruedas en cinco categorías: (1) Ruedas estándar fija y (2) Ruedas estándar orientables, (3) Ruedas locas, (4) Ruedas Suecas, y (5) Ruedas esféricas. Fijándose en las restricciones cinemáticas de la rueda en las ecuaciones 8.17, 8.18, y 8.19 que la rueda loca, la rueda sueca, y la rueda esférica no imponen restricciones cinemáticas sobre el chasis del robot, dado que  $\dot{\xi}_I$  puede oscilar libremente en todos estos casos debido a los grados de libertad internos de las ruedas.

Por tanto, sólo las ruedas estándar fijas y las ruedas estándar orientables tienen un impacto sobre la cinemática del chasis del robot y por lo tanto, requieren una consideración al calcular las restricciones cinemáticas del robot. Supongamos que el robot cuenta con un total de  $N$  ruedas estándar, que comprende  $N_f$  ruedas estándar fijas y  $N_S$  ruedas estándar orientables. Usaremos  $\beta_s(t)$  para denotar los ángulos de orientación variables de las  $N_S$  (Ruedas estándar orientables). En contraste  $\beta_f$  se refiere a la orientación de las  $N_f$  (Ruedas estándar fijas) como se representa en la figura 8.11. En el caso de giro de la rueda, tanto las ruedas fijas como las orientables tienen posiciones de rotación alrededor del eje horizontal que varían como una función del tiempo. Se indican los casos fijos y orientables por separado como  $\varphi_f(t)$  y  $\varphi_s(t)$  y usamos  $\varphi(t)$  como una matriz agregada que combina ambos valores:

$$\varphi(t) = \varphi \begin{bmatrix} \varphi_f(t) & \varphi_s(t) \end{bmatrix} \quad (8.23)$$

Las restricciones de rodadura de todas las ruedas pueden ahora ser reunidas en una sola expresión:

$$J_1(\beta_s)R(\theta)\dot{\xi}_I - J_2\dot{\varphi} = 0 \quad (8.24)$$

Esta expresión tiene un gran parecido a la de restricción de rodadura de una sola rueda, pero se sustituye por matrices en lugar de valores individuales, teniendo en cuenta así todas las ruedas.  $J_2$  es una matriz  $N \times N$  diagonal constante cuyas entradas son radios  $r$  de todas las ruedas estándar.  $J_1(\beta_s)$  denota una matriz con proyecciones para todas las ruedas a sus movimientos a lo largo de sus planos individuales de la rueda:

$$J_1(\beta_s) = \begin{bmatrix} J_{1f} \\ J_{1s}(\beta_s) \end{bmatrix} \quad (8.25)$$

Nótese que  $J_1(\beta_s)$  es solo una función de  $\beta_s$  y no de  $\beta_f$ . Esto es debido a que las orientaciones de las ruedas estándar orientables varían como una función del tiempo, mientras que las orientaciones de ruedas estándar fijas son constantes.  $J_{1f}$ , es por tanto una matriz constante de proyecciones para todas las ruedas estándar fijas. Esta tiene un tamaño de  $N_f \times 3$  con cada fila consistente de tres términos en la matriz de tres a partir de la ecuación 8.12 para cada rueda estándar fija.  $J_{1s}(\beta_s)$  es una matriz de tamaño  $N_s \times 3$  con cada fila que consta de los tres términos en la matriz  $3 \times 3$  a partir de la ecuación 8.15 para cada rueda estándar orientable.

En resumen, la ecuación 8.24 representa la restricción de que todas las ruedas estándar deben girar alrededor de su eje horizontal con una cantidad apropiada basada en sus movimientos a lo largo del plano de la rueda de modo que se produzca rodadura en el punto de contacto con el suelo. Utilizamos la misma técnica para reunir las restricciones de deslizamiento de las cuatro ruedas estándar en una única expresión con la misma estructura que las ecuaciones 8.13 y 8.16:

$$C_1(\beta_s)R(\theta)\dot{\xi}_I = 0 \quad (8.26)$$

$$C_1(\beta_s) = \begin{bmatrix} C_{1f} \\ C_{1s}(\beta_s) \end{bmatrix} \quad (8.27)$$

$C_{1f}$  y  $C_{1s}$  son matrices ( $N_f \times 3$ ) y ( $N_s \times 3$ ) cuyas filas son los tres términos en la matriz triple de las ecuaciones 8.13 y 8.16 para todas las ruedas estándar fijas y orientables. Así la ecuación 8.26 es una restricción sobre todas las ruedas estándar y sus componentes de movimiento ortogonal a los planos de sus ruedas debe ser cero. Esta restricción de deslizamiento sobre todas las ruedas estándar tiene el impacto más significativo en la definición de la maniobrabilidad general del chasis del robot, como se explica más adelante.

### 8.3.5. Algunos ejemplos de modelos y restricciones cinemáticas del robot

Anteriormente se presentó una solución cinemática para  $\dot{\xi}_I$  en el caso de un robot diferencial simple mediante la combinación de la contribución de cada rueda al movimiento del robot. Ahora se utilizarán tales herramientas para construir la misma expresión cinemática por aplicación directa de las restricciones de rodadura para cada tipo de rueda. Luego se aplica esta técnica de nuevo al robot unidad diferencial, para realizar la verificación del método en comparación con los resultados de la sección anterior.

#### Un ejemplo de Robot de conducción diferencial.

Primero, nos referimos a las ecuaciones 8.24 y 8.26. Estas ecuaciones relacionan el movimiento del robot a las restricciones de rodadura y deslizamiento  $J_1(\beta_s)$  y  $C_1(\beta_s)$ , y la velocidad de giro de las ruedas del robot,  $\dot{\varphi}$ . Al fusionar estas dos ecuaciones se obtiene la siguiente expresión:

$$\begin{bmatrix} J_1(\beta_s) \\ C_1(\beta_s) \end{bmatrix} R(\theta)\dot{\xi}_I = \begin{bmatrix} J_2\dot{\varphi} \\ 0 \end{bmatrix} \quad (8.28)$$

Una vez más, se debe tener en cuenta el robot de conducción diferencial en la figura 8.10. Se construirá  $J_1(\beta_s)$  y  $C_1(\beta_s)$  directamente de las restricciones de rodadura de cada rueda. La rueda loca no está alimentada y es libre de moverse en cualquier dirección, por lo que se deja de lado este tercer punto de contacto por completo. Las dos ruedas conductoras restantes no son orientables, y por lo tanto  $J_1(\beta_s)$  y  $C_1(\beta_s)$  se simplifican a  $J_{1f}$  y  $C_{1f}$  respectivamente. Para emplear la fórmula de restricción de rodadura de la rueda estándar fija, la ecuación 8.12, primero se deben identificar los valores de  $\alpha$  y  $\beta$  de cada una de las ruedas. Se supone que el marco local de referencia del robot está alineado de tal manera que el robot se mueve hacia adelante a lo largo de  $+X_R$ . En este caso para la rueda derecha  $\alpha = -\pi/2, \beta = \pi$ , y para la rueda izquierda  $\alpha = \pi/2, \beta = 0$ . Nótese que en el valor de  $\beta$  para la rueda derecha es necesario garantizar que el giro positivo cause el movimiento en la dirección  $+X_R$  (figura 8.11). Ahora se pueden calcular las matrices  $J_{1f}$  y  $C_{1f}$  utilizando los términos de las matrices de las ecuaciones 8.12 y 8.13. Debido a que las dos ruedas estándar fijas son paralelas, la ecuación 8.13 se traduce en una sola ecuación independiente, y la ecuación 8.28 genera:

$$\begin{bmatrix} \begin{bmatrix} 1 & 0 & l \\ 1 & 0 & -l \\ 0 & 1 & 0 \end{bmatrix} \end{bmatrix} R(\theta) \dot{\xi}_I = \begin{bmatrix} J_2 \varphi \\ 0 \end{bmatrix} \quad (8.29)$$

Al invertir la ecuación 8.29 se obtiene la ecuación cinemática específica de nuestro robot diferencial:

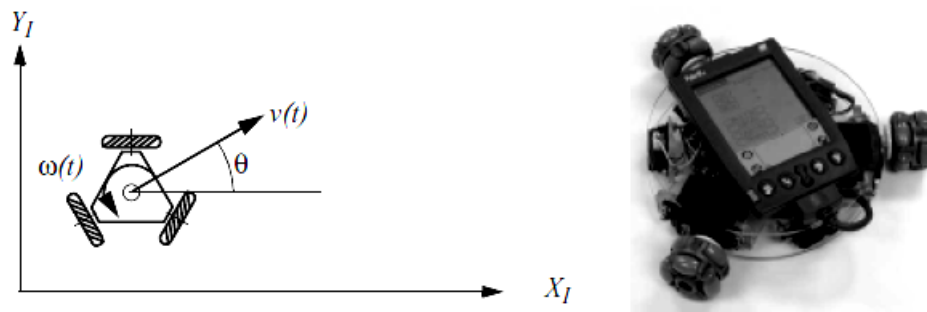
$$\dot{\xi}_I = R(\theta)^{-1} \begin{bmatrix} 1 & 0 & l \\ 1 & 0 & -l \\ 0 & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} J_2 \varphi \\ 0 \end{bmatrix} = R(\theta)^{-1} \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \\ \frac{1}{2l} & -\frac{1}{2l} & 0 \end{bmatrix} \begin{bmatrix} J_2 \varphi \\ 0 \end{bmatrix} \quad (8.30)$$

Esto demuestra que, para el simple caso de conducción diferencial, la combinación de las restricciones de deslizamiento y rodadura de la rueda describe el comportamiento cinemático, basado en el cálculo manual hecho anteriormente.

### Un ejemplo de robot omnidireccional.

Se considera el robot de ruedas omnidireccionales que se muestra en la figura 8.17. Este robot tiene tres ruedas Suecas 90, dispuestas radialmente de forma simétrica, con los rodillos perpendiculares a cada rueda principal.

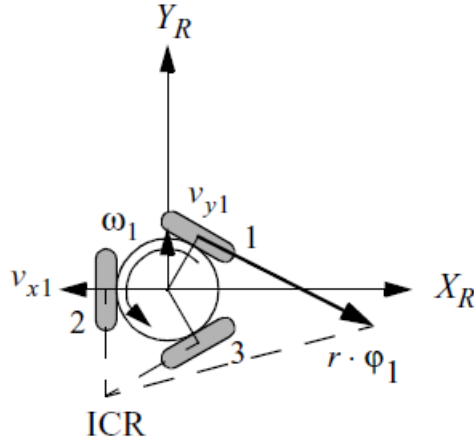
Figura 8.17: Un robot Omnidireccional de tres ruedas desarrollado por la Universidad Carnegie Mellon.



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition.)

En primer lugar se tiene que imponer un marco específico de referencia local en el robot. Esto se hace eligiendo el punto  $P$  en el centro del robot, a continuación, alineando el robot con el marco de referencia local tal que  $X_R$  sea colineal con el eje de rueda 2. La figura 8.18 muestra el robot y su marco de referencia local dispuestos de esta manera.

Figura 8.18: El marco local de referencia y los parámetros detallados para la rueda 1.



(Fuente: Sistemas de locomoción de robots móviles, Automatización y Robótica Móvil.)

Se supone que la distancia entre cada rueda y  $P$  es  $l$ , y que las tres ruedas tienen el mismo radio,  $r$ . Una vez más el valor de  $\dot{\xi}_I$  puede calcularse como una combinación de las restricciones de rodadura de las tres ruedas omnidireccionales del robot, como en la ecuación 8.28. Al igual que con el robot diferencial, dado que este robot no tiene ruedas orientables  $J_1(\beta_s)$  se simplifica a  $J_{1f}$ :

$$\dot{\xi}_I = R(\theta)^{-1} J_{1f}^{-1} J_2 \dot{\varphi} \quad (8.31)$$

Se calcula  $J_{1f}$  utilizando los elementos de la matriz de las restricciones de rodadura de la rueda Sueca, dada por la ecuación 8.19. Pero para usar estos valores se deben establecer los valores de  $\alpha, \beta, \gamma$  para cada rueda. Haciendo referencia a la figura 8.15, se puede ver que  $\gamma = 0$  para la rueda sueca 90. Nótese que esto inmediatamente simplifica la ecuación 8.19 a la ecuación 8.12, las restricciones de rodadura de una rueda estándar fija. Dando una colocación particular del marco de referencia local, el valor de  $\alpha$  para cada rueda se calcula fácilmente:  $(\alpha_1 = \pi/3)$ ,  $(\alpha_2 = \pi)$ ,  $(\alpha_3 = -\pi/3)$ . Además,  $\beta = 0$  para todas las ruedas porque las ruedas son tangentes al cuerpo circular del robot. Construyendo y simplificando  $J_{1f}$  usando la ecuación 8.12, queda

$$J_{1f} = \begin{bmatrix} \sin \frac{\pi}{3} & -\cos \frac{\pi}{3} & -l \\ 0 & -\cos \pi & -l \\ \sin -\frac{\pi}{3} & -\cos -\frac{\pi}{3} & -l \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{2} & \frac{1}{2} & -l \\ 0 & 1 & -l \\ \frac{\sqrt{3}}{2} & \frac{1}{2} & -l \end{bmatrix} \quad (8.32)$$

Una vez más calculando el valor de  $\dot{\xi}_I$  se requiere calcular la inversa  $J_{1f}^{-1}$  según sea

necesario en la ecuación 8.31. Un enfoque consistiría en aplicar los métodos rutinarios para el cálculo de la inversa de una matriz cuadrada 3 x 3. Una segunda aproximación sería calcular la contribución de cada rueda Sueca en el movimiento del chasis. Se deja este proceso como un ejercicio para el estudiante. Una vez que la inversa se obtiene,  $\dot{\xi}_I$  puede ser despejada:

$$\dot{\xi}_I = R(\theta)^{-1} \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & -\frac{1}{\sqrt{3}} \\ -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} \\ -\frac{1}{3l} & -\frac{1}{3l} & -\frac{1}{3l} \end{bmatrix} \quad (8.33)$$

Se considera un chasis específico omnidireccional con  $l = 1$  y  $r = 1$  para todas las ruedas. El marco de referencia local del robot y el marco de referencia global están alineados, por lo que  $\theta = 0$ . Si las ruedas 1, 2, y 3 giran a velocidades ( $\varphi_1 = 4$ ), ( $\varphi_2 = 1$ ), ( $\varphi_3 = 2$ ), ¿cuál es el movimiento resultante de todo el robot? Utilizando la ecuación anterior, la respuesta se puede calcular fácilmente:

$$\dot{\xi}_I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & -\frac{1}{\sqrt{3}} \\ -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} \\ -\frac{1}{3l} & -\frac{1}{3l} & -\frac{1}{3l} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} \frac{2}{\sqrt{3}} \\ -\frac{4}{3} \\ -\frac{7}{3} \end{bmatrix} \quad (8.34)$$

Así que este robot se mueve instantáneamente a lo largo del eje  $x$  con velocidad positiva y a lo largo del eje  $y$  con velocidad negativa mientras gira en sentido horario. Se puede ver en los ejemplos anteriores que el movimiento del robot se puede predecir por la combinación de las restricciones de rodadura de las ruedas individuales. Las restricciones de deslizamiento comprenden  $C_1(\beta_s)$  que se puede utilizar para ir aún más lejos, lo que permite evaluar la capacidad de maniobra y espacio de trabajo del robot en lugar de sólo predecir su movimiento. A continuación, se examinan los métodos para el uso de las restricciones de deslizamiento, a veces en conjunción con las restricciones de rodadura, para generar un análisis de gran alcance de la maniobrabilidad del cuerpo de un robot.

## 8.4. MANIOBRABILIDAD DEL ROBOT MÓVIL

La movilidad cinemática del cuerpo de un robot es su capacidad para moverse directamente en su entorno. La restricción básica que limita la movilidad es la regla de que cada rueda tiene que satisfacer su restricción de deslizamiento. Por lo tanto, formalmente se puede derivar la movilidad del robot a partir de la ecuación 8.26. Además de movimiento cinemático instantáneo, un robot móvil es capaz de manipular aún más su posición, en el tiempo, para dirigir las ruedas orientables.

### 8.4.1. Grado de movilidad

La ecuación 8.26 impone la restricción de que todas las ruedas deban evitar cualquier deslizamiento lateral. Por supuesto, esto se mantiene por separado para cada de las ruedas, y por lo tanto es posible especificar esta restricción por separado para ruedas estándar fijas y orientables:

$$C_1 R(\theta)^{-1} \dot{\xi}_I = 0 \quad (8.35)$$

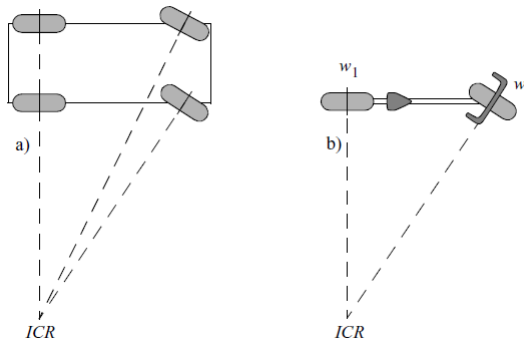
$$C_{1S}(\beta_S) R(\theta)^{-1} \dot{\xi}_I = 0 \quad (8.36)$$

Para que estas dos restricciones sean satisfechas, el vector de movimiento  $R(\theta)\dot{\xi}_I$  debe pertenecer al espacio nulo de la matriz de proyección  $C_1(\beta_S)$ , la cual es una simple combinación de  $C_{1f}$  y  $C_{1S}$ . Matemáticamente el espacio nulo de  $C_1(\beta_S)$  es el espacio  $N$  tal que para cada vector  $n$  en  $N$   $C_1(\beta_S)n = 0$ . Si las restricciones cinemáticas deben ser respetadas, entonces el movimiento del robot siempre debe estar dentro de este espacio  $N$ . Las restricciones cinemáticas (ecuaciones 8.35 y 8.36) también se pueden demostrar geoméricamente, utilizando el concepto de centro instantáneo de rotación del robot (ICR).

Se considera la posibilidad de una rueda estándar única. Esta es obligada por la restricción deslizante a tener movimiento lateral cero. Esto se puede demostrar geoméricamente dibujando una línea de movimiento cero a través de su eje horizontal, perpendicular al plano de la rueda (figura 8.19). En cualquier momento dado, el movimiento de la rueda a lo largo de la línea de movimiento cero debe ser cero. En otras palabras, la rueda debe moverse instantáneamente a lo largo de algún círculo de radio  $R$  de tal manera que el centro de ese círculo está situado en la línea de movimiento cero. Este punto central, llamado centro instantáneo de rotación, puede encontrarse en cualquier punto a lo largo de la línea de movimiento cero. Cuando  $R$  tiende a infinito, la rueda se mueve en una línea recta. Un robot como el vehículo Ackerman en la figura 8.19a puede tener varias ruedas, pero siempre debe tener un solo *ICR*. Debido a que todas sus líneas de movimiento cero se reúnen en un solo punto, hay una solución única para el movimiento del robot, colocando el *ICR* en este punto de encuentro.

Esta construcción geométrica del *ICR* muestra cómo la movilidad del robot es una función del número de restricciones sobre el movimiento del robot, no el número de ruedas. En la figura 8.19b, la bicicleta que se muestra tiene dos ruedas,  $w_1$  y  $w_2$ . Cada rueda contribuye con una restricción o una línea de movimiento cero. Tomando las dos restricciones, resulta en un solo punto como la única solución que queda para el *ICR*. Esto es porque las dos restricciones son independientes, y por lo tanto cada una limita aún más el movimiento del robot en general. Pero en el caso del robot diferencial en

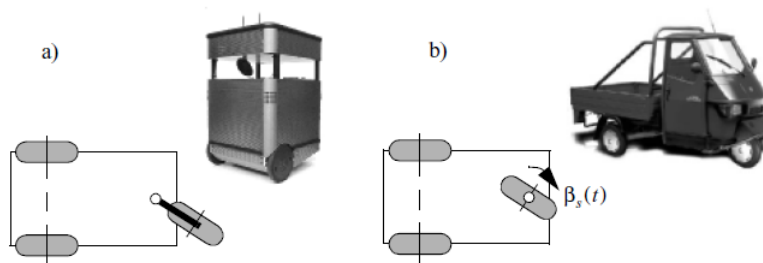
Figura 8.19: (a) Coche de cuatro ruedas con dirección Ackerman. (b) Bicicleta.



(Fuente: Sistemas de locomoción de robots móviles, Automatización y Robótica Móvil.)

la figura 8.20a, las dos ruedas están alineadas a lo largo del mismo eje horizontal. Por lo tanto, el *ICR* se ve obligado a recostarse a lo largo de una línea, no en un punto específico. De hecho, la segunda rueda no impone restricciones cinemáticas adicionales de movimiento al robot dado que su línea de movimiento cero es idéntica a aquella de la primera rueda. Así, aunque la bicicleta y los cuerpos de dirección diferencial tienen el mismo número de ruedas no omnidireccionales, el primero tiene dos restricciones cinemáticas independientes mientras que el segundo tiene sólo una.

Figura 8.20: (a) Robot diferencial con dos ruedas motorizadas individualmente y una rueda loca, e.g., el robot Pigmalión en EPFL. (b) triciclo con dos ruedas fijas estándar y una rueda estándar orientable, e.g., Piaggio minitransporter.



(Fuente: Sistemas de locomoción de robots móviles, Automatización y Robótica Móvil.)

El vehículo Ackerman de la figura 8.19a demuestra otra manera en la que una rueda puede ser incapaz de contribuir con una restricción independiente a la cinemática

del robot. Este vehículo cuenta con dos ruedas orientables estándar. Dada la posición instantánea de sólo una de estas ruedas orientables y la posición de las ruedas traseras fijas, sólo hay una única solución para el *ICR*. La posición de la segunda rueda dirigitible está absolutamente restringida por el *ICR*. Por lo tanto, no ofrece restricciones independientes al movimiento del robot. La ecuación 8.26 representa todas las restricciones de deslizamiento impuestas por las ruedas del robot móvil. Por tanto  $rank[C_1(\beta_s)]$  es el número de restricciones independientes.

Cuanto mayor sea el número de restricciones independientes, y por tanto cuanto mayor sea el grado de  $C_1(\beta_s)$ , estará más restringida la movilidad del robot. Por ejemplo, se considera un robot con una sola rueda estándar fija. Entonces se consideran sólo las ruedas estándar. Este robot puede ser un monociclo o puede tener varias ruedas suecas, sin embargo, tiene exactamente una rueda estándar fija. La rueda está en una posición especificada por los parámetros  $\alpha, \beta, l$  con relación al marco de referencia local del robot.  $C_1(\beta_s)$ , está compuesta por  $C_{1f}$  y  $C_{1s}$ . Sin embargo, puesto que no hay ruedas estándar orientables  $C_{1s}$  está vacía y por tanto  $C_1(\beta_s)$  contiene solo a  $C_{1f}$ . Debido a que hay rueda estándar fija, esta matriz tiene un rango de uno y por lo tanto, este robot tiene una única e independiente restricción en su movilidad:

$$C_1(\beta_s) = C_{1f} = \begin{bmatrix} \cos(\alpha + \beta) & \sin(\alpha + \beta) & l \sin \beta \end{bmatrix} \quad (8.37)$$

Sin perder la generalidad, se puede colocar el punto  $P$  en el punto medio entre los centros de las dos ruedas. Dados  $\alpha_1, \beta_1, l_1$  para la rueda  $w_1$  y  $\alpha_2, \beta_2, l_2$  para la rueda  $w_2$ , se mantiene geoméricamente que  $(l_1 = l_2), (\beta_1 = \beta_2 = 0), (\alpha_1 + \pi = \alpha_2)$ . Por tanto, en este caso, la matriz  $C_1(\beta_s)$  tiene dos restricciones pero un rango de uno:

$$C_1(\beta_s) = C_{1f} = \begin{bmatrix} \cos(\alpha_1) & \sin(\alpha_1) & 0 \\ \cos(\alpha_1 + \pi) & \sin(\alpha_1 + \pi) & 0 \end{bmatrix} \quad (8.38)$$

Alternativamente, considerando el caso cuando  $w_2$  está situada en el plano de la rueda  $w_1$  pero con la misma orientación, como en una bicicleta con la dirección bloqueada en la posición delantera. Se vuelve a colocar el punto  $P$  entre los dos centros de las ruedas, y se orientan las ruedas de tal forma que se encuentran en el eje  $X_1$ . Esta geometría implica que  $(l_1 = l_2), (\beta_1 = \beta_2 = \pi/2), (\alpha_1 = 0), (\alpha_2 = \pi)$ , y por lo tanto la matriz  $C_1(\beta_s)$  conserva dos restricciones independientes y tiene un rango de dos:

$$C_1(\beta_s) = C_{1f} = \begin{bmatrix} \cos(\frac{\pi}{2}) & \sin(\frac{\pi}{2}) & l_1 \sin(\frac{\pi}{2}) \\ \cos(\frac{3\pi}{2}) & \sin(\frac{3\pi}{2}) & l_1 \sin(\frac{\pi}{2}) \end{bmatrix} = \begin{bmatrix} 0 & 1 & l_1 \\ 0 & -1 & l_1 \end{bmatrix} \quad (8.39)$$

En general, si  $rank[C_{1f}] > 1$  entonces el vehículo puede, a lo mejor, sólo viajar a lo largo de un círculo o a lo largo de una línea recta. Esta configuración hace que el robot

tenga dos o más restricciones independientes debido a las ruedas estándar fijas que no comparten el mismo eje horizontal de rotación. En general, un robot tendrá cero o más ruedas estándar fijas y cero o más ruedas estándar orientables. Por lo tanto, se puede identificar la serie posible de valores de rango para cualquier robot:  $0 \leq \text{rank}[C_1(\beta_s)] \leq 3$ . Considerando el caso  $\text{rank}[C_1(\beta_s)] = 0$ . Esto sólo es posible si hay cero restricciones cinemáticas independientes en  $C_1(\beta_s)$ . En este caso no hay ruedas estándar ni fijas ni orientables unidas al marco del robot:  $N_f = N_s = 0$ .

Considerando el otro extremo,  $\text{rank}[C_1(\beta_s)] = 3$ . Este es el rango máximo posible puesto que las restricciones cinemáticas se especifican a lo largo de tres grados de libertad (es decir, la matriz de restricción es de tres columnas de ancho). Por lo tanto, no puede haber más de tres restricciones independientes. De hecho, cuando  $\text{rank}[C_1(\beta_s)] = 3$ , entonces, el robot está completamente restringido en todas las direcciones y es, por tanto, degenerado dado que el movimiento en el plano es totalmente imposible.

Ahora se puede definir formalmente el grado de movilidad de un robot  $\delta_m$  como:

$$\delta_m = \text{dim}N[C_1(\beta_s)] = 3 - \text{rank}[C_1(\beta_s)] \quad (8.40)$$

La dimensionalidad del espacio nulo ( $\text{dim}N$ ) de la matriz  $C_1(\beta_s)$  es una medida del número de grados de libertad del cuerpo del robot que puede ser inmediatamente manipulado a través de cambios en la velocidad de la rueda. Es lógico, pues, que  $\delta_m$  debe oscilar entre 0 y 3.

Se considera entonces un chasis diferencial común. En tal robot hay dos ruedas estándar fijas que comparten un eje horizontal común. Como se vio anteriormente, la segunda rueda no añade restricciones cinemáticas independientes para el sistema. Por tanto,  $\text{rank}[C_1(\beta_s)] = 1$  y  $\delta_m = 2$ . En contraste, se considera un chasis de bicicleta. Esta configuración se compone de una rueda estándar fija y una rueda estándar orientable. En este caso, cada rueda contribuye una restricción deslizante independiente para  $C_1(\beta_s)$ . Entonces  $\delta_m = 1$ . Se debe tener en cuenta que la bicicleta tiene el mismo número total de ruedas no omnidireccionales como los chasis de dirección diferencial, y de hecho una de sus ruedas es orientable. Sin embargo, tiene un menor grado de movilidad. Pensándolo bien esto es adecuado. Una bicicleta sólo tiene control sobre su velocidad de avance/retroceso por la manipulación directa de las velocidades de las ruedas. Sólo por la dirección la bicicleta puede cambiar su *ICR*. Como era de esperarse, en base a la ecuación 8.40 cualquier robot que consiste solamente de ruedas omnidireccionales tales como ruedas suecas o esféricas tendrá la máxima movilidad,  $\delta_m = 3$  y tal robot puede manipular directamente todos los tres grados de libertad.

## 8.4.2. El grado de maniobrabilidad

El grado de movilidad se ha definido sobre la cuantificación de los grados de libertad controlables basados en los cambios en la velocidad de la rueda. La dirección también puede tener un impacto eventual sobre la posición del chasis del robot  $\xi$ , aunque el impacto es indirecto, porque después de cambiar el ángulo de una rueda estándar dirigitible, el robot debe moverse para que el cambio de ángulo de dirección tenga un impacto en la posición. Al igual que con la movilidad, la preocupación es por el número de parámetros de dirección controlables de forma independiente para definir el grado de maniobrabilidad  $\delta_s$ :

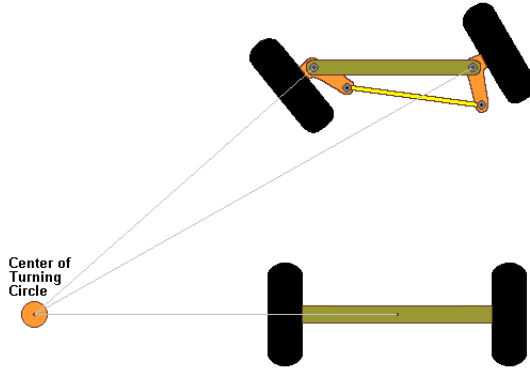
$$\delta_s = \text{rank}[C_{1s}(\beta_s)] \quad (8.41)$$

En el caso de la movilidad, un aumento en el rango de  $C_1(\beta_s)$  implicaba más restricciones cinemáticas y por lo tanto un sistema menos móvil. En el caso de maniobrabilidad, un aumento en el rango de  $C_{1s}(\beta_s)$  implica más grados de libertad de dirección y por lo tanto una mayor maniobrabilidad eventual. Dado que  $C_{1s}(\beta_s)$  incluye a  $C_{1s}(\beta_s)$ , esto significa que una rueda estándar dirigitida puede tanto disminuir la movilidad como incrementar la maniobrabilidad: su orientación particular, en cualquier instante impone una restricción cinemática, pero su capacidad para cambiar esa orientación puede dar lugar a trayectorias adicionales. El rango de  $\delta_s$  se puede especificar como  $0 \leq \delta_s \leq 2$ . El caso de  $\delta_s = 0$  implica que el robot no tiene ruedas estándar orientables,  $N_s = 0$ . El caso de  $\delta_s = 1$  es más común cuando la configuración del robot incluye una o más ruedas estándar orientables. Por ejemplo considerando un automóvil ordinario.

En este caso  $N_f = 2$  y  $N_s = 2$ . Pero las ruedas fijas comparten un eje común así que  $\text{rank}[C_{1f}] = 1$ . Las ruedas fijas y cualquiera de las ruedas orientables limitan el *ICR* a ser un punto a lo largo de la línea que se extiende desde el eje trasero. Por lo tanto, la segunda rueda dirigitible no puede imponer ninguna restricción cinemática independiente y entonces  $\text{rank}[C_{1s}(\beta_s)] = 1$ . En este caso  $\delta_s = \delta_m = 1$ . En el caso  $\delta_s = 2$ , solo es posible en robots con ruedas estándar no fijas:  $N_f = 0$ . Es decir se considera un robot que tiene una rueda estándar orientable centrada y dos ruedas estándar orientables, es decir,  $N_s = 2$ . Bajo estas circunstancias, es posible crear un cuerpo con dos ruedas separadas estándar orientables, como una pseudo bicicleta (de dos direcciones) en la que ambas ruedas traseras son orientables.

Entonces, la orientación de una rueda restringe al *ICR* a una línea, mientras que la segunda rueda puede limitar el *ICR* a cualquier punto a lo largo de esa línea. Curiosamente, esto significa que el  $\delta_s = 2$  implica que el robot puede ubicar su *ICR* en cualquier lugar sobre el plano del suelo.

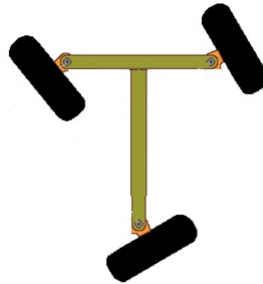
Figura 8.21: Centro Instantáneo de rotación.



$$C_{1s}(\beta_s) = C_{1f} = \begin{bmatrix} \cos(\alpha + \beta) & \sin(\alpha + \beta) & l \sin \beta \end{bmatrix} \quad (8.42)$$

(Fuente: Sistemas de locomoción de robots móviles, Automatización y Robótica Móvil.)

Figura 8.22: Centro Instantáneo de rotación.



$$C_{1s}(\beta_s) = C_{1f} = \begin{bmatrix} \cos(\alpha_1 + \beta_1) & \sin(\alpha_1 + \beta_1) & l \sin(\beta_1) \\ \cos(\alpha_2 + \beta_2) & \sin(\alpha_2 + \beta_2) & l \sin(\beta_2) \end{bmatrix} \quad (8.43)$$

(Fuente: Sistemas de locomoción de robots móviles, Automatización y Robótica Móvil.)

### 8.4.3. Maniobrabilidad del Robot

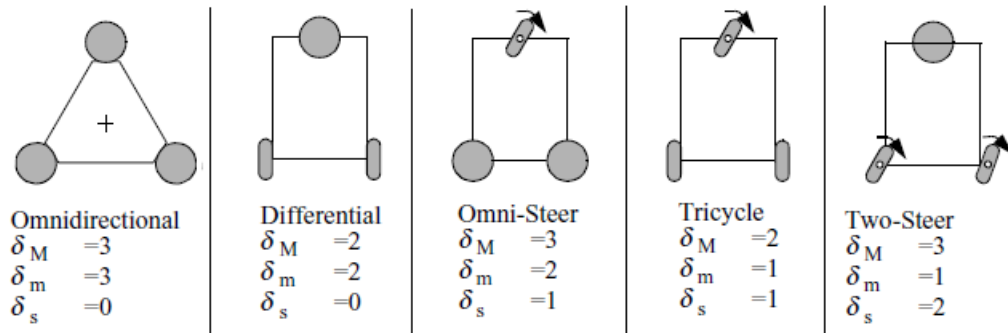
Los grados de libertad totales que un robot puede manipular, conocido como el grado de maniobrabilidad  $\delta_M$ , puede ser fácilmente definido en términos de movilidad

y maniobrabilidad:

$$\delta_M = \delta_m + \delta_s \quad (8.44)$$

Por lo tanto la maniobrabilidad incluye tanto a los grados de libertad que el robot manipula directamente a través de la velocidad de la rueda y los grados de libertad que manipula de forma indirecta, cambiando la configuración de dirección y movimiento. Con base en las investigaciones de las secciones anteriores, se pueden dibujar los tipos básicos de configuración de las ruedas. Se representan en la figura 8.23. Nótese que dos robots con el mismo  $\delta_M$  no son necesariamente equivalentes. Por ejemplo, las geometrías de dirección diferencial y de triciclo (Figura 8.22) tienen una maniobrabilidad igual  $\delta_M = 2$ . En la de dirección diferencial toda la maniobrabilidad es el resultado de la movilidad directa ya que  $\delta_m = 2$  y  $\delta_s = 0$ . En el caso de un vehículo de tres ruedas la maniobrabilidad resulta de la dirección:  $\delta_m = 1$  y  $\delta_s = 1$ . Ninguna de estas configuraciones permite al *ICR* oscilar en el plano. En ambos casos, el *ICR* debe estar en una línea predefinida con respecto al marco de referencia del robot.

Figura 8.23: Los cinco tipos básicos de configuraciones de tres ruedas.

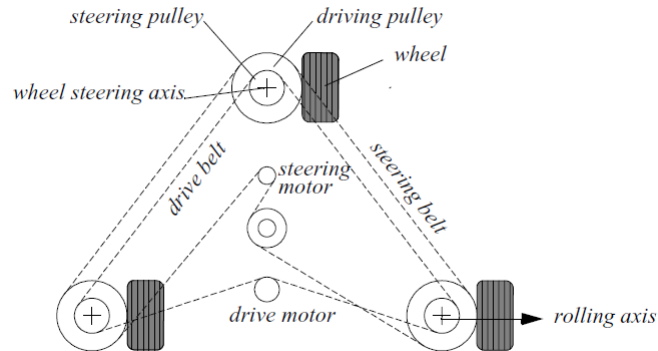


(Fuente: Sistemas de locomoción de robots móviles, Automatización y Robótica Móvil.)

Mas generalmente, para cada robot con  $\delta_M = 2$ , el *ICR* está siempre limitado a situarse en una línea y para cualquier robot con  $\delta_M = 3$ , el *ICR* puede ser colocado en cualquier punto del plano. Una configuración del robot común para la investigación de robótica móvil en interiores es la configuración de conducción sincronizada (Figura 8.24). Tal un robot tiene dos motores y tres ruedas que están bloqueadas juntas. Un motor proporciona la energía para hacer girar las tres ruedas, mientras que el segundo motor proporciona la energía para dirigir las tres ruedas.

En un robot de tres ruedas, la unidad de sincronización es  $N_f = 0$  y  $N_s = 3$ . Por tanto,  $rank[C_{1s}(\beta_s)]$  puede ser usado para determinar  $\delta_m$  y  $\delta_s$ . Las tres ruedas no com-

Figura 8.24: Configuración de conducción sincronizada



(Fuente: Sistemas de locomoción de robots móviles, Automatización y Robótica Móvil.)

parten un eje común, por lo tanto, dos de las tres contribuyen independientemente a las restricciones de deslizamiento. La tercera debe ser dependiente de estas dos restricciones para que el movimiento sea posible. Por tanto  $rank[C_{1s}(\beta_s)] = 2$  y  $\delta_m = 1$ . Esto es intuitivamente correcto. Un robot de diferencial sincronizado con la dirección congelada manipula un solo grado de libertad, que consiste en viajar de ida y vuelta en una línea recta. Sin embargo una complicación interesante ocurre cuando se considera  $\delta_s$ . Basándose en la ecuación 8.41 el robot debería tener  $\delta_s = 2$ .

En efecto, para un robot con dirección de tres ruedas con la configuración geométrica de un robot diferencial sincronizado esto sería correcto. Sin embargo, se tiene información adicional: en una configuración diferencial sincronizada solo un motor dirige las tres ruedas con una transmisión por correa. Por lo tanto, aunque idealmente, si las ruedas fuesen orientables de forma independiente, entonces el sistema podría lograr  $\delta_s = 2$ , en el caso de la conducción sincronizada el sistema de conducción, limita aún más la cinemática de tal manera que en realidad  $\delta_s = 1$ . Finalmente, se puede calcular la maniobrabilidad sobre la base de estos valores:  $\delta_M = 2$  para un robot de conducción sincronizada.

## 8.5. ESPACIO DE TRABAJO DEL ROBOT MÓVIL

Para un robot, la maniobrabilidad es equivalente a sus grados de libertad de control. Pero el robot está situado en algún entorno, y la siguiente pregunta es situar el análisis

en el medio donde se desenvuelve. Nos preocupamos por la forma en que el robot puede utilizar sus grados de control de libertad para posicionarse en el entorno. Por ejemplo, considere un vehículo Ackerman, o un automóvil. El número total de grados de libertad de control para ese vehículo es  $\delta_M = 2$ , uno para la dirección y el segundo para el accionamiento de las ruedas motrices. Pero, ¿qué son los grados de libertad totales del vehículo en su entorno? De hecho, son tres: el vehículo puede situarse en el plano en cualquier punto  $x, y$  y con cualquier ángulo  $\theta$ .

Así identificar el espacio de un robot de posibles configuraciones es importante porque sorprendentemente puede exceder  $\delta_M$ . Además del espacio de trabajo, nos preocupamos por cómo el robot es capaz de moverse entre las diferentes configuraciones, además, nos alejamos de detalles cinemáticos más profundos, como las ruedas y en cambio nos centramos en la posición del chasis del robot y los grados de libertad del mismo. De esta manera, vamos a colocar el robot en el contexto de su espacio de trabajo.

### 8.5.1. Grados de libertad

Al definir el espacio de trabajo de un robot, es útil examinar en primer lugar su velocidad espacial admisible. Dadas las limitaciones cinemáticas del robot, su velocidad espacial describe las componentes independientes de movimiento del robot que el robot puede controlar. Por ejemplo, la velocidad espacial de un monociclo puede ser representada con dos ejes, uno que representa la velocidad instantánea hacia adelante del monociclo y el segundo representa el cambio instantáneo en la orientación  $\dot{\theta}$  del mismo. El número de dimensiones en la velocidad espacial de un robot es el número de velocidades alcanzables de forma independiente. Esto también conoce como los grados diferenciables de libertad, en inglés Differentiable Degrees Of Freedom (*DDOF*). Un *DDOF* del robot es siempre igual a su grado de movilidad  $\delta_m$ . Por ejemplo, una bicicleta tiene el siguiente grado de maniobrabilidad:  $\delta_M = \delta_m + \delta_s = 1 + 1 = 2$ . El *DDOF* de una bicicleta es en realidad 1.

En contraste con una bicicleta, consideremos un Omnibot, un robot con tres ruedas Suecas. Sabemos que en este caso hay cero ruedas estándar y por lo tanto  $\delta_M = \delta_m + \delta_s = 3 + 0 = 3$ . Así, el Omnibot tiene tres grados de libertad diferenciales. Esto es apropiado, dado que debido a que un robot no tiene limitaciones de movimiento cinemáticas, es capaz de ajustar de forma independiente todas las tres variables de posición:  $(\dot{x}, \dot{y}, \dot{\theta})$ . Dada la diferencia de *DDOF* entre una bicicleta y un Omnibot, se tiene en cuenta los grados generales de libertad en el espacio de trabajo de cada configuración. El Omnibot puede alcanzar cualquier posición  $(x, y, \theta)$  en su entorno y lo pueden hacer directamente logrando las posiciones objetivo de los tres ejes al mismo tiempo ya que *DDOF* = 3.

Claramente este tiene un espacio de trabajo con  $DOF = 3$ .

Puede una bicicleta lograr cualquier posición  $(x, y, \theta)$  en su entorno? Puede hacerlo, pero alcanzar algunos puntos de destino puede requerir más tiempo y energía que un Omnibot equivalente. Por ejemplo, si una configuración de bicicleta debe moverse lateralmente de  $1m$ , la maniobra más simple de éxito implicaría o bien una espiral o un movimiento hacia adelante y hacia atrás similar al parqueo en paralelo de los automóviles. No obstante, una bicicleta puede lograr cualquier  $(x, y, \theta)$  y por lo tanto el espacio de trabajo de una bicicleta tiene  $DOF = 3$  también. Es evidente que existe una relación desigualdad en el trabajo:

$$DDOF \leq \delta_M \leq DOF$$

Aunque la dimensionalidad de espacio de trabajo de un robot es un atributo importante, es evidente a partir del ejemplo anterior que las trayectorias particulares disponibles para un robot también importan. Así como los  $DOF$  del espacio de trabajo gobiernan la capacidad del robot para alcanzar varias posiciones, así los  $DDOF$  del robot gobiernan su capacidad para lograr diferentes trayectorias.

### 8.5.2. Robots Holonómicos

En robótica móvil, el término se refiere específicamente a las restricciones cinemáticas del chasis del robot. Un robot holonómico es un robot que tiene cero restricciones cinemáticas no holonómicas. A la inversa, un robot no holonómico es un robot con una o más restricciones cinemáticas no holonómicas. Una restricción cinemática holonómica se puede expresar como una función explícita de las variables de posición solamente. Por ejemplo, en el caso de un robot móvil con una sola rueda estándar fija, una restricción cinemática holonómica sería expresable utilizando  $\alpha_1, \theta_1, l_1, r_1, \varphi_1, x, y, \theta$  solamente. Tal restricción puede no usar los derivados de estos valores, como  $\dot{\varphi}$  o  $\dot{\xi}$ . Una restricción cinemática no holonómica requiere una relación diferencial, tal como la derivada de una variable de posición. Teniendo en cuenta la restricción de deslizamiento de la rueda estándar fija:

$$\begin{bmatrix} \cos(\alpha + \beta) & \sin(\alpha + \beta) & l \sin \beta \end{bmatrix} R(\theta) \dot{\xi}_I = 0 \quad (8.45)$$

Esta restricción debe utilizar movimiento del robot  $\dot{\xi}$  en lugar de representar  $\xi$  porque el punto es restringir el movimiento del robot perpendicular al plano de la rueda a cero. La restricción no es integrable, dependiendo de forma explícita del movimiento del robot. Por lo tanto, la restricción de deslizamiento es una restricción no holonómica. Considere una configuración de bicicleta, con una rueda estándar fija y otra rueda

estándar orientable. Debido a que la restricción de deslizamiento de la rueda fija estará en vigor por este robot, se puede concluir que la bicicleta es un robot no holonómico.

Pero supóngase ahora que se bloquea el sistema de dirección de la bicicleta, de modo que se convierte en dos ruedas fijas estándar con ejes separados pero paralelos. Para simplificar fórmulas, se supone que esta línea infinita está alineada con  $X_1$  en el marco de referencia global, y que  $(\beta_{1,2} = \pi/2, \alpha_1 = 0, \alpha_2 = \pi)$ . En este caso, las restricciones de deslizamiento de las dos ruedas pueden ser remplazadas con un conjunto igualmente completo de restricciones sobre la posición del robot ( $Y = 0, \theta = 0$ ). Esto elimina dos restricciones no holonómicas, correspondientes a las restricciones de deslizamiento de las dos ruedas. Las únicas restricciones cinemáticas no holonómicas que quedan son las restricciones de rodadura para cada rueda:

$$\begin{bmatrix} -\sin(\alpha + \beta) & \cos(\alpha + \beta) & l \cos \beta \end{bmatrix} R(\theta) \dot{\xi}_I + r \dot{\varphi} = 0 \quad (8.46)$$

Esta restricción es necesaria para cada rueda para relacionar la velocidad de giro de la rueda a la velocidad de movimiento proyectada a lo largo del plano de la rueda. Pero en el caso de la bicicleta bloqueada, dada la posición inicial de rotación de una rueda en el origen,  $\varphi_0$ , se puede sustituir esta restricción con una que se relacione directamente con la posición en la línea,  $x$ , con ángulo de rotación de la rueda,  $\varphi : \varphi = (x/r) + \varphi_0$ . La bicicleta bloqueada es un ejemplo del primer tipo de robot holonómico - donde las restricciones no existen, pero son todas las restricciones cinemáticas holonómicas. Este es el caso para todos los robots holonómicos con  $\delta_M < 3$ . El segundo tipo de robot holonómico existe cuando no hay restricciones cinemáticas, lo cual es,  $N_f = 0$  y  $N_s = 0$ . Dado que no existen restricciones cinemáticas, tampoco hay restricciones cinemáticas no holonómicas y así tal robot siempre es holonómico. Este es el caso para todos los robots holonómicos con  $\delta_M = 3$ .

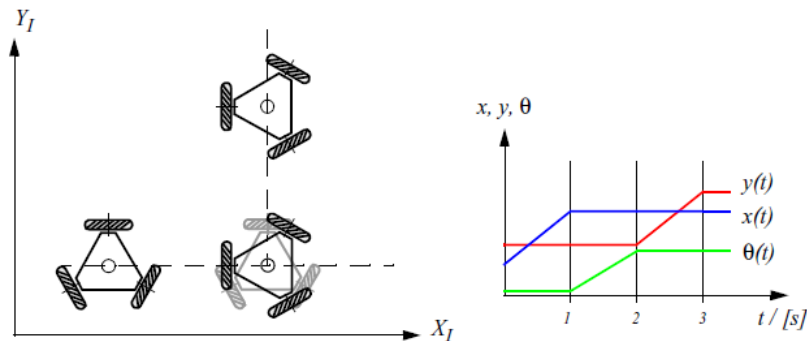
En robótica móvil, los chasis útiles en general, deben lograr posicionarse en un espacio de trabajo con dimensionalidad 3, entonces en general se requiere  $DOF = 3$  para el chasis. Sin embargo, las “habilidades” holonómicas para maniobrar alrededor de obstáculos, sin afectar la orientación y hacer un seguimiento a un objetivo mientras se sigue una ruta arbitraria son consideraciones adicionales importantes. Por estas razones, la forma particular de holonomía más relevante para la robótica móvil es esa de  $DOF = DDOF = 3$  Se define este tipo de configuraciones del robot, como omnidireccional: un robot omnidireccional es un robot holonómico con  $DDOF = 3$ .

### 8.5.3. Ruta de acceso y consideraciones de trayectoria

En robótica móvil, hay que preocuparse no sólo por la capacidad del robot para llegar a las configuraciones necesarias finales, sino también acerca de cómo se llega allí. Se considera el problema de la capacidad del robot para seguir un camino: en el mejor de los casos, un robot debe ser capaz de rastrear cualquier trayectoria de acceso a través de su área de trabajo de posiciones. Considere un vehículo omnidireccional, conduciendo a gran velocidad en una curva con un diámetro constante. Durante tal movimiento del vehículo estará expuesto a una fuerza centrípeta no despreciable. Esta fuerza lateral empujando el vehículo fuera de la curva tiene que ser contrarrestada por el par del motor de las ruedas omnidireccionales.

Pero tomando un ejemplo de alta maniobrabilidad con ruedas estándar: una bicicleta en la que las dos ruedas son orientables. Este vehículo alcanza un grado de controlabilidad de 2, lo que resulta en un alto grado de maniobrabilidad:  $\delta_M = \delta_m + \delta_s = 1 + 2 = 3$ . Curiosamente, esta configuración no es holonómica, sin embargo, tiene un alto grado de maniobrabilidad en un espacio de trabajo con  $DOF = 3$ .

Figura 8.25: Ejemplo de la trayectoria de un robot omnidireccional.



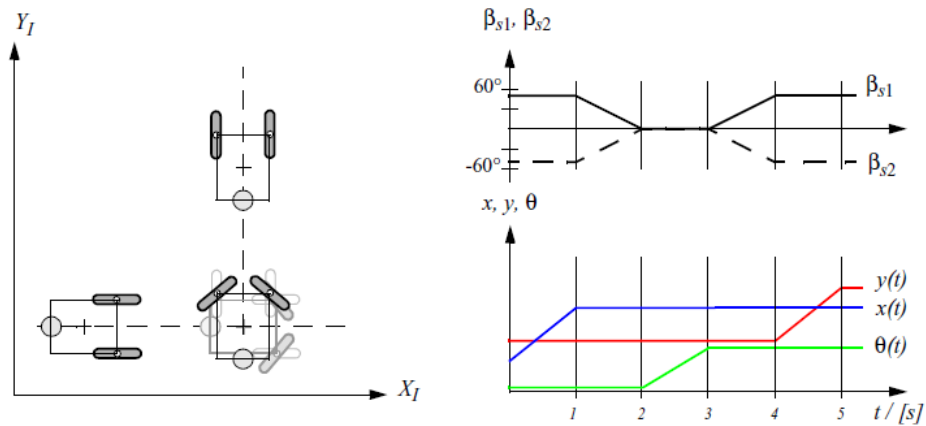
(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition.)

La maniobrabilidad resulta  $\delta_M = 3$  lo cual significa que las bicicletas de doble dirección pueden seleccionar cualquier ICR orientando apropiadamente sus dos ruedas. Entonces, ¿cómo se compara esto con un robot omnidireccional? La capacidad de manipular la ICR en el plano significa que la bicicleta de doble dirección puede seguir cualquier trayectoria en su área de trabajo. Mas generalmente cualquier robot con  $\delta_M = 3$  puede seguir cualquier camino en su espacio de trabajo desde su posición inicial hasta su posición final. Un robot omnidireccional también puede seguir cualquier ruta en su espacio de trabajo y, como es lógico, ya que  $\delta_m = 3$  en un robot omnidireccional,

entonces se debe cumplir que  $\delta_M = 3$ .

Se considera ahora una trayectoria objetivo en la cual el robot se mueve a lo largo del eje  $X_I$  a una velocidad constante de  $1m/s$  durante 1 segundo, luego la orientación cambia en sentido contrario a las manecillas del reloj 90 grados también en 1 segundo, luego se mueve paralelo al eje  $Y_I$  por un segundo final. La trayectoria deseada de 3 segundos se muestra en la figura 8.25, utilizando gráficos de  $x, y$  y  $\theta$  con relación al tiempo.

Figura 8.26: Ejemplo de la trayectoria del robot similar a la figura 8.25, con dos ruedas motrices.



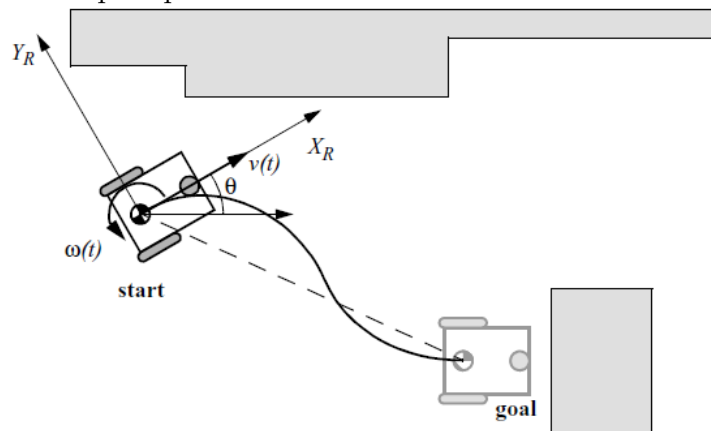
(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition.)

Se supone que el robot puede alcanzar una cierta velocidad finita y arbitraria en cada rueda. Por simplicidad, suponemos además que la aceleración es infinita, es decir, que toma un tiempo cero para llegar a cualquier velocidad deseada. Bajo estos supuestos, el robot omnidireccional de hecho puede seguir la trayectoria de la figura 8.25. La transición entre el movimiento de 2 segundos y 1 segundo, por ejemplo, implica solo cambios en las velocidades de las ruedas. En resumen, las bicicletas de doble dirección requieren cambios en los grados de libertad internos y debido a que estos cambios llevan tiempo, las trayectorias arbitrarias no son alcanzables. La figura 8.26 representa la trayectoria más parecida que una bicicleta de doble dirección puede lograr. En contraste con las tres fases de movimiento deseadas, esta trayectoria tiene cinco fases.

## 8.6. MÁS ALLÁ DE LA CINEMÁTICA BÁSICA

Muchos robots móviles, tales como tipo tanque y sistemas de deslizamiento/derrape de cuatro ruedas violan los modelos cinemáticos mencionados anteriormente. Al analizar esos sistemas, a menudo es necesario modelar explícitamente la dinámica de la fricción viscosa entre el robot y el plano del suelo. Más importante aún, el análisis cinemático de un sistema robótico móvil ofrece resultados sobre el espacio de trabajo teórico de ese robot móvil. Sin embargo, para moverse con eficacia en esta área de trabajo de un robot móvil debe contar con el movimiento adecuado de sus grados de libertad. Este problema, se conoce como motorización, requiere un mayor análisis de las fuerzas que deben estar activamente suministradas para generar el rango cinemático de movimiento disponible para el robot.

Figura 8.27: Caso típico para el control de retroalimentación de un robot móvil.



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition.)

## 8.7. CONTROL DE MOVIMIENTO

### 8.7.1. Introducción

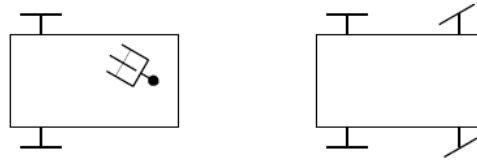
El control de robots móviles con ruedas ha sido, y sigue siendo, objeto de numerosos estudios de investigación. En particular, las restricciones no holonómicas asociadas con estos sistemas han motivado el desarrollo de técnicas de control no lineales. Estos enfoques se tratan en esta sección, pero su exposición se limita deliberadamente a fin

de dar la prioridad a las técnicas más clásicas, cuyas bases, tanto práctica como teórica, están mejor establecidas. Recordemos (ver figura 8.28) que:

- Un robot móvil tipo monociclo está compuesto esquemáticamente de dos ruedas independientes accionadas en un eje común, cuya dirección está rígidamente relacionada con el chasis del robot, y una o varias ruedas orientables o caster, que no se controlan y sirven para fines de sustentación.
- Un robot móvil tipo carro de tracción trasera se compone de un eje de ruedas motriz, en la parte trasera del chasis y una (o un par de) ruedas orientables de dirección delantera(s).

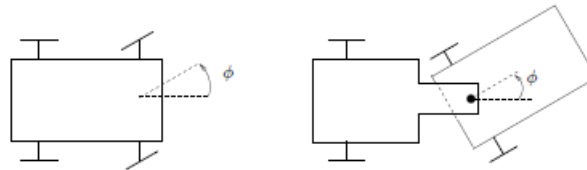
Nótese también, como se ilustra en el siguiente diagrama, que un robot móvil tipo carro se puede ver (al menos cinemáticamente) como un robot monociclo tipo móvil al que está unido un remolque.

Figura 8.28: Robots móviles Monociclo y Tipo carro.



(Fuente: Motion control of wheeled mobile robots, Pascal Morin, Claude Samson, 2007.)

Figura 8.29: Analogía del carro / monociclo con remolque.



(Fuente: Motion control of wheeled mobile robots, Pascal Morin, Claude Samson, 2007.)

**Se analizan tres problemas genéricos en esta sección: Seguimiento de ruta:** Dada una curva  $C$  en el plano, una velocidad longitudinal (no cero)  $v_0$  para el chasis

del robot, y un punto  $P$  sujeto al chasis, el objetivo es que el punto  $P$  siga la curva  $C$ , cuando el robot se mueve con la velocidad  $v_0$ . La variable que se tiene que estabilizar a cero es, pues, la distancia entre el punto  $P$  y la curva (es decir, la distancia entre  $P$  y el punto más cercano  $M$  en  $C$ ). Este tipo de problema normalmente corresponde a la conducción en una carretera al intentar mantener la distancia entre el chasis del vehículo y el lado de la carretera constante. Seguimiento automático de muro es otra posible aplicación.

**La estabilización de las trayectorias:** Este problema difiere del anterior en que la velocidad longitudinal del vehículo ya no se predetermina porque tiene también por objeto un seguimiento de la distancia a lo largo de la curva  $C$ . Este objetivo supone que la curva  $C$  geométrica se complementa con un tiempo-horario, es decir, que está parametrizada con la variable de tiempo  $t$ .

Esto se acerca a la definición de una trayectoria  $t \rightarrow ((x_r(t), y_r(t)))$  con respecto a un marco de referencia  $F_0$ . Entonces, el objetivo es estabilizar el vector error de posición  $((x(t) - x_r(t), y(t) - y_r(t)))$  a cero, con  $(x(t), y(t))$  que denota las coordenadas del punto  $P$  dentro de  $F_0$  en el tiempo  $t$ . El problema también se puede formular como el de un control del vehículo con el fin de realizar un seguimiento de un vehículo de referencia cuya trayectoria está dada por  $t \rightarrow ((x_r(t), y_r(t)))$ .

**Estabilización de posiciones fijas:**  $F_1$  denota un marco fijo al chasis del robot. En esta sección, llamamos una posición del robot (o situación) a la asociación de la posición de un punto  $P$  situado sobre el chasis del robot con la orientación  $\theta(t)$  de  $F_1$  con respecto a un marco fijo  $F_O$  en el plano de movimiento. Para este último problema, el objetivo es estabilizar a cero el vector posición  $\xi(t) = (x(t), y(t), \theta(t))$ , con  $(x(t); y(t))$  que denota la posición de  $P$  expresada en  $F_O$ . Aunque una posición fija deseada (o de referencia) es obviamente un caso particular de una trayectoria factible, este problema no puede resolverse por los métodos de control clásicos.

## 8.7.2. Los modelos de control

### Cinemática contra dinámica.

Existe un modelo dinámico para la configuración general WMRs. Su particularización al caso de los robots tipo monociclo y coche como móviles da:

$$H(q)\dot{u} + F(q, u)u = \Gamma'(\phi)\tau \quad (8.47)$$

Con  $q$  denota vector de configuración de un robot,  $u$  el vector de variables de velocidad independiente asociadas con los grados de libertad del robot,  $H(q)$  una matriz de

inercia reducida (lo cual es invertible para cualquier  $q$ ),  $F(q, u)u$ , un vector de fuerzas combinando la contribución de Coriolis y las fuerzas de contacto rueda-tierra,  $\phi$  el ángulo de orientación de volante del vehículo,  $\Gamma$  una matriz de control invertible (que es constante en el caso de un vehículo tipo monociclo), y  $\tau$  un vector de independientes pares de motor (cuya dimensión es igual al número de grados de libertad en el caso de accionamiento completo, es decir, igual a dos de los vehículos aquí considerados). En el caso de un vehículo tipo monociclo,  $\xi$  un vector de configuración se compone de los componentes del vector de postura del chasis y los ángulos de orientación de las ruedas giratorias (con respecto al chasis). En el caso de un coche-como vehículo, un vector de configuración se compone de los componentes de  $\xi$  y el ángulo de un volante  $\phi$ . Para ser completo, este modelo dinámico debe ser complementado con las ecuaciones de la cinemática en la forma:

$$\dot{q} = S(q)u \quad (8.48)$$

$$\dot{z} = B(z)u \quad (8.49)$$

Con  $z = \xi$ , en el caso de un vehículo de tipo monociclo, y  $z = (z = (\xi, \phi); a)$  en el caso de un vehículo automóvil similar.

En el control automático de la terminología, el modelo dinámico completo 8.47 y 8.48 forma un sistema de control, que se puede escribir como  $X = f(X, \tau)$  con  $X = (q, u)$  denota el vector de estado de este sistema, y  $\tau$  el vector de entradas de control. Los modelos cinemáticos 8.48 y 8.49 también están los sistemas de control con los respectivos vectores de estado de  $q$  y  $z$ , y el control de vectores  $u$ . Cualquiera de estos modelos pueden ser utilizados para el diseño de control y con fines de análisis. En el resto de este capítulo, hemos optado por trabajar con el modelo cinemático 8.49. Por analogía con el control de movimiento de los brazos manipuladores, esto viene a la utilización de un modelo con entradas de control de velocidad, en lugar de un modelo con control de par entradas. Las principales razones de esta elección son los siguientes.

- El modelo cinemático es más simple que el dinámico. En particular, que no implica un cierto número de funciones con valores de matriz cuya precisa determinación se basa en el conocimiento de numerosos parámetros asociados con el vehículo y sus actuadores (repartición geométrica de los órganos constitutivos, masas y momentos de inercia de masa, coeficientes de reducción en la transmisión de pares producidos por los motores, etc.
- En el caso de los robots accionados con motores eléctricos, estos motores son frecuentemente suministrados con “bajo nivel” de bucles de control de velocidad

que tengan una velocidad angular deseada como entrada y estabilizar la velocidad angular del motor en este valor.

- Si el servo-bucles evocado más arriba, cuya función es separar la cinemática de la dinámica del vehículo, no está presente, uno puede diseñar e incluso mejorar su rendimiento mediante el uso de la información que se tiene de los términos que intervienen en la ecuación dinámica 8.47.

$$\tau = \Gamma(\theta)^{-1}[H(q)\omega + F(q, u)u]$$

Esto produce el sencillo sistema de control lineal desacoplado  $\dot{u} = w$  con la variable  $w$ , homogéneo a un vector de aceleraciones, interpretando el papel de un vector de control de entrada nueva. Esta última ecuación indica que el problema de controlar el vehículo con pares de motor puede ser llevado de nuevo a un problema con las entradas de control de aceleración. Generalmente no es difícil deducir una solución de control a este problema de una solución de control de velocidad diseñado mediante el uso de un modelo cinemático. Por ejemplo,

$$\omega = -k(u - u^*(k, t)) + \frac{\delta u^*}{\delta z}(z, t)B(z)u + \frac{\delta u^*}{\delta z}$$

Con  $k > 0$  es una solución si  $u^*$  es una solución cinemática y diferenciable

$$u = u^*(z, t) + (u(0) - u^*(z_0, 0))e^{(-kt)}$$

Es también una solución.

Para el robot móvil tipo monociclo, el modelo cinemático 8.48 usado a partir de ahora es:

$$\begin{aligned} \dot{x} &= u_1 \cos \theta \\ \dot{y} &= u_1 \sin \theta \\ \dot{\theta} &= u_2 \end{aligned} \tag{8.50}$$

Donde  $(x, y)$  representa las coordenadas del punto  $P_m$  situado a una distancia media de las ruedas accionadas, y el ángulo  $\theta$  caracteriza la orientación del chasis del robot (véase la figura 8.30). En esta ecuación,  $u_1$  representa la intensidad de la velocidad longitudinal del vehículo, y  $u_2$  es la velocidad instantánea de rotación del chasis. Las variables  $u_1$  y  $u_2$  están ellas mismas relacionadas con la velocidad angular de las ruedas accionadas a través de las relaciones uno-a-uno

$$u_1 = \frac{r}{2}(\dot{\Psi}_r - \dot{\Psi}_l)$$

$$u_1 = \frac{r}{2R}(\dot{\Psi}_r - \dot{\Psi}_l)$$

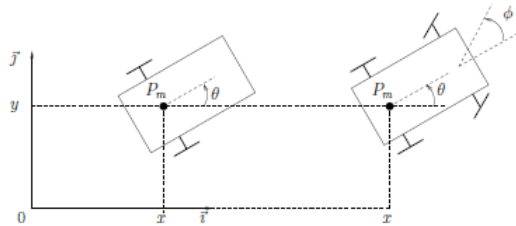
Con un radio  $r$  de las ruedas,  $R$  la distancia entre las dos ruedas accionadas, y  $\dot{\Psi}_r$  (resp.  $\dot{\Psi}_l$ ) la velocidad angular de la derecha (resp. izquierda) de la rueda trasera. Para el coche-robot móvil, el modelo cinemático 8.49 usado a partir de ahora es

$$\begin{aligned} \dot{x} &= u_1 \cos \theta \\ \dot{y} &= u_1 \sin \theta \end{aligned} \tag{8.51}$$

$$\dot{\theta} = \frac{u_1}{L} \tan \phi$$

Donde  $\phi$  representa el ángulo del volante del vehículo, y  $L$  es la distancia entre la parte posterior y los ejes de las ruedas delanteras.

Figura 8.30: Las variables de configuración.



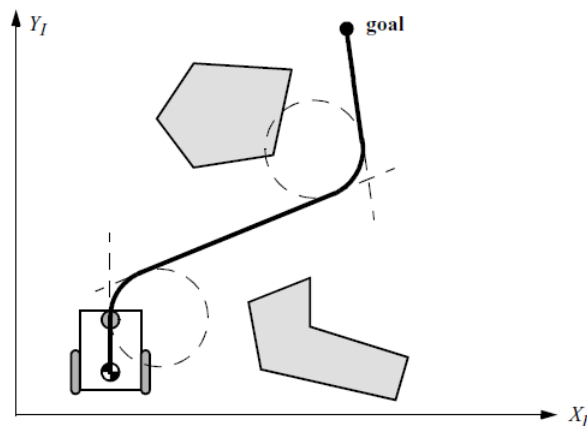
(Fuente: Motion control of wheeled mobile robots, Pascal Morin, Claude Samson, 2007.)

### 8.7.3. Control de lazo abierto (Siguiendo una trayectoria)

El objetivo de un controlador cinemático es seguir una trayectoria descrita por su perfil de posición o velocidad como una función del tiempo. El problema de control es, pues, pre calcular una trayectoria suave basada en segmentos de línea y de círculo que llevan al robot desde la posición inicial a la final (figura 8.31). Este enfoque puede ser considerado como el control de movimiento de lazo abierto, porque la posición del robot medida no es realimentada para el control de posición o velocidad. Hay varias desventajas:

1. No es en absoluto una tarea fácil pre calcular una trayectoria viable si todas las limitaciones y restricciones de las velocidades y aceleraciones del robot tienen que ser consideradas.
2. El robot no se adaptará automáticamente o corregirá la trayectoria si se hacen cambios dinámicos en el entorno.
3. Las trayectorias resultantes no son generalmente suaves, ya que las transiciones de un segmento de trayectoria una a otra son, para la mayoría de los segmentos de uso común (por ejemplo, líneas y parte de los círculos), no fluidos. Esto significa que hay una discontinuidad en la aceleración del robot.

Figura 8.31: Control de lazo abierto de un robot móvil basado en líneas rectas y segmentos de trayectoria circulares.



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition.)

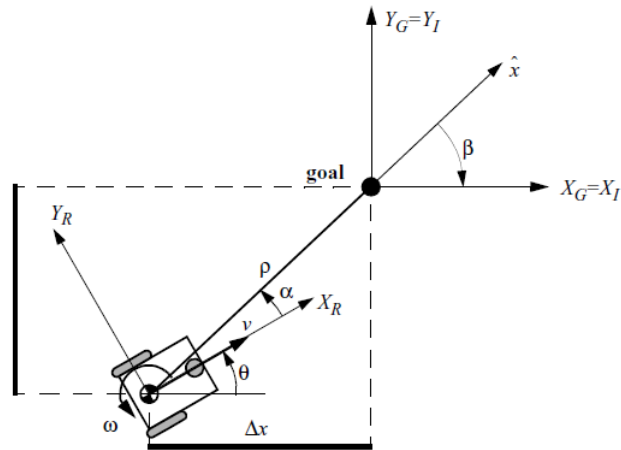
#### 8.7.4. Control de retroalimentación

Un enfoque más apropiado en el control de movimiento de un robot móvil es utilizar un controlador de realimentación de estado real. Con ese controlador la planificación de ruta del robot se reduce a fijar posiciones intermedias (sub-objetivos) situados en la ruta solicitada. Una solución útil para estabilizar un control de realimentación de un robot móvil diferencial se explica a continuación.

### Enunciado del problema.

Se considera la situación mostrada en la figura 8.32, con una posición y orientación arbitraria del robot y una posición y orientación objetivo predefinidas. El verdadero vector error de posición dado en el marco de referencia del robot  $(X_R, Y_R, \theta)$  es  $e = R(x, y, \theta)^T$  con  $x, y$ , y  $\theta$  siendo las coordenadas objetivo del robot.

Figura 8.32: Cinemática de un robot y sus marcos de intereses.



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition.)

La tarea del diseño del controlador es encontrar una matriz de control  $K$ , si es que existe

$$K = \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{123} \end{bmatrix} \quad (8.52)$$

Con  $k_{ij} = k(t \cdot e)$

De tal manera que el control de  $v(t)$  y  $\omega(t)$ :

$$K = \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} = k \cdot e = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad (8.53)$$

Conduce el error  $e$  a cero

$$\lim_{x \rightarrow \infty} e(t) = 0 \quad (8.54)$$

### Modelo cinemático.

Se asume como siempre, que el objetivo está en el origen del sistema de referencia inercial (figura 8.32). En lo siguiente, la posición del vector  $[x, y, \theta]^T$  esta siempre representada en el sistema de referencia inercial. La cinemática de un robot móvil diferencial descrita en el sistema de referencia inercial  $\{X_I, Y_I, \theta\}$  esta dada por:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (8.55)$$

Donde  $\dot{x}$ ,  $\dot{y}$  son las velocidades lineales en la dirección de  $X_I$  y  $Y_I$  del marco de referencia inercial.  $\alpha$  denota el angulo entre el eje  $X_R$  del marco de referencia del robot y el vector  $\hat{x}$  que conecta el centro del eje de las ruedas con la posición final. Si  $\alpha \in I_1$ , donde

$$I_1 = \left( \frac{-\pi}{2}, \frac{\pi}{2} \right] \quad (8.56)$$

Luego considere la transformación de coordenadas en coordenadas polares con su origen en la posición objetivo.

$$\rho = \sqrt{\Delta x^2 + \Delta y^2} \quad (8.57)$$

$$\alpha = -\theta + \text{atan2}(\Delta y, \Delta x) \quad (8.58)$$

$$\beta = -\theta - \alpha \quad (8.59)$$

Esto produce una descripción del sistema, en las nuevas coordenadas polares, usando una ecuación de matriz

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -\cos \alpha & 0 \\ \frac{\sin \alpha}{\rho} & -1 \\ -\frac{\sin \alpha}{\rho} & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (8.60)$$

Donde  $\rho$  es la distancia entre el centro del eje de la rueda del robot y la posición meta,  $\theta$  denota el ángulo entre el eje  $X_R$  del marco de referencia del robot y el eje  $X_I$  asociado con la posición final,  $v$  y  $w$  son la velocidad tangencial y angular respectivamente. Por otra parte, si  $\alpha \in I_2$ , donde

$$I_2 = (-\pi, -\pi/2] \cup (\pi/2, \pi] \quad (8.61)$$

Redefiniendo la dirección de avance del robot mediante la configuración  $v = -v$ , se obtiene un sistema descrito por una matriz ecuación así

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} \cos \alpha & 0 \\ -\frac{\sin \alpha}{\rho} & 1 \\ \frac{\sin \alpha}{\rho} & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (8.62)$$

**Observaciones sobre el modelo cinemático en coordenadas polares (ecuaciones 8.60 y 8.62).**

- La transformación de coordenadas no está definida en  $x = y = 0$ , ya que en tal momento, el determinante de la matriz jacobiana de la transformación no está definido, no tiene límites.
- Para  $\alpha \in I_1$  la dirección de avance del robot apunta hacia el objetivo, para  $\alpha \in I_2$ , es la situación inversa.
- Definiendo adecuadamente la dirección de avance del robot en su configuración inicial, siempre es posible tener  $\alpha \in I_1$  en  $t = 0$ . Sin embargo esto no significa que  $\alpha$  permanezca en  $I_1$  por todo el tiempo  $t$ . Por lo tanto, para evitar que el robot cambie de dirección mientras se acerca al objetivo, es necesario determinar, si es posible, el controlador de tal manera que  $\alpha \in I_1$ , para todo  $t$  siempre y cuando  $\alpha(0) \in I_1$ . Lo mismo se aplica para la dirección inversa.

**La ley de control.**

Las señales de control  $v$  y  $w$  ahora deben ser diseñadas para conducir el robot desde su configuración real, por ejemplo  $(\rho_0, \alpha_0, \beta_0)$  a la posición objetivo. Es obvio que la ecuación 8.60 presente una discontinuidad en  $\rho = 0$ ; por lo tanto el teorema de Brockett no obstruye la estabilidad suave. Si consideramos ahora la ley de control lineal

$$v = K_\rho \cdot \rho \quad (8.63)$$

$$\omega = K_\alpha \alpha + K_\beta \beta \quad (8.64)$$

Se obtiene con la ecuación 8.60 un sistema de lazo cerrado descrito por

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} K_\rho \rho \cos \alpha \\ K_\rho \sin \alpha - K_\alpha \alpha - K_\beta \beta \\ -K_\rho \sin \alpha \end{bmatrix} \quad (8.65)$$

El sistema no tiene ninguna singularidad en  $\rho = 0$  y tiene un único punto de equilibrio en  $(\rho, \alpha, \beta) = (0, 0, 0)$ . Por lo tanto, conducirá el robot a este punto, que es la posición objetivo.

- En el sistema de coordenadas cartesianas, la ley de control (ecuación 8.64) conduce a ecuaciones que no están definidas en  $x = y = 0$ .
- Se debe ser consciente del hecho de que los ángulos  $\alpha, \beta$  tienen que ser expresados siempre en el rango  $(-\pi, \pi)$ .
- Observe que la señal de control  $v$  tiene siempre una señal constante, es decir, es positivo siempre y cuando  $\alpha(0) \in I_1$  y es siempre negativo de lo contrario. Esto implica que el robot lleva a cabo su maniobra de estacionamiento siempre en una sola dirección y sin invertir su movimiento.

En la figura 8.32a se encuentran los caminos que resultan cuando el robot se encuentra inicialmente en un círculo en el plano  $xy$ . Todos los movimientos tienen trayectorias suaves hacia el objetivo en el centro. Los parámetros de control para esta simulación se establece en

$$K = (K_\rho, K_\alpha, K_\beta) = (3, 8, -1, 5) \quad (8.66)$$

### Problema de estabilidad local.

Esto puede demostrar, que el sistema de control de lazo cerrado (ecuación 8.65) es local y exponencialmente estable si

$$K_\rho > 0; K_\beta < 0; K_\alpha - K_\rho > 0 \quad (8.67)$$

Prueba:

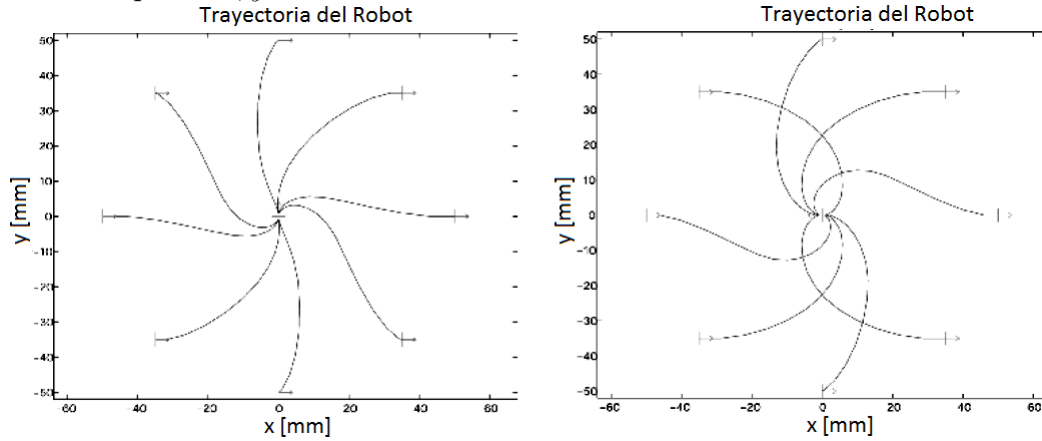
Linealizada alrededor de la posición de equilibrio ( $\cos x = 1, \sin x = x$ ), la ecuación 8.65 puede ser escrita como:

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -K_\rho & 0 & 0 \\ 0 & -(K_\alpha - K_\rho) & -K_\beta \\ 0 & -K_\rho & 0 \end{bmatrix} \begin{bmatrix} \rho \\ \alpha \\ \beta \end{bmatrix} \quad (8.68)$$

Por lo que es local y exponencialmente estable si los valores propios de la matriz:

$$A = \begin{bmatrix} -K_\rho & 0 & 0 \\ 0 & -(K_\alpha - K_\rho) & -K_\beta \\ 0 & -K_\rho & 0 \end{bmatrix} \quad (8.69)$$

Figura 8.33: Caminos resultantes cuando el robot se encuentra inicialmente en el círculo unitario en el plano  $x, y$ .



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition.)

Todos tienen una parte real negativa. El polinomio característico de la matriz  $A$  es

$$(\lambda + K_\rho)(\lambda^2 + \lambda(K_\alpha - K_\rho) - K_\rho K_\beta) \quad (8.70)$$

Y todas las raíces tienen parte real negativa si

$$K_\rho > 0; -K_\beta > 0; K_\alpha - K_\rho > 0 \quad (8.71)$$

Lo que demuestra la solicitud.

Para el control de posición robusto, podría ser recomendable aplicar la condición de estabilidad fuerte, lo que garantiza que el robot no cambie de dirección durante su aproximación al objetivo:

$$K_\rho > 0; K_\beta < 0; K_\alpha + \frac{5}{3}K_\beta - \frac{2}{\pi}K_\rho > 0 \quad (8.72)$$

Esto implica que  $\alpha \in I_1$  para todo  $t$  siempre y cuando  $\alpha(0) \in I_1$  y  $\alpha \in I_2$  para todo  $t$  siempre y cuando  $\alpha(0) \in I_2$  respectivamente. Esta condición de estabilidad sólida también se ha verificado en las aplicaciones.

## 8.8. CINEMATICA DE ROBOTS RODANTES

Los robots rodantes son básicamente de dos tipos, dependiendo de si son suplidos con ruedas convencionales o con omnidireccionales. Los más simples robots con ruedas

convencionales sólo son capaces de hacer movimientos de 2-DOF, y por lo tanto, son cinemáticamente equivalentes a los vehículos convencionales terrestres. Sin embargo, los robots con ruedas omnidireccionales (ODWs) son capaces de hacer movimientos de 3 DOF, lo que aumenta considerablemente su maniobrabilidad. A continuación detallamos la cinemática de los dos tipos de robots.

### 8.8.1. Robots con ruedas convencionales

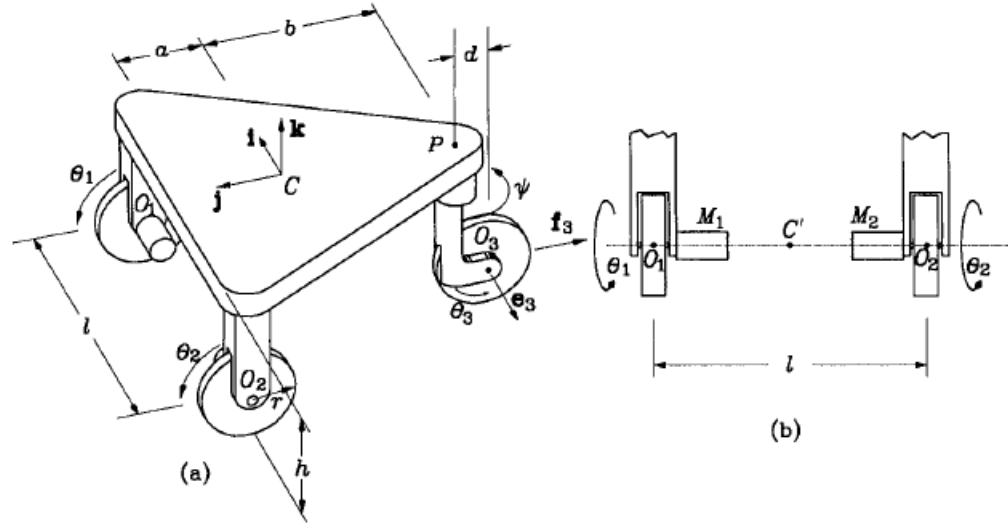
Comenzamos con robots rodantes con ruedas convencionales. Debido a que estos tienen dos grados de libertad, estos necesitan sólo dos actuadores, los diseños disponibles varían esencialmente en donde estos actuadores se encuentren. La arquitectura básica de este tipo de robot se muestra en la figura 8.34a, en la que se distingue un chasis o cuerpo de robot, representado como una placa triangular: dos ruedas coaxiales que están acopladas al chasis por medio de ejes giratorios que pasan a través de los puntos de  $O_1$  y  $O_2$ ; y una tercera rueda montada en un soporte. Este tren se requiere para permitir diferentes velocidades angulares de las dos ruedas coaxiales. Además, la orientación de plano medio de la rueda de dirección, definido por el ángulo  $\Psi$ , se controla con el segundo actuador.

Este diseño tiene algunos inconvenientes a saber, (i) los dos motores que sirven esencialmente para dos tareas diferentes requieren características operativas esencialmente diferentes, hasta el punto de que ambos no pueden estar disponibles en el mismo fabricante, (ii) el motor de propulsión da el control de la velocidad, el motor de dirección proporciona control de posición, lo que da lugar a dos sistemas de control independientes que pueden terminar funcionando sin coordinación y, finalmente, (iii) el uso de un tren diferencial de engranajes aumenta el costo, el peso, y provoca una reacción inherente de los engranajes.

En la segunda matriz de accionamiento, que se muestra en la figura 8.34b, las dos ruedas coaxiales están alimentadas de forma independiente, eliminando así el tren diferencial y sus efectos secundarios indeseables, la tercera rueda es una rueda libre. Además, la orientación de la última está determinada por la fricción y las fuerzas de restricción, haciendo así innecesario el sistema de control de la dirección de la primera matriz. A continuación se analiza la cinemática de un robot con esta forma de accionamiento.

Se hace el punto  $C$  de la plataforma el punto de operación, su proyección sobre un plano horizontal  $H$  que contiene el eje común de las dos ruedas accionadas como  $C'$  como se indica en la figura 8.34b. Que, por otra parte, el vector de posición de  $C'$  en un marco fijo al suelo, con origen acostado en  $H$ , se denota por  $c$ . Además, sea  $\omega$  la

Figura 8.34: Un robot rodante de 2-DOF: (a) Su disposición general, y (b) Detalle de las ruedas accionadas.



(Fuente: Fundamentals of Robotic Mechanical Systems, Theory, Methods, and Algorithms, Third Edition, Jorge Angeles.)

velocidad escalar angular de la plataforma alrededor de un eje vertical. En virtud del movimiento de 2-DOF de este robot, podemos controlar tanto la velocidad  $\dot{c}$  de  $C$  o de  $C'$  para el caso- o una combinación  $\omega$  y de una función escalar de  $\dot{c}$  especificando correctamente las dos velocidades de articulación  $\theta_1$  y  $\theta_2$ . Sin embargo, no podemos controlar los dos componentes de  $\dot{c}$  y  $\omega$  simultáneamente.

A fin de proceder con el análisis cinemático del sistema que nos ocupa, una tríada ortogonal de vectores cuya orientación se fija con respecto al chasis. Esta tríada se denota por  $i, j, k$ , con  $k$  apuntando en la dirección vertical hacia arriba. Por lo tanto, las velocidades de los puntos  $\dot{o}_i O_i$ , para  $i = 1, 2$ , están dadas por

$$\dot{o}_i = r\dot{\theta}_i j, \quad i = 1, 2 \quad (8.73a)$$

Además, la velocidad de  $C$  puede escribirse ahora en forma 2-dimensional como:

$$\dot{c} = \dot{o}_i + \omega E(c - \dot{o}_i), \quad i = 1, 2 \quad (8.73b)$$

Con  $E$  que se define como:

$$E = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Por lo tanto, todos los vectores de la ecuación 8.73b son de 2 dimensiones. Sustituyendo la ecuación 8.73a en la ecuación 8.73b y restando la ecuación lateral 8.73b para  $i = 1$  y para  $i = 2$ , obtenemos

$$[r(\dot{\theta}_1 - \dot{\theta}_2) - \omega l]j = O_2$$

Por lo tanto, la velocidad angular  $\omega$  de la línea  $O_1O_2$  en movimiento plano, la cual es la misma que la de la plataforma, puede ser fácilmente expresada como

$$\omega = \frac{r}{l}(\dot{\theta}_1 - \dot{\theta}_2) \quad (8.73c)$$

Su dirección positiva que es la misma de  $k$ . Al sustituir las ecuaciones 8.73a y 8.73c en la ecuación 8.73b, obtenemos expresiones para  $\dot{c}$  en términos de las velocidades de articulación, similar a las ecuación 8.73b, para  $i = 1, 2$ . Además, tras la adición de las dos expresiones como resultado, se obtiene  $\dot{c}$  en la forma deseada, es decir,

$$\dot{c} = a\frac{r}{l}(\dot{\theta}_1 - \dot{\theta}_2)i + \frac{r}{l}(\dot{\theta}_1 - \dot{\theta}_2)j \quad (8.73d)$$

Las ecuaciones ( 8.73c y d) expresan ahora las relaciones cinemáticas directas diferenciales del robot en estudio. En forma compacta, estas relaciones se convierten

$$t = L\dot{\theta}_a \quad (8.73e)$$

Con la matriz  $3 \times 2$ ,  $L$  se define como:

$$L \equiv \begin{bmatrix} \frac{r}{l} & -\frac{r}{l} \\ (\frac{ar}{l}i) + (\frac{r}{2})i & -(\frac{ar}{l}i) + (\frac{r}{2})i \end{bmatrix} \quad (8.73f)$$

Además, la torsión plana  $t$  de la plataforma y el vector de 2 dimensiones  $\theta_a$  velocidades de articulación de accionamiento se definen como

$$L \equiv \begin{bmatrix} \omega \\ \dot{c} \end{bmatrix}, \quad \theta_a \equiv \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \quad (8.73g)$$

Calcular las velocidades de articulación de las ecuaciones anteriores, es decir, resolver el problema de cinemática inversa asociado, es ahora una tarea trivial. Las relaciones cinemáticas inversas se calculan abajo al notar que la ecuación 8.73a ofrece una relación para la diferencia de velocidades de articulación. Por lo tanto, todo lo que necesitamos ahora es una segunda ecuación para la suma de velocidad de articulación. Por inspección de la ecuación 8.73d, es evidente que se puede derivar esta relación multiplicando ambos lados de esta ecuación por  $j$ , obteniendo de este modo

$$\dot{c} \cdot j = \frac{r}{2}(\dot{\theta}_1 + \dot{\theta}_2) \quad (8.74)$$

Las dos ecuaciones 8.73c y 8.74 ahora se puede convertir en la forma usual

$$j\theta_a = kt \quad (8.75a)$$

Donde los dos jacobianos del robot  $J$  y  $K$  se dan aqui:

$$J \equiv \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \frac{l}{r} & 0^T \\ 0 & \frac{2}{r}j^T \end{bmatrix} \quad (8.75b)$$

Nótese que  $J$  es una matriz  $2 \times 2$ , pero  $K$  es una matriz  $2 \times 3$

Las relaciones cinemáticas inversas se derivan fácilmente de la ecuación 8.75a,

$$\dot{\theta}_1 = \frac{1}{2} \left( \frac{l}{r}\omega + \frac{2}{r}\dot{y} \right)$$

$$\dot{\theta}_2 = -\frac{1}{2} \left( \frac{l}{r}\omega - \frac{2}{r}\dot{y} \right)$$

Donde  $\dot{y} \equiv \dot{c} \cdot j$ . Ahora, con el fin de completar el análisis cinemático del robot, se calculan las velocidades de las articulaciones no accionadas,  $\dot{\theta}_3$  y  $\dot{\psi}$ . Con este fin, se denota a  $\omega_i$ , para  $i = 1, 2, 3$ , y  $\omega_3$  el vector de velocidad angular de 3 dimensiones de la  $i$ -ésima rueda y el vector de velocidad de 3 dimensiones del centro de la rueda loca. Asimismo,  $\omega_4$  denota la velocidad angular escalar del soporte.

Tenemos así, para los vectores de velocidad angular de las dos ruedas accionadas.

$$\omega_1 = -\dot{\theta}_1 + \omega k = -\dot{\theta}_1 + \frac{r}{l}(\dot{\theta}_1 - \dot{\theta}_2)k = \left[ -i + \left(\frac{r}{l}\right)k - \left(\frac{r}{l}\right)k \right] \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \quad (8.76a)$$

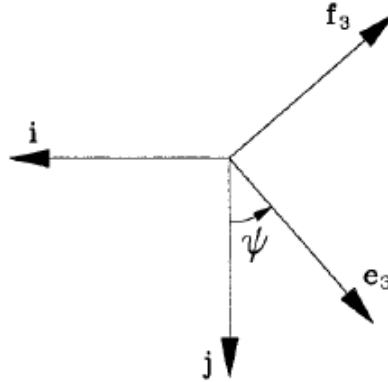
$$\omega_2 = -\dot{\theta}_2 + \omega k = -\dot{\theta}_2 + \frac{r}{l}(\dot{\theta}_1 - \dot{\theta}_2)k = \left[ \left(\frac{r}{l}\right)k - i - \left(\frac{r}{l}\right)k \right] \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \quad (8.76b)$$

En las derivaciones subsiguientes, se necesitaran las velocidades de los centros de las dos ruedas accionadas, las cuales fueron derivadas en la ecuación 8.73a. Además, la velocidad angular de la rueda loca puede ser fácilmente escrita en el marco fijo al soporte,  $\{e_3, f_3, k\}$ , es decir:

$$\omega_3 = \theta_3 e_3 + (\omega + \dot{\psi})k \quad (8.77)$$

Con  $\dot{\psi}$  que caracteriza el ángulo entre los vectores  $j$  y  $e_3$  de la figura 8.34a, medido en la dirección positiva de  $k$ , tal como se indica en el esquema de la figura 8.35.

Figura 8.35: Presentación de los vectores unitarios fijados a la plataforma y al soporte.



(Fuente: Fundamentals of Robotic Mechanical Systems, Theory, Methods, and Algorithms, Third Edition, Jorge Angeles.)

Tenga en cuenta que el vector  $e_3$  es paralelo al eje de rodadura de la rueda loca, mientras que  $f_3$  es un vector horizontal perpendicular a  $e_3$ . Estos dos conjuntos de vectores unitarios están relacionados por

$$e_3 = -\sin \psi i + \cos \psi j \quad (8.78a)$$

$$f_3 = -\cos \psi i - \sin \psi j \quad (8.78b)$$

Sus relaciones inversas son

$$i = -\sin \psi e_3 + \cos \psi f_3 \quad (8.79a)$$

$$j = \cos \psi e_3 - \sin \psi f_3 \quad (8.79b)$$

Además, la velocidad del centro de la rueda loca se deriva como

$$\dot{o}_3 = \omega_3 \times rk = -r\dot{\theta}_3 f_3$$

Mientras que la velocidad angular escalar Del soporte,  $\omega_4$ , está dada por

$$\omega_4 = (\omega + \dot{\psi}) = \frac{r}{l}(\dot{\theta}_1 - \dot{\theta}_2) + \dot{\psi} \quad (8.80)$$

Si en algún momento necesitamos a  $\dot{c}$  en coordenadas del soporte. Tal expresión se obtiene a partir de las ecuaciones 8.73a b,c etc. y (8.79a y d), es decir,

$$\dot{c} = \left[ -a\frac{r}{l}(\dot{\theta}_1 - \dot{\theta}_2) \sin \psi + \frac{r}{2}(\dot{\theta}_1 + \dot{\theta}_2) \cos \psi \right] e_3 - \left[ a\frac{r}{l}(\dot{\theta}_1 - \dot{\theta}_2) \cos \psi + \frac{r}{2}(\dot{\theta}_1 + \dot{\theta}_2) \cos \psi \right] f_3 \quad (8.81)$$

Las expresiones para las velocidades de dependencia en términos de las independientes,  $\dot{\theta}_1$  y  $\dot{\theta}_2$ , se deriva fácilmente. Con este fin, expresamos la velocidad de P en dos formas independientes, uno en términos de la velocidad de  $O_3$  y el otro en términos de la velocidad de C, es decir,

$$\dot{p} = \dot{o} + \omega_4 \times (p - o_3) \quad (8.82a)$$

$$\dot{p} = \dot{c} + \omega_4 \times (-bj) \quad (8.82b)$$

Al igualar los lados derechos de las ecuaciones anteriores, obtenemos una ecuación vectorial de 3 dimensiones en relación de dependencia con las velocidades independientes, a saber,

$$-r\dot{\theta}_3 f_3 + (\omega + \dot{\psi}) k \times (p - o_3) = \dot{c} + b\omega i$$

Donde hemos recordado las expresiones deducidas anteriormente para  $o_3$  y  $\omega_4$ . Además, se ha re escrito la ecuación anterior con las velocidades desconocidas,  $\omega_3$  y  $\dot{\psi}$ , en el lado izquierdo, es decir,

$$-r\dot{\theta}_3 f_3 + \dot{\psi} k \times (p - o_3) = \dot{c} + b\omega i - \omega k \times (p - o_3) \quad (8.83)$$

Por otra parte, observamos que, a partir de la figura 8.34,

$$p - o_3 = -df_3 + (h - r) k$$

Y por lo tanto

$$kx(p - o_3) = de_3$$

La ecuación 8.83 en consecuencia llega a ser

$$-r\dot{\theta}_3 f_3 + \dot{\psi} de_3 = \dot{c} + \omega(bi - de_3) \quad (8.84)$$

Ahora bien, es una cuestión simple de resolver para  $\theta_3$  y  $\dot{\psi}$  de la ecuación 8.84. De hecho, se resuelve el  $\theta_3$  multiplicando ambos lados de la ecuación anterior por  $f_3$ . Del mismo modo, se resuelve para  $\dot{\psi}$  multiplicando ambos lados de la misma ecuación por  $e_3$ , obteniendo así

$$-r\dot{\theta}_3 = \dot{c} \cdot f_3 + \omega bi \cdot f_3$$

$$d\dot{\psi} = \dot{c}e_3 + \omega(bi \cdot e_3 - d)$$

Ahora, al recordar las expresiones deducidas anteriormente para  $\omega$  y  $\dot{c}$ , se obtiene

$$\begin{aligned}\dot{c}f_3 &= -a\frac{r}{l}(\dot{\theta}_1 - \dot{\theta}_2)\cos\psi - \frac{r}{2}(\dot{\theta}_1 + \dot{\theta}_2)\sin\psi \\ \dot{c}e_3 &= -a\frac{r}{l}(\dot{\theta}_1 - \dot{\theta}_2)\sin\psi - \frac{r}{2}(\dot{\theta}_1 + \dot{\theta}_2)\cos\psi \\ if_3 &= -\cos\psi, \quad ie_3 = -\sin\psi\end{aligned}$$

Por lo tanto,

$$\dot{\theta}_3 = a\cos\psi(\dot{\theta}_1 - \dot{\theta}_2) + \frac{1}{2}(\sin\psi)(\dot{\theta}_1 + \dot{\theta}_2) \quad (8.85a)$$

$$\dot{\psi} = \rho \left[ -(a\sin\psi + \delta)(\dot{\theta}_1 - \dot{\theta}_2) + \frac{1}{2}(\cos\psi)(\dot{\theta}_1 + \dot{\theta}_2) \right] \quad (8.85b)$$

Con las definiciones dadas a continuación:

$$a \equiv \frac{a+b}{l}, \quad \delta \equiv \frac{d}{l}, \quad \rho \equiv \frac{r}{d} \quad (8.86)$$

Por lo tanto, si dejamos que  $\dot{\theta}_u = [\theta_3 \dot{\psi}]^T$  el vector de velocidades de articulación no accionadas, entonces tenemos

$$\dot{\theta}_u = \Theta \dot{\theta}_\alpha \quad (8.87a)$$

Con  $\Theta$  definido como

$$\Theta = \begin{bmatrix} \alpha\cos\psi + \frac{\sin\psi}{2} & -\alpha\cos\psi + \frac{\sin\psi}{2} \\ \rho \left[ -\alpha\sin\psi + \frac{\cos\psi}{2} - \delta \right] & \rho \left[ \alpha\sin\psi + \frac{\cos\psi}{2} - \delta \right] \end{bmatrix} \quad (8.87b)$$

Completando de esta manera el análisis de la cinemática previsto.

## 8.8.2. Los robots con ruedas omnidireccionales

En general, las ruedas omnidireccionales (ODWs) permiten dos movimientos independientes de traslación sobre el suelo de soporte y un movimiento independiente de rotación alrededor de un eje vertical. Basado en las formas de las ruedas, por otra parte, las ODWs se pueden clasificar en las ruedas esféricas y ruedas Mekanum, estas últimas también se conoce como “ilonators”.

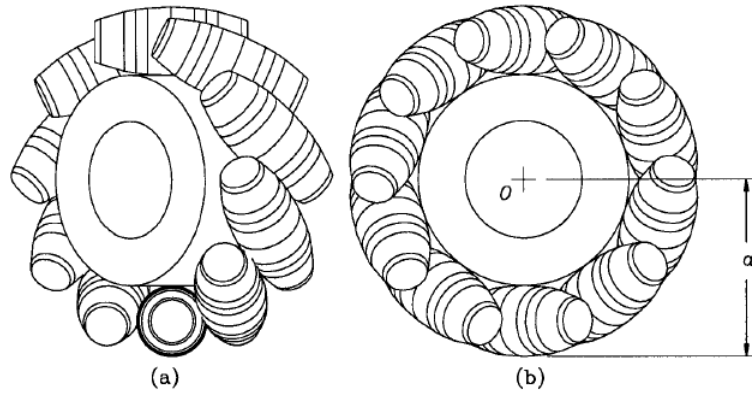
La rueda Mekanum lleva un conjunto de rodillos montados a lo largo de la periferia del cubo de la rueda en un ángulo dado, tal como se muestra en la figura 8.36a. Además, los rodillos están conformados de manera que la rueda aparece como circular en su vista lateral, como se muestra en la figura 8.36b, con el fin de asegurar un movimiento suave. Los vectores unitarios ortogonales por parejas  $e_i$ ,  $f_i$  y  $g_i$ ,  $h_i$  se definen en los planos horizontales medios del cubo de la rueda y del rodillo en contacto con el suelo,

respectivamente. Este rodillo se denomina activo en la discusión que sigue. Ahora se pretende encontrar la relación cinemática entre las velocidades de articulación de la rueda  $\{\dot{\theta}\}_{i_1}^n$  y las variables de velocidad cartesianas del robot, es decir, la velocidad angular escalar  $\omega$  y el vector velocidad de 2 dimensiones  $\dot{c}$  del centroide de la plataforma. Para ello, expresamos la velocidad  $\dot{o}_i$  del centroide  $O_i$  de la  $i$ -ésima rueda en dos formas diferentes: en primer lugar nos fijamos en esta velocidad del rodillo activa hasta el centroide  $O_i$ , luego, desde el centroide de la plataforma  $C$  a  $O_i$ .

Si se relaciona la velocidad de  $O_i$  con la del punto de contacto del rodillo activo con el suelo, entonces podemos escribir, con la ayuda de la figura 8.37,

$$\dot{o}_i = \dot{p}_i + \dot{v}_i \quad (8.88)$$

Figura 8.36: (a) La Rueda Mekanum, (b) Su vista lateral.



(Fuente: Fundamentals of Robotic Mechanical Systems, Theory, Methods, and Algorithms, Third Edition, Jorge Angeles.)

Con  $v_i$  definida como la velocidad relativa de  $O_i$  con respecto a  $P_i$ . Ahora  $\omega_h$  y  $\omega_r$  denotan los vectores de velocidad angular del cubo y el rodillo, respectivamente, es decir.

$$\omega_h = \omega k + \dot{\theta}_i e_i, \quad \omega_r = \omega_h + \dot{\phi}_i g_i$$

Tenemos así

$$\dot{p}_i = \omega_r \times \overrightarrow{Q_i P_i} = (\omega k + \dot{\theta}_i e_i + \dot{\phi}_i g_i) \times bk$$

Donde  $b$  es el radio de los rodillos en el punto de contacto con el suelo. Además,  $\theta_i$  denota la velocidad del cubo de la rueda, mientras que  $\dot{\phi}_i$  denota la del rodillo activo,

las cuales son positivas en las direcciones de los vectores  $e_i$  y  $g_i$ , respectivamente. Por lo tanto,

$$\dot{p}_i = -b(\dot{\theta}_i f_i + \dot{\phi}_i h_i) \quad (8.89)$$

Por otra parte,

$$v_i = \omega_h \times \overrightarrow{Q_i P_i} = (\omega k + \dot{\theta}_i e_i) \times (a - b)k$$

$a$  denota la altura del eje del cubo de la rueda, como se muestra en la figura 8.36(b). Por lo tanto,

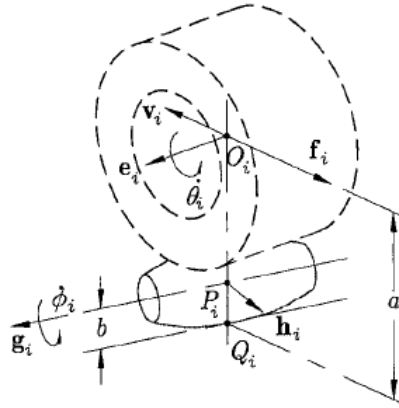
$$v_i = -\dot{\theta}_i(a - b)f_i \quad (8.90)$$

Obteniendo de este modo la expresión deseada para  $\dot{o}_i$ ,

$$\dot{o}_i = -a\dot{\theta}_i f_i - b\dot{\phi}_i h_i \quad (8.91)$$

Una disposición general de la  $i$ -ésima ODW con ejes de los rodillos en un ángulo  $\alpha_i$  con respecto a la normal  $e_i$  al plano vertical medio del cubo correspondiente se muestra en la figura 8.38.

Figura 8.37: El rodillo activo de la  $i$ -ésima rueda.

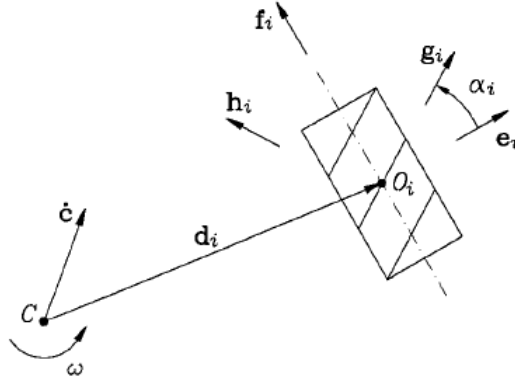


(Fuente: Fundamentals of Robotic Mechanical Systems, Theory, Methods, and Algorithms, Third Edition, Jorge Angeles.)

El subíndice  $i$  se asocia tanto con la  $i$ -ésima rueda y su rodillo activo. Además, la velocidad  $\dot{o}_i$  de la  $i$ -ésima rueda se puede expresar en términos de las variables de velocidad cartesianas,  $\dot{c}$  y  $\omega$ , como

$$\dot{o}_i = \dot{c} + \omega E d_i \quad (8.92)$$

Figura 8.38: El diseño de la  $i$ -ésima rueda con respecto a la plataforma robot.



(Fuente: Fundamentals of Robotic Mechanical Systems, Theory, Methods, and Algorithms, Third Edition, Jorge Angeles.)

Donde se ha utilizado una representación vectorial de 2 dimensiones, con  $d_i$  definido como el vector dirigido desde el punto  $C$  hasta el centroide  $O_i$  del cubo y  $E$  se definido como en la ecuación  $E$ . Además, puesto que todos los rodillos no están accionados y giran sin hacer nada, el valor de  $\phi_i$  es irrelevante para nuestro estudio. Por lo tanto, eliminar esta variable a partir de las ecuaciones anteriores, que se realiza multiplicando ambos lados de la ecuación 8.91 por  $g_i$ ,  $h_i$  normales, con lo que se deduce

$$g_i^T \dot{o}_i = -a \dot{\theta}_i g_i^T f_i$$

Pero

$$g_i^T f_i = \sin \alpha_i$$

Por lo tanto,

$$g_i^T \dot{o}_i = -a(\sin \alpha_i) \dot{\theta}_i \quad (8.93)$$

La misma multiplicación se realizó en la ecuación 8.90 se obtiene

$$g_i^T \dot{o}_i = (g_i^T E d_i) \omega + g_i^T \dot{c} \quad (8.94)$$

Al igualar los lados derechos de las ecuaciones 8.93 y 8.94, obtenemos la relación deseada,

$$-a(\sin \alpha_i) \dot{\theta}_i = k_i^T t \quad i = 1, \dots, n \quad (8.95)$$

Donde el vector de 3 dimensiones  $k_i$  se define como

$$k_i = \begin{bmatrix} g_i^T E d_i \\ g_i \end{bmatrix}$$

Y el vector  $t$  giro es como se definió en la ecuación 8.73g. Definimos ahora el vector de tasas de ruedas  $\theta$  en la forma

$$\begin{bmatrix} \dot{\theta}_1 & \dot{\theta}_2 & \dots & \dot{\theta}_n \end{bmatrix}^T \quad (8.96)$$

Si las  $n$  ecuaciones de la ecuación 8.95 ahora están ensambladas, se obtiene

$$J\dot{\theta} = Kt \quad (8.97)$$

Donde, si se supone que todos los ángulos  $\alpha_i$  son idénticos y etiquetados como  $\alpha$ , entonces el jacobiano  $nxn$   $J$  y el Jacobiano  $nx3$   $K$  adoptan las formas

$$J \equiv -a \sin \alpha \quad (8.98a)$$

$$K \equiv \begin{bmatrix} g_1^T E d_1 & g_1^T \\ g_n^T E d_n & g_n^T \end{bmatrix} \quad (8.98b)$$

Con  $1$  denotando la matriz identidad  $nxn$ . Dadas las ecuaciones 8.98a y 8.98b, la cinemática inversa diferencial se puede resolver como

$$\dot{\theta} = -\frac{1}{\alpha \sin \alpha} Kt \quad (8.99)$$

Donde es evidente que  $\sin \alpha$  debe ser diferente de cero, es decir, los ejes de los rodillos no deben ser paralelos al eje del cubo. Si estos ejes son paralelos, entonces las ODWs se reducen a las ruedas convencionales.

---

---

# CAPÍTULO 9

---

## DINÁMICA DE LOS ROBOTS MÓVILES

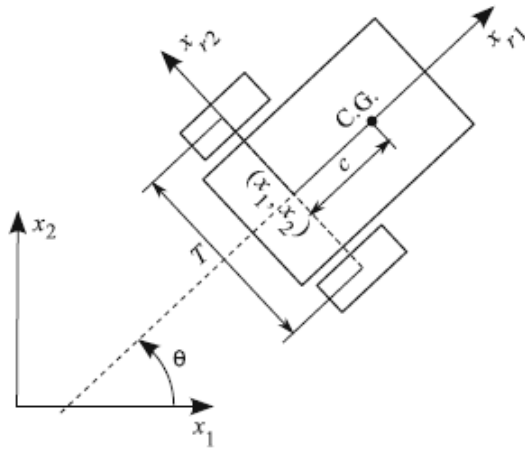
### 9.1. ROBOTS MÓVILES HILARE

Una figura esquemática de un robot móvil Hilare se muestra en la figura 9.1. Este tipo de robot se utiliza sobre todo para aplicaciones en interiores. El mecanismo de accionamiento de un robot tipo Hilare tiene dos motores independientes. Cada uno de estos motores alimenta una de las ruedas del robot. Por lo tanto, las entradas reales cinemáticas que impulsan al robot y afectan a su velocidad y dirección del movimiento son las dos velocidades de las ruedas. Con esto en mente, a primera vista parece intuitivo escribir las ecuaciones cinemáticas del movimiento de un robot móvil Hilare en términos de estas velocidades. Sin embargo, en los robots móviles más comerciales, existe un controlador de bajo nivel que controla la velocidad lineal y angular del robot.

Por lo tanto, para fines de aplicación, es más conveniente elegir la velocidad lineal y angular del robot móvil como las entradas del modelo cinemático. Cuando una ley de control se encuentra más adelante sobre la base de este modelo, se puede aplicar más fácilmente utilizando los paquetes de desarrollo disponibles para los robots comerciales.

Consideremos el robot móvil tipo Hilare mostrado en la figura 9.1. El sistema de coordenadas de cuerpo  $x_{r1} - x_{r2}$  está unido al cuerpo del robot en el centro de la línea que conecta el centro de las dos ruedas. Nótese que el centro de gravedad del robot se supone que está en el eje longitudinal del cuerpo  $x_{r1}$  con una distancia  $c$  del origen del sistema de coordenadas del cuerpo. Para el modelado dinámico de un robot móvil, uno

Figura 9.1: Robot móvil Hilare



Fuente: Autonomous Robots, Modeling, Path Planning, and Control, Farbod Fahimi, 2009

tiene que asumir los pares ejercidos por los motores a las ruedas como las entradas. La masa de las ruedas del robot tiene generalmente el mismo orden de magnitud que la del cuerpo del robot. Por lo tanto, despreciando la rotación de las ruedas y suponiendo que el robot entero como un cuerpo rígido puede resultar en un modelo dinámico incorrecto. Lo mejor es incluir el efecto de las rotaciones de la rueda en el modelo dinámico. Hay varios métodos disponibles para derivar las ecuaciones dinámicas de movimiento para robots móviles. En este caso, estamos usando las ecuaciones de Newton-Euler del movimiento. Vamos a utilizar un enfoque más sistemático. Los pasos de este método sistemático comprenden:

1. Las coordenadas y las velocidades generalizadas que definen el estado dinámico del sistema están definidas.
2. La aceleración de los centros de masa del robot y las ruedas se expresan en términos de la derivada de las velocidades generalizadas.
3. El diagrama de cuerpo libre del robot y las ruedas se dibujan y todas las fuerzas implicadas se muestran en el diagrama.
4. Las ecuaciones de movimiento se escriben usando el balance de todas las fuerzas y los momentos con las fuerzas y momentos de inercia.

5. La ecuación de restricción no integrable (no holonómica), la cual se deriva de la condición de no deslizamiento, se utiliza para completar las ecuaciones dinámicas de movimiento.
6. Finalmente, las ecuaciones de movimiento se convierten en la forma de primer orden para permitir simulaciones numéricas más simples.

### 9.1.1. Coordenadas generalizadas y velocidades generalizadas

Las coordenadas generalizadas para un sistema dinámico son las variables que definen de manera única la configuración geométrica del sistema en cualquier momento dado. Estas coordenadas deben ser independientes, en cuyo caso el número de ellas es igual al número de DOFs del sistema dinámico. Son necesarias tres variables independientes para definir únicamente la configuración geométrica del robot. Estas variables son las dos componentes del vector posición inercial del punto medio de la línea transversal que sujeta los centros de las ruedas,  $(x_1, x_2)$ , y el ángulo entre el eje longitudinal del robot con el eje inercial  $x_1, \theta$  (Figura 9.1).

Las coordenadas generalizadas únicamente definen la configuración geométrica (es decir, la posición y orientación) del sistema dinámico, sin embargo, para definir completamente el estado dinámico de un sistema. Para un robot móvil, dos de las velocidades generalizadas se seleccionan para ser las componentes de la velocidad del punto  $(x_1, x_2)$  que se define en términos del eje del cuerpo,  $(\dot{x}_{r1}, \dot{x}_{r2})$ . La velocidad angular del cuerpo del robot,  $\dot{\theta}$ , se selecciona como la tercera velocidad generalizada. Tenga en cuenta que debido a la condición de no deslizamiento, una ecuación de restricción de velocidad se puede escribir en términos de  $\dot{x}_{r1}$  y  $\dot{x}_{r2}$ . Las coordenadas anteriormente definidas y velocidades generalizadas se agrupan en un solo vector,  $\mathbf{q}$ , conocido como el vector de estado del robot móvil.

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \theta \\ \dot{x}_{r1} \\ \dot{x}_{r2} \\ \dot{\theta} \end{bmatrix} \quad (9.1)$$

La condición de no deslizamiento para el movimiento lateral resulta en la siguiente restricción cinemática.

$$\dot{x}_{r2} = -\dot{x}_1 \sin\theta + \dot{x}_2 \cos\theta = 0 \quad (9.2)$$

Vamos a utilizar esta restricción cinemática más tarde, cuando se vayan a escribir las

ecuaciones de movimiento. Esta ecuación no se puede integrar con respecto al tiempo porque  $\theta$  no se conoce como una función del tiempo antes de cualquier simulación dinámica. Por lo tanto, esta restricción no se puede convertir en una restricción entre las coordenadas generalizadas por integración. Tal restricción no integrable en componentes de la velocidad se denomina restricción no holonomica. En general, cualquier sistema dinámico que tenga una ecuación de restricción no integrable se llama sistema dinámico no holonomico.

### 9.1.2. La aceleración de los centros de masa del robot y las ruedas

El segundo paso del método sistemático para encontrar las ecuaciones de movimiento utilizando el método de Newton-Euler es encontrar las aceleraciones de los centros de masa de todos los cuerpos. Aquí, el robot como un sistema dinámico consta de tres cuerpos rígidos: el cuerpo del robot y dos ruedas. La forma de las ecuaciones será más simple si se usa el sistema de coordenadas del cuerpo para expresar las aceleraciones inerciales de los tres cuerpos. Supongamos que la aceleración inercial del origen del sistema de coordenadas del cuerpo es:

$$a_0 = \ddot{x}_{r1}\hat{i}_r + \ddot{x}_{r2}\hat{j}_r \quad (9.3)$$

Donde  $\hat{i}_r$  y  $\hat{j}_r$  son los vectores unitarios del sistema de coordenadas del cuerpo. La velocidad y la aceleración angular del sistema de coordenadas del cuerpo son:

$$\begin{aligned} w &= \dot{\theta}\hat{k}_r, \\ \alpha &= \ddot{\theta}\hat{k}_r \end{aligned}$$

Si la distancia del centro de masa del cuerpo del robot con el origen del marco del cuerpo del robot se denota por  $c = c\hat{j}_r$ , entonces la aceleración inercial del centro de masa del cuerpo del robot expresada en el sistema de coordenadas cuerpo es

$$\begin{aligned} a_r &= a_0 + \alpha \cdot c - w^2 c \\ a_{r1}\hat{i}_r + a_{r2}\hat{j}_r &= (\ddot{x}_{r1} - c\dot{\theta}^2)\hat{i}_r + (\ddot{x}_{r2} + c\ddot{\theta})\hat{j}_r \end{aligned} \quad (9.4)$$

Si la distancia del centro de masa de la rueda derecha del robot con el origen del marco del cuerpo del robot se denota por  $d_r = -(\frac{T}{2})\hat{i}_r$ , entonces la aceleración inercial del centro de masa de la rueda derecha del robot expresada en el marco del cuerpo es

$$\begin{aligned} a_{wr} &= a_0 + \alpha \cdot d_r - w^2 d_r \\ a_{wr1}\hat{i}_r + a_{wr2}\hat{j}_r &= (\ddot{x}_{r1} + \frac{T}{2}\dot{\theta}^2)\hat{i}_r + (\ddot{x}_{r2} + \frac{T}{2}\ddot{\theta})\hat{j}_r \end{aligned} \quad (9.5)$$

### 9.1.3. Diagramas de cuerpo libre y cinético

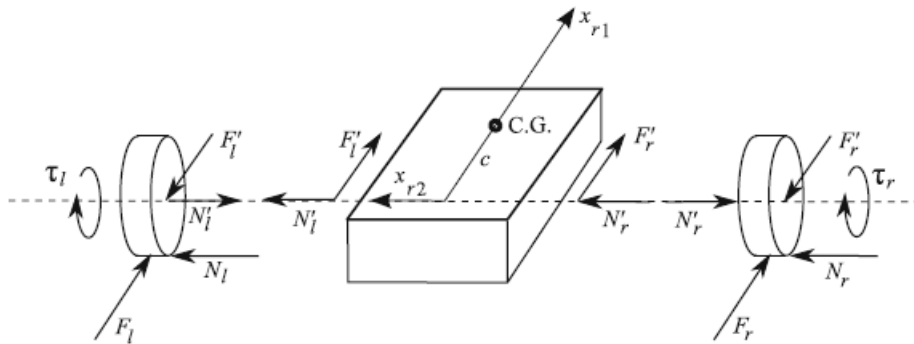
Dado que en este modelo el efecto dinámico de la rotación de las ruedas está siendo considerado en la obtención de las ecuaciones de movimiento, las ruedas y el cuerpo deben ser separados en el diagrama de cuerpo libre. El diagrama de cuerpo libre del cuerpo del robot móvil y las ruedas se muestra en la figura 9.2. Este diagrama de cuerpo libre nos ayuda a visualizar mejor las fuerzas en las ruedas. Ahora, la versión 2D de los diagramas cinéticos y de cuerpo libre para el robot completo y las ruedas como partes separadas pueden ser dibujados. Estos diagramas se muestran en las figuras 9.3 y 9.4.

### 9.1.4. Las ecuaciones de movimiento

Los diagramas de cuerpo libre y diagramas cinéticos mostrados en las figuras 9.3 y 9.4 simplifican la escritura de las ecuaciones de movimiento. Observando la figura 9.3, se puede escribir el balance de fuerzas de la fuerza externa y de inercia en la dirección  $x_{r1}$ .

$$\begin{aligned}
 F_r + F_l &= m_w a_{wr1} + m_w a_{wl1} + m_r a_{wr1} \\
 F_r + F_l &= m_w (\ddot{x}_{r1} + \frac{T}{2} \ddot{\theta}) + m_w (\ddot{x}_{r1} - \frac{T}{2} \ddot{\theta}) + m_r (\ddot{x}_{r1} - c \dot{\theta}^2) \\
 F_r + F_l &= (m_r + 2m_w) \ddot{x}_{r1} - m_r c \dot{\theta}^2
 \end{aligned} \tag{9.6}$$

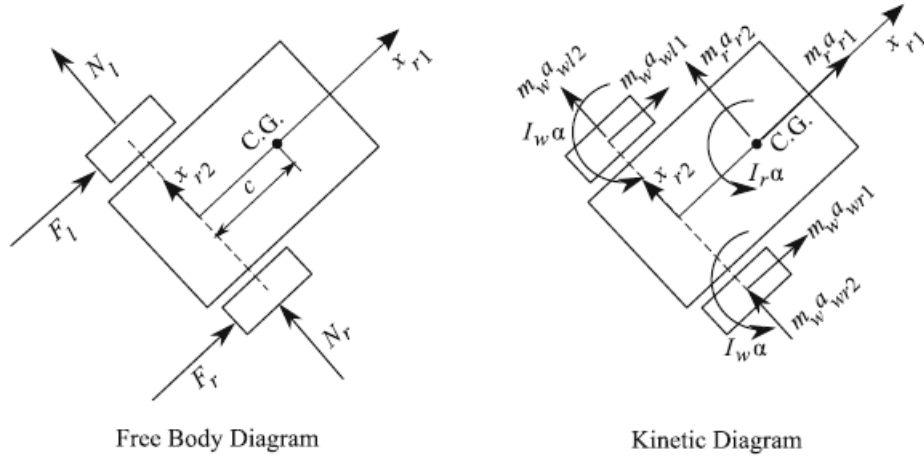
Figura 9.2: El diagrama de cuerpo libre detallado para un robot móvil Hilare



Fuente: Autonomous Robots, Modeling, Path Planning, and Control, Farbod Fahimi, 2009.

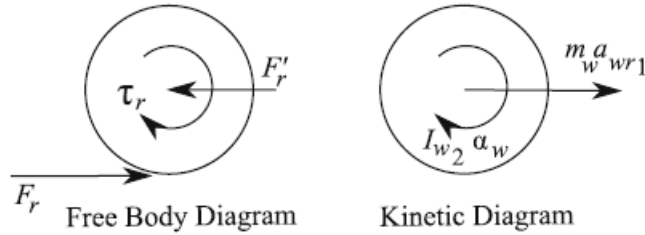
Observando la figura 9.3, se puede escribir el balance de fuerzas de la fuerza externa

Figura 9.3: El diagrama de cuerpo libre y cinético para un robot móvil Hilare como un único sistema



Fuente: Autonomous Robots, Modeling, Path Planning, and Control, Farbod Fahimi, 2009.

Figura 9.4: El diagrama de cuerpo libre y cinético de la rueda derecha



Fuente: Autonomous Robots, Modeling, Path Planning, and Control, Farbod Fahimi, 2009.

y de inercia en la dirección  $x_{r2}$ .

$$\begin{aligned}
 N_r + N_l &= m_w a_{wr2} + m_w a_{wl1} + m_r a_{r2} \\
 N_r + N_l &= m_w (\ddot{x}_{r2} + \frac{T}{2} \dot{\theta}^2) + m_w (\ddot{x}_{r2} - \frac{T}{2} \dot{\theta}^2) + m_r (\ddot{x}_{r2} - c\ddot{\theta}) \\
 N_r + N_l &= (m_r + 2m_w) \ddot{x}_{r2} - m_r c\ddot{\theta}
 \end{aligned} \tag{9.7}$$

Observando la figura 9.3, se puede escribir balance de momentos, de momento externo y momentos de inercia en la dirección  $x_{r3}$ .

$$\begin{aligned} (F_r + F_l)\frac{T}{2} &= (I_r + 2I_w)\alpha + m_w a_{wr1}\frac{T}{2} + m_w a_{wl1}\frac{T}{2} + m_r a_r 2c \\ (F_r + F_l)\frac{T}{2} &= (I_r + 2I_w)\dot{\theta} + m_w(\ddot{x}_{r1} + \frac{T}{2}\ddot{\theta})\frac{T}{2} - m_w(\ddot{x}_{r1} - \frac{T}{2}\ddot{\theta})\frac{T}{2} + m_r(\ddot{x}_{r2} - c\dot{\theta})c \\ (F_r + F_l)\frac{T}{2} &= (I_r + 2I_w + m_w\frac{T^2}{2})\theta + (\ddot{x}_{r2} + c\dot{\theta})c \end{aligned} \quad (9.8)$$

Observando la figura 9.4, se puede escribir el balance de momentos de momento externo y momentos de inercia en la dirección  $x_{r2}$ .

$$\begin{aligned} -F_r r + \tau_r &= I_{w2} + \alpha_w \\ -F_r r + \tau_r &= I_{w2}\frac{1}{r}(\ddot{x}_{r1} + \frac{T}{2}\ddot{\theta}) \end{aligned}$$

Esto se traduce en lo siguiente para la fuerza de tracción,  $F_r$ , en la rueda derecha:

$$F_r = \frac{1}{r}(\tau_r - \frac{I_{w2}}{r}(\ddot{x}_{r1} + \frac{T}{2}\ddot{\theta})) \quad (9.9)$$

Un resultado similar para la fuerza de tracción de la rueda izquierda,  $F_l$ , se puede lograr.

$$F_l = \frac{1}{r}(\tau_l - \frac{I_{w2}}{r}(\ddot{x}_{r1} + \frac{T}{2}\ddot{\theta})) \quad (9.10)$$

En las ecuaciones (9.9) y (9.10),  $\tau_r$  y  $\tau_l$  son el par motor de la rueda derecha e izquierda, respectivamente. Puesto que estos pares son la entrada al sistema, vamos a escribir las ecuaciones del movimiento en términos de estos pares, en lugar de en términos de las tracciones de las ruedas. Este objetivo se consigue mediante la sustitución de las ecuaciones (9.9) y (9.10) en las ecuaciones (9.6), (9.7), y (9.8). Las ecuaciones resultantes dinámicas del movimiento son:

$$\begin{aligned} (m_r + 2m_w + \frac{2}{r^2}I_{w2})\ddot{x}_{r1} - m_r c\dot{\theta}^2 &= \frac{1}{r}(\tau_r + \tau_l) \\ (m_r + 2m_w)\ddot{x}_{r2} + m_r c\dot{\theta} &= N_r + N_l \\ (I_r + 2I_w + m_w\frac{T^2}{2} + I_{w2}\frac{T^2}{2r^2} + m_r c)\ddot{\theta} + m_r \ddot{x}_{r2} &= \frac{T}{2r}(\tau_r - \tau_l) \end{aligned} \quad (9.11)$$

Aunque las ecuaciones dinámicas (9.11) son completas y precisas, no pueden ser utilizadas en esta forma para simular el movimiento del robot. La razón de este hecho es que la fuerza lateral del neumático  $N_r + N_l$  es desconocida. Esta fuerza lateral del neumático es una función de la aceleración lateral y angular del robot y esta cambia dinámicamente. Una ecuación adicional es necesaria para nosotros para ser capaces de determinar esta fuerza lateral como una función del tiempo. Observando de las ecuaciones dinámicas (9.11) más de cerca, una vez que se puede ver que si se puede encontrar una relación cinemática de la aceleración lateral, se puede calcular esta fuerza lateral desconocida durante la simulación de las ecuaciones dinámicas (9.11). Encontrar esta relación cinemática se discute a continuación.

### 9.1.5. El papel de la ecuación de restricción No holonómica

Conocer la aceleración lateral del robot,  $\ddot{x}_{r2}$ , es uno de los requisitos para la integración de las ecuaciones dinámicas 9.11. La restricción no holonomica definida en la ecuación 9.2 se puede utilizar para la determinación de la componente de aceleración lateral  $\ddot{x}_{r2}$ . La restricción no holonomica se repite aquí.

$$\dot{x}_{r2} = -\dot{x}_1 \sin\theta + \dot{x}_2 \cos\theta = 0 \quad (9.12)$$

Nótese que aunque la velocidad lateral  $\dot{x}_{r2}$  es cero debido a la condición no deslizante,  $\ddot{x}_{r2}$  podría ser distinto de cero bajo ciertas condiciones. Aquí, vamos a investigar si  $\ddot{x}_{r2}$  es, de hecho, cero. La aceleración lateral se puede encontrar derivando la ecuación 9.12.

$$\begin{aligned} \ddot{x}_{r2} &= -\ddot{x}_1 \sin\theta + \ddot{x}_2 \cos\theta - \dot{\theta}(\dot{x}_1 \cos\theta + \dot{x}_2 \sin\theta) \\ \ddot{x}_{r2} &= -\ddot{x}_1 \sin\theta + \ddot{x}_2 \cos\theta - \dot{\theta} \dot{x}_{r1} \end{aligned} \quad (9.13)$$

Para completar el cálculo de  $\ddot{x}_{r2}$ , vamos a tratar de determinar los componentes de aceleración del robot expresadas en el sistema de coordenadas inercial,  $\ddot{x}_{r1}$  y  $\ddot{x}_{r2}$ . La ecuación 9.12 implica que

$$\dot{x}_2 = \dot{x}_1 \tan\theta \quad (9.14)$$

Derivando la anterior expresión se tiene

$$\ddot{x}_2 = \ddot{x}_1 \tan\theta + \dot{x}_1 \dot{\theta} (1 + \tan^2\theta) \quad (9.15)$$

Sustituyendo la relación anterior en la ecuación 9.13 se obtiene

$$\ddot{x}_{r2} = \dot{\theta}(\dot{x}_1(1 + \tan^2\theta)\cos\theta - \dot{x}_{r1}) \quad (9.16)$$

Sin embargo, sabemos que  $\dot{x}_1 = \dot{x}_{r1}\cos\theta - \dot{x}_{r2}\sin\theta$  y  $\dot{x}_{r2} = 0$ . Usando estas relaciones con la ecuación anterior muestra que la componente de la aceleración lateral del robot se vuelve

$$\dot{x}_{r2} = 0 \quad (9.17)$$

La ecuación 9.17 es un resultado importante. Esto nos permite pasar por alto la segunda ecuación dinámica en 9.11 y simplificar la tercera ecuación dinámica en 9.11. Finalmente, las ecuaciones dinámicas de movimiento para el robot se reducen a lo siguiente:

$$\begin{aligned} (m_r + 2m_w + \frac{2}{r^2}I_w)\ddot{x}_{r1} - m_r c \dot{\theta}^2 &= \frac{1}{r}(\tau_r + \tau_l) \\ (I_r + 2I_w + m_w \frac{T^2}{2} + I_w \frac{T^2}{2r^2} + m_r c)\ddot{\theta} + m_r \ddot{x}_{r2} &= \frac{T}{2r}(\tau_r - \tau_l) \end{aligned} \quad (9.18)$$

### 9.1.6. La forma de primer orden de las ecuaciones dinámicas

Para fines de simulación, es deseable escribir las ecuaciones del movimiento del robot móvil en la forma de primer orden. Vamos a definir un nuevo conjunto de variables de estado para excluir a  $\dot{x}_{r2}$ , la cual es cero.

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \theta \\ \dot{x}_{r1} \\ \theta \end{bmatrix} \quad (9.19)$$

Utilizando la definición de la ecuación 9.19 para los estados dinámicos del robot, las ecuaciones de movimiento se escriben como sigue:

$$\begin{aligned} \dot{q}_1 &= q_4 \cos q_3 \\ \dot{q}_2 &= q_4 \sin q_3 \\ \dot{q}_3 &= q_5 \\ \dot{q}_4 &= \frac{1}{m} \left( \frac{1}{r} (\tau_r + \tau_l) + m_r c q_5^2 \right) \\ \dot{q}_5 &= \frac{T}{2I} \frac{1}{r} (\tau_r - \tau_l) \end{aligned} \quad (9.20)$$

Donde

$$m = m_r + 2(m_w + \frac{1}{r^2} I_w 2)I = I_r + 2I_w + (m_w + \frac{1}{r^2} I_w 2) \frac{T^2}{2} + m_r c \quad (9.21)$$

La simulación de estas ecuaciones dinámicas se pueden encontrar en el Anexo C página 612.

## 9.2. DINÁMICA DE ROBOTS MÓVILES NO HOLONOMICOS

En general, un sistema no holonomico de robot móvil que tiene una n configuración dimensional con coordenadas generalizadas  $q = [q_1, \dots, q_n]^T$  y sujeto a  $(n - m)$  restricciones puede ser descrito por:

$$M(q)\dot{q} + C(q, \dot{q})\dot{q} + G(q) = B(q)\tau + f + \tau_d \quad (9.22)$$

Donde  $M(q) \in R^{n \times n}$  es la matriz de inercia y  $M(q)T = M(q) > 0$ ,  $C(q, \dot{q}) \in R^{n \times n}$  es la matriz centrípeta y de Coriolis,  $G(q) \in R^n$  es el vector de fuerza gravitación,

$B(q) \in R^{n \times r}$  es la matriz transformación de entrada de rango completo y se supone que se conocen, ya que es una función de la geometría fija del sistema,  $\tau \in R^r$  es el vector de entrada de fuerzas y momentos,  $f \in R^n$  es el vector de fuerza restringida, y  $\tau_d \in R^n$  denota perturbaciones acotadas desconocidas incluyendo dinámica no estructurada y no modelada. El sistema dinámico 9.22 tiene las siguientes propiedades:

**Propiedad 1.** Las matrices  $M(q), G(q)$  son uniformemente acotadas y continuas si  $q$  es uniformemente acotada y continua, respectivamente. La matriz  $C(q, \dot{q})$  es uniformemente acotada y continua si  $\dot{q}$  es uniformemente acotada y continua.

**Propiedad 2.** La matriz  $\dot{M} - 2C$  es antisimétrica, es decir,  $X^T(\dot{M} - 2C)x = 0, \forall x \neq 0$ .

Cuando el sistema está sometido a restricciones no holonomicas, las  $(n - m)$  velocidades independientes y no integrables se pueden expresar como:

$$J(q)\dot{q} = 0 \quad (9.23)$$

Donde  $J(q) \in R^{(n-m)n}$  es la matriz asociada con la restricción. La restricción 9.23 se conoce como la restricción no holonomica clásica cuando no es integrable. En esta sección, la restricción 9.23 se supone que es completamente no holonomica y se conoce con exactitud. El efecto de las restricciones puede ser visto como una restricción de la dinámica en el colector  $\Omega_{nh}$  como

$$\Omega_{nh} = \{(q, \dot{q}) | J(q)\dot{q} = 0\}$$

Se hace notar que, dado que la restricción no holonomica 9.23 es no integrable, no hay ninguna restricción explícita en los valores de las variables de configuración. En base a la restricción no holonomica 9.23, las fuerzas de restricción generalizadas en el sistema mecánico 9.22 pueden ser dadas por

$$f = J^T(q)\lambda \quad (9.24)$$

Donde  $\lambda \in R^{(n-m)}$  se conoce como fuerza de rozamiento en el punto de contacto entre el cuerpo rígido y las superficies del entorno. Dado que  $J(q) \in R^{(n-m)n}$  siempre es posible encontrar una matriz de rango  $m$   $R(q) \in R^{(n \times m)}$  formada por un conjunto de campos de vectores suaves y linealmente independientes que abarcan el espacio nulo de  $J(q)$ , es decir,

$$R^T(q)J^T(q) = 0 \quad (9.25)$$

Donde  $R(q) = [r_1(q), \dots, r_m(q)]$  y define una función auxiliar de tiempo  $\dot{z}(t) = [\dot{z}_1(t), \dots, \dot{z}_m(t)]^T \in R^m$  de manera que

$$\dot{q} = R(q)\dot{z}(t) = r_1(q)\dot{z}_1(t) + \dots + r_m(q)\dot{z}_m(t) \quad (9.26)$$

La ecuación 9.26 es el también llamado modelo cinemático de los sistemas de no holonomicos en la literatura. Por lo general,  $\dot{z}(t)$  tiene significado físico, que consiste en la velocidad lineal  $v$  y la velocidad angular  $w$ , es decir,  $\dot{z}(t) = [w \ v]^T$ . La ecuación 9.26 describe la relación cinemática entre el vector de movimiento  $q(t)$  y el vector de velocidad  $\dot{z}(t)$ .

Derivando 9.26 se obtiene:

$$\dot{q} = \dot{R}(q)\dot{z} + R(q)\dot{z} \quad (9.27)$$

De 9.26  $\dot{z}$  se puede obtener desde  $q$  y  $\dot{q}$  así:

$$\dot{z} = [R^T(q)R(q)]^{(-1)} R^T(q)\dot{q} \quad (9.28)$$

La ecuación dinámica 9.22, que satisface la restricción no holonomica 9.23, se puede re escribir en términos de la variable de estado interna  $\dot{z}$  como:

$$M(q)R(q)\ddot{z} + [M(q)\dot{R}(q) + C(q, \dot{q})R(q)]\dot{z} + G(q) = B(q)\tau + J^T(q)\lambda + \tau_d \quad (9.29)$$

Sustituyendo 9.26 y 9.27 en 9.22 y, a continuación pre multiplicando 9.22 por  $R^T(q)$ , la matriz de restricción  $J^T(q)\lambda$  puede ser eliminada en virtud de 9.25. Como consecuencia de ello, se tiene el sistema no holonómico transformado:

$$\dot{q} = R(q)\dot{z} = r_1(q)\dot{z}_1 + \dots + r_m(q)\dot{z}_m \quad (9.30)$$

$$M_1(q)\ddot{z} + C_1(q, \dot{q})\dot{z} + G_1(q) = B_1(q)\tau + \tau_{d1} \quad (9.31)$$

Donde:

$$\begin{aligned} M_1(q) &= R^T M(q) R \\ C_1(q, \dot{q}) &= R^T [M(q)\dot{R} + C(q, \dot{q})R] \\ G_1(q) &= R^T G(q) \\ B_1(q) &= R^T B(q) \\ \tau_{d1} &= R^T \tau_d \end{aligned}$$

Que es más apropiado para el diseño del controlador ya que la restricción  $\lambda$  ha sido eliminada de la ecuación dinámica. Aprovechando la estructura de la ecuación dinámica 9.31, algunas propiedades se muestran a continuación.

**Propiedad 3.** La matriz  $D_1(q)$  es simétrica y definida positiva.

**Propiedad 4.** La matriz  $\dot{D}_1(q) - 2C_1(q, \dot{q})$  es antisimétrica.

**Propiedad 5.**  $D(q), G(q), J(q)$  y  $R(q)$  son limitadas y continuas si  $z$  es acotada y uniformemente continua.  $C(q, \dot{q})$  y  $\dot{R}(q)$  están limitados si  $\dot{z}$  está limitada.  $C(q, \dot{q})$  y  $\dot{R}(q)$  son uniformemente continuas si  $\dot{z}$  es uniformemente continua.

El subsistema no holonómico cinemático 9.26 se convierte en la forma canónica encadenada. El sistema encadenado no holonómico en cuenta en esta sección es la  $m$ -entrada,  $(m-1)$ -cadena, forma encadenada de generador único dada por Walsh y Bushnell.

$$\dot{x}_1 = u_1 \dot{x}_{j,i} = u_1 x_{j,i+1} (2 \leq i \leq n_j - 1) (1 \leq j \leq m - 1) \dot{x}_{j,n_j} = u_{j+1} \quad (9.32)$$

Note que en la ecuación 9.32  $X = [x_1, X_2, \dots, X_m]^T \in R^n$  con  $X_j = [X_{j-1,2}, X_{j-1}, n_{j-1}]$  ( $2 \leq j \leq m$ ) son los estados y  $u = [u_1, u_2, \dots, u_m]^T$  son las entradas del subsistema cinemático. La clase de sistemas no holonómicos en forma encadenada se introdujo primero y ha sido estudiado como un ejemplo de referencia en la literatura. Es la forma canónica más importante que se utiliza comúnmente en el estudio de los sistemas de control no holonómicos. Retos teóricos e intereses prácticos han proporcionado una motivación importante para el estudio amplio de los sistemas no holonómicos en forma encadenada. Se plantea la siguiente hipótesis.

**Supuesto 1.** El modelo cinemático del sistema no holonómico dada por 9.26 se puede convertir en la forma encadenada 9.32 por alguna transformación difeomorfa de coordenadas  $X = T_1(q)$  y realimentación del estado  $v = T_2(q)u$  donde  $u$  es un nuevo control de entrada.

**Proposición 1.** Sea el sistema no holonómico sin derivar

$$\dot{q} = r_1(q)\dot{z}_1 + \dots + r_m(q)\dot{z}_m$$

Donde  $r_i(q)$  son campos fluidos, de entrada de vectores linealmente independientes. Existe transformación del Estado  $X = T_1(q)$  y la retroalimentación  $\dot{z} = T_2(q)u$  en algún conjunto abierto  $U \subset R^n$  para transformar el sistema en una cadena  $(m-1)$ , forma encadenada de generador único, si y sólo si existe una base  $f_1, \dots, f_m$  para  $\Delta_0 = \text{span}\{r_1, \dots, r_m\}$  la cual tiene la forma:

$$f_1 = \left(\frac{\delta}{\delta q_1}\right) + \sum_{i=2}^n f_1^i(q) \frac{\delta}{\delta q_i}$$

$$f_j = \sum_{i=1}^n f_j^i(q) \frac{\delta}{\delta q_i}, \quad 2 \leq j \leq m$$

De tal manera que las distribuciones

$$G_j = \text{span}\{ad_{f_1}^i f_2, \dots, ad_{f_1}^i f_m : 0 \leq i \leq j\}, \quad 0 \leq j \leq n - 1$$

Tienen una dimensión constante en  $U$  y son todas involutivas, y  $G(n-1)$  tiene dimensión  $n-1$  en  $U$ . Para un sistema controlable de dos entradas, un método constructivo se reproduce y se muestra completo aquí. Considerar

$$\dot{q} = r_1(q)\dot{z}_1 + r_2(q)\dot{z}_2 \quad (9.33)$$

Donde  $r_1(q), r_2(q)$  son linealmente independientes y fluidos,  $q \in R^n$  y  $\dot{z} = [\dot{z}_1, \dot{z}_2]^T$  definen:

$$\begin{aligned} \Delta_0 &:= \text{span}\{r_1, r_2, ad_{r_1}r_2 \dots, ad_{r_1}^{n-2}r_2\} \\ \Delta_1 &:= \text{span}\{r_2, ad_{r_1}r_2 \dots, ad_{r_1}^{n-2}r_2\} \\ \Delta_2 &:= \text{span}\{r_2, ad_{r_1}r_2 \dots, ad_{r_1}^{n-3}r_2\} \end{aligned}$$

Si,  $\Delta_0(q) = R^n, \forall q \in U$ , (donde  $U$  es un conjunto abierto de  $R^n$ ),  $\Delta_1$  y  $\Delta_2$  son involutivas en  $U$  y  $r_1(q)$  satisface  $[r_1, \Delta_1] \subset \Delta_1$ , entonces existen dos funciones independientes  $h_1 : U \rightarrow R$  y  $h_2 : U \rightarrow R$  que satisfacen las siguientes relaciones:

$$\begin{aligned} dh_1 \cdot \Delta_1 &= 0 & dh_1 \cdot r_1 &= 1 \\ dh_2 \cdot \Delta_2 &= 1, & dh_2 \cdot ad_{r_1}^{n-2}r_2 &\neq 0 \end{aligned}$$

Hacemos que  $T_1(q) : q \rightarrow X$  así

$$\begin{aligned} x_1 &= h_1 \\ x_2 &= L_{r_1}^{n-2}h_2 \\ x_{n-1} &= L_{r_1}h_2 \\ x_n &= h_2 \end{aligned}$$

Se puede verificar que  $T_1(q)$  es un cambio válido de coordenadas evaluando el jacobiano de  $T_1(q)$  en el origen. Desde que  $L_{r_2}L_{r_1}^{n-2}h_2 \neq 0$ , se hace que  $T_2(q) : \dot{z} \rightarrow u$  así:

$$\begin{aligned} \dot{z}_1 &:= u_1 \\ \dot{z}_2 &:= \frac{1}{L_{r_2}L_{r_1}^{n-2}h_2} [u_2 - (L_{r_1}^{n-1}h_2)u_1] \end{aligned}$$

Luego, la transformación de coordenadas local  $X = T_1(q)$  y la realimentación de estado  $\dot{z} = T_2(q)u$  traducen el sistema 9.33 en la forma encadenada

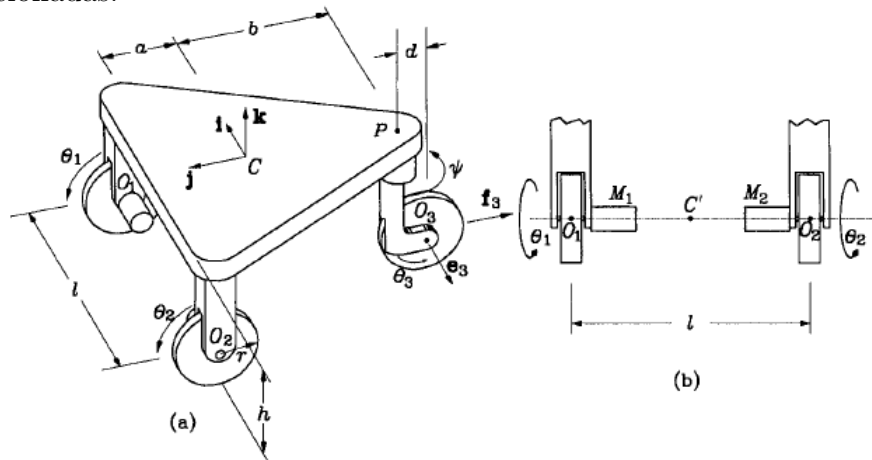
$$\begin{aligned} \dot{x}_1 &= u_1 \\ \dot{x}_2 &= u_2 \\ \dot{x}_3 &= x_2u_1 \\ \dot{x}_n &= x_{n-1}u_1 \end{aligned}$$

**Observación 1.** Bajo ciertas condiciones que se ha señalado en la Proposición 1, el modelo cinemático 9.26 se puede convertir en una forma encadenada impulsado por los integradores.

## 9.3. DINÁMICA DE ROBOTS RODANTES

La dinámica de los robots rodantes, de forma similar a la de otros sistemas mecánicos robóticos, comprende dos problemas principales, la dinámica inversa y directa. Vamos a estudiar ambos usando el mismo modelo matemático. Por lo tanto, la tarea principal es deducir este modelo. Se estudiarán los robots con ruedas convencionales y omnidireccionales. De estos últimos, nos centraremos en los robots con ruedas Mecanum.

Figura 9.5: Un robot rodante de 2 DOF: (a) Su disposición general, y (b) Detalle de las ruedas accionadas.



Fuente: Fundamentals of Robotic Mechanical Systems Theory, Methods, and Algorithms Third Edition, Jorge Angeles, 2007.

### 9.3.1. Robots con ruedas convencionales

Se estudia aquí el robot de la figura 9.5, bajo la suposición de que se manipula mediante motores colocados en los ejes de las dos ruedas coaxiales, indicados como  $M_1$  y  $M_2$  en la figura 9.5. Nuestro enfoque será uno de dinámica de multicuerpo, por esta razón, se distinguen cinco cuerpos rígidos que componen el sistema mecánico robótico en mano. Estas son las tres ruedas (dos ruedas accionadas y una loca), el soporte que lleva la rueda loca, y la plataforma. Se etiquetan estos cuerpos con números a partir del 1 a 5, en el orden anterior, mientras se nota que los cuerpos 4 y 5, el soporte y la plataforma, se someten a movimiento plano, y por lo tanto, merecen un tratamiento especial.

Las matrices de masa 6 x 6 de los tres primeros cuerpos están etiquetadas  $M_1$  a  $M_3$ , con un etiquetado similar para sus correspondientes torsiones de 6 dimensiones, los otros elementos para los cuerpos 4 y 5 se designan por  $M_4, M_5, t'_4$  y  $t'_5$ , los primeros indican las matrices de masa 3 x 3 - a diferencia de la 6 x 6 en el caso general - y las matrices de torsión de 3 dimensiones - a diferencia de las 6 dimensiones en el caso general-. Se lleva a cabo una formulación del modelo matemático del sistema mecánico en estudio, que es de la forma general de la ecuación 9.34 derivados para sistemas holonómicos. La no holonomía del sistema trae características especiales que se destacan en las siguientes deducciones.

$$I\ddot{\theta}_a + C\dot{\theta}_a = \tau + \delta + \gamma + J^T + w^W \quad (9.34)$$

Como primer paso en nuestra formulación, distinguimos entre variables de articulación accionadas y no accionadas, agrupadas en los vectores  $\theta_a$  y  $\theta_u$ , respectivamente, sus derivadas respecto al tiempo son las velocidades de articulación accionadas y no accionadas,  $\dot{\theta}_a$  y  $\dot{\theta}_u$  respectivamente. Desde el análisis cinemático de este sistema en el estudio cinemático visto anteriormente, es evidente que los anteriores vectores son todos de 2 dimensiones,

$$\theta_a = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}, \quad \theta_u = \begin{bmatrix} \theta_3 \\ \psi \end{bmatrix} \quad (9.35)$$

Además, ponemos a derivar expresiones para los giros de los cinco cuerpos que se mueven en función de las tasas de participación, es decir accionados, escribimos esos giros como transformaciones lineales  $\dot{\theta} - a$ ,

$$t_i = T_i \dot{\theta}_a, \quad i = 1, 2, 3 \quad (9.36a)$$

$$t'_i = T'_i \dot{\theta}_a, \quad i = 4, 5 \quad (9.36b)$$

Donde, a partir de las ecuaciones. (8.73a y b), (8.75a y b), y 8.80 vistas en la anteriormente en la sección de cinemática.

$$T_1 = \begin{bmatrix} -i + \rho\delta k & -\rho\delta k \\ rj & 0 \end{bmatrix} \quad (9.37)$$

$$T_2 = \begin{bmatrix} \rho\delta k & -i + \rho\delta k \\ 0 & rj \end{bmatrix} \quad (9.38)$$

$$T_3 = \begin{bmatrix} \theta_3 \\ G_3 \end{bmatrix} \quad (9.39)$$

$$T'_4 = \begin{bmatrix} \theta_4^T \\ G_4 \end{bmatrix} \quad (9.40)$$

$$T'_5 = \begin{bmatrix} \rho\delta & -\rho\delta \\ r(\lambda i + (\frac{1}{2})j) & r(-\lambda i + (\frac{1}{2})j) \end{bmatrix} \equiv \begin{bmatrix} \theta_5^T \\ G_5 \end{bmatrix} \quad (9.41)$$

Con  $\theta_3, G_3, \theta_4$  y  $G_4$  aún no deducidas. En la secuela, encontramos conveniente trabajar con unos pocos parámetros adimensionales,  $\alpha, \delta, \rho$  ya definidos anteriormente en la ecuación 8.85a y b y  $\lambda$ , que se introduce ahora, y se muestra a continuación con los tres primeros parámetros para una referencia rápida:

$$\alpha \equiv (a+b)/l, \quad \delta \equiv d/l, \quad \rho \equiv r/d, \quad \lambda \equiv a/l \quad (9.42)$$

En las deducciones abajo, recurrimos a la notación introducida en la subsección 8.8.1. En primer lugar, observamos que, a partir de las ecuaciones 8.76a y ( 8.84a y, b), Se puede escribir, con  $\theta_{ij}$  que denota la entrada  $(i, j)$  de  $\theta$ , como se dedujo de la subsección 8.8.1,

$$w_3 = (\theta_{11}\dot{\theta}_1 + \theta_{12}\dot{\theta}_2)e_3 + [\rho\delta(\dot{\theta}_1 - \dot{\theta}_2) + \theta_{21}\dot{\theta}_1 + \theta_{22}\dot{\theta}_2]k \quad (9.43)$$

o

$$w_3 = \theta_3\dot{\theta}_a \quad (9.44)$$

Con  $\theta_3$  definida como:

$$\theta_3 = [(\theta_{11}e_3) + (\theta_{21} + \rho\delta)k \quad \theta_{12}e_3 + (\theta_{22} - \rho\delta)k]$$

En una forma más compacta:

$$\theta_3 = [(\theta_{11}e_3 + \bar{\theta}_{21}k \quad \theta_{12}e_3 + \bar{\theta}_{22}k)] \quad (9.45a)$$

Con  $\bar{\theta}_{21}$  y  $\bar{\theta}_{22}$  definido, a su vez, como

$$\bar{\theta}_{21} \equiv \theta_{21} + \rho\delta, \quad \bar{\theta}_{22} \equiv \theta_{22} - \rho\delta \quad (9.45b)$$

Además,

$$\dot{c}_3 = -r\dot{\theta}_3f_3 = -r(\theta_{11}\dot{\theta}_1 + \theta_{12}\dot{\theta}_2)f_3$$

Y por lo tanto,

$$G_3 = r [-\theta_{11}f_3 \quad -\theta_{12}f_3] \quad (9.46)$$

Además, es evidente a partir de la figura 9.5 que la velocidad angular escalar el soporte,  $w_4$ , está dada por:

$$w_4 = w + \dot{\psi}$$

Y por lo tanto,

$$w_4 = \rho\delta(\dot{\theta}_1 - \dot{\theta}_2) + \theta_{21}\dot{\theta}_1 + \theta_{22}\dot{\theta}_2 = \bar{\theta}_{21}\dot{\theta}_1 + \bar{\theta}_{22}\dot{\theta}_2$$

Por lo tanto, se puede escribir

$$w_4 = \theta_4^T \dot{\theta}_a \quad (9.47a)$$

Donde  $\theta_4$  se define como:

$$\theta_4 \equiv [\bar{\theta}_{21} \quad \bar{\theta}_{22}]^T \quad (9.47b)$$

Ahora, puesto que se dan las propiedades inerciales del soporte en coordenadas de soporte, tiene sentido expresar  $\dot{c}_4$  en esas coordenadas, teniendo en cuenta que el punto  $C_4$  se encuentra en el medio de la línea  $(\bar{P}O)_3$ . Tal expresión se obtiene a continuación:

$$\dot{c}_4 = \dot{o}_3 + w_4 \times \frac{1}{2} [-df_3 + (h-r)k] = -r\dot{\theta}_3 f_3 \frac{d}{2} (w + \dot{\psi}) e_3$$

Al expresar  $\dot{\theta}_3$  y  $\dot{\psi}$  en términos de  $\dot{\theta}_1$  y  $\dot{\theta}_2$ , obtenemos

$$\dot{c}_4 = d \left( \frac{1}{2} \bar{\theta}_{21} e_3 - \rho \theta_{11} f_3 \right) \dot{\theta}_1 + d \left( \frac{1}{2} \bar{\theta}_{22} e_3 - \rho \theta_{12} f_3 \right) \dot{\theta}_2 \quad (9.48)$$

Donde es evidente que

$$G_4 = d \left[ \begin{array}{cc} \left( \frac{1}{2} \right) \bar{\theta}_{21} e_3 - \rho \theta_{11} f_3 & \bar{\theta}_{22} e_3 - \rho \theta_{12} f_3 \end{array} \right] \quad (9.49)$$

Por lo tanto,

$$T_4' = \left[ \begin{array}{cc} \bar{\theta}_{21} & \bar{\theta}_{22} \\ d \left[ \left( \frac{1}{2} \right) \bar{\theta}_{21} e_3 - \rho \theta_{11} f_3 \right] & d \left[ \left( \frac{1}{2} \right) \bar{\theta}_{22} e_3 - \rho \theta_{12} f_3 \right] \end{array} \right] \quad (9.50)$$

Completando así todas las matrices de formación de torsión necesarias. La matriz 2 x 2 de inercia generalizada,  $I(\theta)$  se obtiene ahora. Aquí hemos escrito esta matriz como una función de todas las variables, independientes y dependientes, dispuestas en el vector de 4 dimensiones  $\theta$ , porque no se puede obtener una expresión para  $\theta_u$  en términos de  $\theta_a$ , dada la no holonomía del sistema en cuestión. Por lo tanto, I es, en general, una función de  $\theta_1, \theta_2, \theta_3$ , y  $\psi$ . Para estar seguros, a partir de las expresiones anteriores para las matrices de conformación de torsión  $T_i$  y  $T_i'$ , es evidente que dicha matriz de inercia es una función explícita solamente de  $\psi$ , su dependencia de  $\theta_1$  y  $\theta_2$  está implícitamente dada a través de los vectores  $e_3$  y  $f_3$ . Se deduce la expresión buscada para I a partir de la energía cinética, es decir:

$$T = \sum_1^3 \frac{1}{2} t_i^T M_i t_i + \frac{1}{2} \sum_4^5 (t'_i)^T M'_i t'_i$$

O

$$T = \frac{1}{2} \dot{\theta}_\alpha^T \left( \sum_1^3 T_i^T M_i T_i \right) \dot{\theta}_\alpha + \frac{1}{2} \dot{\theta}_\alpha^T \left( \sum_4^5 (T'_i)^T M'_i T'_i \right) \dot{\theta}_\alpha \quad (9.51)$$

Y por lo tanto

$$I = \sum_1^3 T_i^T M_i T_i + \sum_4^5 (T'_i)^T M'_i T'_i \quad (9.52)$$

Con el fin de expandir la expresión anterior,  $J_w$  y  $J_c$  son las matrices 3 x 3 de inercia de las dos ruedas accionadas y la rueda loca, respectivamente, los momentos de inercia escalares del soporte y la plataforma, que se someten a movimiento plano, denotados por  $I_b$  e  $I_p$ . Del mismo modo, dejamos que  $m_w$ ,  $m_b$ ,  $m_c$ , y  $m_p$ , denoten las masas de los cuerpos correspondientes. Por lo tanto,

$$M_1 = \begin{bmatrix} J_w & 0 \\ 0 & m_w 1_3 \end{bmatrix} = M_2$$

$$M_3 = \begin{bmatrix} J_c & 0 \\ 0 & m_c 1_3 \end{bmatrix}$$

$$M'_4 = \begin{bmatrix} I_b & 0^T \\ 0 & m_b 1_2 \end{bmatrix}$$

$$M'_5 = \begin{bmatrix} I_p & 0^T \\ 0 & m_p 1_2 \end{bmatrix}$$

Con  $O$  y  $1_3$  que denota las matrices 3 x 3 cero e identidad, mientras que  $0$  y  $I_2$  el vector cero de 2 dimensiones y la matriz 2 x 2 identidad. Por otra parte, en el supuesto de que las ruedas accionadas están balanceadas dinámicamente, tenemos

$$J_w = \begin{bmatrix} I & 0 & 0 \\ 0 & H & 0 \\ 0 & 0 & H \end{bmatrix}$$

Además, suponemos que la rueda loca puede ser modelada como un disco rígido de material uniforme de masa dada  $m_c$  y radio  $r$ , y por lo tanto, en coordenadas de soporte fijo  $\{e_3, f_3, k\}$ ,

$$J_c = \frac{1}{4} m_c r^2 \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Ahora es una cuestión simple para calcular

$$T_1^T M_1 T_1 = \begin{bmatrix} I + (\rho\delta)^2 H + m_w r^2 & -(\rho\delta)^2 H \\ -(\rho\delta)^2 H & (\rho\delta)^2 H \end{bmatrix}$$

$$T_2^T M_2 T_2 = \begin{bmatrix} (\rho\delta)^2 H & -(\rho\delta)^2 H \\ -(\rho\delta)^2 H & I + (\rho\delta)^2 H + m_w r^2 \end{bmatrix}$$

Donde la simetría entre las dos expresiones anteriores, es de destacar: es decir, la segunda expresión se deduce si las entradas diagonales de la primera expresión se intercambian, lo que es físicamente plausible, porque tal intercambio es equivalente a un reetiquetado de las dos ruedas. El cálculo de los productos restantes es menos sencillo, pero se puede conseguir fácilmente. De las expresiones para  $T_3$  y  $M_3$  tenemos

$$T_3^T M_3 T_3 = \begin{bmatrix} \theta_3^T & G_3^T \end{bmatrix} \begin{bmatrix} J_c & 0 \\ 0 & m_c 1_3 \end{bmatrix} \begin{bmatrix} \theta_3 \\ G_3 \end{bmatrix} = \theta_3^T J_c \theta_3 + m_c G_3^T G_3$$

Para el cálculo de los productos anteriores, se escribe  $J_c$  y  $\theta_3$  en forma de componentes, es decir,

$$J_c \theta_3 = \frac{1}{4} m_c r^2 \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \theta_{11} & \theta_{12} \\ 0 & 0 \\ \bar{\theta}_{21} & \bar{\theta}_{22} \end{bmatrix} = \frac{1}{4} m_c r^2 \begin{bmatrix} 2\theta_{11} & 2\theta_{12} \\ 0 & 0 \\ \bar{\theta}_{21} & \bar{\theta}_{22} \end{bmatrix}$$

Y por lo tanto,

$$\theta_3^T J_c \theta_3 = \frac{1}{4} m_c r^2 \begin{bmatrix} 2\theta_{11}^2 + \bar{\theta}_{21}^2 & 2\theta_{11}\theta_{12} + \bar{\theta}_{21}\bar{\theta}_{22} \\ 2\theta_{11}\theta_{12} + \bar{\theta}_{21}\bar{\theta}_{22} & 2\theta_{12}^2 + \bar{\theta}_{22}^2 \end{bmatrix}$$

Del mismo modo,

$$m_c G_3^T G_3 = m_c r^2 \begin{bmatrix} \theta_{11}^2 & \theta_{11}\theta_{12} \\ \theta_{11}\theta_{12} & \theta_{12}^2 \end{bmatrix}$$

Además,

$$(T_4')^T M_4' T_4' = \begin{bmatrix} \theta_4^T & G_4^T \end{bmatrix} \begin{bmatrix} I_b & 0^T \\ 0 & m_b 1_2 \end{bmatrix} \begin{bmatrix} \theta_4^T \\ G_4 \end{bmatrix} = I_b \theta_4^T \theta_4 + m_b G_4^T G_4$$

Tras la expansión, tenemos

$$(T_4')^T M_4' T_4' I_b = \begin{bmatrix} \bar{\theta}_{21}^2 & \bar{\theta}_{21}\bar{\theta}_{22} \\ \bar{\theta}_{21}\bar{\theta}_{22} & \bar{\theta}_{22}^2 \end{bmatrix} + \frac{1}{4} m_b d^2 \begin{bmatrix} \bar{\theta}_{21}^2 + 4\rho^2 \bar{\theta}_{11}^2 & \bar{\theta}_{21}\bar{\theta}_{22} + 4\rho^2 \theta_{11}\theta_{12} \\ \bar{\theta}_{21}\bar{\theta}_{22} + 4\rho^2 \theta_{11}\theta_{12} & \bar{\theta}_{22}^2 + 4\rho^2 \theta_{12}^2 \end{bmatrix}$$

Finalmente,

$$(T'_5)^T M'_5 T'_5 = [\theta_5^T \quad + G_5^T] \begin{bmatrix} I_p & 0^T \\ 0 & m_p 1_2 \end{bmatrix} \begin{bmatrix} \theta_5^T \\ G_5 \end{bmatrix} = I_p \theta_5 \theta_5^T + m_b G_5^T G_5$$

Que puede ser fácilmente expandido como

$$(T'_5)^T M'_5 T'_5 = I_p (\rho\delta)^2 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} + m_p r^2 \begin{bmatrix} (\frac{1}{4}) + \lambda^2 & (\frac{1}{4}) - \lambda^2 \\ (\frac{1}{4}) - \lambda^2 & (\frac{1}{4}) + \lambda^2 \end{bmatrix}$$

De este modo podemos expresar la matriz de inercia generalizada como

$$I = I_w + I_c + I_b + I_p$$

Donde  $I_w$ ,  $I_c$ ,  $I_b$ , e  $I_p$  denotan las contribuciones de las ruedas accionadas, la rueda loca, el soporte y la plataforma, respectivamente, es decir,

$$I_w = \sum_1^2 T_i^T M_i T_i = \begin{bmatrix} I + 2(\rho\delta)^2 H + m_w r^2 & -2(\rho\delta)^2 H \\ -2(\rho\delta)^2 H & I + 2(\rho\delta)^2 H + m_w r^2 \end{bmatrix}$$

$$I_c = \frac{m_c r^2}{4} \begin{bmatrix} 6\theta_{11}^2 + \bar{\theta}_{21}^2 & 6\theta_{11}\theta_{12} + \bar{\theta}_{21}\bar{\theta}_{22} \\ 6\theta_{11}\theta_{12} + \bar{\theta}_{21}\bar{\theta}_{22} & 6\theta_{12}^2 + \bar{\theta}_{22}^2 \end{bmatrix}$$

$$I_b = I_b \begin{bmatrix} \bar{\theta}_{21}^2 & \bar{\theta}_{21}\bar{\theta}_{22} \\ \bar{\theta}_{21}\bar{\theta}_{22} & \bar{\theta}_{22}^2 \end{bmatrix} + (m_b d^2)/4 \begin{bmatrix} \bar{\theta}_{21}^2 + 4\rho^2 \bar{\theta}_{11}^2 & \bar{\theta}_{21}\bar{\theta}_{22} + 4\rho^2 \theta_{11}\theta_{12} \\ \bar{\theta}_{21}\bar{\theta}_{22} + 4\rho^2 \theta_{11}\theta_{12} & \bar{\theta}_{22}^2 + 4\rho^2 \theta_{12}^2 \end{bmatrix}$$

$$I_p = I_p (\rho\delta)^2 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} + m_p r^2 \begin{bmatrix} (\frac{1}{4}) + \lambda^2 & (\frac{1}{4}) - \lambda^2 \\ (\frac{1}{4}) - \lambda^2 & (\frac{1}{4}) + \lambda^2 \end{bmatrix}$$

Ahora es evidente que las contribuciones de las ruedas accionadas y la plataforma son constantes, mientras que las de la rueda loca y el soporte dependen de la configuración. Por lo tanto, sólo el último contribuye a las fuerzas generalizadas de Coriolis y centrífugas. Tenemos así

$$T^T M \dot{T} = T_3^T M_3 \dot{T}_3 + (T_4)^T M_4 (\dot{T}_4)$$

De la expresión para  $T_3^T M_3 T_3$ , obtenemos

$$T_3^T M_3 \dot{T}_3 = \theta_3^T J_c \dot{\theta}_3 + m_3 C_3^T \dot{C}_3$$

De las derivadas de tiempo se muestran a continuación:

$$\dot{\theta}_3 = \left[ \dot{\theta}_{11} e_3 + \theta_{11} w_4 f_3 + \dot{\theta}_{21} k \quad \dot{\theta}_{21} e_3 + \theta_{12} w_4 f_3 + \dot{\theta}_{22} k \right]$$

$$\dot{C}_3 = r \begin{bmatrix} -\dot{\theta}_1 f_3 + \theta_{11} w_4 e_3 & -\dot{\theta}_{12} f_3 + \theta_{12} w_4 e_3 \end{bmatrix}$$

Con las derivadas de tiempo de las entradas de  $\Theta$  se dan como

$$\dot{\theta} = \dot{\psi} \begin{bmatrix} -\alpha \sin \psi + \frac{(\cos \psi)}{2} & \alpha \sin \psi + \frac{(\cos \psi)}{2} \\ \rho \left[ -\alpha \cos \psi - \frac{(\sin \psi)}{2} \right] & \rho \left[ \alpha \cos \psi - \frac{(\sin \psi)}{2} \right] \end{bmatrix} \quad (9.53)$$

Sus parámetros se definen en la ecuación 9.42. Tras la expansión, los productos incluidos en la expresión para  $T_3^T M_3 \dot{T}_3$  se convierte en

$$\theta_3^T J_c \dot{\theta}_3 = \frac{m_c r^2}{4} \begin{bmatrix} 2\theta_{11} \dot{\theta}_{11} + \bar{\theta}_{21} \dot{\theta}_{21} & \theta_{11} \dot{\theta}_{12} + \bar{\theta}_{21} \dot{\theta}_{22} \\ 2\theta_{12} \dot{\theta}_{11} + \bar{\theta}_{22} \dot{\theta}_{21} & 2\theta_{12} \dot{\theta}_{12} + \bar{\theta}_{22} \dot{\theta}_{22} \end{bmatrix}$$

$$m_3 C_3^T \dot{C}_3 = m_c r^2 \begin{bmatrix} \theta_{11} \theta_{11} & \theta_{11} \theta_{12} \\ \theta_{12} \dot{\theta}_{11} & \theta_{12} \dot{\theta}_{12} \end{bmatrix}$$

Por consiguiente,

$$T_3^T M_3 \dot{T}_3 = \frac{m_c r^2}{4} \begin{bmatrix} 6\theta_{11} \dot{\theta}_{11} + \bar{\theta}_{21} \dot{\theta}_{21} & 6\theta_{11} \dot{\theta}_{12} + \bar{\theta}_{21} \dot{\theta}_{22} \\ 6\theta_{12} \dot{\theta}_{11} + \bar{\theta}_{22} \dot{\theta}_{21} & 6\theta_{12} \dot{\theta}_{12} + \bar{\theta}_{22} \dot{\theta}_{22} \end{bmatrix}$$

Del mismo modo,

$$(T_4')^T M_4' \dot{T}_4' = I_b \theta_4 \dot{\theta}_4^T + m_b C_4^T \dot{C}_4$$

Las anteriores derivadas con el tiempo son

$$\dot{\theta}_4^T = \begin{bmatrix} \dot{\theta}_{21} & \dot{\theta}_{22} \end{bmatrix}$$

$$\dot{C}_4 = d [c_{11} e_3 + c_{12} f_3 \quad c_{21} e_3 + c_{22} f_3]$$

Con los coeficientes  $c(i, j)$  dados a continuación:

$$c_{11} = \frac{1}{2} \dot{\theta}_{21} + \rho \theta_{11} w_4, \quad c_{12} = \frac{1}{2} \bar{\theta}_{21} w_4 - \rho \dot{\theta}_{11}$$

$$c_{21} = \frac{1}{2} \dot{\theta}_{22} + \rho \theta_{12} w_4, \quad c_{22} = \frac{1}{2} \bar{\theta}_{22} w_4 - \rho \dot{\theta}_{12}$$

Por lo tanto,

$$I_b \theta_4 \dot{\theta}_4^T = I_b \begin{bmatrix} \theta_{21} \dot{\theta}_{21} & \theta_{21} \dot{\theta}_{22} \\ \theta_{22} \dot{\theta}_{21} & \theta_{22} \dot{\theta}_{22} \end{bmatrix}$$

$$m_b C_4^T \dot{C}_4 = \frac{m_b d^2}{2} \begin{bmatrix} \dot{\theta}_{21} c_{11} - 2\rho \theta_{11} c_{12} & \dot{\theta}_{21} c_{21} - 2\rho \theta_{11} c_{22} \\ \dot{\theta}_{21} c_{11} - 2\rho \theta_{12} c_{12} & \dot{\theta}_{22} c_{21} - 2\rho \theta_{12} c_{22} \end{bmatrix}$$

Por consiguiente,

$$(T_4')^T M_4' \dot{T}_4' = I_b \begin{bmatrix} \theta_{21} \dot{\theta}_{21} & \theta_{21} \dot{\theta}_{22} \\ \theta_{22} \dot{\theta}_{21} & \theta_{22} \dot{\theta}_{22} \end{bmatrix} + \frac{m_b d^2}{2} \begin{bmatrix} \bar{\theta}_{21} c_{11} - 2\rho \theta_{11} c_{12} & \bar{\theta}_{21} c_{21} - 2\rho \theta_{11} c_{22} \\ \bar{\theta}_{21} c_{11} - 2\rho \theta_{12} c_{12} & \bar{\theta}_{22} c_{21} - 2\rho \theta_{12} c_{22} \end{bmatrix}$$

En los pasos finales, calculamos  $T^T W M T$ . Como vimos anteriormente, sólo la rueda loca y el soporte pueden contribuir a este término, por las contribuciones de los otros cuerpos a la matriz de inercia generalizada son constantes. Sin embargo, el soporte se somete a movimiento plano, se anula su contribución a este término. Por lo tanto,

$$T^T W M T = T_3^T W_3 M_3 T_3$$

Tras la expansión del producto anterior, tenemos

$$T_3^T W_3 M_3 T_3 = \begin{bmatrix} \theta_3^T & G_3^T \end{bmatrix} \begin{bmatrix} \Omega_3 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} I_c & 0 \\ 0 & m_c 1_3 \end{bmatrix} \begin{bmatrix} \theta_3 \\ G_3 \end{bmatrix} = \theta_3^T \Omega_3 I_c \theta_3 \quad (9.54)$$

El término anterior se anula, como demostramos a continuación. En primer lugar, se observa que

$$\Omega_3 w_3 = 0$$

Sin embargo, de la ecuación 9.44,

$$w_3 = \theta_3 \dot{\theta}_a$$

Y por lo tanto

$$\Omega_3 \theta_3 \dot{\theta}_a = 0$$

Para cada  $\dot{\theta}_a$ , por lo cual

$$\Omega_3 \theta_3 = O_{32}$$

Con  $O_{32}$  que denota la matriz 3 x 2 cero. Al transponer la expresión anterior, se obtiene

$$\theta_3^T \Omega_3 = O_{23}$$

Donde hemos recordado que  $\Omega_3$  es antisimétrica. La sustitución de la expresión anterior en la ecuación 9.54 demuestra fácilmente que el término en cuestión en realidad se anula, es decir, en resumen, los términos de fuerzas de Coriolis y centrífugas del sistema a mano toman la forma

$$C(\theta, \dot{\theta}_a)\dot{\theta}_a = \frac{m_c r^2}{4} \begin{bmatrix} 6\theta_{11}(\dot{\theta}_{11}\dot{\theta}_1 + \dot{\theta}_{12}\dot{\theta}_2) + \bar{\theta}_{21}(\dot{\theta}_{12}\dot{\theta}_1) + \dot{\theta}_{22}\dot{\theta}_2 \\ 6\theta_{12}(\dot{\theta}_{11}\dot{\theta}_1 + \dot{\theta}_{12}\dot{\theta}_2) + \dot{\theta}_{22}(\dot{\theta}_{12}\dot{\theta}_1 + \dot{\theta}_{22}\dot{\theta}_2) \end{bmatrix} + I_b(\dot{\theta}_{21}\dot{\theta}_1 + \dot{\theta}_{22}\dot{\theta}_2) \begin{bmatrix} \theta_{21} \\ \theta_{22} \end{bmatrix} + \frac{m_b d^2}{2}(c_1\dot{\theta}_1 + c_{21}\dot{\theta}_2) \begin{bmatrix} \bar{\theta}_{21} \\ \bar{\theta}_{22} \end{bmatrix} - m_b d^2 \rho(c_{12}\theta_1 + c_{22}\theta_2) \begin{bmatrix} \theta_{11} \\ \theta_{12} \end{bmatrix}$$

Si se recuerda que los coeficientes  $c_{i,j}$  son lineales en las velocidades de articulación, entonces la expresión anterior muestra claramente la naturaleza cuadrática de los términos de Coriolis y centrífugos con respecto a los velocidades de articulación. La deducción de las fuerzas suministradas por los actuadores es sencilla:

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$$

La fuerza generalizada disipativa es menos sencilla, pero su cálculo no es demasiado largo. De hecho, si se suponen amortiguadores lineales en todas las articulaciones, entonces la función de disipación es:

$$\Delta = \frac{1}{2}c_1\dot{\theta}_1^2 + \frac{1}{2}c_2\dot{\theta}_2^2 + \frac{1}{2}c_3\dot{\theta}_3^2 + \frac{1}{2}c_4\dot{\psi}^2 = \frac{1}{2}\dot{\theta}_a^T C_{12}\dot{\theta}_a + \frac{1}{2}\dot{\theta}_u^T C_{34}\dot{\theta}_u$$

Con  $C_{12}$  y  $C_{34}$  definidos como:

$$C_{12} \equiv \begin{bmatrix} c_1 & 0 \\ 0 & c_2 \end{bmatrix}, \quad C_{34} \equiv \begin{bmatrix} c_3 & 0 \\ 0 & c_4 \end{bmatrix}$$

Ahora bien, si recordamos la expresión de  $\dot{\theta}_u$  en términos de  $\dot{\theta}_a$ , nos encontramos con

$$\Delta = \frac{1}{2}\dot{\theta}_a^T D\dot{\theta}_a$$

D se define, a su vez, como la matriz de amortiguamiento equivalente, dada por

$$D = C_{12} + \theta^T C_{34} \theta$$

Dado que  $\theta = \theta(\psi)$ ,  $D = D(\psi)$ , el modelo dinámico bajo estudio entonces toma la forma

$$I(\theta)\dot{\theta}_a + C(\theta, \dot{\theta}_a)\dot{\theta}_a = \tau - D(\psi)\dot{\theta}_a$$

Con  $I$  y  $C(\theta, \dot{\theta}_a)$  dadas, tal como en el caso de los sistemas holonómicos, como

$$I(\theta) = T^T M T$$

$$C(\theta, \dot{\theta}_a) = T^T M \dot{T} + T^T W M T$$

Completando de esta manera el modelo matemático que rige el movimiento del sistema en cuestión. Nótese aquí que  $\theta$  denota el vector de 4 dimensiones de variables de articulación que contienen los cuatro ángulos que aparecen como componentes de  $\theta_a$  y  $\theta_u$ . Debido a la no holonomía del sistema, una expresión para el último en términos de que el primero no se puede deducir, y por lo tanto todo el vector de 4 dimensiones  $\theta$  se deja como un argumento de ambos  $I$  y  $C$ .

### 9.3.2. Robots con ruedas omnidireccionales

Consideremos ahora un robot con 3 DOF con tres ruedas accionadas del tipo Mecanum, como se muestra en las Figura 8.56, con la configuración de la figura 9.6, que se denominará, en adelante, el arreglo  $\Delta$ . Este sistema se ilustra en la figura 9.7. A continuación se adopta la notación del capítulo de cinemática, con  $\alpha = \pi/2$  y  $n = 3$ . Ahora recordar que la torsión de la plataforma se ha representado en forma plana como se muestra

$$t' = \begin{bmatrix} w \\ \dot{c} \end{bmatrix} \quad (9.55)$$

Donde  $w$  es la velocidad escalar angular de la plataforma y  $\dot{c}$  es el vector de posición de 2 dimensiones de su centro de masa, el cual se supone que coincide con el centroide del conjunto de puntos  $\{C_i\}_1^3$ . Además, las tres ruedas son accionadas, y por lo tanto, el vector de 3 dimensiones de las velocidades de articulación accionadas se define como

$$\dot{\theta}_a = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \quad (9.56)$$

La relación entre  $\dot{\theta}_a$  y  $t'$  se deduce en general en la subsección 8.8.2, tenemos:

$$J \dot{\theta}_a = K t' \quad (9.57a)$$

Con los dos Jacobianos  $3 \times 3$   $J$  y  $K$  definidos como

$$J \equiv -a1 \quad , K \equiv \begin{bmatrix} r & f_1^T \\ r & f_2^T \\ r & f_3^T \end{bmatrix} \quad (9.57b)$$

Donde, se recuerda,  $a$  es la altura del eje del cubo de la rueda y  $r$  es la distancia horizontal de los puntos de contacto con el suelo para el centro de masa  $C$  de la plataforma, como se indica en la figura 9.6a. Por otra parte, los vectores  $\{e_i\}_1^3$  y  $\{f_i\}_1^3$ , definida en la subsección 8.8.2, se muestran en la figura 9.6. A continuación se deducen las expresiones para  $w$  y  $\dot{c}$ , de la ecuación 9.57a, en términos de las velocidades de articulación. Para este fin, se amplían estas tres ecuaciones, obteniendo así

$$rw + f_1^T \dot{c} = -a\dot{\theta}_1 \quad (9.58a)$$

$$rw + f_2^T \dot{c} = -a\dot{\theta}_2 \quad (9.58b)$$

$$rw + f_3^T \dot{c} = -a\dot{\theta}_3 \quad (9.58c)$$

Tras la adición de los lados correspondientes de las tres ecuaciones anteriores, se obtiene

$$3rw + \dot{c}^T \sum_1^3 f_i = -a \sum_1^3 \dot{\theta}_i \quad (9.59)$$

Pero a partir de la figura 9.6b, es evidente que

$$e_1 + e_2 + e_3 = 0 \quad (9.60a)$$

$$f_1 + f_2 + f_3 = 0 \quad (9.60b)$$

Del mismo modo,

$$e_1 = \frac{\sqrt{3}}{3}(f_3 - f_2), \quad e_2 = \frac{\sqrt{3}}{3}(f_1 - f_3), \quad e_3 = \frac{\sqrt{3}}{3}(f_2 - f_1) \quad (9.60c)$$

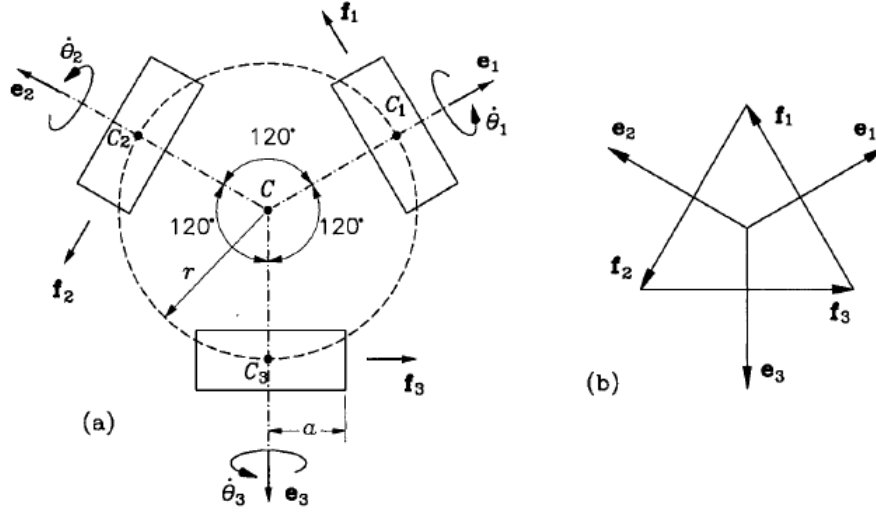
$$f_1 = \frac{\sqrt{3}}{3}(e_2 - e_3), \quad f_2 = \frac{\sqrt{3}}{3}(e_3 - e_1), \quad f_3 = \frac{\sqrt{3}}{3}(e_1 - e_2) \quad (9.60d)$$

Y por lo tanto, la ecuación anterior para  $w$  y  $\dot{c}$  conduce a

$$w = \frac{-a}{3r} \sum_1^3 \dot{\theta}_i \quad (9.61)$$

Ahora se deduce una expresión para  $\dot{c}$  en términos de las velocidades de articulación accionadas. Esto se hace restando, la ecuación 9.58b de la ecuación 9.58a y la ecuación 9.58c de la ecuación 9.58b, obteniendo así un sistema de dos ecuaciones lineales con dos incógnitas, los dos componentes del vector de 2 dimensiones  $\dot{c}$ , es decir,

Figura 9.6: Robot rodante con ODWs en arreglo  $\Delta$ .



Fuente: Fundamentals of Robotic Mechanical Systems Theory, Methods, and Algorithms Third Edition, Jorge Angeles, 2007.

$$A\dot{c} = b$$

Con la matriz  $A$  y el vector  $b$  definidos como

$$A \equiv \begin{bmatrix} (f_1 - f_2)^T \\ (f_2 - f_3)^T \end{bmatrix} \equiv -\sqrt{3} \begin{bmatrix} e_3^T \\ e_1^T \end{bmatrix}, \quad b \equiv -a \begin{bmatrix} \dot{\theta}_1 - \dot{\theta}_2 \\ \dot{\theta}_2 - \dot{\theta}_3 \end{bmatrix}$$

Donde se han utilizado las relaciones 9.60c. Dado que  $A$  es una matriz  $2 \times 2$ , su inversa se puede encontrar fácilmente:

$$\dot{c} = \frac{2}{3}a \begin{bmatrix} -Ee_1 & Ee_3 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 - \dot{\theta}_2 \\ \dot{\theta}_2 - \dot{\theta}_3 \end{bmatrix}$$

Ahora de la figura 9.6b,

$$Ee_1 = f_1, Ee_3 = f_3$$

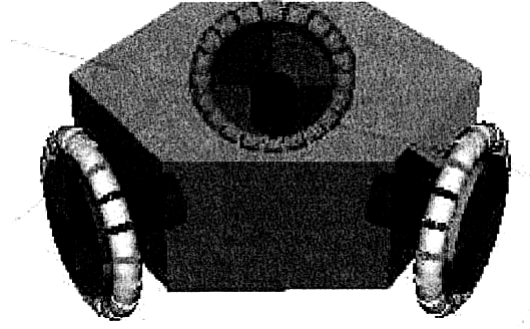
Y, por tanto,  $\dot{c}$  se reduce a

$$\dot{c} = \frac{2}{3}a \left[ (\dot{\theta}_2 - \dot{\theta}_1)f_1 + (\dot{\theta}_2 - \dot{\theta}_3)f_3 \right] \equiv \frac{2}{3}a [\dot{\theta}_2(f_1 + f_3) - \dot{\theta}_1 f_1 - \dot{\theta}_3 f_3]$$

Pero en virtud de la ecuación 9.60b,

$$f_1 + f_3 = -f_2$$

Figura 9.7: Una vista del robot de tres ruedas con ruedas Mekanum en un arreglo  $\Delta$ .



Fuente: Fundamentals of Robotic Mechanical Systems Theory, Methods, and Algorithms Third Edition, Jorge Angeles, 2007.

La expresión anterior para  $(\dot{c})$  se convierte así

$$\dot{c} = -\frac{2}{3}a \sum_1^3 \dot{\theta}_i f_i \quad (9.62)$$

Así,  $w$  es proporcional al valor medio de  $\{\dot{\theta}_i\}_1^3$ , mientras que  $\dot{c}$  es proporcional al valor medio de  $\{\dot{\theta}_i f_i\}_1^3$ . Al obtener el modelo matemático del robot a mano, vamos a recurrir al complemento ortogonal natural, y por lo tanto, se requerirán expresiones para las torsiones de todos los cuerpos implicados en términos de las velocidades de ruedas accionadas. Empezamos por el etiquetado de las ruedas como cuerpos 1, 2 y 3, con la plataforma como cuerpo 4. Además, se desprecia la inercia de los rodillos, y por lo tanto no tienen que ser etiquetados. Además, los cubos de las ruedas se someten a rotaciones con velocidades angulares en dos direcciones ortogonales, y por lo tanto, será requerida una representación total de torsión de 6 dimensiones de éstos. En adelante, se considerará la velocidad angular de la plataforma y la velocidad de su centro de masa como vectores de 3 dimensiones. Por lo tanto,

$$t_4 \equiv T_4 \dot{\theta}_a, \quad T_4 \equiv -\lambda \begin{bmatrix} k & k & k \\ 2rf_1 & 2rf_2 & 2rf_3 \end{bmatrix} \quad (9.63)$$

Con  $\lambda$  definido, a su vez, como la relación

$$\lambda \equiv \frac{a}{3r} \quad (9.64)$$

Ahora, las velocidades angulares de las ruedas se dan simplemente como

$$w_i = \dot{\theta}_i e_i + wk = \dot{\theta}_i e_i - \lambda \left( \sum_1^3 \dot{\theta}_i \right) k \quad (9.65)$$

O

$$w_1 = \dot{\theta}_1(e_1 - \lambda k) - \lambda \dot{\theta}_2 k - \lambda \dot{\theta}_3 k \quad (9.66a)$$

$$w_2 = -\lambda \dot{\theta}_1 k + \dot{\theta}_2(e_2 - \lambda k) - \lambda \dot{\theta}_3 k \quad (9.66b)$$

$$w_3 = -\lambda \dot{\theta}_1 k - \lambda \dot{\theta}_2 k + (e_3 - \lambda k) \dot{\theta}_3 \quad (9.66c)$$

Expresiones similares se obtienen para los vectores  $\dot{c}_i$ . Para este fin, se recurre a la geometría de la figura 9.6, de la cual se deducen las relaciones

$$\dot{c}_i = \dot{c} + wr f_i = -2\lambda r(\sum_1^3 \dot{\theta}_i f_j) - \lambda r(\sum_1^3 \dot{\theta}_j) f_i$$

Y por lo tanto

$$\dot{c}_1 = -\lambda r[(3\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)f_1 + 2(\dot{\theta}_2 f_2 + \dot{\theta}_3 f_3)] \quad (9.67a)$$

$$\dot{c}_2 = -\lambda r[2\dot{\theta}_1 f_1 + (\dot{\theta}_1 + 3\dot{\theta}_2 + \dot{\theta}_3)f_2 + 2\dot{\theta}_3 f_3] \quad (9.67b)$$

$$\dot{c}_3 = -\lambda r[(2(\dot{\theta}_1 f_1 + \dot{\theta}_2 f_2) + (\dot{\theta}_1 + \dot{\theta}_2 + 3\dot{\theta}_3)f_3] \quad (9.67c)$$

De las relaciones anteriores, y aquellas de las velocidades angulares de las ruedas, Ecuaciones (9.66a, 9.66b, 9.66c), ahora se pueden escribir las torsiones de las ruedas en la forma

$$t_i = T_i \dot{\theta}_a, \quad i = 1, 2, 3 \quad (9.68)$$

Donde:

$$T_1 \equiv \begin{bmatrix} e_1 - \lambda k & -\lambda k & -\lambda k \\ -3\lambda r f_1 & -\lambda r(f_1 + 2f_2) & -\lambda r(f_1 + 2f_3) \end{bmatrix}$$

$$T_2 \equiv \begin{bmatrix} -\lambda k & e_2 - \lambda k & -\lambda k \\ -\lambda r(f_2 + 2f_1) & -3\lambda r f_2 & -\lambda r(f_2 + 2f_3) \end{bmatrix}$$

$$T_3 \equiv \begin{bmatrix} -\lambda k & -\lambda k & e_3 - \lambda k \\ -\lambda r(f_3 + 2f_1) & -\lambda r(f_3 + 2f_2) & -3\lambda r f_3 \end{bmatrix}$$

Por otra parte, similar a lo que tenemos en la ecuación 9.62, una interesante relación entre las velocidades angulares de las ruedas se plantea aquí. En efecto, después de sumar los lados correspondientes de las tres ecuaciones (9.66a, 9.66b, 9.66c), se obtiene

$$\sum_1^3 w_i = \sum_1^3 \dot{\theta}_i e_i - 3\lambda k \sum_1^3 \dot{\theta}_i$$

Además, por producto punto de los dos lados de la ecuación anterior por  $k$ , se obtiene, después de reordenación de los términos.

$$3\lambda \sum_1^3 \dot{\theta}_i = -k \sum_1^3 w_i$$

Y en virtud de la ecuación 9.61

$$w = k\bar{w}, \quad \bar{w} \equiv \frac{1}{3} \sum_1^3 w_i \quad (9.69)$$

Es decir, la componente vertical de la velocidad media angular de la rueda es igual a la velocidad escalar angular de la plataforma. Ahora se procede a establecer el modelo matemático que rige la dinámica del sistema bajo estudio. La matriz de inercia generalizada se calcula como

$$I = \sum_1^4 T_i^T M_i T_i \quad (9.70)$$

Donde, si  $I_w$ ,  $j$  y  $m_w$  denotan la matriz momento de inercia, en coordenadas del cuerpo fijo, y la masa de cada una de las tres ruedas, con definiciones similares para  $I_p$  y  $m_p$  línea de contacto como perteneciente a la plataforma.

$$M_i = \begin{bmatrix} I_w & 0 \\ 0 & m_w 1 \end{bmatrix}, \quad i = 1, 2, 3, \quad M_4 = \begin{bmatrix} I_p & 0 \\ 0 & m_p 1 \end{bmatrix} \quad (9.71)$$

También necesitaremos las parejas de velocidad angular,  $W_i$ , que se calculan como

$$W_i = \begin{bmatrix} \Omega_i & 0 \\ 0 & 0 \end{bmatrix}, \quad i = 1, 2, 3 \quad (9.72)$$

Donde  $W_4$  no será necesario, dado que la plataforma se somete a movimiento plano. Se tiene entonces

$$M_1 T_1 = \begin{bmatrix} I_w(e_1 - \lambda k) & -\lambda I_w k & -\lambda I_w k \\ -3m_w \lambda r f_1 & -m_w \lambda r (f_1 + 2f_2) & -m_w \lambda r (f_1 + 2f_3) \end{bmatrix}$$

Además, suponemos que en un marco de coordenadas locales  $\{e_i, f_i, k\}$ ,

$$I_w = \begin{bmatrix} I & 0 & 0 \\ 0 & J & 0 \\ 0 & 0 & J \end{bmatrix}$$

En los cuales  $I$  y  $J$  son constantes, por lo tanto

$$T_1^T M_1 T_1 = \begin{bmatrix} I + \lambda^2 K & \lambda^2 J & \lambda^2 J \\ \lambda^2 J & \lambda^2 L & \lambda^2 M \\ \lambda^2 J & \lambda^2 M & \lambda^2 L \end{bmatrix}$$

Donde

$$K \equiv J + 9m_w r^2$$

$$L \equiv J + 3m_w r^2$$

$$M \equiv J - 3m_w r^2$$

Del mismo modo,

$$T_2^T M_2 T_2 = \begin{bmatrix} \lambda^2 L & \lambda^2 J & \lambda^2 M \\ \lambda^2 J & I + \lambda^2 K & \lambda^2 J \\ \lambda^2 M & \lambda^2 J & \lambda^2 L \end{bmatrix}$$

$$T_3^T M_3 T_3 = \begin{bmatrix} \lambda^2 L & \lambda^2 J & \lambda^2 J \\ \lambda^2 J & \lambda^2 L & \lambda^2 J \\ \lambda^2 J & \lambda^2 J & I + \lambda^2 K \end{bmatrix}$$

Además,

$$M_4 T_4 = -\lambda \begin{bmatrix} I_p k & I_p k & I_p k \\ 2m_p r f_1 & 2m_p r f_2 & 2m_p r f_3 \end{bmatrix}$$

Es evidente que, en virtud del movimiento de plano experimentado por la plataforma, sólo se necesita su momento de inercia  $H$  sobre la vertical que pasa por su centro de masa. Entonces,

$$T_4^T M_4 T_4 = \lambda^2 \begin{bmatrix} H + 4m_p r^2 & H - 2m_p r^2 & H - 2m_p r^2 \\ H - 2m_p r^2 & H + 4m_p r^2 & H - 2m_p r^2 \\ H - 2m_p r^2 & H - 2m_p r^2 & H + 4m_p r^2 \end{bmatrix}$$

Al sumar los cuatro productos calculados anteriormente, obtenemos

$$I = \lambda^2 \begin{bmatrix} \alpha & \beta & \beta \\ \beta & \alpha & \beta \\ \beta & \beta & \alpha \end{bmatrix}$$

Con las siguientes definiciones:

$$\alpha \equiv I + \lambda^2 K(H + 3J + 15m_w r^2 + 4m_p r^2)$$

$$\beta \equiv (H + 3J - 3m_w r^2 - 2m_p r^2)$$

La cual es una matriz constante. Además, observe que la simetría geométrica y de inercia asume desde el principio es evidente en la forma de la matriz de inercia anterior, su inversa se obtiene fácilmente en forma cerrada, es decir,

$$I^{-1} = \frac{1}{\Delta} \begin{bmatrix} \alpha + \beta & -\beta & -\beta \\ -\beta & \alpha + \beta & -\beta \\ -\beta & -\beta & \alpha + \beta \end{bmatrix}, \quad \Delta \equiv (\alpha + \beta)\alpha - 2\beta^2$$

A continuación, nos dirigimos al cálculo del término  $T^T M \dot{T}$ . Se encuentra que

$$T^T M \ddot{T} = \sum_1^4 T_i^T M_i \dot{T}_i$$

Cada uno de los productos antes mencionados se amplía a continuación. Tenemos, en primer lugar,

$$\dot{T}_1 = \begin{bmatrix} wf_1 & 0 & 0 \\ 3\lambda r w e_1 & -\lambda r w (e_3 - e_2) & \lambda r w (e_3 - e_2) \end{bmatrix}$$

$$\dot{T}_2 = \begin{bmatrix} 0 & wf_2 & 0 \\ \lambda r w (e_1 - e_3) & 3\lambda r w e_2 & -\lambda r w (e_1 - e_3) \end{bmatrix}$$

$$\dot{T}_3 = \begin{bmatrix} 0 & 0 & wf_3 \\ -\lambda r w (e_2 - e_1) & \lambda r w (e_2 - e_1) & 3\lambda r w e_3 \end{bmatrix}$$

$$\dot{T}_4 = \begin{bmatrix} 0 & 0 & 0 \\ 2r w e_1 & 2r w e_2 & 2r w e_3 \end{bmatrix}$$

Por lo tanto, para la primera rueda.

$$M_1 \dot{T}_1 = \begin{bmatrix} I_w w f_1 & 0 & 0 \\ -3\lambda m_w r w e_1 & -\lambda m_w r w (e_3 - e_2) & \lambda m_w r w (e_3 - e_2) \end{bmatrix}$$

Por lo tanto,

$$T_1^T M_1 \dot{T}_1 = 3\sqrt{3}\lambda^2 m_w r^2 w \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}$$

Donde la matriz antisimétrica es la matriz producto cruz del vector  $[0 \ 1 \ 1]^T$ . Por simetría, los otros dos productos,  $T_i^T M_i \dot{T}_i$  para  $i = 1, 2$ , toman formas similares, con la matriz antisimétrica, llegando a ser, en consecuencia, la matriz producto vectorial de los vectores  $[1, 0, 1]^T$  y  $[1, 1, 0]^T$ . Esto significa que el primero de estos tres productos es afectado por la rotación de la segunda y tercera rueda, pero no por el de la primera, el segundo de dichos productos se ve afectado por la rotación de la primera y tercera rueda, pero no por la segunda, el tercer producto se ve afectado, a su vez, por la rotación de las dos primeras ruedas, pero no por aquel de la tercera rueda. Tenemos así

$$T_2^T M_2 \dot{T}_2 = 3\sqrt{3}\lambda^2 m_w r^2 w \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$T_3^T M_3 \dot{T}_3 = 3\sqrt{3}\lambda^2 m_w r^2 w \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix}$$

Además,

$$M_4 \dot{T}_4 = \begin{bmatrix} 0 & 0 & 0 \\ -2m_p r w e_1 & -2m_p r w e_2 & -2m_p r w e_3 \end{bmatrix}$$

Y por lo tanto

$$T_4^T M_4 \dot{T}_4 = 2\sqrt{3}\lambda^2 m_p r^2 w \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix} \quad (9.73)$$

Cuya matriz antisimétrica se identifica fácilmente como la matriz producto cruz del vector  $[1, 1, 1]^T$ , lo que indica una participación igual de las tres ruedas en este término, más bien un resultado plausible. Tras la adición de los cuatro productos calculados anteriormente, se obtiene

$$T^T M \dot{T} = 2\sqrt{3}\lambda^2 (3m_w + m_p) r^2 w \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix} \quad (9.74)$$

La participación equitativa de las tres ruedas en el producto anterior es evidente. Además, observe que el término entre paréntesis puede ser considerado como una masa equivalente, que es simplemente la suma de las cuatro masas implicadas, los momentos de inercia de las ruedas no juegan ningún papel en este término. Pasamos ahora al cálculo del término  $T^T W M T$ , que puede ser expresada como una suma,

$$T^T W M T = \sum_1^3 T_i^T W_i M_i T_i$$

Donde no hemos considerado la contribución de la plataforma, ya que esta se somete a movimiento plano. Además, las matrices  $W_i$ , para  $i = 1, 2, 3$ , toman las formas obvias

$$W_i \equiv \begin{bmatrix} \Omega_i & 0 \\ 0 & 0 \end{bmatrix}$$

Tenemos entonces, para la primera rueda,

$$W_1 M_1 T_1 = \begin{bmatrix} w_1 \times [I_w(e_1 - \lambda k)] & -w_1 \times (I_w \lambda k) & -w_1 x(I_w \lambda k) \\ 0 & 0 & 0 \end{bmatrix}$$

Ahora, no se requiere de mucho esfuerzo para calcular el primer producto completo, el cual simplemente se anula, es decir,

$$T_1^T W_1 M_1 T_1 = O_{33}$$

Con  $O_{33}$  definido como la matriz cero 3 x 3. Por simetría, los otros dos productos también desaparecen, y, por tanto, la suma también lo hace, es decir.

$$T^T W M T = O_{33} \quad (9.75)$$

Ahora, el cálculo de las fuerzas generalizadas disipativas y activas es sencillo. Se desprezará aquí la disipación de energía que se produce en los cojinetes de los rodillos, y por lo tanto, si se supone que el lubricante de los cubos de rueda produce torques lineales disipativos, entonces tenemos

$$\delta = c \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}, \quad \tau = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \quad (9.76)$$

Donde  $c$  es el coeficiente de amortiguación común para los tres cubos de las ruedas. Ahora tenemos todos los elementos necesarios para establecer el modelo matemático que rige la dinámica del robot, es decir.

$$I \dot{\theta}_a + C(w) \dot{\theta}_a = \tau - \delta \quad (9.77)$$

Donde  $C(w) = T^T M T + T^T W M T$

$$C(w) = 2\sqrt{3}\pi^2(3m_w + m_P)r^2w \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix} \quad (9.78)$$

Dado que  $w = -a/(3r)(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)$ , la naturaleza cuadrática del segundo término de la ecuación 9.77 en las velocidades de articulación se hace evidente. También es evidente que el modelo matemático deducido anteriormente no depende de  $\theta_a$ . Lo que esto significa es que el modelo matemático permite la integración de las aceleraciones de articulación accionadas para producir historias de velocidad de articulación  $\dot{\theta}_a(t)$ , pero este modelo no puede proporcionar historias variables de articulación  $\theta_a(t)$ . Para obtener estas, para condiciones iniciales dadas, las historias de velocidad de articulación tienen que ser integradas, lo cual puede hacerse por cuadratura numérica. Por último, con el fin de obtener las historias cartesianas de la posición de la plataforma, dadas por el ángulo  $\sigma$  que una línea específica de la plataforma hace con una línea fija en un marco de inercia, y el vector de posición del centro de masa,  $c$ , las ecuaciones 9.61 y 9.62 deben ser integradas. Aunque la integración de la primera puede ser fácilmente realizada por cuadratura, la de esta última requiere el conocimiento de los vectores  $f_i$ , para  $i = 1, 2, 3$ , y estos vectores depende de  $\sigma$ . Así, la integración de la ecuación 9.61 se puede hacer una vez que las historias de velocidad de articulación son conocidas; aquella de la ecuación 9.62 requiere el conocimiento de un ángulo  $\sigma$ . Estas características son inherentes a los sistemas no holonómicos.

---

---

# CAPÍTULO 10

---

## PERCEPCIÓN EN LOS ROBOTS MÓVILES

Una de las tareas más importantes de un sistema autónomo de cualquier tipo es adquirir conocimientos sobre su entorno. Esto se hace tomando mediciones utilizando diversos sensores y luego extrayendo información significativa de esas mediciones. En este capítulo se presentan los sensores más comunes que se utilizan en los robots móviles y se discutirán estrategias para la extracción de información de los sensores.

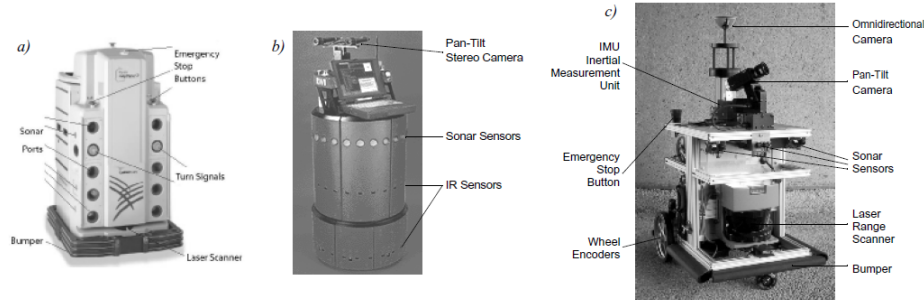
### 10.1. SENSORES PARA ROBOTS MÓVILES

Hay una gran variedad de sensores utilizados en robots móviles (figura 10.1). Algunos sensores se utilizan para medir los valores simples como la temperatura interna de la electrónica de un robot o de la velocidad de rotación de los motores. Otros sensores más sofisticados, pueden ser utilizados para adquirir información sobre el entorno del robot o incluso para medir directamente la posición global de un robot.

#### 10.1.1. Clasificación de los sensores

Se clasifican los sensores por medio de dos ejes funcionales importantes: propioceptivo / exteroceptivo y pasivo / activo. Los sensores *propioceptivos* miden valores internos al sistema (robot), por ejemplo, velocidad del motor, carga de la rueda, los ángulos de las articulaciones del brazo del robot, voltaje de la batería. Los sensores *exteroceptivos*

Figura 10.1: Ejemplos de robots con sistemas de sensores múltiples: (a) HelpMate del Transition Research Corporation, (b) B21 del Real World Interface, (c) BIBA Robot, BlueBotics SA



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

adquieren información de entorno del robot, por ejemplo, mediciones de la distancia, la intensidad de luz, amplitud del sonido. Por lo tanto las mediciones de los sensores exteroceptivos son interpretadas por el robot con el fin de extraer características ambientales significativas.

Los sensores pasivos miden la energía ambiental del entorno que entra en al sensor. Ejemplos de sensores pasivos incluyen sondas de temperatura, micrófonos y cámaras CCD o CMOS. Los sensores activos emiten energía en el medio ambiente, y luego miden la reacción del medio ambiente. Ejemplos de sensores activos incluyen codificadores de cuadratura de ruedas, sensores ultrasónicos, y telémetros láser. La tabla 10.1 proporciona una clasificación de los sensores más útiles para las aplicaciones de robots móviles. Los sensores más interesantes son discutidos más adelante.

### 10.1.2. Caracterización del rendimiento del sensor

Algunos sensores proporcionan una precisión extrema en entornos de laboratorio bien controlados, pero se llenan con errores cuando se somete a las variaciones ambientales del mundo real. Otros sensores proporcionan, estrechos datos de alta precisión de en una amplia variedad de entornos.

#### Clasificaciones básicas de respuesta del sensor

Un número de características del sensor se puede clasificar cuantitativamente en un entorno de laboratorio. El rango dinámico se utiliza para medir la diferencia entre

Cuadro 10.1: Clasificación de los sensores usados en aplicaciones de robótica móvil. (A, Activo; P, Pasivo; P/A, pasivo/activa; PC, propioceptivos; EC, exteroceptivos)

Clasificación general (uso típico)	Sensor Sistema de sensor	PC o EC	A o P
<i>Sensores táctiles</i> (detección de contacto físico o proximidad; interruptores de seguridad)	Interruptores de contacto, Parachoques	EC	P
	Barreras ópticas	EC	A
	Sensores de proximidad sin contacto	EC	A
<i>Sensores de rueda/motor</i> (velocidad y posición de rueda/motor)	Encoders Brush	PC	P
	Potenciómetros	PC	P
	Sincronizadores de resolución	PC	A
	Encoders ópticos	PC	A
	Encoders magnéticos	PC	A
	Encoders inductivos	PC	A
Encoders capacitivos	PC	A	
<i>Sensor de rumbo</i> (orientación del robot en relación con un marco de referencia fijo)	Brújula	EC	P
	Giroscopios	PC	P
	Inclinómetros	EC	A/P
<i>Basado en balizas</i> (localización en un marco de referencia fijo)	GPS	EC	A
	Óptica Activa o balizas RF	EC	A
	Balizas ultrasónicas activas	EC	A
	Balizas reflectantes	EC	A
<i>Alcance Activo.</i> (reflectividad, tiempo de vuelo, y triangulación geométrica)	Sensores de reflectividad	EC	A
	Sensor ultrasónico	EC	A
	Telémetro láser	EC	A
	Triangulación óptica (1D)	EC	A
	Luz estructurada (2D)	EC	A
<i>Sensores de Movimiento / velocidad</i> (velocidad relativa de objetos fijos o móviles)	Radar Doppler	EC	A
	Sonido Doppler	EC	A
<i>Sensores basados en Visión</i> (alcance visual, análisis de imagen completa, segmentación, reconocimiento de objetos)	Cámara (s) CCD / CMOS Paquetes de Visual Los paquetes de seguimiento de objeto	EC	P

Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition, 2004

los límites inferior y superior de los valores de entrada al sensor mientras se mantiene la operación del sensor normal. Formalmente, el rango dinámico es la relación entre el

valor máximo de entrada al valor de entrada mínimo medible. Debido a que esta relación bruta puede ser difícil de manejar, por lo general se mide en decibeles, que se calculan como diez veces el logaritmo común de la gama dinámica. Sin embargo, hay confusión potencial en el cálculo de decibeles, que tienen el propósito de medir la relación entre poderes, como vatios o caballos de fuerza. Suponga que su sensor mide la corriente del motor y puede registrar valores entre un mínimo de  $1\text{ mA}$  a  $20\text{ A}$ . El rango dinámico de este sensor de corriente se define como:

$$10 \cdot \log \left[ \frac{20}{0,001} \right] = 43\text{ dB} \quad (10.1)$$

Ahora suponga que tiene un sensor de voltaje que mide el voltaje de la batería del robot, que mide cualquier valor de  $1\text{ mV}$  a  $20\text{ V}$ . El voltaje no es una unidad de potencia, pero el cuadrado del voltaje es proporcional a la potencia. Por lo tanto, se usa 20 en lugar de 10:

$$20 \cdot \log \left[ \frac{20}{0,001} \right] = 86\text{ dB} \quad (10.2)$$

La **resolución** es la diferencia mínima entre dos valores que pueden ser detectados por un sensor. Por lo general, el límite inferior de la gama dinámica de un sensor es igual a su resolución. Sin embargo, en el caso de sensores digitales, esto no es necesariamente así. Por ejemplo, supongamos que tiene un sensor que mide tensión, realiza una conversión de analógico a digital (A/D), y envía el valor convertido como un número de 8-bit lineal correspondiente a entre 0 y  $5\text{ V}$ . Si este sensor es realmente lineal, entonces tiene  $2^8 - 1$  valores totales de salida, o una resolución de  $5\text{ V}(225) = 20\text{ mV}$ .

La **linealidad** es una medida importante que determina el comportamiento de la señal de salida del sensor a medida que la señal de entrada varía. Una respuesta lineal indica que si dos entradas  $x$  e  $y$  resultan en dos salidas  $f(x)$  y  $f(y)$ , entonces para cualquier valor de  $a$  y  $b$ ,  $f(ax + by) = af(x) + bf(y)$ . Esto significa que una gráfica de la respuesta de entrada/salida del sensor es simplemente una línea recta.

El **ancho de banda** o **frecuencia** se utiliza para medir la velocidad con la que un sensor puede proporcionar una corriente de lecturas. Formalmente, el número de mediciones por segundo se define como la frecuencia del sensor en hertz. Debido a la dinámica de movimiento a través de su medio ambiente, los robots móviles a menudo son limitados en velocidad máxima por el ancho de banda de sus sensores de detección de obstáculos. Por lo tanto, aumentando el ancho de banda de sensores de rango y basados en la visión, ha sido un objetivo prioritario de la comunidad robótica.

## Rendimiento del sensor in situ

Las características de los sensores anteriores pueden ser medidas razonablemente en un entorno de laboratorio con extrapolación segura para el desempeño en el despliegue en el mundo real. Sin embargo, una serie de medidas importantes no pueden ser confiablemente adquiridas sin conocimiento profundo de la compleja interacción entre todas las características ambientales y los sensores correspondientes. Esto es más relevante para los sensores más sofisticados, incluyendo los sensores de alcance activo y sensores visuales de interpretación.

La **sensibilidad** en sí misma es un rasgo deseable. Esta es una medida del grado en el que un cambio incremental en la señal de entrada del objetivo cambia la señal de salida. Formalmente, la sensibilidad es la proporción de cambio de la salida a cambio de entrada. Desafortunadamente, sin embargo, la sensibilidad de los sensores exteroceptivos se confunde a menudo por la sensibilidad indeseable y el rendimiento de acoplamiento a otros parámetros ambientales.

**Sensibilidad cruzada (cross-sensitivity)** es el término técnico para la sensibilidad a los parámetros ambientales que son ortogonales a los parámetros objetivo para el sensor. Por ejemplo, una brújula de puerta de flujo puede demostrar alta sensibilidad al norte magnético y es por lo tanto de gran uso para la navegación móvil del robot. Sin embargo, la brújula también demuestra una alta sensibilidad a los materiales de construcción ferrosos, tanto así que su sensibilidad cruzada a menudo hace inútil al sensor en algunos ambientes interiores. Una alta sensibilidad cruzada de un sensor es generalmente indeseable, especialmente cuando no se puede modelar.

El **error** de un sensor se define como la diferencia entre las mediciones de la salida del sensor y los valores reales que se están midiendo, dentro de un cierto contexto operativo específico. Dado un valor real  $v$  y un valor medido  $m$ , se puede definir el error como  $error = m - v$ .

La **exactitud** se define como el grado de conformidad entre la medida del sensor y el valor verdadero, y a menudo se expresa como una proporción del valor real (por ejemplo, 97.5% de exactitud). Así un pequeño error corresponde a una gran exactitud y viceversa:

$$accuracy = 1 - \frac{|error|}{v} \quad (10.3)$$

Es importante distinguir entre dos diferentes fuentes de error:

- Los *errores sistemáticos* son causados por factores o procesos que pueden ser modelados en teoría. Estos errores son, por lo tanto, deterministas (predecibles).

Una mala calibración de un telémetro de láser, una inclinación no modelada del suelo de un pasillo, y una cabeza de cámara estéreo torcida debido a una colisión anterior son todas las posibles causas de errores de sensores sistemáticos.

- Los *errores aleatorios* no pueden predecirse utilizando un sofisticado modelo ni pueden ser mitigados por una maquinaria de sensor más precisa. Estos errores sólo pueden ser descritos en términos probabilísticos (es decir, estocásticamente). La inestabilidad Hue en una cámara de color, falsos errores de telemetría, y el ruido de nivel negro en una cámara son todos ejemplos de errores aleatorios.

La precisión se confunde a menudo con exactitud, y ahora tenemos las herramientas para distinguir claramente estos dos términos. Intuitivamente, de alta precisión se refiere a la reproducibilidad de los resultados del sensor. Por ejemplo, un sensor que toma múltiples lecturas de la misma situación ambiental tiene una alta precisión si se produce la misma salida. Si el error aleatorio de un sensor se caracteriza por un valor medio  $\mu$  y la desviación estándar  $\sigma$ . La definición formal de precisión es la proporción del rango de salida del sensor para la desviación estándar:

$$precision = \frac{range}{\sigma} \quad (10.4)$$

Nótese que solo  $\sigma$  y no  $\mu$  tiene un impacto en la precisión. En contraste, el error medio  $\mu$  es directamente proporcional al error general del sensor e inversamente proporcional a la exactitud del sensor.

### **Caracterización del error: los desafíos en robótica móvil**

Puesto que los sistemas que estudiamos son móviles, su posición siempre cambiante y su movimiento tienen un impacto significativo sobre el comportamiento del sensor en general.

**Desenfoco de errores sistemáticos y aleatorios.** Los sensores de alcance activos tienden a tener modos de fallo que se desencadenan en gran medida por determinadas posiciones relativas de los objetivos del sensor y el medio ambiente. Por ejemplo, un sensor de sonar producirá reflexiones especulares, produciendo mediciones bastante inexactas de rango, en ángulos específicos a una pared de yeso liso. Durante el movimiento del robot, dichos ángulos relativos se producen a intervalos estocásticos. Las posibilidades de que un sonar entre en este modo de error durante el movimiento del robot son altas. Una vez que el robot no se mueve, el error parece ser sistemático y de alta precisión.

**Distribuciones multimodales de error.** Es común caracterizar el comportamiento del error aleatorio de un sensor en términos de una distribución de probabilidad sobre los valores de salida diferentes. Es importante por el momento reconocer el hecho de que con frecuencia se asume simetría y una distribución unimodal. Estos son supuestos fuertes que permiten a poderosos principios matemáticos ser aplicados a los problemas de robot móviles, pero es importante darse cuenta de lo equivocadas que son estas suposiciones por lo general.

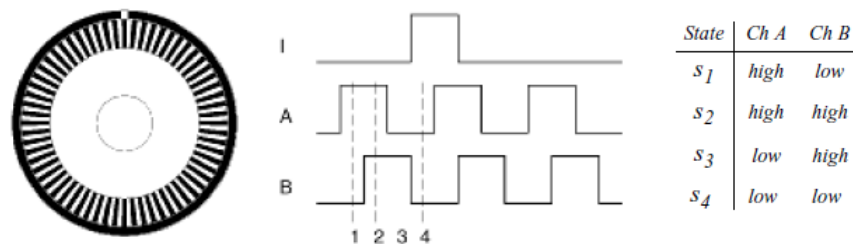
### 10.1.3. Sensores de Ruedas/motor

Los sensores de rueda/motor son dispositivos utilizados para medir el estado interno y la dinámica de un robot móvil. Se muestra sólo uno de tales sensores, el codificador incremental óptico.

#### Codificadores ópticos

Los codificadores ópticos incrementales se han convertido en el dispositivo más popular para medir la posición y velocidad angular dentro de un motor de accionamiento o en el eje de una rueda o un mecanismo de dirección. En robótica móvil, los codificadores se utilizan para controlar la posición o la velocidad de las ruedas y otras articulaciones accionadas por motor. Debido a que estos sensores son propioceptivos, su estimación de posición es la mejor en el marco de referencia del robot y, cuando se aplica al problema de la localización del robot, se requieren correcciones importantes. Un codificador

Figura 10.2: Codificador de cuadratura óptico de la rueda



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

óptico es básicamente un chopper de luz mecánico que produce un cierto número de pulsos de onda sinusoidales o cuadrados para cada revolución del eje. Se compone de una fuente de iluminación, una rejilla fija que enmascara la luz, un disco de rotor con

una rejilla fina óptica que gira con el eje, y detectores ópticos fijos. A medida que el rotor se mueve, la cantidad de luz que incide en los detectores ópticos varía basándose en la alineación de las rejillas fijas y en movimiento.

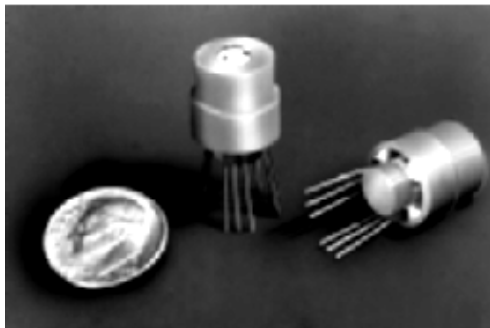
#### 10.1.4. Detectores de rumbo

Los detectores de rumbo que pueden ser propioceptivos (giroscopio, inclinómetro) o exteroceptivos (brújula). Se utilizan para determinar la orientación y la inclinación del robot. Ellos nos permiten, junto con información adecuada sobre la velocidad, integrar el movimiento a una estimación de la posición. Este procedimiento, que tiene sus raíces en la navegación de buques y barcos, se llama navegación por estima.

##### Brújulas

Los dos sensores modernos más comunes para la medición de la dirección de un campo magnético son el efecto Hall y brújulas de compuerta de flujo. Cada uno tiene ventajas y desventajas. El efecto Hall describe el comportamiento del potencial eléctrico en un semiconductor cuando está en presencia de un campo magnético. Cuando una corriente constante se aplica a través de la longitud de un semiconductor, habrá una diferencia de tensión en la dirección perpendicular, en todo el ancho del semiconductor, basado en la orientación relativa de los semiconductores hacia líneas de flujo magnético. Además, la señal del potencial de voltaje identifica la dirección del campo magnético.

Figura 10.3: Brújula digital: Se muestran los sensores tales como el sensor de efecto Hall analógico/digital



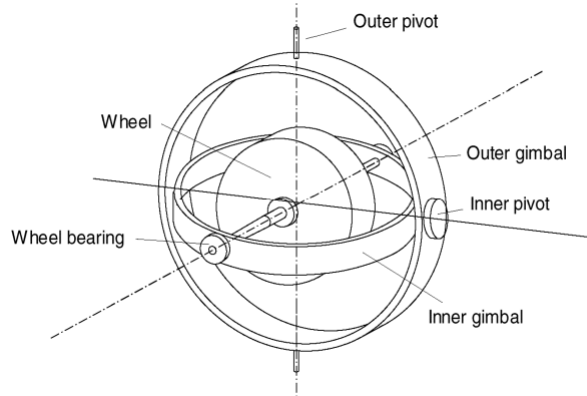
Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

Así, un solo semiconductor proporciona una medición de flujo y la dirección a lo largo de una dimensión. Las brújulas de efecto Hall digitales son populares en la robótica

móvil, y contienen dos semiconductores, tales en ángulo recto, proporcionando dos ejes de dirección de campo magnético, produciendo de esta manera una de las ocho posibles direcciones de la brújula. Los instrumentos son baratos, pero sufren también de una serie de desventajas. La resolución de una brújula digital de efecto Hall es pobre. Las fuentes internas de error incluyen la no linealidad del sensor básico y los errores sistemáticos de polarización a nivel del semiconductor. La circuitería resultante debe realizar un filtrado significativo, y esto disminuye el ancho de banda de las brújulas de efecto Hall a valores que son lentos en términos de robots móviles. Por ejemplo, la brújula de efecto Hall representada en la figura 10.3 necesita 2.5 segundos hasta estabilizarse después de un giro de 90 grados.

La brújula de puerta de flujo funciona según un principio diferente. Dos pequeñas bobinas están enrolladas sobre núcleos de ferrita y están fijados perpendicularmente entre sí. Cuando la corriente alterna se activa en ambas bobinas, el campo magnético causa cambios en la fase en función de su alineación relativa con cada bobina. Mediante la medición de los dos desplazamientos de fase, se puede calcular la dirección del campo magnético en dos dimensiones. La brújula de puerta de flujo puede medir con precisión la fuerza de un campo magnético y ha mejorado la resolución y precisión, sin embargo, es tanto más grande y más caro que una brújula efecto Hall.

Figura 10.4: Giroscopio mecánico de dos ejes



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

## Giroscopios

Los giroscopios son sensores de rumbo que conservan su orientación en relación con un marco de referencia fijo. Por tanto, proporcionan una medida absoluta del rumbo de

un sistema móvil. Los giroscopios pueden ser clasificados en dos categorías, giroscopios mecánicos y giroscopios ópticos.

**Giroscopios mecánicos.** El concepto de un giroscopio mecánico se basa en las propiedades de inercia de un rotor de giro rápido. La característica de interés se conoce como la precesión giroscópica. Si se intenta girar una rueda de giro rápido alrededor de su eje vertical, se sentirá una dura reacción en el eje horizontal. Esto es debido al momento angular asociado con una rueda de giro y mantendrá el eje del giroscopio inercialmente estable. El torque reactivo  $\tau$  y por lo tanto la estabilidad de seguimiento con el marco de inercia son proporcionales a la velocidad de giro  $\omega$ , la velocidad de precesión  $\Omega$ , y la inercia de la rueda  $I$ .

$$\tau = I\omega\Omega \quad (10.5)$$

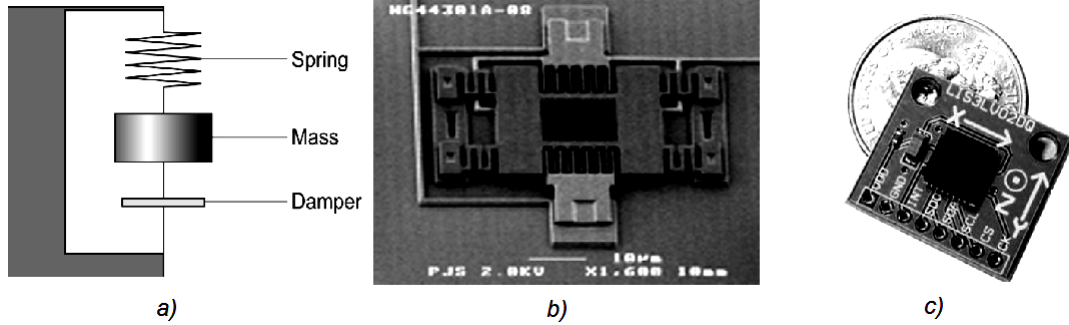
Mediante la disposición de una rueda que gira, como se ve en la figura 10.4, ningún par de torsión puede ser transmitido desde el pivote exterior al eje de rueda. Para navegación, el eje de giro tiene que ser seleccionado inicialmente. Si el eje de giro está alineado con el meridiano norte-sur, la rotación de la tierra no tiene efecto en el eje horizontal de giro. Si apunta hacia el este-oeste, el eje horizontal lee la rotación de la tierra. Los giroscopios de velocidad tienen la misma disposición básica como se muestra en la figura 10.4, pero con una ligera modificación. Los cardanes están sujetos por un resorte de torsión con amortiguación viscosa adicional. Esto permite al sensor medir las velocidades angulares en lugar de la orientación absoluta.

**Giroscopios ópticos.** Son sensores de velocidad angular que utilizan dos haces de luz monocromática, o láseres, que son emitidos a partir de la misma fuente, en lugar de las piezas mecánicas móviles. Trabajan sobre el principio de que la velocidad de la luz no ha cambiado y, por lo tanto, el cambio geométrico puede causar que a la luz le tome una cantidad variable de tiempo para llegar a su destino.

### 10.1.5. Acelerómetros

Un acelerómetro es un dispositivo utilizado para medir todas las fuerzas externas que actúan sobre él, incluyendo la gravedad. Los acelerómetros pertenecen a la clase de sensores propioceptivos. Conceptualmente, un acelerómetro es un sistema de resorte-masa-amortiguador (figura 10.5a) en la que la posición tridimensional de la masa respecto a la prueba de la carcasa del acelerómetro puede medirse con algún mecanismo. Supongamos que una fuerza externa se aplica sobre la carcasa del sensor (por ejemplo,

Figura 10.5: Acelerómetros: (a) Principio de funcionamiento del acelerómetro mecánico, (b) Un acelerómetro MEMS producido por Sandia National Laboratories, (c) Un acelerómetro MEMS comercial



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

gravedad) y que tenemos un resorte ideal con una fuerza proporcional a su desplazamiento. Entonces, podemos escribir:

$$F_{aplicada} = F_{inercial} + F_{amortiguador} + F_{resorte} = m\ddot{x} + c\dot{x} + kx \quad (10.6)$$

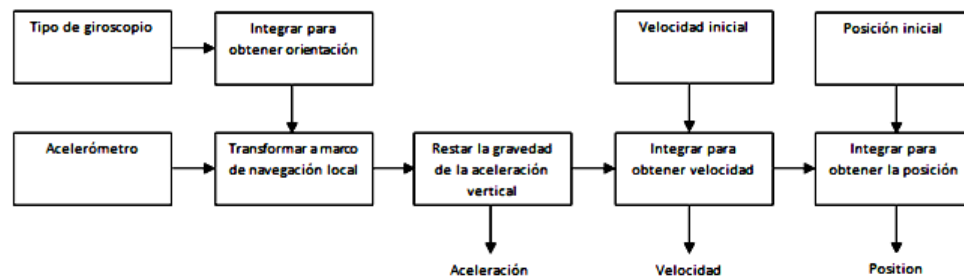
Donde  $m$  es la masa de prueba,  $c$  es el coeficiente de amortiguación,  $k$  es la constante del resorte, y  $x$  es la posición relativa de equilibrio de la carcasa. Cuando el valor se estabiliza, entonces  $\ddot{x} = 0$  y la aceleración aplicada se puede obtener como:

$$a_{aplicada} = \frac{kx}{m} \quad (10.7)$$

Este es el principio de funcionamiento de un acelerómetro mecánico. Los acelerómetros modernos son a menudo pequeños micro sistemas Electro-Mecánicos (Micro Electro-Mechanical Systems - MEMS) que consiste en una estructura de tipo resorte con una masa de prueba. La amortiguación resulta del gas residual en el dispositivo de sellado. Cuando una fuerza externa es aplicada, la masa de prueba se desvía de su posición neutral. Según el principio físico utilizado para medir esta desviación, se pueden tener diferentes tipos de acelerómetros. Los acelerómetros capacitivos miden la desviación mediante la medición de la capacitancia entre una estructura fija y la masa de prueba. Estos acelerómetros son fiables y de bajo costo (figura 10.5b-c).

Finalmente, los acelerómetros se clasifican en dos categorías de acuerdo con su ancho de banda: acelerómetros para mediciones estáticas y dinámicas. Esto es típico para acelerómetros mecánicos y capacitivos. Los usos típicos son la medición de la aceleración

Figura 10.6: (a) Diagrama de bloques IMU, (b) Un IMU comercial producido por Xsens



(a)



(b)

Fuente: Inertial sensors, GPS, and odometry, Dudek, G., Jenkin, M , 2008

de la gravedad o la de un vehículo en movimiento. La segunda categoría de acelerómetros se utilizan para medir aceleraciones de vibración de los objetos o aceleraciones durante los choques. Los acelerómetros típicos de esta categoría son aquellos que fueron realizados con la tecnología piezoeléctrica.

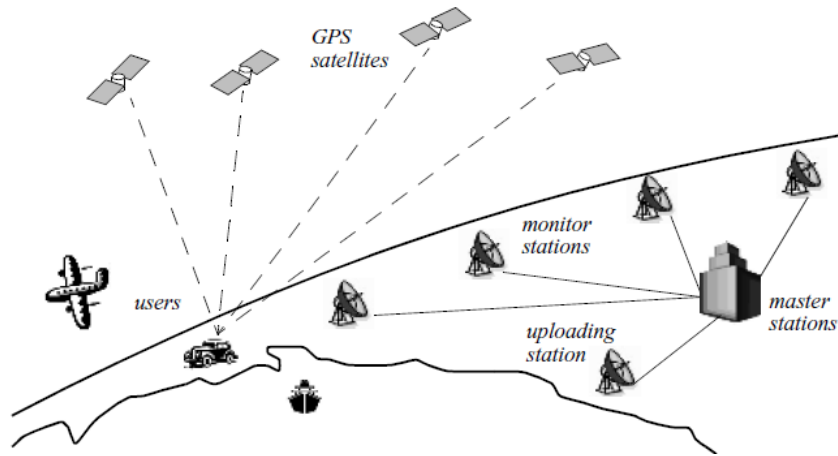
### 10.1.6. Unidad de medición inercial (IMU)

Una unidad de medición inercial (IMU) es un dispositivo que utiliza giroscopios y acelerómetros para estimar la posición relativa, la velocidad y aceleración de un vehículo en movimiento. Una IMU también se conoce como un sistema de navegación inercial (INS), y se ha convertido en un componente común de navegación de aeronaves y buques. Una IMU estima la posición de seis grados de libertad (DOF) del vehículo: la posición  $(x, y, z)$  y la orientación (roll, pitch, yaw). Sin embargo, los sensores de orientación como brújulas y giroscopios, que por el contrario sólo estiman la orientación, a menudo son llamados incorrectamente llamado IMU. El principio de funcionamiento de una IMU se muestra en la figura 10.6.

### 10.1.7. Balizas de tierra

Un método elegante para resolver el problema de localización en la robótica móvil es utilizar balizas activas o pasivas. Usando la interacción de los sensores “a bordo” y las balizas en el medio ambiente, el robot puede identificar su posición con exactitud. La tecnología moderna ha permitido crear sensores para localizar un robot al aire libre con una precisión mejor que 5 cm dentro de áreas que son de kilómetros de tamaño.

Figura 10.7: Cálculo de la posición y rumbo basado en GPS



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

### Sistema de posicionamiento global

El sistema de posicionamiento global (GPS) fue desarrollado inicialmente para uso militar, pero ahora está libremente disponible para la navegación civil. Hay por lo menos veinticuatro satélites GPS operativos en todo momento. Los satélites orbitan cada 12 horas a una altura de 20190 kilómetros. Cuatro satélites se encuentran en cada uno de los seis planos inclinados 55 grados con respecto al plano del ecuador de la Tierra (figura 10.7).

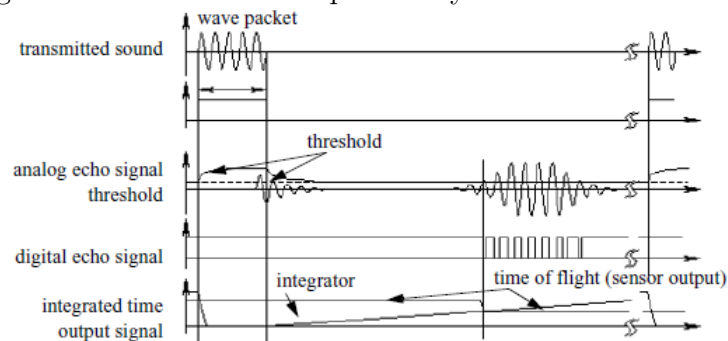
Cada satélite transmite continuamente los datos que indican su ubicación y la hora actual. Por lo tanto, los receptores GPS son sensores completamente pasivos pero exteroceptivos. Los satélites GPS sincronizan sus transmisiones para que sus señales se envíen al mismo tiempo. Cuando un receptor GPS lee la transmisión de dos o más satélites, las diferencias de tiempo de llegada informan al receptor como a su distancia relativa a cada satélite. Mediante la combinación de la información sobre la hora

de llegada y la localización instantánea de cuatro satélites, el receptor puede deducir su propia posición. En teoría, tal triangulación requiere sólo tres puntos de datos. Sin embargo, el tiempo es extremadamente crítico en la aplicación de GPS, porque los intervalos de tiempo que se miden son en nanosegundos. Es, por supuesto, obligatorio que los satélites estén bien sincronizados. Para este fin, se actualizan regularmente por las estaciones de tierra y cada satélite lleva a bordo relojes atómicos para la medida de tiempo. El GPS ha sido un sensor popular en robótica móvil, pero ha sido relegado a proyectos de robots móviles que involucren espacios abiertos y máquinas voladoras autónomas.

### 10.1.8. Alcance activo

Muchos sensores de oscilación tienen un precio bajo, y, lo más importante, todos estos sensores proporcionan salidas de fácil interpretación: mediciones directas de la distancia del robot a los objetos de su entorno. Pero la información de espacio libre local proporcionada por sensores de ubicación también puede ser acumulada en representaciones más allá del marco de referencia local actual del robot. Así, estos sensores de alcance también se encuentran comúnmente en el marco de la localización y los procesos ambientales de modelado de robots móviles.

Figura 10.8: Cálculo de la posición y rumbo basado en GPS



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

### Oscilación activa del tiempo de vuelo

La oscilación de tiempo de vuelo hace uso de la velocidad de propagación del sonido o una onda electromagnética. En general, la distancia de recorrido de una onda de

sonido electromagnética está dada por:

$$d = c \cdot t \quad (10.8)$$

Dónde:

$d$  = Distancia recorrida (por lo general de ida y vuelta);

$c$  = Velocidad de propagación de la onda;

$t$  = Tiempo de vuelo.

El tiempo de vuelo de una distancia típica, por ejemplo 3 m, es de 10 ms para un sistema de ultrasonidos, pero sólo 10 ns para un telémetro láser. La calidad de los sensores de alcance de tiempo de vuelo depende principalmente de:

- Las incertidumbres en la determinación del momento exacto de llegada de la señal reflejada.
- Inexactitudes en la medición del tiempo de vuelo (en particular con los sensores de láser).
- La dispersión del cono del haz transmitido (principalmente con sensores de nivel ultrasónico).
- Interacción con el objetivo (por ejemplo, la absorción de superficie, reflexiones especulares).
- Variación de la velocidad de propagación.
- La velocidad del robot móvil y el objetivo (en el caso de un objetivo dinámico).

**El sensor ultrasónico (tiempo-de-vuelo, sonido).** El principio básico de un sensor ultrasónico es transmitir un paquete de ondas de presión (ultrasónicas) y medir el tiempo que le toma a este paquete de ondas en reflejarse y volver al receptor. La distancia  $d$  del objeto que causa la reflexión se puede calcular sobre la base de la velocidad de propagación del sonido  $c$  y el tiempo de vuelo  $t$ :

$$d = \frac{c \cdot t}{2} \quad (10.9)$$

La velocidad del sonido  $c$  en el aire es dada por:

$$c = \sqrt{\gamma RT} \quad (10.10)$$

Dónde:

$\gamma$  = Relación de calores específicos.

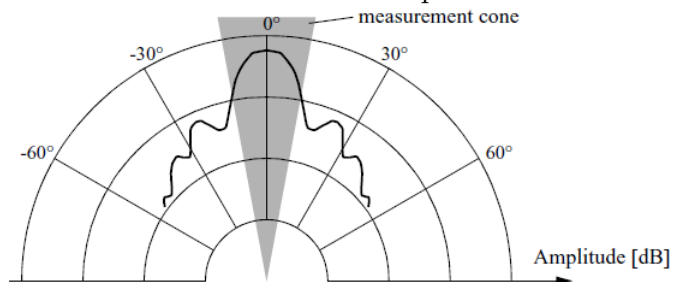
$R$  = Constante del gas.

$T$  = Temperatura en grados Kelvin.

En aire a presión normal y  $20^{\circ} C$ , la velocidad del sonido es aproximadamente  $c = 343m/s$ . La figura 10.8 muestra la salida de señal y entrada de un sensor ultrasónico. En primer lugar, una serie de pulsos de sonido son emitidos que comprenden el paquete de ondas. Un integrador también comienza a subir linealmente en valor, midiendo el tiempo de la transmisión de estas ondas de sonido para la detección de un eco. La onda ultrasónica tiene típicamente una frecuencia entre 40 y 180 kHz y se genera normalmente por un transductor piezoeléctrico o electrostático. Los sensores ultrasónicos más utilizados por los robots móviles tienen un alcance efectivo de unos 12 cm a 5 m. La precisión publicada de los sensores ultrasónicos comerciales varía entre 98 % y 99.1 %. En las aplicaciones de robots móviles, las implementaciones específicas generalmente logran una resolución de aproximadamente 2 cm.

En la mayoría de los casos uno puede desear un ángulo de abertura estrecha para el

Figura 10.9: Distribución de intensidad típica de un sensor ultrasónico

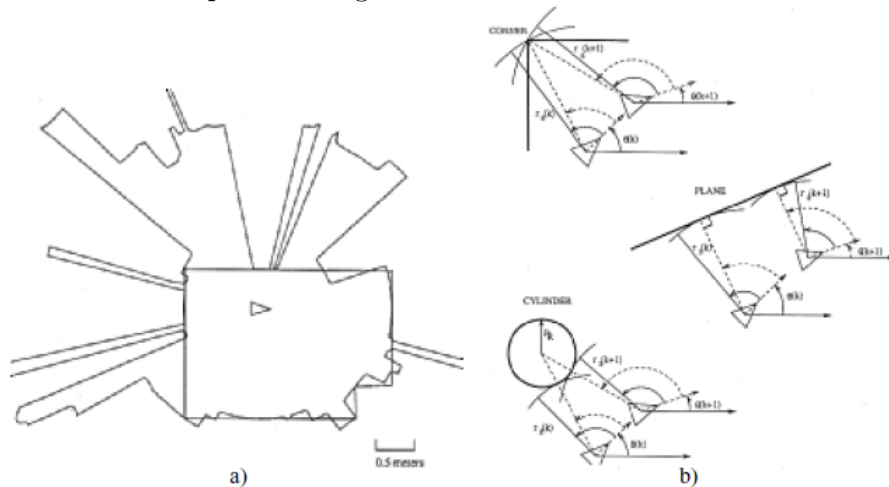


Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

haz de sonido con el fin de obtener también información direccional precisa sobre los objetos que se encuentran. Esta es una limitación importante ya que el sonido se propaga en forma de cono (figura 10.9) con ángulos de apertura de alrededor de 20 a 40 grados. Por consiguiente, cuando se utiliza ultrasonido no se adquieren puntos de datos de profundidad, sino más bien, regiones enteras de profundidad constante. Las lecturas del sensor deben registrarse como segmentos de una esfera de arco (3D) y no como mediciones puntuales (figura 10.10). Sin embargo, desarrollos recientes de la investigación muestran una mejora significativa de la calidad de la medida en el uso sofisticado de procesamiento de señal.

Los sensores ultrasónicos sufren de varios inconvenientes adicionales, es decir, en las áreas de error, ancho de banda, y la sensibilidad a las interferencias. Los valores

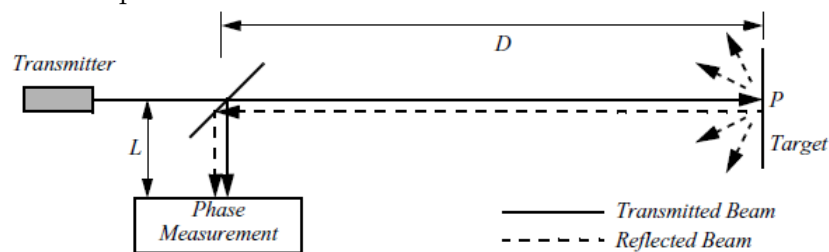
Figura 10.10: Lecturas típicas de un sistema ultrasónico: (a) 360° de exploración; (b) Resultados de diferentes primitivas geométricas.



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

de exactitud revelados por ultrasonidos son valores nominales basados en exitosas reflexiones perpendiculares, de la onda de sonido que sale de un material acústicamente reflectante. Esto no refleja la modalidad de error efectivo vista en un robot móvil que se desplaza a través de su entorno superficie lisa. Una última limitación de alcance

Figura 10.11: Esquema de telemetría láser mediante medición de cambio de fase

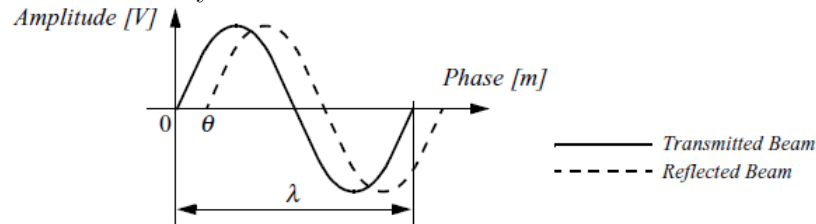


Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

ultrasónico se refiere al ancho de banda. Particularmente en espacios moderadamente abiertos, un único sensor ultrasónico tiene un tiempo de ciclo relativamente lento. Para llevar a cabo el movimiento de un robot a una velocidad moderada, evitando obstáculos por ultrasonidos, la tasa de actualización puede tener un impacto medible en la máxima

velocidad posible sin dejar de detectar y evitar obstáculos con seguridad.

Figura 10.12: Rango de estimación mediante la medición del desplazamiento de fase entre señales transmitidas y recibidas

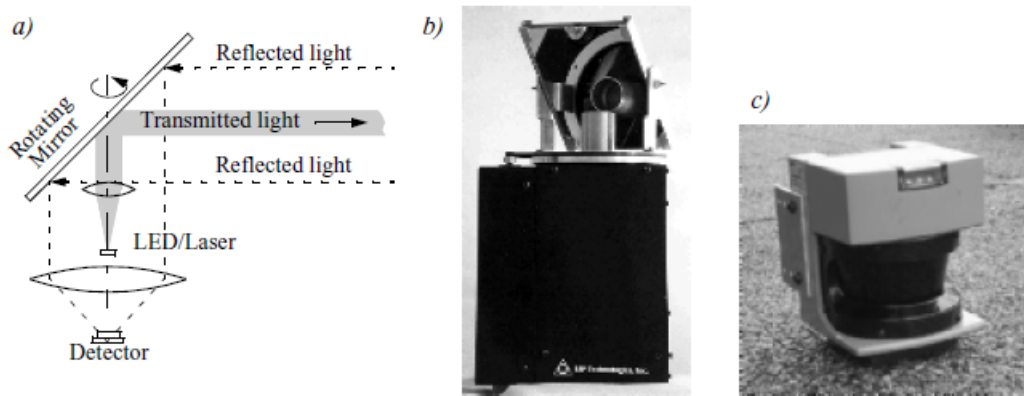


Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

**Telémetro láser (tiempo de vuelo, electromagnético).** El telémetro láser es un sensor de tiempo de vuelo que logra mejoras significativas sobre el alcance del sensor ultrasónico debido a la utilización de la luz láser en lugar de sonido. Este tipo de sensor se compone de un transmisor que ilumina un objetivo con un haz colimado (por ejemplo, láser), y un receptor capaz de detectar la componente de luz que es esencialmente coaxial con el haz transmitido. A menudo se refiere como radar óptico o LIDAR (*Light Detection and Ranging*), estos dispositivos producen una estimación de intervalo basado en el tiempo necesario para que la luz llegue al destino y regrese. Un dispositivo mecánico con un espejo barre el haz de luz para cubrir la escena requerida en un plano o incluso en tres dimensiones, utilizando un espejo que rota y que se inclina.

**Medición de desplazamiento de fase.** Luz del infrarrojo cercano (procedente de un diodo [LED] emisor de luz o láser) es colimada y transmitida desde el transmisor en la figura 10.11 y golpea un punto  $P$  en el entorno. La longitud de onda de la luz infrarroja emitida es de 824 nm y para la mayoría de las superficies. El componente de la luz infrarroja que cae dentro de la abertura de recepción del sensor volverá casi paralela al haz transmitido para objetos distantes. El sensor transmite 100% de luz modulada en amplitud a una frecuencia conocida y mide el desplazamiento de fase entre las señales transmitidas y reflejadas. La figura 10.12 muestra cómo esta técnica se puede utilizar para medir rango. La longitud de onda de la señal de modulación obedece a la ecuación  $c = f \cdot \lambda$  donde  $c$  es la velocidad de la luz y  $f$  la frecuencia de modulación. Para  $f = 5 \text{ MHz}$  (como en el sensor de AT&T),  $\lambda = 60 \text{ m}$ . La distancia

Figura 10.13: (a) Esquema del sensor de alcance láser con espejo giratorio, (b) Sensor de nivel de barrido de EPS Technologies Inc., (c) Sensor Industrial láser de 180 grados de Sick Inc., Alemania



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

total  $D'$  recorrida por la luz emitida es:

$$D' = L + 2D = L + \frac{\theta}{2\pi}\lambda \quad (10.11)$$

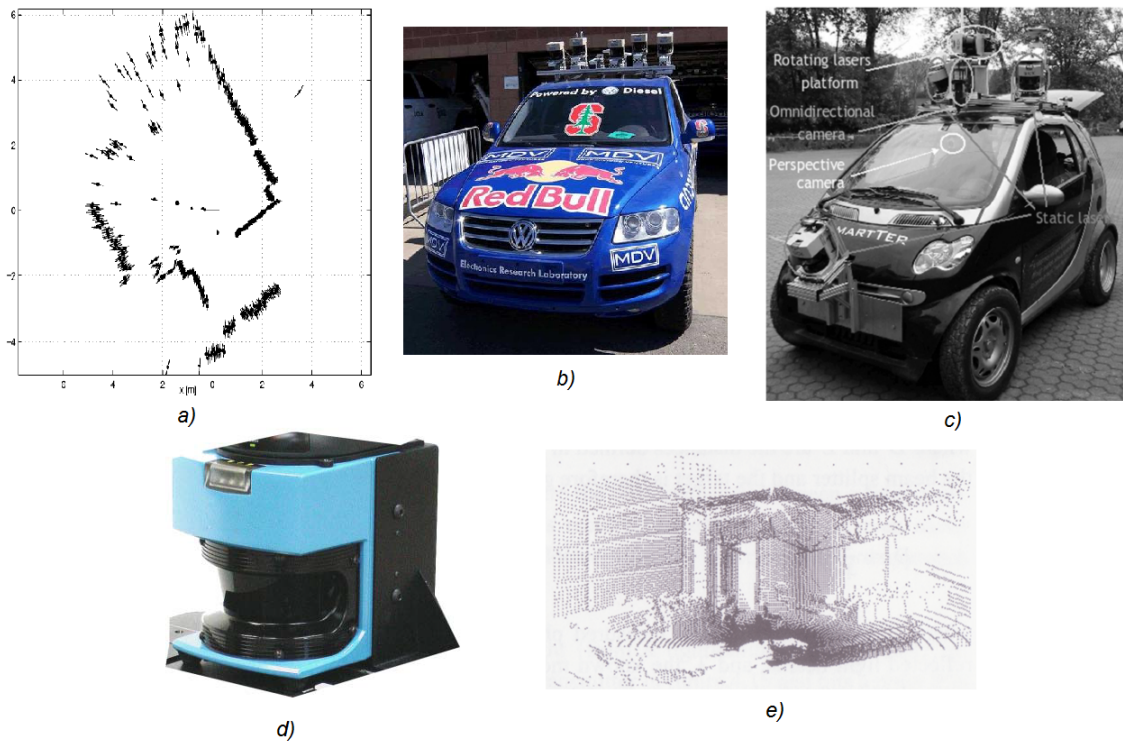
Donde  $D$  y  $L$  son las distancias definidas en la figura 10.11. La distancia requerida  $D$ , entre el divisor de haz y el objetivo, viene dada por:

$$D = \frac{\lambda}{4\pi}\theta \quad (10.12)$$

Donde  $\theta$  es la diferencia de fase medida electrónicamente entre los haces de luz transmitidos y reflejados, y  $\lambda$  la longitud de onda de modulación conocida. Se puede observar que la transmisión de una onda modulada en frecuencia única, teóricamente, puede resultar en estimaciones de rango ambigua, ya que, por ejemplo, si  $\lambda = 60 \text{ m}$ , un objetivo a una distancia de 5 m daría una medida de fase indistinguible desde un objetivo a 65 m, dado que cada ángulo de fase sería  $360^\circ$  por separado. Por lo tanto, se define un “intervalo de ambigüedad” de  $\lambda$ , pero en la práctica se observa que el alcance del sensor es mucho menor que  $\lambda$ , debido a la atenuación de la señal en el aire.

En la figura 10.13 se muestran el esquema de un sensor láser típico de rango de 360 grados y dos ejemplos. La figura 10.14a muestra una imagen de alcance típico de una toma escaneada 360 grados con un sensor de rango láser. El escáner láser Sick que se muestra en la figura 10.13 consigue una resolución angular de 0.5 grados. Resolución en profundidad de aproximadamente 5 cm, en un rango desde 5 cm hasta 20 m o

Figura 10.14: (a) Imagen rango típico de un sensor láser de rangos 2D con un espejo giratorio. (b) Stanley, el coche autónomo de Stanford ganó el 2005 Darpa Grand Challenge, (c) El Smarter, el coche autónomo desarrollado en la ASL (ETH Zurich). (d) Una vista de cerca a los Sicks utilizados en el Smarter. (e) Una nube de puntos láser 3D construida a partir de los Sicks giratorios



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

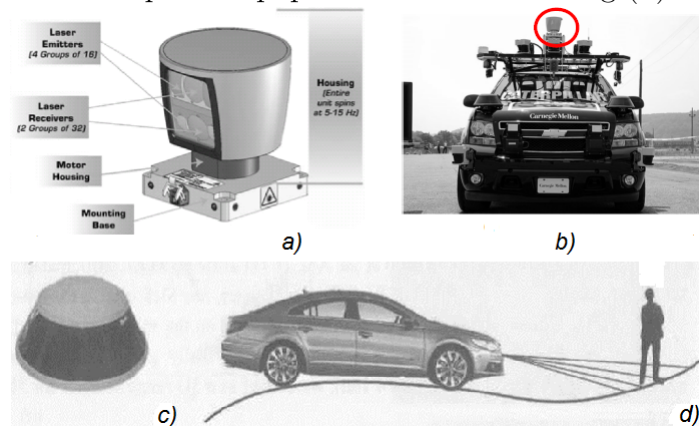
más, dependiendo de la luminosidad del objeto que se recorrió. Este dispositivo realiza veinticinco exploraciones de  $180^\circ/s$ , pero no tiene capacidad de espejo giratorio para la dimensión vertical.

Como un ejemplo de su uso en robótica móvil, cinco rayos láser Sick fueron utilizados para la detección de corto alcance en Stanley (figura 10.14b), el coche autónomo que ganó en 2005 el DARPA Grand Challenge. En una configuración diferente, cinco rayos láser Sick también se utilizaron en el Smarter (figura 10.14c), el coche autónomo desarrollado en la ASL (ETH Zurich), que participó en ELROB 2006, la prueba Robot Land Europea. En el Smarter, un láser Sick en la parte delantera inferior se utiliza para la evasión de obstáculos cercanos, mientras que dos rayos láser en el techo (figura

10.14d, ligeramente inclinados hacia los lados, se utilizan para la navegación local. Por último, otros dos rayos láser, montados verticalmente sobre una plataforma giratoria (figura 10.14d), se utilizaron como un escáner de alcance 3D para mapeo 3D.

**Telémetros láser 3D.** Un telémetro láser 3D es un escáner láser que adquiere datos de escáner en más de un plano. Un ejemplo de escáner 3D hecho a la medida ha sido desarrollado en el ASL para el más inteligente (figura 10.14d). En este caso, dos láseres Sick fueron posicionados para mirar en direcciones opuestas. De esta manera, después de media rotación de la plataforma giratoria, un escaneo completo 3D del entorno alrededor del vehículo podría ser adquirido. Estos datos se utilizan principalmente para calcular un modelo 3D consistente digital del terreno del medio ambiente (figura 10.14e). Al reducir la velocidad de rotación de la plataforma giratoria, la resolución angular en la dirección horizontal puede hacerse tan pequeña como se desee.

Figura 10.15: (a) La unidad Velodyne HDL-64E cuenta con 64 rayos láser y gira hasta 15 Hz para recopilar los datos (imagen cortesía de Velodyne). (d) El principio de trabajo del Ibeo Alasca XT (c), el cual utiliza un haz de cuatro capas de láser. El Alasca y Velodyne fueron utilizados por el equipo CMU Tartan Racing (b)

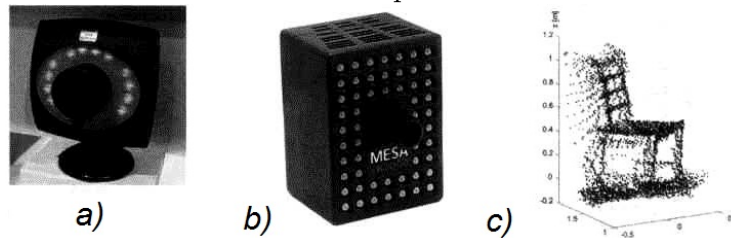


Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

La ventaja de esta configuración es que el campo completo de visión esférico puede ser cubierto ( $360^\circ$  en azimut y  $90^\circ$  en elevación). El inconveniente es que el tiempo de adquisición de un escaneo completo 3D puede tomar hasta varios segundos, dependiendo de la resolución deseada. El Velodyne HDL-64E (figura 10.15a) supera los inconvenientes de los telémetros láser 3D hechos a medida. Este sensor es un lidar 3D que utiliza 64 emisores láser en lugar de único utilizado en los Sick. Este dispositivo gira a una

velocidad de 5-15 Hz y ofrece más de 1.3 millones de puntos de datos por segundo. El campo de visión es de  $360^\circ$  en azimut y  $26,8^\circ$  en elevación y la resolución angular es  $0,09^\circ$  y  $0.4$ , respectivamente. La precisión de la distancia es mejor que 2 cm y se puede medir la profundidad de hasta 50 m, con 10 % de reflectividad, o de hasta 120 m, con 80 % de reflectividad. El Velodyne es actualmente todavía mucho más caro que los telémetros láser Sick.

Figura 10.16: (a) La ZCAM producida por el desarrollador israelí sistemas 3DV, (b) La Swiss Ranger SR3000 producida por la empresa suiza MESA, (c) Imagen de alcance de una silla capturada con una cámara de tiempo de vuelo



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

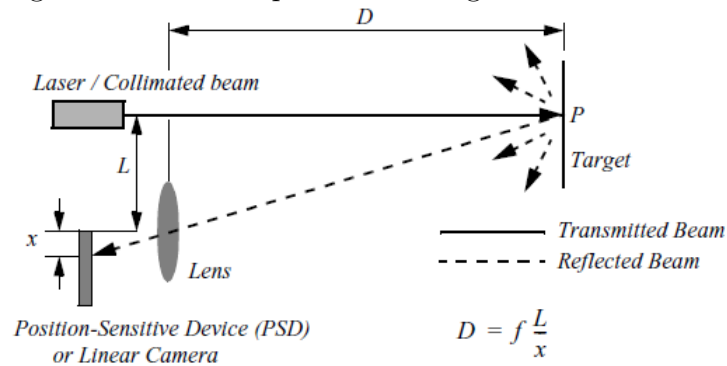
El escáner láser Alasca XT, producido por Ibeo (figura 10.15c-d), por otra parte, divide el haz de láser en cuatro capas verticales. Debido a su principio de exploración de múltiples capas, nos permite cualquier cabeceo del vehículo para ser completamente compensado.

**Cámara de tiempo-de-vuelo.** Una cámara de Tiempo de Vuelo (TOF cámara, figura 10.16) funciona de manera similar a un lidar con la ventaja de que toda la escena 3D se captura al mismo tiempo y que no hay piezas móviles. Este dispositivo utiliza una fuente de luz infrarroja modulada para determinar la distancia de cada píxel de un sensor de dispositivo mezclador fotónico (Photonic Mixer Device - PMD). Como la fuente de iluminación se coloca justo al lado de la lente (figura 10.16), todo el sistema es muy compacto comparado con lidares, visión estéreo, o sensores de triangulación .

### La triangulación basada en rango activo

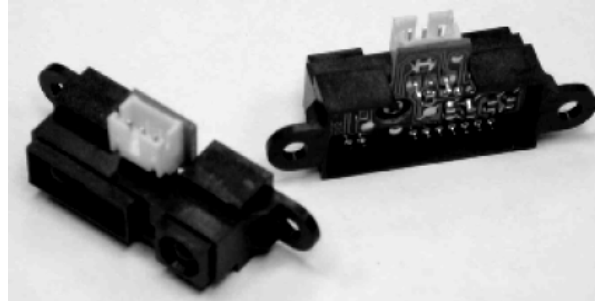
Los sensores de alcance basados en triangulación utilizan propiedades geométricas manifiestas en su estrategia de medición para establecer lecturas de distancia a los objetos. La clase más simple de base de exploradores basados en triangulación son activos

Figura 10.17: Principio de la triangulación de laser 1D



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

Figura 10.18: Sensor de triangulación óptica Sharp



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

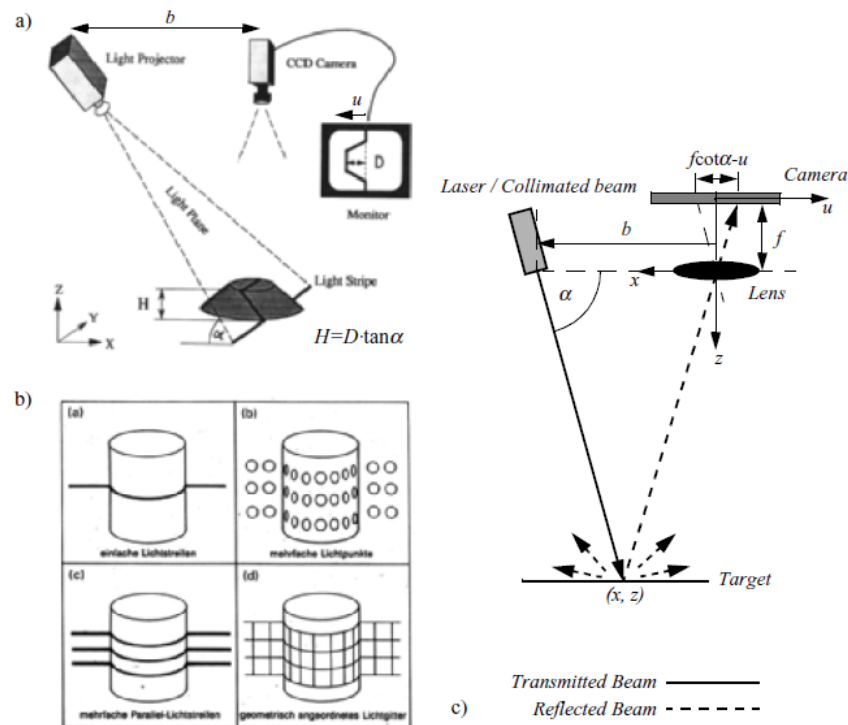
porque proyectan un patrón de luz conocido (por ejemplo, un punto, una línea o una textura) sobre el medio ambiente. La reflexión del patrón conocido es capturada por un receptor y, junto con los valores geométricos conocidos, el sistema puede utilizar la triangulación simple para establecer mediciones de alcance.

**Triangulación óptica (sensor 1D).** El principio de la triangulación óptica en 1D es sencillo, como se representa en la figura 10.17. Un haz colimado se transmite hacia el objetivo. La luz reflejada es recogida por un lente y se proyecta en un dispositivo sensible a la posición (PSD) o una cámara lineal. Dada la geometría de la figura 10.17, la distancia  $D$  está dada por:

$$D = \frac{fL}{x} \quad (10.13)$$

La distancia es proporcional a  $1/x$ , por lo que la resolución del sensor es mejor para los objetos cercanos y se vuelve pobre a una distancia. Los sensores basados en este principio se utilizan en el rango de detección de hasta 1 o 2 m, pero también en medidas de alta precisión industriales con resoluciones muy por debajo de 1 m. El rango de funcionamiento de un dispositivo es normalmente bastante limitado por la geometría. Por ejemplo, el sensor de triangulación óptica representado en la figura 10.18 opera en un rango de distancia de entre 8 y 80 cm. Aunque más limitado en alcance que el sonar, el sensor de triangulación óptica tiene gran ancho de banda y no sufre de sensibilidades cruzadas que son más comunes en el dominio de sonido.

Figura 10.19: (a) Principio de triangulación activa de dos dimensiones. (b) Otras estructuras ligeras posibles. (c) Esquemática 1D del principio.



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

**Luz estructurada (sensor 2D).** Si se sustituye la cámara lineal o PSD de un sensor de triangulación óptica con un receptor 2D, como una cámara CCD o CMOS, luego se puede recuperar la distancia a un gran conjunto de puntos en lugar de un solo punto. El emisor debe proyectar un patrón conocido, o la luz estructurada, sobre

el medio ambiente. Existen muchos sistemas que o proyecta texturas de luz (figura 10.19b) o emiten luz colimada (posiblemente láser) por medio de un espejo giratorio. Sin embargo, otra alternativa popular es el de proyectar una franja de láser (figura 10.19a) pasando un rayo láser en un plano utilizando un prisma. Independientemente de cómo se crea, la luz proyectada tiene una estructura conocida, y por lo tanto la imagen tomada por el receptor CCD o CMOS puede ser filtrada para identificar la reflexión del patrón.

La figura 10.19c muestra una geometría de triangulación activa 1D. Podemos examinar la compensación en el diseño de sistemas de triangulación mediante el examen de la geometría en la figura 10.19c. Los valores medidos en el sistema son  $\alpha$  y  $u$ , la distancia del punto de iluminación desde el origen en el sensor de imagen. El sensor de imagen aquí puede ser una cámara o un conjunto de fotodiodos de un dispositivo sensible a la posición, por ejemplo, un PSD 2D. De la figura 10.19c, la geometría simple muestra que:

$$x = \frac{b \cdot u}{f \cot \alpha + u} \quad z = \frac{b \cdot f}{f \cot \alpha + u} \quad (10.14)$$

Donde  $f$  es la distancia de la lente al plano de la imagen. En el límite, la relación de resolución de imagen a resolución de alcance se define como la ganancia de triangulación  $G_p$  y de la ecuación 10.14 está dada por:

$$\frac{\partial u}{\partial z} = G_p = \frac{c \cdot f}{z^2} \quad (10.15)$$

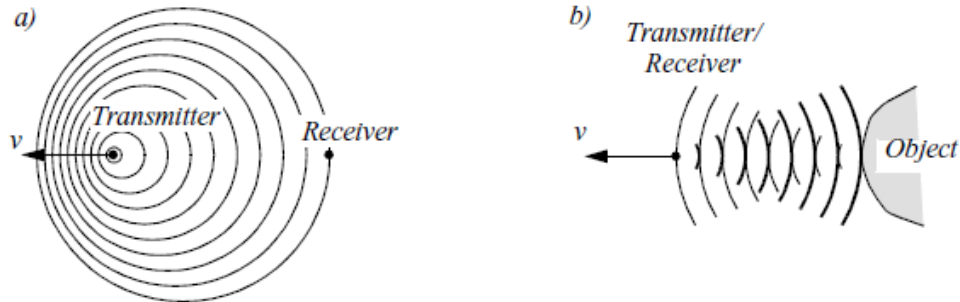
Esto muestra que la precisión de alcance, para una resolución de la imagen dada, es proporcional a la separación  $b$  de fuente/detector y la distancia focal  $f$ , y disminuye con el cuadrado del rango  $z$ . En un sistema de exploración que oscila, existe un efecto adicional en la precisión de alcance, causada por la medición del ángulo de proyección  $\alpha$ . De la ecuación 10.14 se observa que:

$$\frac{\partial \alpha}{\partial z} = G_\alpha = \frac{b \cdot \sin \alpha^2}{z^2} \quad (10.16)$$

Se pueden resumir los efectos de los parámetros de la precisión del sensor de la siguiente manera:

- Longitud de línea de base ( $b$ ): entre más pequeño sea  $b$ , más compacto el sensor puede ser. Entre mayor sea  $b$ , mejor será la resolución en distancia será.
- Detector de longitud y la longitud focal ( $f$ ): Una longitud de detector más grande puede proporcionar ya sea un gran campo de visión o un rango de resolución mejorada o beneficios parciales para ambos.

Figura 10.20: Efecto Doppler entre dos objetos móviles (a) o un objeto en movimiento y fijas a (b)



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

### 10.1.9. Sensores de movimiento y velocidad

Algunos sensores miden directamente el movimiento relativo entre el robot y su entorno.

#### Detección basada en efecto Doppler (Radar o sonido)

Cualquiera que haya notado el cambio en el tono de sirena que se produce cuando un camión de bomberos se acerca, pasa y se aleja está familiarizado con el efecto Doppler. Un transmisor emite una onda electromagnética o de sonido con una frecuencia  $f_t$ . Es bien recibida por un receptor (figura 10.20a) o reflejada por un objeto (figura 10.20b). La frecuencia medida  $f_r$  en el receptor es una función de la velocidad relativa  $v$  entre el transmisor y el receptor según:

$$f_r = f_t \frac{1}{1 + \frac{v}{c}} \quad (10.17)$$

Si el transmisor se está moviendo y:

$$f_r = f_t \left(1 + \frac{v}{c}\right) \quad (10.18)$$

Si el receptor esta en movimiento. En el caso de una onda reflejada (figura 10.20b) hay un factor de 2 introducido, ya que cualquier cambio  $x$  en la separación relativa afecta a la longitud del camino de ida y vuelta por  $2x$ . Además, en tales situaciones, generalmente es más conveniente considerar el cambio en la frecuencia  $\Delta f$ , conocido

como el efecto Doppler, en oposición a la notación de frecuencia Doppler mostrada anteriormente:

$$\Delta f = f_t - f_r = \frac{2f_t v \cos \theta}{c} \quad (10.19)$$

$$v = \Delta f \frac{c}{2f_t \cos \theta} \quad (10.20)$$

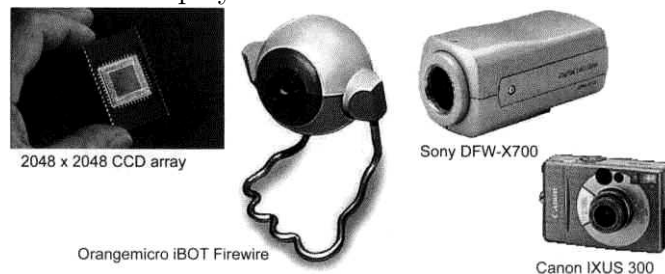
Dónde:  $\Delta f$  = Desplazamiento de frecuencia Doppler.

$\theta$  = ángulo relativo entre la dirección de movimiento y el eje del haz.

El efecto Doppler se aplica a las ondas sonoras y electromagnéticas. Tiene un amplio espectro de aplicaciones:

- Las ondas de sonido: por ejemplo, el control de procesos industriales, la seguridad, la localización de peces, medida de la velocidad de avance.
- Las ondas electromagnéticas: por ejemplo, medición de la vibración, sistemas de radar, seguimiento de objetos.

Figura 10.21: CCD chips y cámaras CCD comercialmente disponibles.



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

## 10.2. FUNDAMENTOS DE LA VISIÓN POR COMPUTADOR

La visión es nuestro sentido más poderoso. Nos proporciona una enorme cantidad de información sobre el medio ambiente y permite la interacción rica e inteligente en entornos dinámicos. Por lo tanto, no es sorprendente que una gran cantidad de esfuerzo se ha dedicado a la provisión de máquinas con sensores que imitan las capacidades del sistema de visión humana. El primer paso en este proceso es la creación de dispositivos

sensores que captan la luz, la misma información en bruto que el sistema de visión humano utiliza. A continuación se describen las dos tecnologías actuales para la creación de sensores de visión: CCD y CMOS.

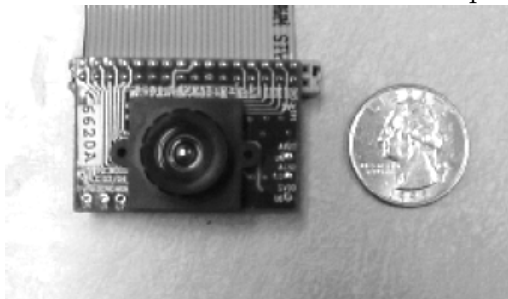
### 10.2.1. La cámara digital

Los dos principales tipos de sensores utilizados en cámaras digitales y de vídeo de hoy son CCD (Charge Coupled Device dispositivo de carga acoplada) y CMOS (Complementary Metal Oxide On Silicon - óxido de metal complementario en el silicio). A continuación, se revisan las ventajas y desventajas de estas dos tecnologías.

**Cámaras CCD.** El chip CCD (figura 10.21) es una matriz de elementos de imagen sensibles a la luz, o píxeles, por lo general entre 20000 a varios millones de píxeles totales. Cada píxel puede ser pensado como un condensador de descarga sensible a la luz, que es 5 a 25 micras de tamaño. En primer lugar, los condensadores de todos los píxeles son completamente cargados, luego, comienza el período de integración. Cuando los fotones de luz golpean cada píxel, liberan electrones, los cuales son capturados por los campos eléctricos y retenido en el píxel. Con el tiempo, cada píxel se acumula un nivel variable de carga basado en el número total de fotones que han golpeado. Después de que el periodo de integración se completa, las cargas relativas de todos los píxeles deben ser congeladas y leídas. En un CCD, el proceso de lectura se realiza en una esquina del chip CCD. La fila inferior de las cargas de pixel se transporta a este rincón y se leen, luego, las filas por encima bajan la marcha y se repite el proceso. Esto significa que cada carga debe ser transportada a través del chip, y es crítico que el valor sea preservado. Esto requiere un circuito de control especializado y técnicas personalizadas de fabricación para garantizar la estabilidad de las cargas transportadas.

**Cámaras CMOS.** El chip semiconductor de óxido de metal complementario es una salida importante de la CCD. Este, también, tiene una matriz de píxeles, pero situada a lo largo del lado de cada píxel hay varios transistores específicos a ese píxel. Como en los chips CCD, todos los píxeles acumulan una carga durante el período de integración. Durante la etapa de recogida de datos, el CMOS toma un nuevo enfoque: el circuito de píxel específico junto a cada píxel mide y amplifica la señal del píxel, todo en paralelo para cada píxel de la matriz. Mediante rastreos más tradicionales de chips semiconductores generales, los valores de los píxeles resultantes se llevan a sus destinos. El chip CMOS, figura 10.22 es mucho más simple ya que consume mucha menos energía; increíblemente, opera con un consumo de energía que es una centésima

Figura 10.22: Una cámara CMOS comercialmente disponible y de bajo costo.



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

parte del consumo de energía de un chip CCD.

Figura 10.23: Filtro Bayer de color de matriz

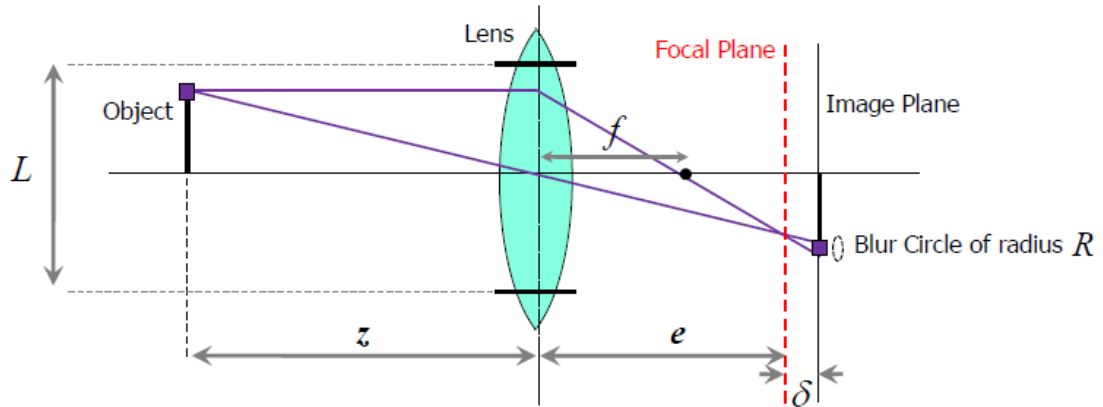


Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

**Cámara a color.** Hay dos métodos comunes para la creación de imágenes en color, que utilizan un único chip o tres chips separados. La tecnología de chip único utiliza el llamado filtro de Bayer. Los píxeles en el chip se agrupan en 4 grupos de 2 x 2, y luego los filtros de color rojo, verde, y azul se aplican de manera que cada píxel individual recibe sólo luz de un color. Normalmente, dos píxeles de cada bloque 2 x 2 mide el verde mientras que los otros dos píxeles miden la intensidad de la luz roja y azul (figura 10.23).

La cámara a color de tres chips evita estos problemas mediante la división de la luz entrante en tres copias completas (menor intensidad). Tres chips separados reciben la luz, con un filtro rojo, verde, o azul sobre cada chip. Por lo tanto, en paralelo, cada chip mide la intensidad de luz de un color, y la cámara debe combinar las salidas de los chips para crear una imagen colectiva en color. La resolución se conserva en esta solución, aunque las cámaras en color de tres chips son, como era de esperar, mucho

Figura 10.24: Representación de la óptica de la cámara y su impacto en la imagen



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

más caras y por lo tanto rara vez se utilizan más en la robótica móvil.

## 10.2.2. Formación de la imagen

### Óptica

Una vez que la forma de la luz de la escena llega a la cámara, todavía debe pasar a través de la lente antes de llegar al sensor. La figura 10.24 muestra un diagrama del modelo de lente más básico, que es la lente delgada. Esta lente se compone de una única pieza de vidrio con muy baja curvatura igual en ambos lados. De acuerdo con la ley de lente, la relación entre la distancia a un objeto  $z$  y la distancia detrás de la lente en la que se forma una imagen enfocada  $e$  se expresa como:

$$\frac{1}{f} = \frac{1}{z} + \frac{1}{e} \quad (10.21)$$

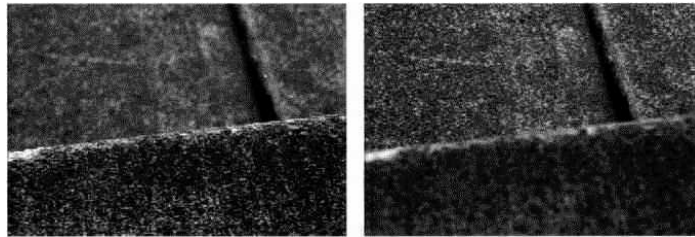
Donde  $f$  es la longitud focal. Como se puede percibir, esta fórmula también se puede utilizar para estimar la distancia a un objeto, conociendo la longitud focal y la distancia actual del plano de la imagen a la lente. Esta técnica se llama profundidad de foco. Si el plano de imagen está situado en la distancia  $e$  de la lente, a continuación, para el voxel específico del objeto representado, toda la luz será enfocada en un punto único en el plano de la imagen y el voxel del objeto se enfoca. Sin embargo, cuando el plano de la imagen no está en  $e$ , como se representa en la figura 10.24, entonces la luz del voxel objeto será emitida en el plano de la imagen como un círculo borroso (o círculo confusión). En una primera aproximación, la luz se distribuye homogéneamente a lo

largo de círculo borroso, y el radio  $R$  del círculo puede ser caracterizado de acuerdo con la ecuación:

$$R = \frac{L\delta}{2e} \quad (10.22)$$

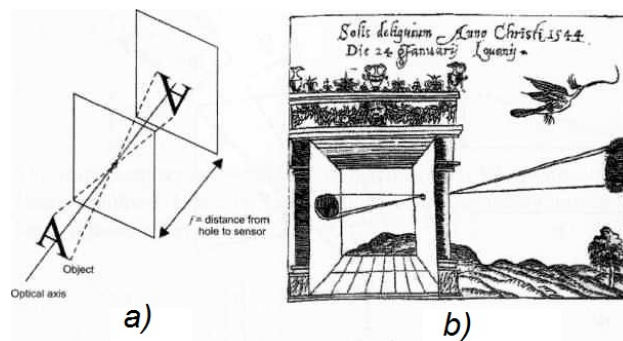
$L$  es el diámetro de la lente o la abertura, y  $\delta$  es el desplazamiento del plano de imagen desde el punto focal. Si la abertura o la lente se reducen a un punto, como en una cámara de agujero de alfiler, entonces el radio del círculo borroso se aproxima a cero. La segunda propiedad que se puede deducir de estas ecuaciones ópticas se refiere a la sensibilidad de desenfoque como una función de la distancia de la lente al objeto.

Figura 10.25: Dos imágenes de la misma escena tomada con una cámara en dos posiciones de enfoque diferentes



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

Figura 10.26: (a) Cuando  $d \gg f$  y  $d \gg l$  la cámara puede ser modelada como una cámara estenopeica. (b) La cámara oscura en un dibujo del matemático Reinerus Gemma-Frisius.

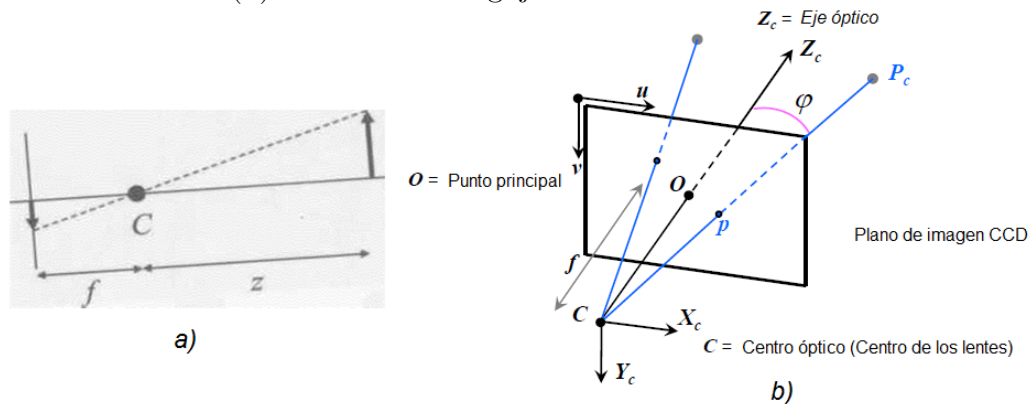


Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

Dados los efectos físicos resumidos en las ecuaciones anteriores, se puede imaginar un sensor visual de alcance que haga uso de varias imágenes en las que se varíe la óptica de

la cámara (por ejemplo, el desplazamiento de plano de la imagen  $\delta$ ) y la misma escena se captura (figura 10.25). La profundidad del método de enfoque es una de las técnicas de alcance visuales más simples. Para determinar el intervalo a un objeto, el sensor se mueve simplemente el plano de la imagen (por medio de enfoque) hasta maximizar la nitidez del objeto. Cuando la nitidez se maximiza, la posición correspondiente del plano de la imagen directamente informa el rango (alcance).

Figura 10.27: (a) Modelo de cámara estenopeica utiliza para representar las cámaras de perspectiva estándar. (b) El modelo de agujero de alfiler



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

### Modelo de cámara estenopeica

Una cámara estenopeica no tiene ninguna lente, sino una sola abertura muy pequeña. En resumen, se trata de una caja a prueba de luz con un pequeño agujero en un lado. La luz de la escena pasa a través de este único punto y proyecta una imagen invertida en el lado opuesto de la caja (figura 10.26).

Este modelo se puede obtener directamente de la ecuación 10.21. De hecho, observe que si  $z \rightarrow \infty$ , es decir, que se ajusta la lente (mover el plano de la imagen), de modo que los objetos en el infinito estén en foco, obtenemos  $e = f$ , por lo que podemos pensar en una lente de distancia focal  $f$ , sea equivalente a una distancia de agujero de alfiler  $f$  a partir del plano focal (figura 10.27a). Por conveniencia, la cámara oscura se representa comúnmente con el plano de la imagen entre el centro de proyección y la escena (figura 10.27b). Esto se realiza para que la imagen preserve la misma orientación que el objeto, es decir, la imagen no se invierte. La intersección  $O$  entre el eje óptico y el plano de la imagen se denomina punto principal. Como se muestra en la figura 10.27b, observar que

una cámara no mide distancias, sino ángulos y por lo tanto puede ser pensada como un sensor de orientación.

### Proyección de perspectiva

Para describir analíticamente la proyección de perspectiva operada por la cámara, tenemos que introducir algún sistema de referencia oportuno en el que podamos expresar las coordenadas 3D del punto de la escena  $P$  y las coordenadas de su proyección  $p$  en el plano de la imagen. Primero se considerará un modelo simplificado y, finalmente, el modelo general.

**Modelo Simplificado.** Sea  $(x, y, z)$  el marco de referencia de cámara con origen en  $C$  y con eje  $z$  coincidente con el eje óptico. Se supone también que el marco de referencia de la cámara coincide con el marco de referencia mundial. Esto implica que las coordenadas del punto de escena  $P$  se expresan ya en el marco de la cámara. Se introducirá también un marco de referencia de dos dimensiones  $(u, v)$  para el plano de imagen  $\Pi$  con origen en  $O$  y los ejes alineados  $u$  y  $v$  como  $x$  y  $y$ , respectivamente, como se muestra en la figura 10.27b. Finalmente, se toma  $P = (x, y, z)$  y  $p = (u, v)$ . Por medio de simples consideraciones sobre la similitud de los triángulos, se puede escribir:

$$\frac{f}{z} = \frac{u}{x} = \frac{v}{y} \quad (10.23)$$

Y por lo tanto:

$$u = \frac{f}{z} \cdot x \quad (10.24)$$

$$v = \frac{f}{z} \cdot y \quad (10.25)$$

Esta es la proyección en perspectiva. El mapeo de coordenadas 3D a coordenadas 2D es claramente no lineal. Sin embargo, usando en cambio coordenadas homogéneas nos permite obtener ecuaciones lineales. Sean:

$$\tilde{p} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad y \quad \tilde{P} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (10.26)$$

Las coordenadas homogéneas de  $p$  y  $P$ , respectivamente. De ahora en adelante se utilizará el superíndice  $--$  para indicar las coordenadas homogéneas. La ecuación de

proyección, en este caso simplificado, se puede escribir como:

$$\begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = \begin{bmatrix} fx \\ fy \\ z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (10.27)$$

$\lambda$  es igual a la tercera coordenada de  $P$  la cual, en este marco de referencia especial, coincide con la distancia del punto al plano  $xy$ . Esta ecuación muestra también que cada punto de la imagen es la proyección de todos los infinitos puntos 3D tendidos en el rayo que pasan por el mismo punto de la imagen y el centro de proyección (figura 10.27b). Por lo tanto, utilizando una cámara estenopeica sola, no es posible calcular la distancia a un punto, pero necesitamos dos cámaras (es decir, una cámara estéreo, sección 10.2.4).

**Modelo general.** Un modelo de cámara realista que describa la transformación de coordenadas 3D a coordenadas de píxeles también debe tener en cuenta:

- La pixelización, esto es, la forma (tamaño) del CCD y su posición con respecto al centro óptico.
- La transformación rígida entre la cámara y la escena (es decir, el mundo).

La pixelización tiene en cuenta el hecho de que:

- El centro óptico de la cámara tiene coordenadas de píxeles  $(u_0, v_0)$  con respecto a la esquina superior izquierda de la imagen, la cual comúnmente se asume como origen del sistema de coordenadas de imagen. Note que el centro óptico en general no corresponde con el centro del CCD.
- Las coordenadas de un punto en el plano de la imagen se mide en píxeles. Por lo tanto, hay que introducir un factor de escala.
- La forma del píxel es, en general, se supone no perfectamente cuadrada y por lo tanto hay que utilizar dos diferentes factores de escala  $k_u$  y  $k_v$  a lo largo de las direcciones horizontal y vertical, respectivamente.
- Los ejes  $u$  y  $v$  podría ser no ortogonal pero desalineados un ángulo  $\theta$ . Estos modelos, por ejemplo, el hecho de que la lente puede no ser paralelo al CCD.

Los tres primeros puntos se tratan por medio de la traslación de centro óptico y el cambio de escala individual de los ejes  $u$  y  $v$ :

$$u = k_u \cdot \frac{f}{z} \cdot x + u_0 \quad (10.28)$$

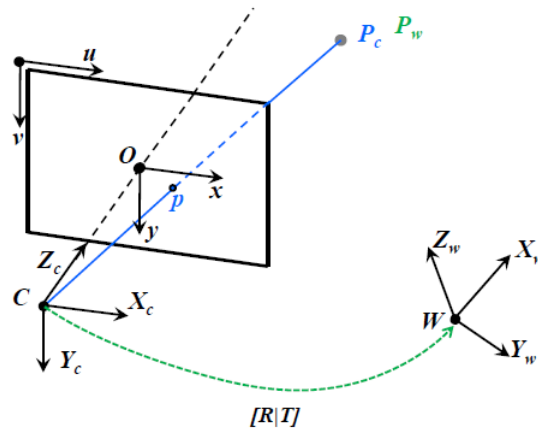
$$v = k_v \cdot \frac{f}{z} \cdot y + v_0 \quad (10.29)$$

Donde  $(u_0, v_0)$  son las coordenadas del punto principal,  $k_u(k_v)$  es la inversa del tamaño de píxel efectivo a lo largo de la dirección  $u(v)$  y se mide en píxeles  $m^{-1}$ . Después de esta actualización las ecuaciones de proyección en perspectiva se convierten en:

$$\begin{bmatrix} \lambda_u \\ \lambda_v \\ \lambda \end{bmatrix} = \begin{bmatrix} fk_u & 0 & u_0 & 0 \\ 0 & fk_v & u_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (10.30)$$

Se observa que podemos plantear  $\alpha_u = fk_u$  y  $\alpha_v = fk_v$  las cuales describen las longitudes focales expresadas en píxeles horizontales y verticales, respectivamente.

Figura 10.28: Cambio de coordenadas entre la cámara y el marco de referencia del mundo



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

Para tener en cuenta el hecho de que en general el sistema de referencia del mundo  $(x_w, y_w, z_w)$  no coincide con el sistema de referencia de la cámara  $(x, y, z)$ , tenemos que introducir la transformación de cuerpo rígido entre los dos marcos de referencia (figura 10.28). Ahora, se introduce un cambio de coordenadas compuesto por una rotación  $R$  seguida de una traslación  $t$ , por lo tanto:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + t \quad (10.31)$$

Usando esta transformación, la ecuación 10.30 se puede reescribir como:

$$\begin{bmatrix} \lambda_u \\ \lambda_v \\ \lambda \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \lambda_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (10.32)$$

O bien, con las coordenadas homogéneas 10.26:

$$\lambda \tilde{p} = A[R|t]\tilde{P}_w \quad (10.33)$$

Donde:

$$A = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \lambda_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (10.34)$$

Es la matriz de parámetros intrínsecos. Como se anticipó, el modelo más general también toma en consideración la posibilidad de que los ejes  $u$  y  $v$  no son ortogonales pero están inclinados un ángulo  $\theta$ . Por lo tanto, la forma más general para  $A$  es:

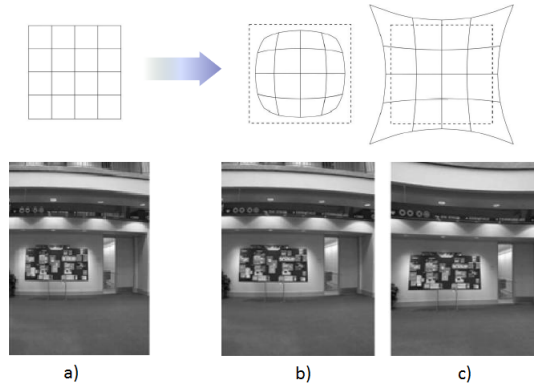
$$A = \begin{bmatrix} \alpha_u & \alpha_u \cot \theta & u_0 \\ 0 & \lambda_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (10.35)$$

Cuando  $\alpha_u \cot \theta$  se puede absorber en un solo parámetro  $\alpha_c$ .  $\alpha_c, \alpha_u, \alpha_u c, \alpha_0, v_0$  se llaman parámetros intrínsecos de la cámara. Los parámetros de rotación y traslación  $R$  y  $t$  se denominan parámetros extrínsecos de la cámara. Los parámetros intrínsecos y extrínsecos pueden estimarse mediante un procedimiento llamado calibración de la cámara que en breve se describen más adelante.

**Distorsión radial.** Un modelo preciso de la cámara por lo tanto, debe también tener en cuenta la distorsión radial de la lente, especialmente para lentes con distancia focal corta (es decir, un gran campo de vista) (figura 10.29). El modelo estándar de distorsión radial es una transformación de las coordenadas ideales (es decir, sin distorsiones)  $(u, v)$  a las coordenadas observables reales (distorsionadas)  $(u_d, v_d)$ . Dependiendo del tipo de distorsión radial, las coordenadas de las imágenes observadas son desplazadas lejos (distorsión de barril) o hacia (distorsión de acerico) el centro de la imagen. La cantidad de distorsión de las coordenadas de la imagen observada es una función no lineal de su distancia radial  $r$ . Para la mayoría de las lentes, un simple modelo cuadrático de distorsión produce buenos resultados:

$$\begin{bmatrix} u_d \\ v_d \end{bmatrix} = (1 + k_1 r^2) \begin{bmatrix} u - u_0 \\ v - v_0 \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} \quad (10.36)$$

Figura 10.29: Ejemplo de distorsión radial de la lente: (a) Ninguna distorsión, (b) Distorsión de barril, (c) acerico



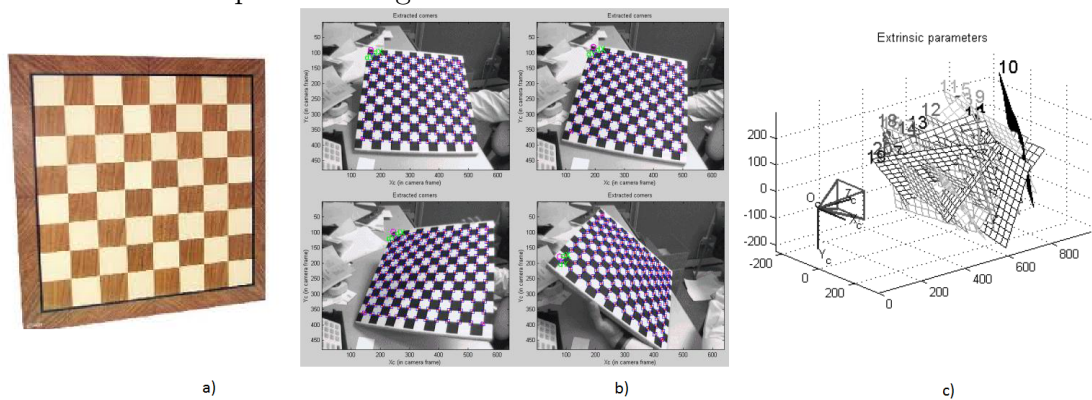
Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

Donde:

$$r^2 = (u - u_0)^2 + (v - v_0)^2 \quad (10.37)$$

Y  $k_1$  es el parámetro de distorsión radial, que puede ser estimado por calibración de la cámara. Las lentes de ojo de pez requieren un modelo diferente a los modelos polinomiales tradicionales de distorsión radial y será presentado en la sección 10.2.3.

Figura 10.30: Imágenes de la Caja de herramientas de calibración de la cámara para Matlab desarrollada por JY Bouguet



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

## Calibración de la cámara

La calibración consiste en medir con precisión los parámetros intrínsecos y extrínsecos del modelo de cámara. A medida que estos parámetros gobiernan la forma en que los puntos de la escena son mapeados a sus puntos de imagen correspondientes, la idea es que al conocer las coordenadas de píxeles de los puntos de imagen  $\tilde{P}$  y las coordenadas 3D de los puntos correspondientes de la escena  $\tilde{P}$ , es posible calcular los parámetros desconocidos  $A$ ,  $R$  y  $T$  mediante la resolución de la ecuación de proyección en perspectiva 10.33.

La red plana más común es un patrón de tablero de ajedrez debido a la facilidad de extracción de sus esquinas, que luego se utilizan para la calibración (figura 10.30) donde (a) es un ejemplo de tablero de ajedrez como patrón utilizado en la calibración de la cámara con esquinas extraídas, en (b) se observan varias fotos de patrón con diferentes orientaciones, (c) es la posición reconstruida y orientaciones del patrón después de la calibración. El método requiere que el usuario tome varias imágenes del patrón mostrado en posiciones y orientaciones diferentes. Al conocer la posición 2D de las esquinas en el patrón real y las coordenadas de píxeles de sus correspondientes esquinas de cada imagen, los parámetros intrínsecos y extrínsecos (incluyendo la distorsión radial y tangencial) se determinan simultáneamente resolviendo una minimización lineal de mínimos cuadrados seguido por un refinamiento no lineales (es decir, Gauss-Newton).

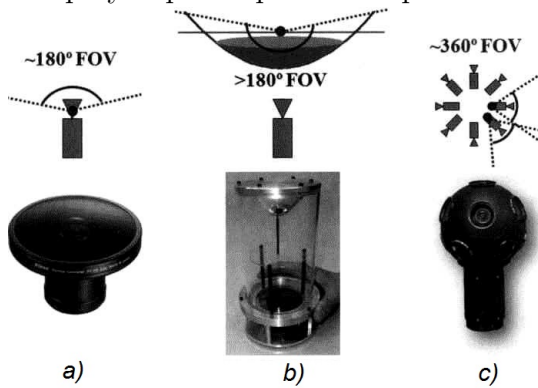
### 10.2.3. Cámaras omnidireccionales

#### Introducción

En la sección anterior, se ha descrito la formación de la imagen de la cámara de agujero de alfiler, que se modela como una proyección en perspectiva. Sin embargo, hay sistemas de proyección cuya geometría no se puede describir mediante el modelo de agujero de alfiler convencional debido a la distorsión muy alta introducida por el dispositivo de formación de imágenes. Algunos de estos sistemas son las cámaras omnidireccionales.

Una cámara omnidireccional es una cámara que ofrece amplio campo de visión, por lo más de 180 grados. Hay varias maneras de construir una cámara omnidireccional. Las cámaras diópticas utilizan una combinación de lentes con forma (por ejemplo, lentes de ojo de pez; véase la figura 10.31a) y típicamente puede alcanzar un campo de visión ligeramente mayor que 180 grados. Las cámaras catadiópticas combinan una cámara estándar con un espejo de forma, como un espejo parabólico, hiperbólico o elíptico y son capaces de proporcionar mucho más que 180 grados de campo de vista en elevación y

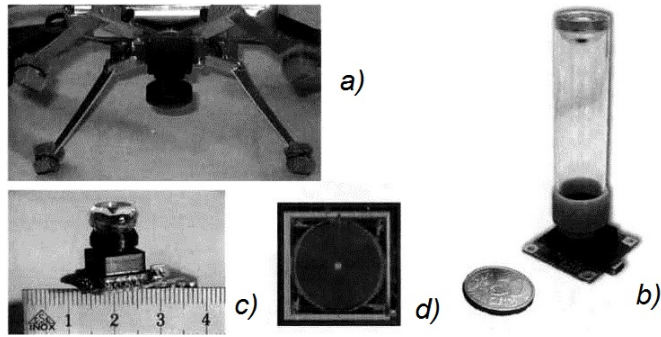
Figura 10.31: Cámara dioptr ca (por ejemplo, ojo de pez), (b) C mara catadi ptrica, (c) una ejemplo de c mara polydi ptrica producida por Immersive Media



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

360 en la direcci n azimutal. En la figura 10.31b se puede ver una c mara catadi ptrica utilizando un espejo hiperb lico. Por  ltimo, las c maras polidi ptricas utilizan varias c maras superpuestas con el campo de visi n (figura 10.31c) y hasta el momento son las  nicas c maras capaces de proporcionar una verdadera vista omnidireccional (esf rica) (es decir,  $4\pi$  radianes).

Figura 10.32: (a) La lente de ojo de pez de Omnitech Rob tica proporciona un campo de visi n de 190 grados. (b) Una c mara miniatura catadi ptrica construida en el ETH Zurich (c) La c mara Mufly construida por CSEM. Adem s, se utiliza un CCD polar (d) donde los p xeles est n dispuestos radialmente



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

S lo a partir de 2005 que estas c maras se han miniaturizado al tama o de 1-2 cent 

metros, y su campo de visión se ha incrementado hasta 190 grados (ver, por ejemplo, figura 10.32a).

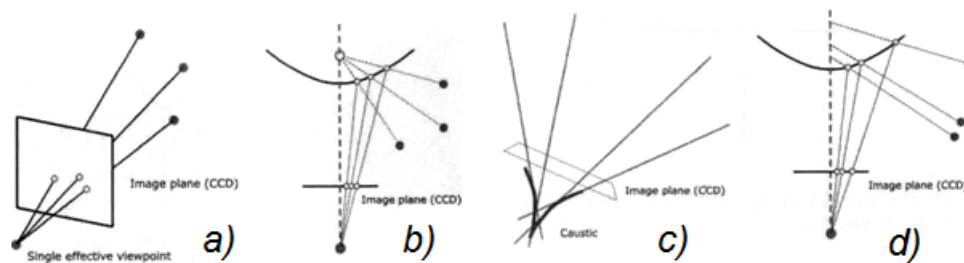
Figura 10.33: (a) Una cámara catadióptrica omnidireccional usando un espejo hiperbólico. (b) Lente de ojo de pez Nikon FC-E8.



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

La figura 10.32 muestra algunos de los ejemplos de cámaras omnidireccionales miniatura utilizadas en el estado del arte de los vehículos micro-aéreos. Dos ejemplos de imágenes omnidireccionales se muestran en la figura 10.33.

Figura 10.34: Ejemplos de cámaras centrales y no centrales



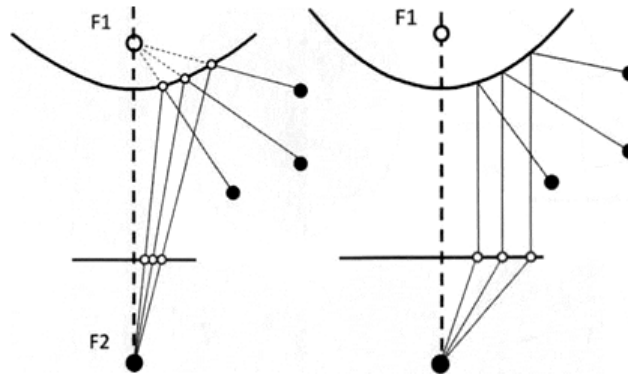
Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

### Cámaras omnidireccionales Centrales

Un sistema de visión se dice que es central cuando los rayos ópticos a los objetos vistos se cruzan en un punto único llamado centro de la proyección 3D o punto de vista efectivo único (figura 10.34). Esta propiedad se llama propiedad punto de vista

efectivo único. La perspectiva de la cámara es un ejemplo de un sistema de proyección central, porque todos los rayos ópticos se cortan en un punto, es decir, el centro óptico de la cámara. Todas las cámaras de ojo de pez modernas son centrales, y por lo tanto, satisfacen la propiedad solo punto de vista efectivo. Las cámaras centrales catadióptricas por lo contrario sólo se pueden construir oportunamente eligiendo la forma del espejo y la distancia entre la cámara y el espejo. La familia de los espejos que satisfacen el punto de vista único adecuado es la clase de secciones cónicas rotadas (barrido), es decir, espejos hiperbólicos, parabólicos, y elípticos. En el caso de los espejos hiperbólicos y elípticos, la propiedad de punto de vista único se consigue asegurando que el centro de la cámara coincida con uno de los focos de la hipérbola (figura 10.35). En el caso de espejos parabólicos, una lente ortográfica debe ser interpuesta entre la cámara y el espejo, esto hace posible que los rayos paralelos reflejados por el espejo parabólico converjan al centro de la cámara (figura 10.35). La razón de un punto de vista único efectivo sea tan

Figura 10.35: Cámaras catadióptricas centrales pueden ser construidas usando espejos hiperbólicos y parabólicos.



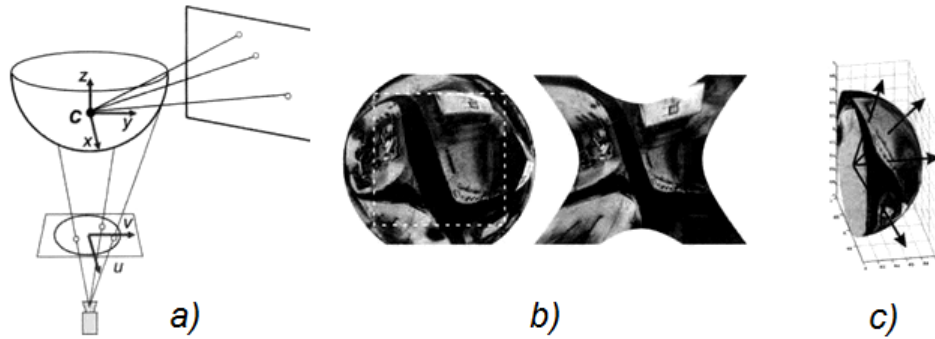
Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

deseable es que nos permite generar imágenes en perspectiva geoméricamente correctas de las imágenes captadas por la cámara omnidireccional (figura 10.36). También la imagen puede ser mapeada a una esfera centrada en el único punto de vista, es decir, la proyección esférica (figura 10.36, parte inferior).

### Modelo de cámara omnidireccional y calibración

Intuitivamente, el modelo de una cámara omnidireccional es un poco más complicado que una cámara de perspectiva estándar. La aproximación de un modelo de lente

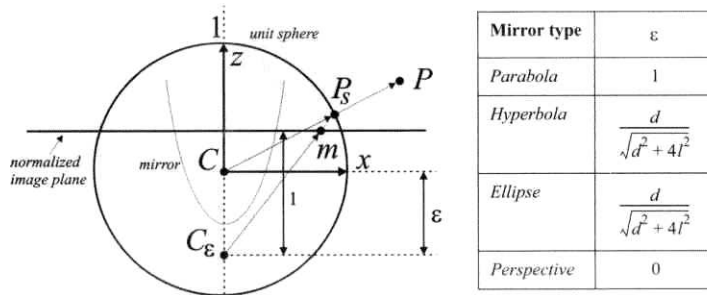
Figura 10.36: Proyecciones de cámaras centrales



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

de ojo de pez por un catadióptrico es generalmente posible, sin embargo, sólo con una precisión limitada.

Figura 10.37: Modelo de proyección unificado para cámaras centrales catadióptricas de Geyer y Daniilidis



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

Por el contrario, el segundo modelo unifica tanto las cámaras centrales catadióptricas y las cámaras de ojo de pez bajo un modelo general también conocido como modelo de Taylor.

**Modelo unificado para las cámaras centrales catadióptricas.** Cada proyección en perspectiva catadióptrica (parabólica, hiperbólica, elíptica) y estándar es equivalente a un mapeo proyectivo de una esfera, centrada en el único punto de vista, a un plano con el centro de proyección en la perpendicular al plano y distante  $\epsilon$  del centro de la esfera. Esto se resume en la figura 10.37. El modelo de proyección de Geyer y

Daniilidis sigue un proceso de cuatro pasos. Sea de nuevo  $P = (x, y, z)$  un punto de la escena en el marco de referencia del espejo centrado en  $C$  (figura 10.37).

1. El primer paso consiste en proyectar el punto de la escena sobre la esfera unidad, por lo tanto:

$$P_s = \frac{P}{\|P\|} = (x_s, y_s, z_s) \quad (10.38)$$

2. Las coordenadas del punto se cambian entonces a un nuevo marco de referencia centrado en  $C_\varepsilon = (0, 0, -\varepsilon)$ , por lo tanto:

$$P_\varepsilon = (x_s, y_s, z_s + \varepsilon) \quad (10.39)$$

$\varepsilon$  oscila entre 0 (espejo plano) y 1 (espejo parabólico). El valor correcto de  $\varepsilon$  puede obtenerse conociendo la distancia  $d$  entre los focos de la cónica y el lado recto  $l$  como se resume en la tabla de la figura 10.37.

3.  $P_\varepsilon$  se proyecta en el plano de la imagen normalizada a distancia 1 de  $C_\varepsilon$ , por lo tanto:

$$\tilde{m} = (x_m, y_m, 1) = \left( \frac{x_s}{z_s + \varepsilon}, \frac{y_s}{z_s + \varepsilon}, 1 \right) = g^{-1}(P_s) \quad (10.40)$$

4. Por último, el punto  $\tilde{m}$  se mapea al punto de imagen de la cámara  $p = (u, v, 1)$  a través de la matriz de parámetro intrínseca  $A$ ; por lo tanto:

$$\tilde{p} = A \cdot \tilde{m} \quad (10.41)$$

Donde  $A$  está dada por 10.35, es decir:

$$A = \begin{bmatrix} \alpha_u & \alpha_u \cot \theta & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (10.42)$$

Es fácil demostrar que la función  $g^{-1}$  es biyectiva y que su  $g$  inversa viene dada por:

$$P_s = g(m) = \begin{bmatrix} x_m \\ y_m \\ 1 - \varepsilon \frac{x_m^2 + y_m^2 + 1}{\sqrt{1 + (1 - \varepsilon^2)(x_m^2 + y_m^2)}} \end{bmatrix} \quad (10.43)$$

Donde  $--$  indica que  $g$  es proporcional a la cantidad en el lado derecho. Para obtener el factor de escala, es suficiente para normalizar  $g(m)$  sobre la esfera unidad. Observe que la ecuación 10.43 es el núcleo del modelo de proyección de las cámaras centrales catadióptricas.

**Modelo unificado para cámaras catadióptricas y de ojo de pez.** Este modelo unificado fue propuesto por Scaramuzza et al., en 2006. La principal diferencia con el modelo anterior reside en la elección de la función  $g$ . Para superar la falta de conocimiento de un modelo paramétrico para cámaras de ojo de pez, los autores propusieron el uso de un polinomio de Taylor, cuyos coeficientes y grado se encuentran a través del proceso de calibración. En consecuencia, la relación entre punto normalizado de la imagen  $\tilde{m} = (x_m, y_m, 1)$  y el vector unidad  $P_s$  en el marco de referencia de ojo de pez (espejo) se puede escribir como:

$$P_s = g(m) -- \begin{bmatrix} x_m \\ y_m \\ a_0 + a_2\rho^2 + \dots + a_N\rho^N \end{bmatrix} \quad (10.44)$$

Donde  $p = \sqrt{x_m^2 + y_m^2}$ . Como se habrá notado, el término de primer orden (es decir, el  $a_1\rho$ ) del polinomio no se encuentra. Esto se deduce de la observación de que la primera derivada del polinomio calculado en  $\rho = 0$  debe ser nulo tanto para la cámara catadióptrica como de ojo de pez (esto es sencillo de verificar para cámaras catadióptricas derivando 10.43).

**Calibración de la cámara omnidireccional.** La calibración de las cámaras omnidireccionales es similar a aquella para calibrar las cámaras de perspectiva estándar, que se ha estudiado anteriormente.

## 10.2.4. Estructura de estéreo

### Introducción

Cualquier chip de visión se derrumba el mundo 3D en un plano imagen 2D, perdiendo así la información de profundidad. La solución general es la recuperación de profundidad al ver varias imágenes de la escena para obtener más información recuperando profundidad. La visión estéreo procesa dos imágenes diferentes tomadas al mismo tiempo, y se supone que la posición relativa entre las dos cámaras se conoce. La estructura de movimiento a la inversa procesa dos imágenes tomadas con la misma cámara o una diferente en momentos diferentes y desde diferentes posiciones desconocidas, el

problema consiste en recuperar tanto el movimiento relativo entre los puntos de vista y la profundidad. La escena en 3D que queremos reconstruir generalmente se llama estructura.

Figura 10.38: (Izquierda) El STH-MDCS3 de Videre Design utiliza sensores CMOS. (Derecha) El Bumblebee2 de Point Grey utiliza sensores CCD.



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

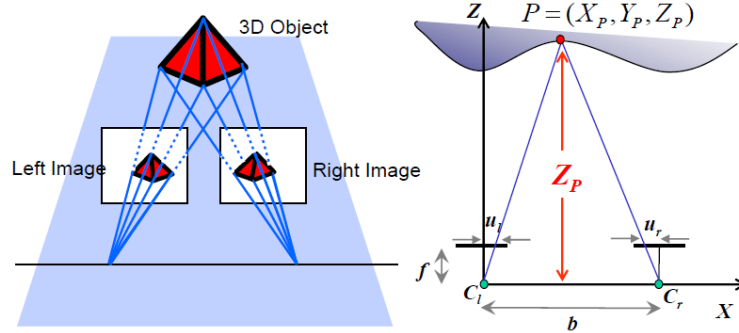
### **Visión estéreo**

La estereopsis es el proceso de percepción visual que conduce a la sensación de profundidad a partir de las dos proyecciones del mundo ligeramente diferentes sobre las retinas de ambos ojos. La diferencia en las dos imágenes de la retina se llama disparidad horizontal, disparidad retiniana, o disparidad binocular. Las diferencias surgen de las diferentes posiciones de los ojos en la cabeza. Es esta disparidad que hace que nuestro cerebro fusione (perciba como una sola imagen) las dos imágenes de la retina que nos hacen percibir el objeto como un uno y sólido. Para tener una idea más clara de lo que es la disparidad, como una simple prueba, mantener un dedo vertical frente a los ojos y cerrar cada ojo alternativamente. Se verá saltos del dedo de izquierda a derecha. La distancia entre la aparición izquierda y derecha del dedo es la disparidad. El mismo fenómeno es visible en el par de la imagen se muestra en la figura 10.44, en la cual los objetos en primer plano cambian de izquierda a derecha en relación con el fondo. La estereopsis computacional, o visión estéreo, es el proceso de obtención de la información de profundidad a partir de un par de imágenes procedentes de dos cámaras que miran en la misma escena desde diferentes posiciones. En la visión estéreo se pueden identificar dos grandes problemas:

1. El problema de correspondencia.
2. Reconstrucción 3D.

El primero consiste en hacer coincidir (emparejamiento) puntos de las dos imágenes que son la proyección del mismo punto en la escena. Estos puntos coincidentes se denominan puntos correspondientes o correspondencias (figura 10.41a). El correspondiente de un punto en una imagen se encuentra en una línea (llamada línea epipolar) en la otra imagen (figura 10.41b). Debido a esta limitación, se verá que la búsqueda se convierte en correspondencia unidimensional en lugar de bidimensional.

Figura 10.39: Geometría de cámara idealizada para visión estéreo



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

Conociendo las correspondencias entre las dos imágenes, sabiendo la orientación relativa y la posición de las dos cámaras, y conociendo los parámetros intrínsecos de las dos cámaras, es posible reconstruir los puntos de la escena (es decir, la estructura). Este proceso de reconstrucción requiere la calibración previa de la cámara estéreo. Ejemplos de cámaras estéreo comercialmente disponibles se muestran en la figura 10.38.

**Caso básico.** En primer lugar, consideramos un caso simplificado en el que dos cámaras tienen la misma orientación y se colocan con sus ejes ópticos paralelos, a una separación de  $b$  (llamada línea de base), que se muestra en la figura 10.39. En esta figura, un punto del objeto se describe como la coordenada  $(x, y, z)$  con respecto al origen situado en la lente de la cámara izquierda. Las coordenadas de la imagen de la izquierda y la derecha son  $(u_l, v_l)$  y  $(u_r, v_r)$ , respectivamente. A partir de la figura 10.39a y usando las ecuaciones 10.24 y 10.25, se puede escribir:

$$\frac{f}{z} = \frac{u_l}{x} \quad (10.45)$$

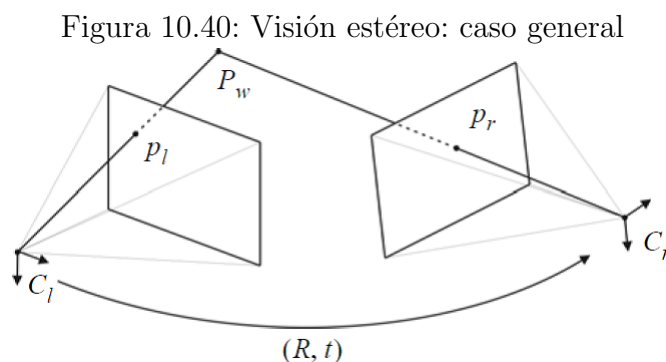
$$\frac{f}{z} = \frac{-u_r}{b - x} \quad (10.46)$$

De donde se obtiene:

$$z = b \frac{f}{u_l - u_r} \quad (10.47)$$

Donde la diferencia en las coordenadas de imagen,  $u_l - u_r$ , se llama disparidad. Este es un término importante en la visión estereoscópica, ya que sólo mediante la medición de disparidad es que se puede recuperar la información de profundidad. Las observaciones de esta ecuación son como sigue:

- La distancia es inversamente proporcional a la disparidad.
- La disparidad es proporcional a  $b$ .
- Así como  $b$  se incrementa, debido a que la separación física entre las cámaras se incrementa, algunos objetos pueden aparecer en una cámara, pero no en la otra.
- Si la línea de base  $b$  es desconocida, es posible reconstruir el punto de la escena sólo hasta cierto nivel. Este es el caso en la estructura de movimiento (sección 10.2.5).
- Un punto en la escena visible para ambas cámaras produce un par de puntos de imagen conocido como par conjugado o par de correspondencia (figura 10.39a).



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

**Caso general.** La suposición de cámaras perfectamente alineadas normalmente se viola en la práctica. Por lo tanto, el modelo de estereo visión general asume que las dos cámaras son diferentes y no alineadas (figura 10.40), pero requiere que la posición relativa y la orientación de las dos cámaras sea conocida. Si la posición relativa no se conoce, la cámara estereo debe ser calibrada usando calibración basada en tablero

de ajedrez tratada en las secciones anteriores. Se supone que las dos cámaras han sido previamente calibradas. Por lo tanto, las matrices de parámetros intrínsecos  $A_l$  y  $A_r$  (véase la ecuación 10.35) para la cámara izquierda y derecha son conocidas, y los parámetros extrínsecos de la cámara, es decir las rotaciones  $R_l$ ,  $R_r$  y las traslaciones  $R_l$ ,  $R_r$  de las dos cámaras con respecto al sistema de coordenadas del mundo, son también conocidas. En estéreo visión, es una práctica común asumir que el origen del sistema de coordenadas del mundo en la cámara izquierda. Por lo tanto, podemos escribir  $R_l = I$  y  $R_r = R$ . Esto nos permite escribir las ecuaciones de proyección en perspectiva de las dos cámaras como:

$$\lambda_l \tilde{p}_l = A_l [I|O] \tilde{P}_w \quad (\text{para la cámara LED}) \quad (10.48)$$

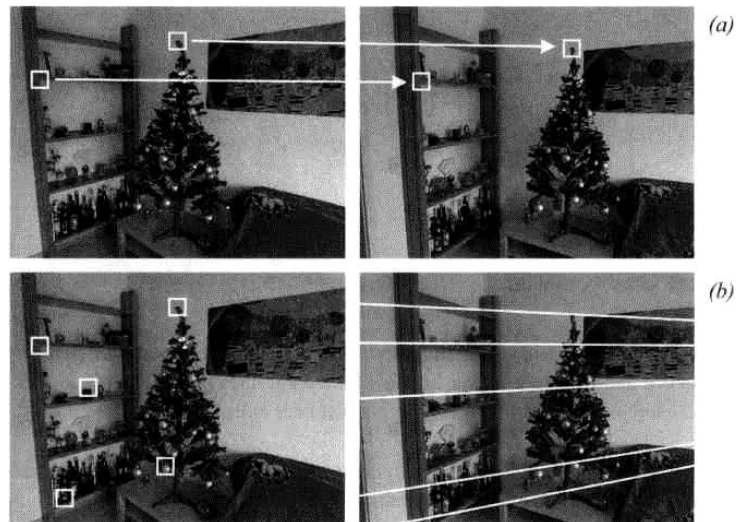
$$\lambda_r \tilde{p}_r = A_r [I|O] \tilde{P}_w \quad (\text{para la cámara derecha}) \quad (10.49)$$

Donde  $\tilde{p}_l = [u_l, v_l, 1]^T$  y  $\tilde{p}_r = [u_r, v_r, 1]^T$  son los puntos de imagen (en coordenadas homogéneas) correspondientes a los puntos de mundo  $\tilde{p}_w = [x, y, z, 1]^T$  (en coordenadas homogéneas) en la cámara izquierda y derecha respectivamente.  $\lambda_l$  y  $\lambda_r$  son los factores de profundidad. Se un sistema de seis ecuaciones con cinco incógnitas, tres para el punto de mundo  $P_w = (x, y, z)$  y dos para factores de profundidad, es decir  $\lambda_l$  y  $\lambda_r$ . El sistema está sobredeterminado y se puede resolver ya sea linealmente, utilizando los mínimos cuadrados, o no linealmente mediante el cálculo del punto 3D que minimiza las distancias entre los dos rayos de luz que pasan por  $\tilde{p}_l$  y  $\tilde{p}_r$ .

**Problema de correspondencia.** Intuitivamente, el problema es: dadas dos imágenes de la misma escena desde diferentes perspectivas, identificar los puntos de un mismo objeto en las dos imágenes. Para cada tal punto de objeto identificado, entonces seremos capaces de recuperar su posición en la escena 3D. **Ver en Anexo C página 631 taller en Matlab** La búsqueda de la correspondencia se basa en la suposición de que las dos imágenes de la misma escena no difieren demasiado, es decir, una característica en la escena se supone que es muy similar en ambas imágenes. Usando una métrica de similitud imagen de oportuna, un punto dado en la primera imagen se puede combinar con un punto en la segunda imagen. El problema de las correspondencias falsas hace que la búsqueda de correspondencia sea un desafío. Otros problemas que hacen difícil la búsqueda de correspondencia son:

- Oclusiones: la escena es vista por dos cámaras en diferentes puntos de vista y por lo tanto hay partes de la escena que aparecen sólo en una de las imágenes.

Figura 10.41: Un par estéreo. Puntos correspondientes son proyecciones del mismo punto de escena.



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

- Distorsión fotométrica: hay superficies de la escena que no son perfectamente lambertiana, es decir, superficies cuyo comportamiento es parcialmente especular. Por lo tanto, la intensidad observada por las dos cámaras es diferente para el mismo punto en la escena cuanto más separadas estén las cámaras.
- Distorsión proyectiva: debido a la distorsión de la perspectiva, un objeto de la escena se proyecta de manera diferente en las dos imágenes, entre más separadas estén cámaras más.

Algunas limitaciones pueden, sin embargo, ser explotadas para mejorar la búsqueda de correspondencia, que son:

- Restricción de similitud: una característica de la imagen aparece similar en la otra imagen.
- Restricción de continuidad: lejos de los bordes de la imagen, la profundidad de los puntos de la escena a lo largo de una superficie continua varía continuamente. Esta restricción limita claramente el gradiente de disparidad.
- Unicidad: un punto de la imagen primero se puede combinar con sólo un único punto en la otra imagen, y viceversa (falla en presencia de oclusiones, especularidades, y transparencia).

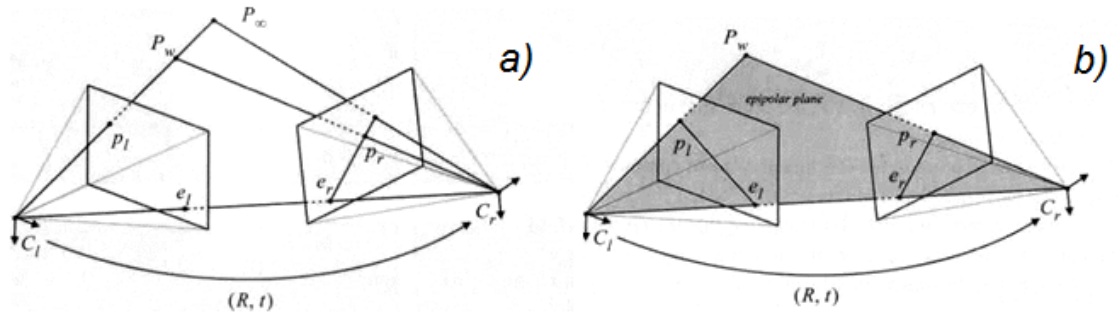
- Orden monótonico: si el punto  $p_l$  en la imagen de la izquierda es el correspondiente de  $p_r$  en la imagen derecha, el correspondiente de un punto a la derecha (izquierda) de  $p_l$  sólo se puede encontrar a la derecha (izquierda) de  $p_r$ . Esto sólo es válido para los puntos que se encuentran sobre un objeto opaco.
- Restricción epipolar: el correspondiente de un punto en la imagen de la izquierda sólo se puede encontrar a lo largo de una línea en la imagen de la derecha, que se llama la línea epipolar (figura 10.41b). Como cuestión de hecho, esta es la limitación más importante, que se explica más adelante.

Los métodos para la búsqueda de correspondencias se pueden distinguir en dos categorías:

- Basados en área: estos algoritmos consideran un pequeño parche (ventana) en una imagen y buscar el parche más similar en la segunda imagen por medio de una medida de correlación apropiada. Esta búsqueda se realiza para cada píxel y permite obtener una reconstrucción densa. La búsqueda puede ser reducida a una sola línea, la línea epipolar, lo que reduce la dimensionalidad de la búsqueda de dos a uno (figura 10.41b).
- Basados en funciones: estos algoritmos extraen las características más destacadas de las imágenes, que posiblemente son estables con respecto al cambio de punto de vista. El proceso de búsqueda se aplica a los atributos asociados a las características. Los bordes, esquinas, segmentos de líneas, y gotas son algunas de las características que se pueden utilizar.

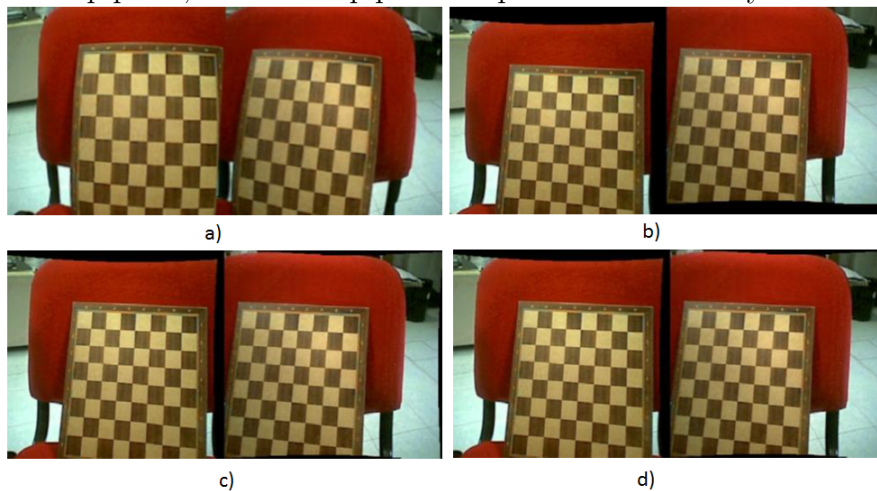
**Geometría epipolar.** Dado un píxel en una imagen (por ejemplo la imagen de la izquierda), ¿cómo podemos calcular su correspondencia con el píxel correcto en la otra imagen? Como se prevé en la sección anterior, una solución sería buscar las correspondencias a través de todos los píxeles de la segunda imagen. La figura 10.42a muestra como un punto de píxel  $p_l$  en una imagen proyecta un segmento de línea epipolar en la otra imagen. El segmento está limitado en un extremo por la proyección de  $P_\infty$  (el rayo de la imagen original en el infinito) y en el otro extremo por la proyección de  $C_l$  en la segunda cámara, la cual se conoce como la epipole  $e_r$ . Al proyectar la línea epipolar en la segunda imagen de nuevo en la primera imagen, se obtiene otra línea que está delimitada por el otro epipole correspondiente  $e_l$ . Nótese que dos líneas epipolares correspondientes (figura 10.42b) se originan a partir de la intersección de los dos planos de la imagen con el plano epipolar que pasa a través de los centros de la cámara  $C_l$  y  $C_r$  y el punto de escena  $P_w$ .

Figura 10.42: Geometría epipolar: (a) Segmento de línea epipolar correspondiente a un rayo, (b) Conjunto correspondiente de líneas epipolar y su plano epipolar



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

Figura 10.43: Rectificación de un par estéreo: (a) Imagen original, (b) Compensación de la distorsión de la lente, (c) Compensación de rotación y traslación, (d) Después de la rectificación epipolar, las líneas epipolares aparecen alineadas y horizontales



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

Para calcular la ecuación de la línea epipolar, debemos proyectar el rayo óptico que pasa a través de  $p_l$  y  $C_l$  a la segunda imagen. Esto es sencillo. La ecuación del rayo óptico que pasa a través  $p_l$  y  $C_l$  se puede obtener de la ecuación de proyección en perspectiva

10.48, la cual se reescribe aquí como:

$$\lambda_l \tilde{p}_l = \lambda_l \begin{bmatrix} u_l \\ v_l \\ 1 \end{bmatrix} \circ A_l [I|O] \tilde{p}_w = A_l \tilde{p}_w = A_l \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (10.50)$$

Y por lo tanto la línea que pasa a través de  $p_l$  y  $C_l$  tiene la ecuación:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \lambda_l A_l^{-1} \begin{bmatrix} u_l \\ v_l \\ 1 \end{bmatrix} \quad (10.51)$$

Que se puede reescribir en una forma más compacta como:

$$P_w = \lambda_l A_l^{-1} \tilde{p}_l \quad (10.52)$$

Por último, para encontrar la ecuación de la línea epipolar, acabamos de proyectar esta línea en la segunda imagen utilizando la ecuación de proyección en perspectiva 10.49:

$$\lambda_r \tilde{p}_r = A_r [R|t] \tilde{P}_w = A_r R P_w + A_r t \quad (10.53)$$

Y por lo tanto, usando 10.52, se obtiene la línea epipolar:

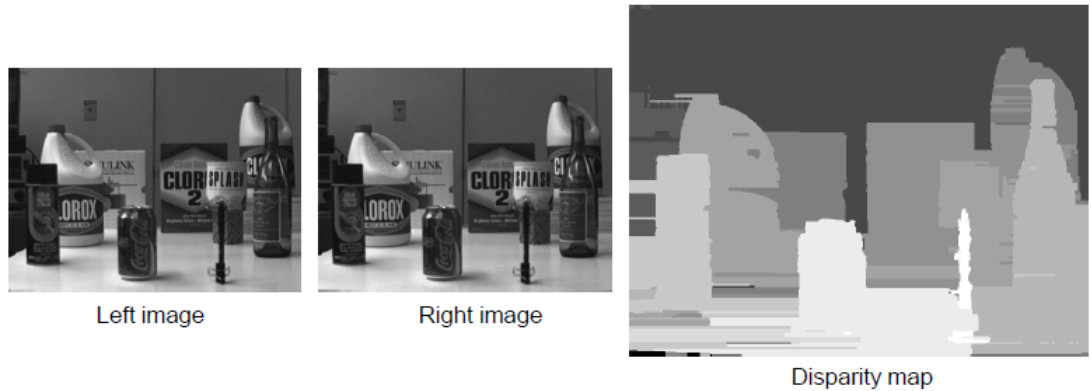
$$\lambda_r \tilde{p}_r = \lambda_l A_r R A_l^{-1} \tilde{p}_l A_r t \quad (10.54)$$

Donde  $A_r t$  es en realidad la epipole  $e_r$  en la segunda imagen, es decir, la proyección del centro óptico  $C_l$  de la cámara izquierda en la imagen derecha. Aplicación la ecuación 10.54 para cada punto de la imagen en la imagen de la izquierda, se puede calcular todas las líneas epipolares en la imagen derecha. La correspondencia de un punto en la imagen izquierda, tendrá que ser buscada sólo a lo largo de su línea epipolar correspondiente.

Aunque para algunas cámaras de campo de visión estrecho de la distorsión radial es más bien pequeña, siempre es oportuno tomar la distorsión radial en cuenta al calcular la ecuación de la línea epipolar. La razón es que si la línea epipolar no se determina con precisión, la búsqueda de correspondencia a lo largo de una línea epipolar no precisa puede conducir a una incertidumbre mayor en el cálculo de la disparidad, así como en la reconstrucción del punto de la escena  $P_w$ .

En lugar de tomar en cuenta la distorsión radial, un procedimiento consolidado común en visión estéreo es aquel de no distorsionar primero las dos imágenes, es decir, remapear la imagen izquierda y derecha en nuevas imágenes sin distorsión. Además, las dos imágenes se puede reasignar de tal manera que todas las líneas epipolares en la imagen izquierda y derecha sean colineales y horizontales (figura 10.43d). El proceso de

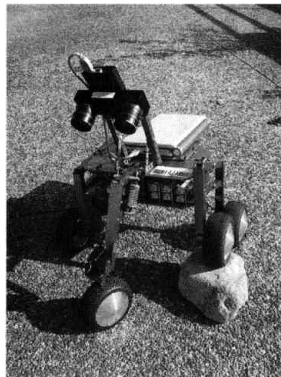
Figura 10.44: Un ejemplo de mapa de disparidad calculado a partir de las dos imágenes superiores



Fuente: <http://www.ait.ac.at>

transformación de un par de imágenes estéreo en un nuevo par sin distorsión radial y con las líneas epipolares horizontales se llama rectificación estéreo o rectificación epipolar.

Figura 10.45: Una cámara estéreo de Videre Design en el robot camarón desarrollado en la ASL



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

**Rectificación epipolar.** Dado un par de imágenes estéreo, la rectificación epipolar es una transformación de cada plano de imagen de tal manera que todas las líneas epipolares correspondientes se vuelvan colineales y paralelas a uno de los ejes de la imagen, por comodidad suele ser el eje horizontal. Los pasos del algoritmo de rectificación epipolar se ilustran en la figura 10.43. Obsérvese que después de la rectificación, todas las

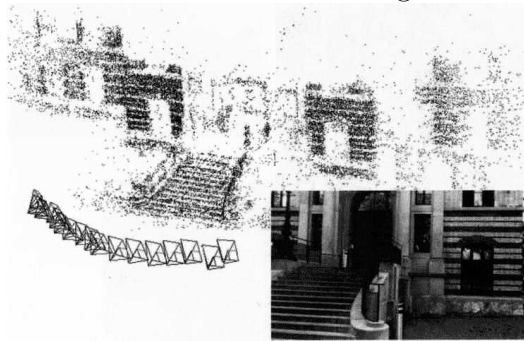
líneas epipolares en la imagen izquierda y derecha son colineales y horizontales (figura 10.43d).

**Mapa de disparidad.** Otro producto popular de la visión estéreo es el mapa de disparidad. Un mapa de disparidad aparece como una escala de grises donde la intensidad de cada punto de pixel es proporcional a la disparidad del píxel en la imagen izquierda y derecha: los objetos que están más cerca de la cámara tienen una apariencia más clara, mientras que los más lejanos aparecen más oscuros. Un mapa de disparidad ejemplo se muestra en la figura 10.44. Los mapas de disparidad son muy útiles para la evasión de obstáculos (figura 10.45). Las cámaras estéreo modernas, como aquellas de Videre Design y Point-Grey (figura 10.38), son capaces de calcular mapas de disparidad directamente en el hardware.

### 10.2.5. Estructura de movimiento

Cuando las dos imágenes son tomadas de la misma cámara, pero en diferentes posiciones y en diferentes momentos, o alternativamente, desde diferentes cámaras, ¿como se recupera la estructura?. Esto implica que la estructura y el movimiento deben ser estimados de forma simultánea. Este problema se conoce como estructura de movimiento (SfM - *Structure from Motion*).

Figura 10.46: Un ejemplo de estructura de movimiento: los puntos de imagen sobresalientes se extraen y se hacen coincidir en varios fotogramas.



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

Se observa que en la estructura de movimiento, las imágenes no necesitan ser pre-calibradas. Esto permite a la SfM trabajar en situaciones difíciles. Los parámetros intrínsecos de hecho, pueden ser estimados de forma automática desde la SfM misma.

Un resultado sugestivo de SfM se ilustra en la figura 10.46. Aquí, la escena fue reconstruida utilizando docenas de imágenes. Usando miles de imágenes desde diferentes puntos de vista, la SfM a veces puede lograr resultados de reconstrucción 3D que son casi comparables en precisión y densidad de puntos que los telémetros láser 3D.

### Estructura de movimiento de dos vistas

Se empieza de nuevo a partir de las dos ecuaciones de proyección en perspectiva 10.48 y 10.49 derivadas del caso de visión estéreo, pero ahora se recuerda que  $R$  y  $t$  denotan el movimiento relativo entre la primera y la segunda posición de la cámara, por lo tanto, se puede escribir:

$$\lambda_1 \tilde{p}_1 = A_1 [I|O] \tilde{P}_w = A_1 P_w \quad (\text{para la primera posición de la cámara}) \quad (10.55)$$

$$\lambda_2 \tilde{p}_2 = A_2 [R|t] \tilde{P}_w \quad (\text{para la segunda posición de la cámara}) \quad (10.56)$$

Con el fin de simplificar nuestro problema, vamos a hacer algunas suposiciones. Supongamos que estamos en la misma cámara para la primera y la segunda posición y que los parámetros intrínsecos no cambian en el medio, por lo tanto  $A_1 = A_2 = A$ . Supongamos también que la cámara está calibrada, y que por lo tanto,  $A$  es conocida. En este caso, es más conveniente trabajar con coordenadas de imagen normalizadas. Sean  $\tilde{x}_1$  y  $\tilde{x}_2$  las coordenadas normalizadas de  $\tilde{p}_1$  y  $\tilde{p}_2$  respectivamente, donde:

$$\tilde{x}_1 = A^{-1} \tilde{p}_1 \quad y \quad \tilde{x}_2 = A^{-1} \tilde{p}_2 \quad (10.57)$$

$\tilde{x}_1 = (x_1, y_1, 1)$   $\tilde{x}_2 = (x_2, y_2, 1)$ , entonces, podemos reescribir 10.55 y 10.56 como:

$$\lambda_1 \tilde{x}_1 = P_w \quad (\text{para la primera posición de la cámara}) \quad (10.58)$$

$$\lambda_2 \tilde{x}_2 = [R|t] \tilde{P}_w = RP_w + t \quad (\text{para la segunda posición de la cámara}) \quad (10.59)$$

Como ya se hizo antes para el cálculo de las líneas epipolares, vamos a trazar el rayo óptico correspondiente a  $x_1$  en la segunda imagen. Por lo tanto, mediante la sustitución de 10.58 en 10.59, se obtiene:

$$\lambda_2 \tilde{x}_2 = \lambda_1 R \tilde{x}_1 + t \quad (10.60)$$

Tomemos ahora el producto vectorial de ambos lados con  $t$ . Esto con el fin de cancelar  $t$  en el lado derecho. Entonces, se obtiene:

$$\lambda_2[t] \times \tilde{x}_2 = \lambda_1([t] \times R) \cdot \tilde{x}_1 \quad (10.61)$$

Donde  $[t]x$  es una matriz antisimétrica definida como:

$$[t]x = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \quad (10.62)$$

Ahora, tomando el producto escalar de los dos lados 10.61 con  $x_2$  genera:

$$\lambda_2[t] \times \tilde{x}_2^T \cdot ([t] \times \tilde{x}_2^T) = \lambda_1 \tilde{x}_2^T ([t] \times R) \cdot \tilde{x}_1 \quad (10.63)$$

Observe que  $\tilde{x}_2^T \cdot ([t] \times \tilde{x}_2^T)$ , y por tanto, de 10.63 se obtiene:

$$\tilde{x}_2^T ([t] \times R) \cdot \tilde{x}_1 \quad (10.64)$$

La cual se llama restricción epipolar. Observe que la restricción epipolar es válida para todos los pares de puntos conjugados. Vamos a definir la matriz esencial  $E = ([t] \times R)$ , la restricción epipolar se lee como:

$$\tilde{x}_2^T \cdot E \cdot \tilde{x}_1 = 0 \quad (10.65)$$

Se puede demostrar que la matriz esencial tiene dos valores singulares que son iguales y otro que es cero.

**Cálculo de la matriz esencial.** Dada esta relación fundamental 10.65, ¿cómo podemos utilizarla para recuperar el movimiento de la cámara codificado en la matriz esencial  $E$ ? Si tenemos  $N$  mediciones correspondientes  $\{(x_1^i, x_2^i)\}$ , podemos formar  $N$  ecuaciones homogéneas en los nueve elementos de  $E = [e_{11}, e_{12}, e_{13}, e_{21}, e_{22}, e_{23}, e_{31}, e_{32}, e_{33}]^T$ , del tipo:

$$\tilde{x}_1^i \tilde{x}_2^i e_{11} + \tilde{y}_1^i \tilde{x}_2^i e_{12} + \tilde{x}_2^i e_{13} + \tilde{x}_1^i \tilde{y}_2^i e_{21} + \tilde{y}_1^i \tilde{y}_2^i e_{22} + \tilde{y}_2^i e_{23} + \tilde{x}_1^i e_{31} + \tilde{y}_1^i e_{32} + e_{33} = 0 \quad (10.66)$$

Esto puede ser reescrito en una forma más compacta como:

$$D \cdot E = 0 \quad (10.67)$$

Dada  $N \geq 8$  como ecuaciones, podemos calcular una estimación (hasta de una escala) para las entradas de  $E$  utilizando la descomposición singular de valores (SVD). La

solución de 10.67 será por lo tanto el vector propio de  $D$  correspondiente al valor propio más pequeño. Debido a que por lo menos ocho correspondencias de puntos son necesarias, este algoritmo se conoce como el algoritmo de ocho puntos. Este algoritmo es uno de los hitos de la visión por computador. Las ventajas principales del algoritmo de ocho puntos son que es muy fácil de implementar y que funciona también para una cámara sin calibrar, es decir, cuando los parámetros intrínsecos de la cámara son desconocidos. El inconveniente es que no funciona para configuraciones puntuales degeneradas, tales como escenas planas, es decir, cuando todos los puntos de la escena son coplanares.

**Descomposición de  $E$  en  $R$  y  $t$ .** Supongamos ahora que la matriz  $E$  esencial se ha determinado a partir de las correspondencias de puntos conocidos. ¿Cómo definimos  $R$  y  $t$ ? Antes de descomponer  $E$ , tenemos que imponer la restricción de que dos de sus valores singulares son iguales y el tercero es cero. De hecho, en presencia de ruido de imagen esta restricción nunca será verificada en la práctica. Para ello, calculamos la matriz esencial más cercana que satisface esta limitación. Una técnica popular es usar SVD y forzar a los dos valores singulares más grandes para que sean iguales y el más pequeño que sea ser cero. Por lo tanto:

$$[U, S, V] = SVD(E) \quad (10.68)$$

Donde  $S = \text{diag}([S_{11} S_{22} S_{33}])$  con  $S_{11} \geq S_{22} \geq S_{33}$ . Entonces, la matriz esencial más cercana  $E$  eo en la norma de Frobenius está dada por:

$$\hat{E} = U \cdot \text{diag} \left( \left[ \frac{S_{11} + S_{22}}{2}, \frac{S_{11} + S_{22}}{2}, 0 \right] \right) \cdot V^T \quad (10.69)$$

Luego, reemplazamos  $E$  con  $\hat{E}$ . En este punto, se puede descomponer  $E$  en  $R$  y  $t$ . La descomposición de  $E$  devuelve cuatro soluciones para  $(R, t)$ , dos para  $R$  y dos para  $t$ . Se define:

$$B = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad y \quad [U, S, V] = SVD(E) \quad (10.70)$$

Donde  $U$ ,  $S$ , y  $V$  son tales que  $U \cdot S \cdot V = E$ . Se puede demostrar en *Multiple View Geometry* de Hartley y Zisserman que las dos soluciones para  $R$  son:

$$R_1 = \det(U \cdot V^T) \cdot U \cdot B \cdot V^T \quad (10.71)$$

$$R_2 = \det(U \cdot V^T) \cdot U \cdot B^T \cdot V^T \quad (10.72)$$

Ahora, vamos a definir:

$$L \equiv U \cdot \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot U^T \quad y \quad M = -U \cdot \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot U^T \quad (10.73)$$

Las dos soluciones para  $t$  son:

$$t_1 = \frac{[L_{32}L_{13}L_{21}]^T}{\|[L_{32}L_{13}L_{21}]\|} \quad (10.74)$$

$$t_2 = \frac{[M_{32}M_{13}M_{21}]}{\|[M_{32}M_{13}M_{21}]^T\|} \quad (10.75)$$

## Odometría visual

La odometría visual está directamente relacionada con la estructura del movimiento. La odometría visual consiste en estimar el movimiento de un robot o de un vehículo mediante la entrada visual solamente. El término “odometría visual” fue ideado en 2004 por Nister con su documento destacado homónimo, donde mostró resultados exitosos en diferentes vehículos (de carretera y off-road) usando ya sea una sola cámara o una cámara estéreo. El principio básico detrás de odometría visual es una iteración simple de una estructura de movimiento de dos vistas que hemos visto en el apartado anterior.

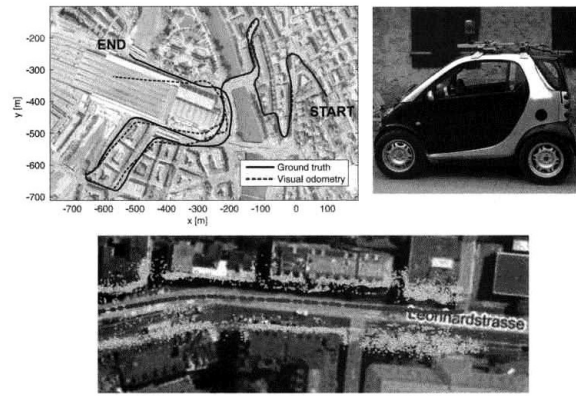
Un ejemplo de resulta de odometría visual usando una única cámara omnidireccional se muestra en la figura 10.47. En esta figura, la odometría visual se realiza sobre una trayectoria de 3 km.

### 10.2.6. Movimiento y flujo óptico

Una gran cantidad de información se puede recuperar mediante el registro de imágenes variables en el tiempo a partir de una cámara fija (o móvil). En primer lugar, se distingue entre el campo de movimiento y el flujo óptico:

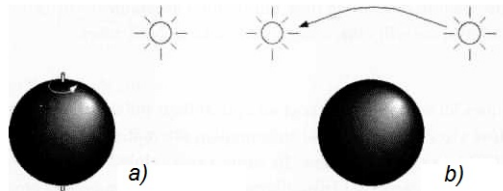
- El campo de movimiento: este asigna un vector de velocidad en cada punto en una imagen. Si un punto en el medio se mueve con velocidad  $v_o$ , entonces esto induce una velocidad  $v_i$  en el plano de la imagen. Es posible determinar matemáticamente la relación entre  $v_i$  y  $v_o$ .
- El flujo óptico, también puede ser cierto que los patrones de brillo de la imagen se muevan como el objeto que causa que se muevan (fuente de luz). Flujo óptico es el movimiento aparente de estos patrones de brillo.

Figura 10.47: (Parte superior izquierda) Un resultado ejemplo de odometría visual con su mapa relacionado (inferior) obtenido utilizando una sola cámara omnidireccional montada en el techo del vehículo (derecha).



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

Figura 10.48: Movimiento de la esfera o la fuente de luz



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

En este análisis se supone que el patrón de flujo óptico corresponderá con el de campo de movimiento, aunque esto no es siempre cierto en la práctica. Esto se ilustra en la figura 10.48a, donde una esfera exhibe una variación espacial de brillo, o sombreado, en la imagen de la esfera ya que su superficie está curvada. Si la superficie se mueve, sin embargo, este patrón de sombreado no se moverá ya que el flujo óptico es cero en todas partes a pesar de que el campo de movimiento no es cero. En la figura 10.48b, ocurre lo contrario. Aquí tenemos una esfera fija con una fuente de luz en movimiento. El sombreado de la imagen cambiará a medida que la fuente se mueve. En este caso, el flujo óptico es distinto de cero, pero el campo de movimiento es cero. Si la única información accesible para nosotros es el flujo óptico y dependemos de esto, vamos a obtener resultados incorrectos en ambos casos.

## Flujo óptico

A continuación se presentan los detalles del método de la ecuación de restricción de flujo óptico. Supongamos primero que el intervalo de tiempo entre las instantáneas sucesivas es tan rápido que se puede asumir que la intensidad medida de una parte del mismo objeto es efectivamente constante. Matemáticamente, sea  $I(x, y, t)$  la irradiancia de la imagen en el tiempo  $t$  en el punto de imagen  $(x, y)$ . Si  $u(x, y)$  y  $v(x, y)$  son las componentes  $x$  y  $y$  del vector de flujo óptico en ese punto, es necesario buscar una nueva imagen para un punto en el que la irradiación será la misma en el tiempo  $t + \delta t$ , es decir, en el punto  $(x + \delta x, y + \delta y)$ , donde  $\delta x = u\delta t, \delta y = v\delta t$ . Es decir:

$$I(x + u\delta t, y + v\delta t, t + \delta t) = I(x, y, t) \quad (10.76)$$

Para un pequeño intervalo de tiempo,  $\delta t$ . Esto capturará el movimiento de un parche de intensidad constante a través del tiempo. Si suponemos además que el brillo de la imagen varía suavemente, entonces se puede ampliar el lado izquierdo de la ecuación 10.76 como una serie de Taylor para obtener:

$$I(x, y, t) = \delta x \frac{\delta I}{\delta x} + \delta y \frac{\delta I}{\delta y} + \delta z \frac{\delta I}{\delta z} + e = I(x, y, t) \quad (10.77)$$

Donde  $e$  incluye términos de segundo y de orden superior en  $\delta x$ , y así sucesivamente. En el límite cuando  $\delta t$  tiende a cero, se obtiene:

$$\frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} + \frac{\partial I}{\partial t} = 0 \quad (10.78)$$

De lo cual se puede abreviar:

$$u = \frac{dx}{dt}; \quad v = \frac{dy}{dt} \quad (10.79)$$

Y,

$$I_x = \frac{\partial I}{\partial x}; \quad I_y = \frac{\partial I}{\partial y}; \quad I_t = \frac{\partial I}{\partial t} = 0 \quad (10.80)$$

De modo que se obtiene:

$$I_x u + I_y v + I_t = 0 \quad (10.81)$$

La derivada  $I_t$  representa que tan rápido la intensidad cambia con el tiempo mientras que las derivadas  $I_x$  y  $I_y$  representan las tasas espaciales de cambio de intensidad (que tan rápido cambia la intensidad a lo largo de la imagen). En total, la ecuación 10.81 es conocida como la ecuación de restricción de flujo óptico, y las tres derivadas pueden ser estimadas para cada píxel dadas imágenes sucesivas. Es necesario calcular tanto  $u$

como  $v$  para cada píxel, pero la ecuación de restricción de flujo óptico sólo proporciona una ecuación por píxel, por tanto esto es insuficiente. La ambigüedad es intuitivamente clara cuando se considera que un número de píxeles de igual intensidad puedan ser inherentemente ambiguos, puede no estar claro cuál píxel es la ubicación resultante para un píxel originario de igual intensidad en la imagen anterior.

La solución a esta ambigüedad requiere una restricción adicional. Porque sabemos que esta restricción suavidad será algo incorrecta, matemáticamente se puede definir el grado en que se viola esta restricción evaluando la fórmula:

$$e_s = \iint (u^2 + v^2) dx dy \quad (10.82)$$

Que es la integral del cuadrado de la magnitud del gradiente del flujo óptico. También se determina el error en la ecuación de restricción de flujo óptico (que en la práctica no del todo será cero).

$$e_s = \iint (I_x u + I_y v + I_t)^2 dx dy \quad (10.83)$$

Ambas de estas ecuaciones debe ser tan pequeñas como sea posible, por lo que queremos minimizar  $e_s + \lambda e_c$ , donde  $\lambda$  es un parámetro que pondera el error en la ecuación de movimiento de la imagen con respecto a la salida de la suavidad. En la práctica, el parámetro  $\lambda$  se ajusta manualmente y de forma interactiva para lograr el mejor rendimiento. El problema resultante es equivale al cálculo de variaciones, y produce las ecuaciones de Euler:

$$\nabla^2 u = \lambda (I_x u + I_y v + I_t) I_x \quad (10.84)$$

$$\nabla^2 v = \lambda (I_x u + I_y v + I_t) I_y \quad (10.85)$$

Donde:

$$\nabla^2 = \frac{\partial^2}{\delta x^2} + \frac{\partial^2}{\delta y^2} \quad (10.86)$$

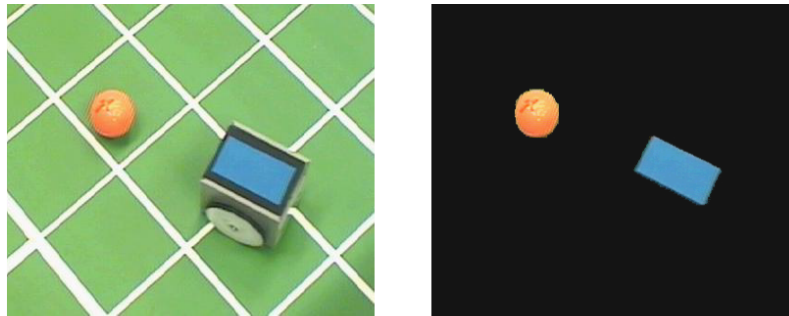
Que es el operador de Laplace. El flujo óptico es un ingrediente importante en algoritmos de visión que combinan las señales a través de múltiples algoritmos. Los sistemas de control de evasión de obstáculos y de navegación para robots móviles (especialmente robots voladores) que utilizan flujo óptico han demostrado ser ampliamente eficaces.

### 10.2.7. Seguimiento de Color

Un aspecto importante de la detección de visión es que el chip de visión puede proporcionar modalidades de detección y señales que ningún otro sensor de robot móvil

proporciona. Una nueva modalidad de detección es la detección y el seguimiento de color en el medio ambiente. El color es una característica del medio ambiente y representa a la vez una señal natural y una señal artificial que puede proporcionar nueva información a un robot móvil, por ejemplo, los eventos de Robot-soccer anuales (RoboCup) hacen un amplio uso del color, tanto para el marcado del medio ambiente como para la localización del robot (figura 10.49).

Figura 10.49: Marcadores de colores en la parte superior de los robots de fútbol de STeam Engine



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

La detección de color tiene dos ventajas importantes. En primer lugar, la detección de color es una función sencilla de una sola imagen, por lo tanto, no hay problema de correspondencia que deba ser resuelto en este tipo de algoritmos. En segundo lugar, ya que la detección de color proporciona una señal nueva, señal independiente del medio ambiente, si se combina (es decir, fusión de sensores) con señales existentes, tales como los datos de la visión estéreo o de telemetría láser, se puede esperar un aumento significativo de la información.

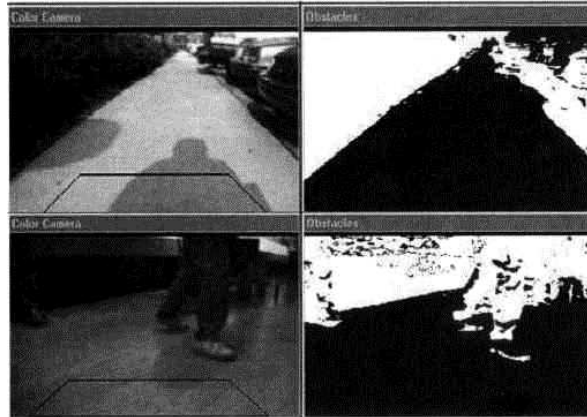
Los sensores eficientes de rastreo de color también están disponibles comercialmente, tales como la CMUcam de la Carnegie Mellon University pero también se puede implementar sencillamente con una cámara estándar. La forma más sencilla de hacer esto es utilizando umbralización constante: un punto de píxel dado se selecciona si y sólo si sus valores  $RGB$  ( $r, g, b$ ) caen simultáneamente en los rangos  $R, G, B$  elegidos, que se definen por seis umbrales  $[R_{min}, R_{max}], [G_{min}, G_{max}], [B_{min}, B_{max}]$  Por lo tanto:

$$R_{min} < r < R_{max} \quad y \quad G_{min} < g < G_{max} \quad y \quad B_{min} < b < B_{max} \quad (10.87)$$

Si se representa el espacio de color RGB como un espacio euclidiano tridimensional, el método mencionado anteriormente selecciona los píxeles cuyas componentes de color

pertenezcan al cubo especificado por los límites indicados. Alternativamente, una esfera se podría utilizar. En este caso, un píxel se seleccionaría sólo si sus componentes  $RGB$  están dentro de una cierta distancia desde un punto dado en el espacio  $RGB$ .

Figura 10.50: Ejemplos de extracción de plano adaptativa del suelo



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

Alternativamente al  $RGB$ , el espacio de color  $YUV$  se puede utilizar. Mientras que los valores  $R, G, y B$  codifican la intensidad de cada color,  $YUV$  separa la medida de color (o crominancia) de la medida del brillo (o luminosidad).  $Y$  Representa la luminosidad de la imagen, mientras que  $U$  y  $V$  capturan su crominancia. Por lo tanto, un cuadro delimitador expresado en el espacio  $YUV$  puede lograr una mayor estabilidad con respecto a los cambios en la iluminación de la que es posible en el espacio  $RGB$ .

Una aplicación popular de segmentación de color en la robótica es la extracción de plano del suelo (figura 10.50). En este caso, se utilizan técnicas de segmentación de color más complejas que umbralización de color, como umbral adaptativo, o  $k$ -means clustering. La extracción de plano de suelo es un enfoque de visión para la identificación de las partes desplazables de la tierra. Como es el caso con todos los algoritmos de visión, la extracción de plano de suelo sólo tiene éxito en ambientes que satisfacen varias suposiciones importantes:

- Los obstáculos difieren en apariencia con los de la tierra.
- El terreno es plano, y su ángulo a la cámara es conocido.
- No hay obstáculos colgantes.

**El taller de seguimiento de color se encuentra en el Anexo C página 616.**

## 10.3. FUNDAMENTOS DEL PROCESAMIENTO DE IMÁGENES

El procesamiento de imágenes es una forma de procesamiento de señal donde la señal de entrada es una imagen (tal como una fotografía o un vídeo) y la salida es una imagen o un conjunto de parámetros asociados a la imagen. La mayoría de las técnicas de procesamiento de imágenes tratan la imagen como una señal bidimensional  $I(x, y)$  donde  $x$  y  $y$  son las coordenadas espaciales de imagen y la amplitud de  $I$  en cualquier par de coordenadas  $(x, y)$  se denomina intensidad o nivel de gris de la imagen en ese punto.

El procesamiento de imágenes es un campo enorme y las operaciones típicas, entre muchos otros, son:

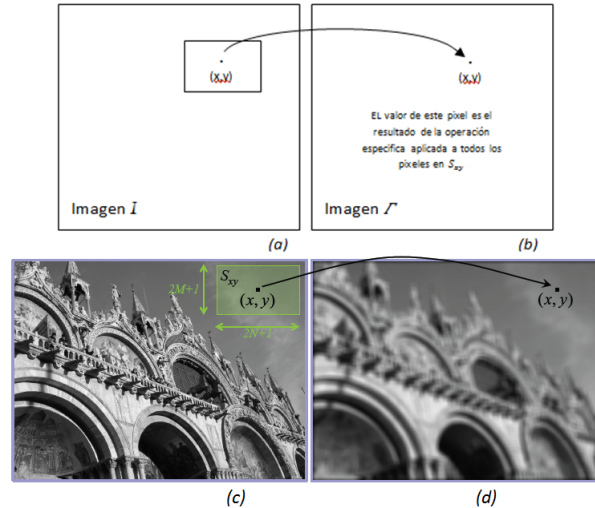
- Filtrado, mejorar la imagen, detección de borde
- Restauración y reconstrucción de la imagen
- Wavelets y el procesamiento multirresolución
- Compresión de imagen (por ejemplo, JPEG)
- Transformaciones geométricas euclidianas como la ampliación, reducción y rotación
- Correcciones de color, como los ajustes de brillo y contraste, cuantización, o la traslación de color a un espacio de color diferente
- Registro de Imagen (alineación de dos o más imágenes)
- Reconocimiento de Imagen (por ejemplo, la extracción de una cara de la imagen mediante el uso de un algoritmo de reconocimiento facial)
- Segmentación de Imagen (partición de la imagen en regiones características de acuerdo al color, bordes u otras características)

### 10.3.1. Filtrado de Imagen

El filtrado de imagen es una de las principales herramientas en el procesamiento de imágenes. El filtro de la palabra viene de procesamiento de dominio de frecuencia, donde “filtrado” se refiere al proceso de aceptación o rechazo de ciertos componentes de frecuencia. Por ejemplo, un filtro que pasa bajas frecuencias es llamado un filtro de paso

bajo. El efecto producido por un filtro de paso bajo es difuminar (suavizar) una imagen, el cual tiene el efecto principal de reducción de ruido de la imagen. A la inversa, un filtro que pasa altas frecuencias se denomina filtro de paso alto y se utiliza típicamente para la detección de bordes.

Figura 10.51: Ilustración del concepto de filtrado espacial, (c) imagen de entrada, (d) la imagen de salida después de la aplicación de filtro de promedio.



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

En la figura 10.51, se explica el principio básico de filtrado espacial. Un filtro espacial consiste en (1) un sector de la imagen bajo examen, y (2) una operación  $T$  predefinida que se realiza en los píxeles de la imagen abarcados por el sector. Sea  $S_{xy}$ , que denota el conjunto de coordenadas de sector centrado en un punto arbitrario  $(x, y)$  en una imagen  $I$ . El filtrado espacial genera un píxel correspondiente en las mismas coordenadas en una imagen de salida  $I'$  donde el valor de ese píxel se determina por una de las operaciones especificadas en los píxeles de  $S_{XY}$ . Por ejemplo, supongamos que la operación especificada es calcular el valor medio de los píxeles en una ventana rectangular de tamaño  $m \times n$  centrada en  $(x, y)$ . Las ubicaciones de los píxeles en esta región constituyen el conjunto  $S_{xy}$ . La figura 10.51a-b ilustra el proceso. Podemos expresar esta operación en forma de ecuación como

$$I'(x, y) = \frac{1}{mn} \sum_{r,c \in S_{xy}} I(r, c) \quad (10.88)$$

Donde  $R$  y  $C$  son las coordenadas de fila y columna de los píxeles en el conjunto  $S_{xy}$ .

La nueva imagen  $I'$  es creada mediante la variación de las coordenadas  $(x, y)$  de manera que el centro de la ventana se mueva de píxel a píxel en la imagen  $I$ . Por ejemplo la imagen de la figura 10.51d fue creada de esta manera utilizando una ventana de tamaño  $21 \times 21$  aplicada sobre la imagen en la figura 10.51c.

$$I'(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) \cdot I(x + s, y + t) \quad (10.89)$$

Donde  $m = 2a + 1$  y  $n = 2b + 1$  se asumen generalmente enteros impares. El filtro  $w$  también es llamado kernel, máscara o ventana. Como se observa en 10.89, el filtrado lineal es el proceso de mover una máscara de filtro sobre la imagen completa, y calcular la suma de los productos en cada lugar. En el procesamiento de señales, esta operación en particular se llama también correlación con el núcleo (o kernel)  $w$ . Esto es, sin embargo, oportuno para especificar que una operación de filtrado lineal equivalente es la convolución.

$$I'(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) \cdot I(x - s, y - t) \quad (10.90)$$

Donde la única diferencia con la correlación es la presencia del signo menos, lo que significa que la imagen debe ser volteada. Observar que para la convolución y la correlación de filtros simétricos devolver el mismo resultado y por tanto, los dos términos se pueden utilizar indistintamente. La operación de convolución con el kernel (núcleo)  $w$  puede escribirse en una forma más compacta como

$$I'(x, y) = w(x, y) * I(x, y) \quad (10.91)$$

Donde  $*$  denota el operador de convolución. La generación de filtros espaciales lineales requiere que se especifiquen los coeficientes  $mn$  del kernel. Estos coeficientes son elegidos en base a lo que el filtro se supone que debe hacer. En la siguiente sección, veremos cómo seleccionar estos coeficientes.

### Filtros de suavizado.

Los filtros de suavizado se utilizan para difuminado y reducción de ruido. El difuminado se utiliza en tareas tales como la remoción de pequeños detalles o llenado de pequeños espacios en las líneas o curvas. Tanto el difuminado como la reducción de ruido se pueden lograr a través de filtros lineales o no lineales. En este artículo examinamos algunos filtros lineales.

En la sección anterior, ya hemos visto un ejemplo de filtro de promediado constante (figura 10.51) que simplemente proporciona el promedio estándar de los píxeles de la máscara. Suponiendo una máscara  $M$ , el filtro se puede escribir como

$$w = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (10.92)$$

Donde todos los coeficientes suman 1. Esta normalización es importante mantener el mismo valor como la imagen original si la región por la cual el filtro se multiplica es uniforme. Tenga en cuenta también que, en lugar de ser  $1/9$ , los coeficientes del filtro son todos 1's. La idea es que los píxeles se sumarán primero y el resultado se divide por 9. En efecto, este es computacionalmente más eficiente que multiplicar cada elemento por  $1/9$ .

Muchos algoritmos de procesamiento de imágenes hacen uso de la segunda derivada de la intensidad de la imagen. Ya que debido a la susceptibilidad de tales algoritmos derivativos de alto orden a cambios en la iluminación en la señal básica, es importante suavizar la señal de modo que los cambios de intensidad se deban a los cambios reales en la luminosidad de los objetos en la escena en vez de variaciones al azar debidas al ruido de imagen. Un enfoque estándar es el uso de un filtro de promediado gaussiano cuyos coeficientes están dados por

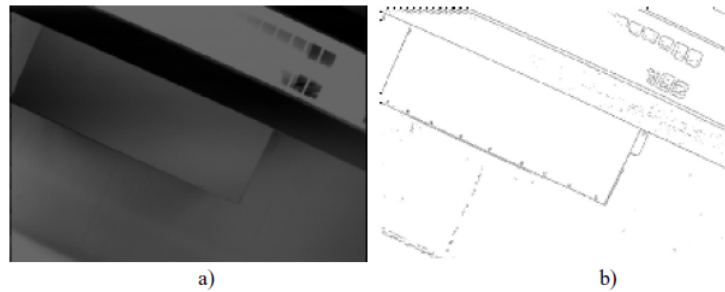
$$G_{\sigma}(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (10.93)$$

Para generar, por ejemplo, una máscara de filtro  $3 \times 3$  de esta función, se toma una muestra alrededor de su centro. Por ejemplo, con  $\sigma = 0,85$ , obtenemos

$$w = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (10.94)$$

Donde, de nuevo, los coeficientes se ajustaron de modo que su suma sea 1. Observe también que los coeficientes son todos potencias de 2, lo que hace que sea muy eficiente para calcular. Este filtro es realmente muy popular. Un filtro de paso bajo elimina eficazmente ruido de alta frecuencia, y esto en vez provoca que la primera derivada y, especialmente, la segunda derivada de la intensidad sean mucho más estables. Debido a la importancia de los gradientes y derivadas para el procesamiento de imágenes, tal preprocesamiento gaussiano de suavización es un primer paso popular de prácticamente todos los algoritmos de visión por ordenador. **El taller en Matlab se encuentra en el Anexo C página 622.**

Figura 10.52: (a) Foto de una lámpara de techo. (b) Calculo de los bordes a partir de (a).



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

### 10.3.2. Detección de bordes

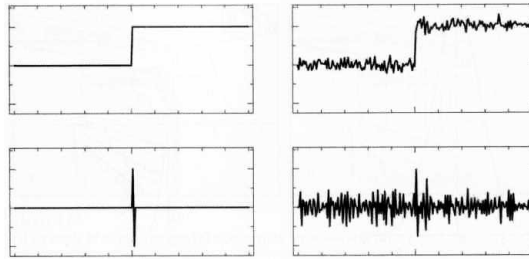
La figura 10.52 muestra una imagen de una escena que contiene una parte de una lámpara de techo, así como los bordes extraídos de esta imagen. Los bordes definen las regiones en el plano de la imagen, donde se lleva a cabo un cambio significativo en el brillo de la imagen. Como se muestra en este ejemplo, la detección de bordes reduce significativamente la cantidad de información en una imagen, y por lo tanto es una característica potencial y útil durante la interpretación de la imagen.

El reto fundamental de la detección de bordes se visualiza en la figura 10.19. La figura 10.19 (arriba a la izquierda) muestra la sección  $1D$  de un borde ideal. Pero la señal producida por una cámara se parecerá más a la figura 10.53 (parte superior derecha). La ubicación del borde todavía está en el valor de  $x$  igual, pero un nivel significativo de ruido de alta frecuencia afecta a la calidad de la señal.

Un detector de bordes simple diferenciaría, desde que un borde, por definición, se encuentre donde hay grandes transiciones en la intensidad. Como se muestra en la figura 10.53 (abajo a la derecha), la diferenciación de la señal ruidosa de la cámara resulta en picos auxiliares que pueden hacer que la detección de bordes sea muy difícil. Una señal derivada mucho más estable puede ser generada simplemente por preprocesamiento de la señal de la cámara utilizando la función de suavizado de Gauss descrito anteriormente. A continuación se presentan varios algoritmos populares de detección de bordes, los cuales operan bajo este mismo principio básico, en que la(s) derivada(s) de intensidad, siguen de alguna forma de suavizado, comprenden la señal básica de la que se extraen características de borde.

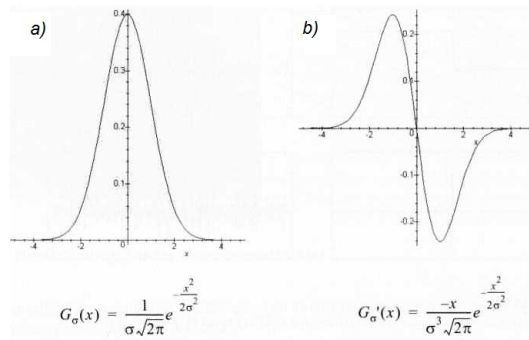
**Detección de bordes óptima de Canny.** Fue inventado por John Canny en 1983.

Figura 10.53: Ejemplo de función Paso de la forma de la segunda derivada y el impacto del ruido.



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

Figura 10.54: (a) Una función gaussiana. (b) La primera derivada de una función gaussiana.



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

Este detector de bordes nació a partir de un enfoque formal en el que Canny trató la detección de borde como un problema de procesamiento de señales en el que hay tres objetivos explícitos:

- Maximizar la relación señal-ruido;
- Lograr la máxima precisión en la ubicación de los bordes;
- Minimizar el número de respuestas de borde asociados con cada borde.

El extractor de borde Canny suaviza la imagen  $I$  a través de convolución gaussiana y luego busca el máximo en la derivada (rectificada). En la práctica, el suavizado y la

diferenciación se combinan en una sola operación porque

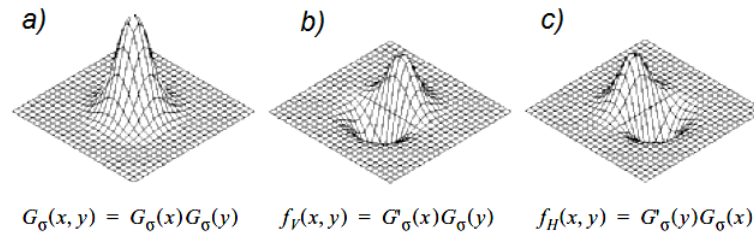
$$(G * I)' = G' * I \quad (10.95)$$

Por lo tanto, suavizar la imagen mediante convolución con una gaussiana  $G_\sigma$  y, a continuación derivarla es equivalente a la convolución de la imagen con  $G'_\sigma$ , la primera derivada de una gaussiana (figura 10.54b). Lo que se desea es detectar bordes en cualquier dirección. Puesto que  $G'$  es direccional, esto requiere la aplicación de dos filtros perpendiculares (figura 10.55). Definimos los dos filtros como  $f_V(x, y) = G'_\sigma(x)G_\sigma(y)$  y  $f_H(x, y) = G'_\sigma(y)G_\sigma(x)$ . El resultado es un algoritmo básico para la detección de bordes en orientaciones arbitrarias.

El algoritmo para la detección de los píxeles del borde en una orientación arbitraria es el siguiente:

1. Convolucionar la imagen  $I(x, y)$  con  $f_v(x, y)$  y  $f_H(x, y)$  para obtener los componentes del gradiente de  $R_v(x, y)$  y  $R_H(x, y)$ , respectivamente.
2. Definir el cuadrado de la magnitud gradiente de  $R(x, y) = R_v^2(x, y) + R_H^2(x, y)$ .
3. Marcar esos picos en  $R(x, y)$  que están por encima de cierto umbral predefinido  $T$ .

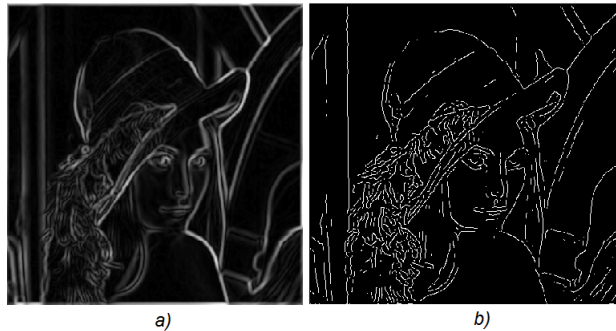
Figura 10.55: (a) Función bidimensional gaussiana. (b) Filtro Vertical. (c) Filtro horizontal.



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

**Detectores de gradiente borde.** En un robot móvil, el tiempo de cálculo debe reducirse al mínimo para conservar el comportamiento en tiempo real del robot. Por lo tanto, operadores discretos de núcleo más simple, se utilizan comúnmente para aproximar el comportamiento del detector de bordes de Canny. Un operador primitivo tal fue desarrollado por Roberts en 1965. Se utilizaron dos máscaras  $2 \times 2$  para calcular el

Figura 10.56: (a) Ejemplo de una imagen de borde; (b) Supresión Nonmaxima de (a).



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

gradiente a través del borde en dos direcciones diagonales. Sea  $r_1$  el valor calculado a partir de la primera máscara y  $r_2$  de la segunda máscara. Roberts obtuvo la magnitud del gradiente  $|G|$  con la ecuación

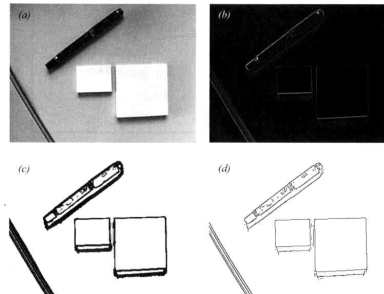
$$|G| \cong \sqrt{r_1^2 + r_2^2}; \quad r_1 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}; \quad r_2 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad (10.96)$$

Prewitt (1970) utiliza dos máscaras  $3 \times 3$  orientadas en las direcciones de fila y columna. Sea  $p_1$  el valor calculado a partir de la primera máscara y  $p_2$  el valor calculado a partir de la segunda máscara. Prewitt obtiene la magnitud del gradiente  $|G|$  y la dirección  $\theta$  del gradiente tomado en un ángulo en sentido horario con respecto al eje de la columna se muestra en la siguiente ecuación.

$$\begin{aligned} |G| &\cong \sqrt{p_1^2 + p_2^2}; \\ \theta &\cong \text{atan}\left(\frac{p_1}{p_2}\right) \\ p_1 &= \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \\ p_2 &= \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \end{aligned} \quad (10.97)$$

En el mismo año Sobel utilizó, como Prewitt, dos máscaras  $3 \times 3$  orientadas en la direc-

Figura 10.57: Ejemplo de extracción de características basada en visión con las diferentes etapas de procesamiento: (a) Datos brutos de imagen; (b) Imagen filtrada mediante un filtro Sobel; (c) Uso de umbral, selección de píxeles de borde (d) Supresión nonmaxima.



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

ción de la fila y la columna. Sea  $s_1$  el valor calculado a partir de la primera máscara y  $s_2$  el valor calculado a partir de la segunda máscara. Sobel obtuvo los mismos resultados que Prewitt para la magnitud del gradiente  $|G|$  y la dirección  $\theta$  del gradiente tomada en un ángulo en sentido horario con respecto al eje de la columna. La figura 10.57 muestra la aplicación del filtro Sobel a una escena visual. **Los talleres en Matlab de detección de bordes en imágenes y video se pueden encontrar en el Anexo C páginas 619 y 630.**

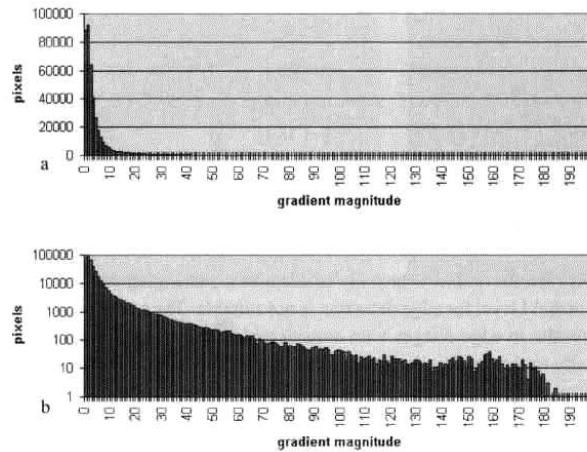
$$|G| \cong \sqrt{s_1^2 + s_2^2};$$

$$\theta \cong \text{atan} \left( \frac{s_1}{s_2} \right); \quad s_1 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}; \quad s_2 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (10.98)$$

**Umbral dinámico.** Un sistema de visión para robots móviles tiene que adaptarse a la iluminación cambiante. Por lo tanto un nivel de umbral constante para la detección de bordes no es adecuado. La misma escena con diferente iluminación resulta en imágenes de borde con diferencias considerables. Para adaptar dinámicamente el detector de borde a la luz ambiental, se requiere un umbral más adaptativo, y un método que incluya el cálculo de umbral basado en un análisis estadístico de la imagen a ser procesada.

Para ello, se calcula un histograma de las magnitudes del gradiente de la imagen procesada (figura 10.58). Con este histograma simple es fácil considerar sólo los píxeles  $n$  con la magnitud de gradiente más alta para etapas de cálculo adicionales. Los píxeles

Figura 10.58: (a) El número de píxeles con una magnitud de gradiente específica en la imagen de la figura 10.42(b). (b) Igual que (a), pero con escala logarítmica.



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

se cuentan hacia atrás a partir de la magnitud más alta. La magnitud del gradiente del punto donde  $n$  es alcanzada se utiliza como el valor de umbral temporal. La motivación de esta técnica es que los píxeles  $n$  con el mayor gradiente se espera que sean los más relevantes para la imagen procesada.

**Extracción borde recto: transformadas de Hough.** En robótica móvil el borde recto se obtiene a menudo como una característica específica. Los bordes verticales rectos, por ejemplo, se puede utilizar como indicios de la ubicación de las puertas y las intersecciones de pasillos. La transformada de Hough es una herramienta sencilla para la extracción de bordes de una forma particular. Aquí se explica su aplicación al problema de la extracción de bordes rectos.

Se supone que un píxel  $(x_p, y_p)$  en la imagen  $I$  que es parte de un borde. Cualquier borde de línea recta incluyendo el punto  $(x_p, y_p)$  debe satisfacer la ecuación:  $y_p = m_1 x_p + b_1$ . Esta ecuación sólo puede ser satisfecha con un conjunto limitado de valores posibles de  $m_1$  y  $b_1$ . En otras palabras, esta ecuación se satisface solamente por las líneas a través de  $I$  que pasan por  $(x_p, y_p)$ .

Consideremos ahora un segundo pixel  $(x_q, y_q)$ , en  $I$ . Cualquier línea que pasa a través de este segundo pixel debe satisfacer la ecuación:  $y_q = m_2 x_q + b_2$ . ¿Qué pasa si  $m_1 = m_2$  y  $b_1 = b_2$ ? Entonces, la línea definida por ambas ecuaciones es una y la misma: es la recta que pasa por ambos  $(x_p, y_p)$  y  $(x_q, y_q)$ .

Más en general, para todos los píxeles que forman parte de una sola línea recta a través de  $I$ , todos ellos debe estar en una línea definida por los mismos valores para  $m$  y  $b$ . La definición general de esta línea es, por supuesto,  $y = mx + b$ . La transformada de Hough utiliza esta propiedad básica, la creación de un mecanismo de manera que cada píxel de borde puede “votar” para diversos valores de los parámetros  $(m, b)$ . Las líneas con mayor número de votos al final son características de borde recto:

- Crear una matriz 2D  $A$  con ejes que teselan los valores de  $m$  y  $b$ .
- Inicializar el vector a cero.  $A[m, b] = 0$  para todos los valores de  $m, b$ .
- Para cada píxel de borde  $(x_p, y_p)$  en  $I$ , un lazo sobre todos los valores de  $m$  y  $b$ : Si  $y_p = mx_p + b$  entonces  $A[m, b] + 1$ .
- Buscar en las celdas de la  $A$  a identificar a aquellos con el mayor valor. Cada célula índices de  $(m, b)$  corresponde a un extraída de la línea recta en el borde  $I$ .

### 10.3.3. Cálculo de similitud de imagen

Los métodos descritos son basados en área. Supongamos que queremos comparar un parche  $m \times n$  en la imagen  $I_1$  centrada en  $(u, v)$  con otro parche del mismo tamaño, centrado en  $(u', v')$  en la imagen  $I_2$ . Suponemos que estos son números enteros impares, por lo tanto,  $m = 2a + 1$  y  $n = 2b + 1$ . La similitud se calcula entonces entre los niveles de intensidad de grises de los dos parches. Algunos de los criterios más populares son:

#### Suma de las diferencias absolutas (SAD)

$$SAD = \sum_{k=-a}^a \sum_{l=-b}^b | I_1(u + k, v + l) - I_2(u' + k', v' + l) | \quad (10.99)$$

#### Suma de los cuadrados de las diferencias (SSD)

$$SSD = \sum_{k=-a}^a \sum_{l=-b}^b [I_1(u + k, v + l) - I_2(u' + k', v' + l)]^2 \quad (10.100)$$

#### La correlación cruzada normalizada

$$NCC = \frac{\sum_{k=-a}^a \sum_{l=-b}^b [I_1(u + k, v + l) - \mu_1] \cdot [I_2(u' + k, v' + l) - \mu_2]}{\sqrt{\sum_{k=-a}^a \sum_{l=-b}^b [I_1(u + k, v + l) - \mu_1]^2 \sum_{k=-a}^a \sum_{l=-b}^b [I_2(u' + k, v' + l) - \mu_2]^2}} \quad (10.101)$$

Donde

$$\mu_1 = \frac{1}{mn} \sum_{k=-a}^a \sum_{l=-b}^b I_1(u+k, v+l) \quad (10.102)$$

$$\mu_2 = \frac{1}{mn} \sum_{k=-a}^a \sum_{l=-b}^b I_2(u'+k, v'+l) \quad (10.103)$$

Son los valores medios de los dos parches de imagen.

La SAD es la más simple entre estas medidas de similitud. La NCC es aún más compleja que los algoritmos SAD y SSD, ya que implica las numerosas operaciones de multiplicación, división y raíces cuadradas.

## 10.4. EXTRACCIÓN DE CARACTERÍSTICAS

Un robot móvil autónomo debe ser capaz de determinar su relación con el medio ambiente, haciendo mediciones con sus sensores y luego entonces, se utilizan las estas señales medidas. Las entradas del sensor deben ser utilizadas en una manera que permitan que el robot interactue con su entorno con éxito a pesar de la incertidumbre de medición.

Figura 10.59: La tubería perceptual: a partir de lecturas de los sensores a modelos de conocimiento.

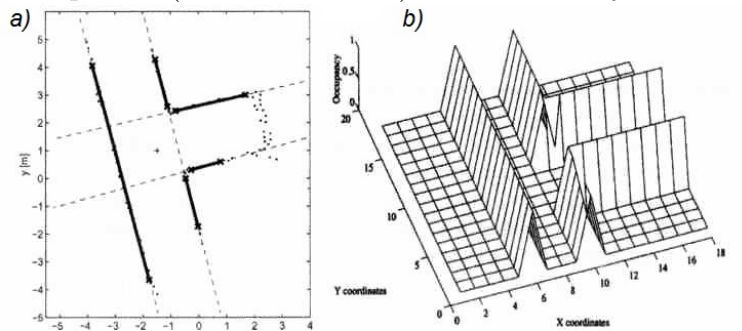


(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

Hay dos estrategias para el uso de una entrada incierta del sensor para guiar el comportamiento del robot. Una estrategia consiste en utilizar cada sensor de medición como un valor bruto e individual. La segunda estrategia es la de extraer información de una o más lecturas de los sensores primero, generando una percepción de nivel superior que puede entonces ser utilizado para reportar el modelo del robot y quizás las acciones del

robot directamente. Llamamos a este proceso extracción de características, y es este el siguiente y opcional paso en la tubería de interpretación perceptual (figura 10.59) que ahora se va a discutir. A medida que se avanza en tareas de percepción a largo plazo más sofisticadas, la extracción de características y aspectos escena de interpretación de la escena de la tubería perceptual se vuelve esencial.

Figura 10.60: Representación del entorno y modelado: (a) basado en funciones (continuo métrico), (b) red ocupación (métrica discreta). Cortesía de Sjur Vestli.



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

**Definición de característica.** Las características son las estructuras reconocibles de elementos en el medio ambiente. En robótica móvil, las características juegan un papel especialmente importante en la creación de modelos ambientales. Permiten descripciones más compactas y sólidas del medio ambiente, ayudan a un robot móvil, tanto durante la construcción de mapas como la localización. En el diseño de un robot móvil, una decisión crítica gira en torno a la elección de las características apropiadas para el robot a usar. Varios factores son esenciales para esta decisión:

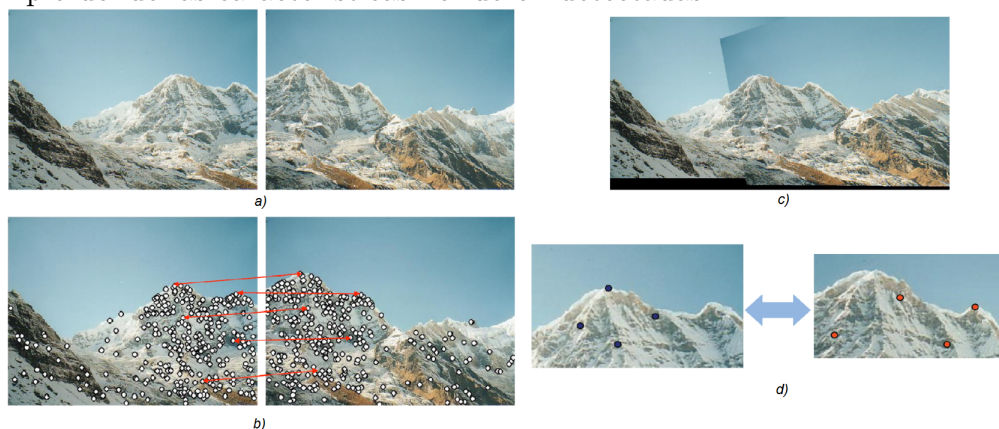
- **Ambiente objetivo.** Para que las características geométricas sean útiles, las geometrías objetivo deben ser fácilmente detectadas en el entorno real.
- **Sensores disponibles.** Obviamente, los sensores específicos y la incertidumbre del sensor del robot afecta a la idoneidad de diversas características.
- **Potencia computacional.** La extracción de características basado en Visión puede efectuar un coste computacional importante, en particular en robots donde el procesamiento del sensor de visión se ejecuta por uno de los procesadores principales del robot.

- Representación del ambiente.** La extracción de características es un paso importante hacia la interpretación de la escena, y por eso mismo las características extraídas deben proporcionar información que está en consonancia con la representación utilizada para el modelo ambiental. La figura 10.60 muestra la aplicación de dos representaciones diferentes a la tarea de modelar un pasillo del edificio de oficinas. Cada método tiene sus ventajas y desventajas, pero la extracción de características de línea y de la esquina tiene mucha más importancia que la representación de la izquierda.

## 10.5. EXTRACCIÓN DE CARACTERÍSTICAS DE IMAGEN: DETECTORES DE PUNTO DE INTERÉS

En esta sección, se define el concepto de la función local y se revisan algunos de los extractores de características más consolidadas. Sólo se describen en detalle los dos detectores de características más populares, es decir, Harris y SIFT y se presentarán brevemente otros, explicando las principales ventajas y desventajas y el dominio de aplicación.

Figura 10.61: (a) Panorama construido a partir de múltiples imágenes superpuestas utilizando software Autostitch, (b) Primer paso, (c) Segunda etapa, (d) Dos imágenes de ejemplo donde las características no fueron detectadas



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

### 10.5.1. Introducción

Una característica local es un patrón de imagen que difiere de su vecindad inmediata en términos de intensidad, color y textura. Las características locales pueden ser pequeños parches de imagen (como regiones de color uniforme), bordes o puntos (tales como esquinas originadas a partir de intersecciones de línea). En la primera categoría están las características que tienen una interpretación semántica tales como, por ejemplo, bordes correspondientes a los carriles de la carretera o manchas que corresponden a células de la sangre en las imágenes médicas.

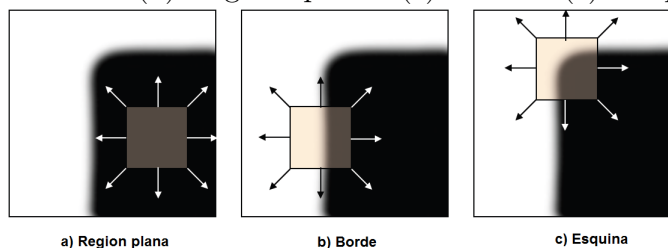
En la segunda categoría están las características que no tienen una interpretación semántica. En este caso, lo que las características realmente representan no es relevante. Lo que importa es que su ubicación se puede determinar con precisión y firmeza en el tiempo. Las aplicaciones típicas son seguimiento de características, calibración de la cámara, reconstrucción 3D, mosaicos de imágenes y composición panorámica. Por último, en la tercera categoría son las características que aún no tienen una interpretación semántica si se toman por separado, pero que se pueden utilizar para reconocer una escena o un objeto si se toman todas juntas. Por ejemplo, una escena podría ser reconocida contando el número de coincidencias entre las características de la escena observada y la imagen de búsqueda.

### 10.5.2. Propiedades del detector de característica ideal

En esta sección se resumen las propiedades que un detector característica ideal debería tener. Comencemos con un ejemplo concreto de la fotografía de la imagen digital. La mayoría de las cámaras digitales para el consumidor de hoy en día vienen con un software para la composición automática de panorámicas de varias fotos. Un ejemplo se muestra en la figura 10.61. El usuario simplemente toma varias fotos de la escena con poca superposición entre imágenes adyacentes y el software automáticamente las alinea y fusiona todas juntas en un panorama cilíndrico (figura 10.61a). El desafío clave es identificar las regiones correspondientes entre las imágenes que se superponen. Una manera de resolver este problema es extraer puntos característicos de las imágenes adyacentes, encontrar los pares correspondientes de acuerdo con alguna medida de similitud (figura 10.61b), y calcular la transformación (por ejemplo, homografía) para alinearlos (figura 10.61c). El primer problema es cómo detectar los mismos puntos de forma independiente en ambas imágenes. En la figura 10.61d, por ejemplo, las características de la imagen a la izquierda no se vuelven a detectar en la imagen derecha. Debido a esto, es necesario un detector de característica “repetible”. El segundo problema es el

siguiente: para cada punto de la imagen primero es necesario reconocer correctamente el correspondiente en la segunda imagen. Así, las características detectadas deben ser muy distintivas.

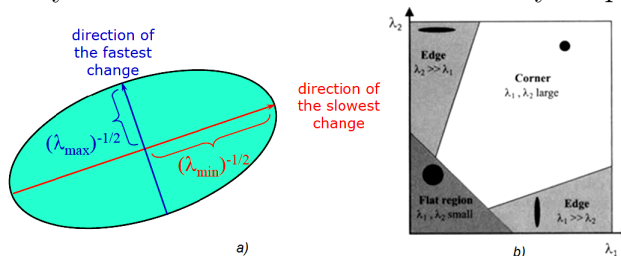
Figura 10.62: (a) Región “plana” (b) “Borde” (c) “Esquina”



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

La “Repetibilidad” es probablemente la propiedad más importante de un detector buena característica. Dadas dos imágenes de la misma escena tomadas en condiciones de visualización y condiciones de iluminación diferentes, es deseable que un alto porcentaje de las características de la primera imagen puedan ser detectadas de nuevo en la segunda imagen. La segunda propiedad importante es “distintividad”, es decir, la información transportada por el parche alrededor del punto de característica debe ser tan distintiva como sea posible para que las características se puedan distinguir y combinarse. Por ejemplo, las esquinas de un tablero de ajedrez no son distintivas porque no pueden ser distinguidos unas de otras.

Figura 10.63: (a) Esta elipse se construye a partir de la matriz de segundo momento y visualiza las direcciones de cambio de intensidad más rápidas y más bajas, (b) La clasificación de esquina y los bordes de acuerdo con Harris y Stephens.



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

Otras propiedades importantes de un detector buena característica son:

- Precisión de localización: las características detectadas deben ser localizadas con precisión, tanto en la posición de la imagen como en la escala.
- Cantidad de funciones: el número ideal de características detectadas depende de la aplicación.
- Invariancia: buenas características deben ser invariantes a los cambios de punto de vista de la cámara, la iluminación del ambiente, y la escala.
- Eficiencia computacional: también es deseable que las características se puedan detectar y emparejar de manera muy eficiente.
- Robustez: las características detectadas debe ser robustas al ruido de imagen, efectos de discretización, artefactos de compresión, difuminado, desviaciones del modelo matemático utilizado para obtener la invariancia, y así sucesivamente.

### 10.5.3. Detectores de Esquina

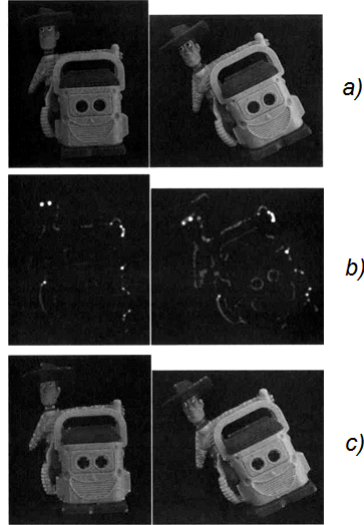
Una esquina en una imagen puede ser definida como la intersección de dos o más bordes. Las esquinas son características con alta repetibilidad.

**El concepto básico de detección de esquina.** Uno de los primeros detectores de esquina fue inventado por Moravec (Towards automatic visual obstacle avoidance, 1977 y Visual mapping by a robot rover, 1979) Él define una esquina como un punto donde hay una gran variación de intensidad en todas las direcciones. Una explicación intuitiva de su algoritmo de detección de ángulo aparece en la figura 10.62. Intuitivamente, se puede reconocer una esquina mirando a través de una pequeña ventana centrada en el píxel. Si el píxel se encuentra en una región “plana” (es decir, una región de intensidad uniforme), entonces las ventanas adyacentes serán similares. Si el píxel está de a lo largo de un borde, entonces las ventanas adyacente en la dirección perpendicular al borde serán diferentes, pero las ventanas adyacentes en una dirección paralela al borde resultarán sólo en un pequeño cambio. Por último, si el píxel está en una esquina, entonces ninguna de las ventanas adyacentes será similar.

#### **El detector esquina de Harris.**

Harris y Stephens mejoraron el detector de esquina Moravec considerando las derivadas parciales de la SSD en lugar de usar ventanas desplazadas (A combined corner and edge detector, 1988).

Figura 10.64: Extracción de las esquinas de Harris, (a) Imagen original, (b) la función Cornerness, (c) Las esquinas de Harris se identifican como máximos locales de la función cornerness



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

Sea  $I$  una imagen en escala de grises. Se considera la adopción de un parche de imagen centrado en  $(u, v)$  y desplazándolo por  $(x, y)$ . La suma de las diferencias de cuadrados SSD entre estos dos parches viene dada por:

$$SSD(x, y) = \sum_u \sum_v ((I(u, v)) - I(u + x, v + y))^2 \quad (10.104)$$

$I(u + x, v + y)$  puede ser aproximada por una expansión de Taylor de primer orden. Sean  $I_x$  e  $I_y$  las derivadas parciales de  $I$ , de forma que

$$I(u + x, v + y) \approx I(u, v) + I_x(u, v)_x + I_y(u, v)_y \quad (10.105)$$

Esto produce la aproximación

$$SSD(x, y) \approx \sum_u \sum_v (I_x(u, v)_x + I_y(u, v)_y)^2 \quad (10.106)$$

Que puede escribirse en forma matricial:

$$SSD(x, y) \approx \begin{bmatrix} x & y \end{bmatrix} M \begin{bmatrix} x \\ y \end{bmatrix} \quad (10.107)$$

Donde  $M$  es la matriz de segundo momento

$$M = \sum_u \sum_v \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \sum \sum I_x^2 & \sum \sum I_x I_y \\ \sum \sum I_x I_y & \sum \sum I_y^2 \end{bmatrix} \quad (10.108)$$

Y puesto que  $M$  es simétrica, podemos reescribir como  $M$

$$M = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R \quad (10.109)$$

Donde  $\lambda_1$ , y  $\lambda_2$  son los valores propios de  $M$ .

Figura 10.65: Modelos de cambios de imagen.

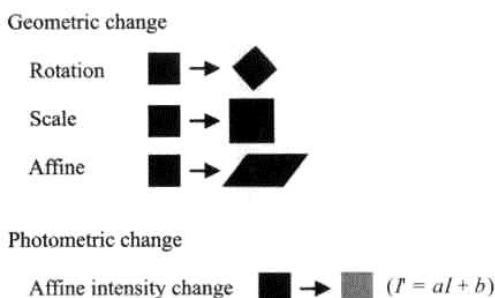


Figure 4.69 Models of image changes.

(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

Como se mencionó antes, una esquina se caracteriza por una gran variación de SSD en todas las direcciones del vector  $(x, y)$ . El detector de Harris analiza los valores propios de  $M$  para decidir si estamos en presencia de una esquina o no. Primero vamos a dar una explicación intuitiva antes de mostrar la expresión matemática. Usando la ecuación 10.107 se puede visualizar  $M$  como una elipse (figura 10.63a) de la ecuación:

$$\begin{bmatrix} x & y \end{bmatrix} M \begin{bmatrix} x \\ y \end{bmatrix} = const \quad (10.110)$$

Las longitudes de eje de esta elipse se determinan por los valores propios de  $M$  y la orientación está determinada por  $R$ . Sobre la base de las magnitudes de los valores propios, las siguientes inferencias pueden hacerse con base en este argumento:

- Si tanto  $\lambda_1$  y  $\lambda_2$  son pequeñas, la SSD es casi constante en todas las direcciones (es decir, estamos en presencia de una región plana).

- Si  $\lambda_1 \gg \gg \lambda_2$  o  $\lambda_2 \gg \gg \lambda_1$ , estamos en presencia de un borde: la SSD tiene una gran variación en una sola dirección, que es la dirección perpendicular al borde.
- Si tanto  $\lambda_1$  y  $\lambda_2$  son grandes, la SSD tiene grandes variaciones en todas las direcciones y entonces estamos en presencia de una esquina.

Las tres situaciones mencionadas anteriormente se resumen pictóricamente en la figura 10.63b.

Debido a que el cálculo de los valores propios es computacionalmente costoso, Harris y Stephens sugirieron en su lugar el uso de la siguiente “función cornerness (esquinicidad)”:

$$C = \lambda_1 \lambda_2 - l(\lambda_1 \lambda_2)^2 = \det(M) - k \cdot \text{trace}^2(M) \quad (10.111)$$

Donde  $k$  es un parámetro de sensibilidad ajustable. De esta manera, en lugar de calcular los valores propios de  $M$ , sólo tenemos que evaluar el determinante y traza de  $M$ . El valor de  $k$  se tiene que determinar empíricamente. En la literatura, los valores se describen con frecuencia en el rango de 0.04-0.15. El último paso del detector de esquina de Harris consiste en extraer los máximos locales de la función cornerness, mediante la supresión nonmaxima (La nonmaxima supresion involucra visitar cada pixel de la función cornerness y determinar si es o no un máximo local. Si no, entonces el valor es ajustado a cero. Esto solo hace que los máximos se preserven). Finalmente sólo los máximos locales que están por encima de un umbral dado se retienen. Los pasos de proceso se ilustra en la figura 10.64. La figura 10.64c muestra las esquinas detectadas en las dos imágenes de ejemplo. **El taller de detección de esquinas se puede encontrar en el Anexo C página 624.**

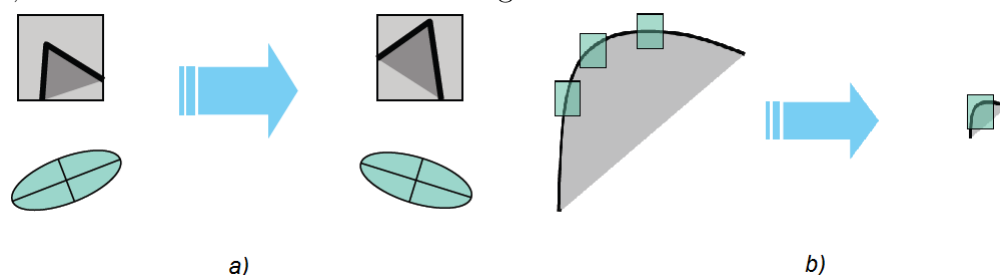
#### 10.5.4. Invariancia a los cambios geométricos y fotométricos

Como se observa en la sección 10.5.2, en general se desea que las características sean detectadas a pesar de los cambios geométricos y fotométricos de la imagen: si tenemos dos versiones transformadas de la misma imagen, las características deben ser detectadas en las ubicaciones correspondientes. Las transformaciones de imagen pueden afectar la geometría o las propiedades fotométricas de una imagen. Los modelos consolidados de transformaciones de imagen son los siguientes (véase también la figura 10.65):

- Cambios geométricos:
  - Rotación 2D
  - Escala (cambio de escala uniforme)

- Afinidad
- Cambios fotométricos
- Intensidad Afín

Figura 10.66: (a) Detector de Harris es invariante a la rotación de la imagen (b) A la inversa, no es invariante a escala de la imagen



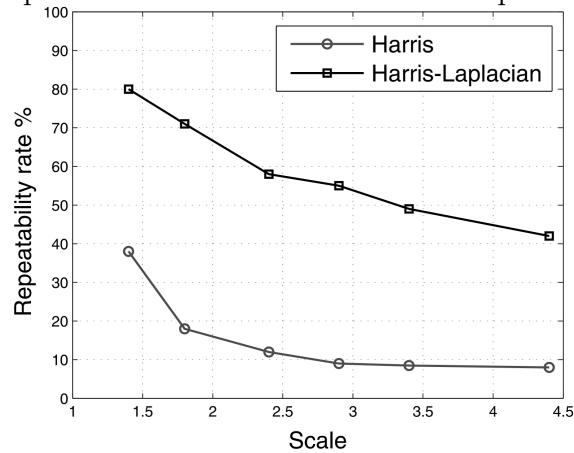
(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

En este caso la transformación válida sólo para objetos planos sería una homografía. Sin embargo, cuando los cambios de punto de vista son pequeños y el objeto es localmente plano, la transformación afín es una buena aproximación de la homografía. La rotación 2D se produce sólo si la cámara gira alrededor de su eje puramente óptico. EL cambio de escala uniforme, en cambio, aparece cuando la cámara se acerca (hacia adentro o fuera) o se traslada a lo largo de la dirección de su eje óptico, pero este último sólo es válido para objetos locamente planos.

A modo de ejemplo, examinemos ahora la invariancia del detector de Harris para las transformaciones antes mencionadas. Podemos observar que el detector de Harris es invariante a las rotaciones de imágenes 2D. Esto puede explicarse por la observación de que los valores propios de la matriz de segundo momento no cambian bajo rotación pura. De hecho, la elipse gira. Pero su forma (es decir, los valores propios) sigue siendo la misma (véase la figura 10.66a). Como cuestión de hecho, observar que para hacer el detector de Harris isotrópico (es decir, uniforme para todas las rotaciones), la matriz de segundo momento debe ser calculada en una región circular en lugar de una ventana cuadrada. Esto se hace generalmente por un promedio de la ecuación 10.108 con una función simétrica circular gaussiana.

El detector de Harris es también invariante a cambios de intensidad afín. En este caso, los valores propios, y por tanto la función cornerness, se reajustarán por un factor

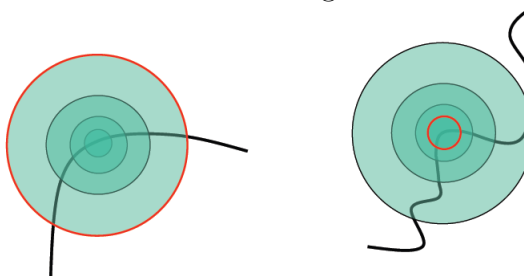
Figura 10.67: Tasa de repetición de detector de Harris. Comparación con Harris-Laplace.



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

constante, pero la posición de los máximos locales de la función cornerness sigue siendo el mismo. A la inversa, el detector de Harris no es invariante a transformaciones afines geométricas o cambios de escala. Intuitivamente, una transformación afín distorsiona la zona de la característica a lo largo de los ejes  $x$ , y  $y$ , en consecuencia, una esquina puede conseguir reducir o aumentar su curvatura. Respecto a los cambios de escala, esto se aclara de inmediato como se observa en la figura 10.66b. En esta figura, la esquina sería clasificada como un borde en una escala alta y como una esquina en una escala más pequeña.

Figura 10.68: Análisis de imagen a diferentes escalas



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

El rendimiento del detector de Harris contra los cambios de escala se muestra en la figura 10.67 de acuerdo con un estudio comparativo realizado por Mikolajczyk (Indexing

base don scale-invariant interest points, 2001). En esta figura, se representa la tasa de repetición contra el factor de escala. La tasa de repetición entre dos imágenes se calcula como la relación entre el número de correspondencias encontradas y el número de todas las correspondencias posibles. Como se observa, después de reescalar la imagen en un factor 2, sólo el 20 % de las correspondencias posibles se vuelven a detectar.

Figura 10.69: Intensidad media como una función del tamaño de la región. (a) Imagen original (b) Imagen recalibrada (cambio de tamaño).



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

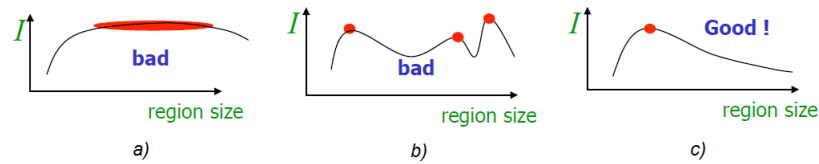
### Detección invariante de escala.

En la figura 10.68, se ve que una forma de detectar la esquina en mayor escala es el uso de un detector de multiescala Harris. Esto significa que el mismo detector se aplica varias veces sobre la imagen, cada vez con un tamaño de ventana (círculo) diferente. Una aplicación eficaz de la detección multiescala utiliza la llamada escala espacio-pirámide: en vez de variar el tamaño de la ventana del detector de característica, la idea es generar versiones superiores o disminuidas del muestreo de la imagen original (es decir, la pirámide).

Al usar detector Harris multiescala, podemos estar seguros de que en algún punto la esquina de la imagen 10.68a será detectada. Una vez que el punto se ha detectado en ambas imágenes se plantea la cuestión: ¿Cómo seleccionar la escala correspondiente? En otras palabras, ¿cómo elegimos correspondientes círculos de forma independiente en cada imagen?.

En la visión por ordenador, la correcta selección de escala se realiza siguiendo el método propuesto en 1998 por Lindeberg: la escala apropiada de un local para una característica puede ser elegida como aquella para la cual una función dada alcanza un máximo o mínimo en escalas (Feature detection with automatic scale selection, 1998). Por cada círculo en las dos imágenes de la figura 10.68, vamos a trazar la intensidad media de los

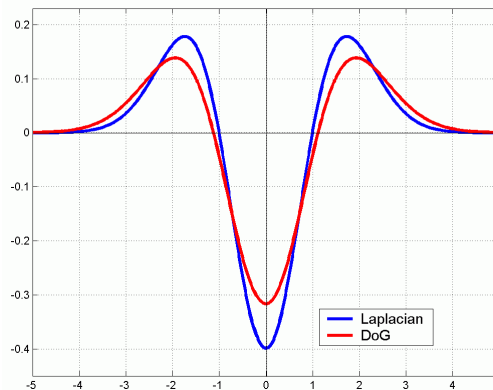
Figura 10.70: (a) – (b) son funciones malas de escala invariante. (c) es una buena.



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

píxeles dentro del círculo como una función del tamaño de círculo. Por estas imágenes, se obtendrán dos gráficos que se muestran en la figura 10.69. Como era de esperar, estas dos funciones lucen igual hasta un cambio de escala en el eje  $x$ . La solución a nuestro problema es, pues, tomar como escalas correspondientes los tamaños de círculo paralelos cuales estas funciones alcanzan su máximo. Dependiendo de la función elegida, se podría tomar el mínimo en lugar del máximo.

Figura 10.71: Comparación entre el Laplaciano de Gauss y diferencia de Gauss.

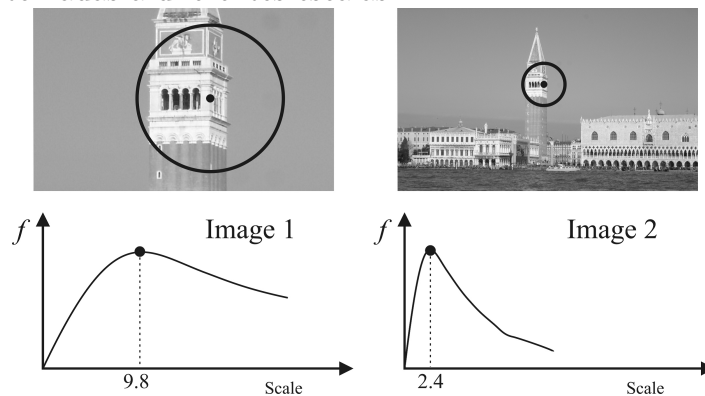


(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

El problema es cómo diseñar una buena función. La primera observación que podemos hacer es que una “buena” función para detección de escala debe tener un pico agudo estable como el de la figura 10.70. A pesar de la utilizada en nuestro ejemplo, la intensidad media no es buena, ya que puede devolver múltiples picos o incluso ningún pico en absoluto. Como cuestión de hecho, resulta que para las imágenes habituales una buena función es la que corresponde a contraste, es decir, los cambios bruscos locales de

intensidad. El Laplaciano de Gaussian (LoG) (figura 10.71) es un buen operador para la identificación de los cambios bruscos de intensidad y actualmente es el utilizado para la selección de escala por el detector de esquina de Harris. En el estudio comparativo de Mikolajczyk presentado anteriormente, el operador de registro ha demostrado que proporciona los mejores resultados con respecto a otras funciones. La respuesta de LoG de más de dos características de la imagen se muestra en la figura 10.72.

Figura 10.72: Respuesta del operador de registro en dos puntos correspondientes a partir de dos imágenes tomadas a diferentes escalas.



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

La comparación entre el Harris estándar y el Laplaciano Harris sobre la escala se muestra en la figura 10.67.

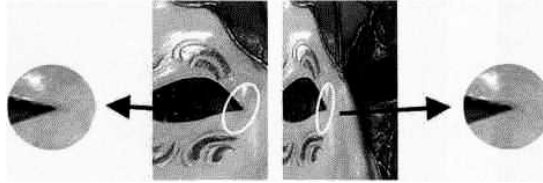
### Detección invariante afín.

En la sección anterior, se consideró que el problema de la detección bajo “reescalado” uniforme. El problema ahora es cómo detectar las mismas características en virtud de transformación afín, que puede ser vista como un cambio de escala no uniforme.

El procedimiento consiste en los pasos siguientes:

- En primer lugar, las características son identificadas usando el detector de escala invariante Harris-Laplaciano.
- Luego, la matriz de segundo momento 10.108 se utiliza para identificar las dos direcciones de cambio de intensidad más lenta y más rápida alrededor de la característica.

Figura 10.73: Cálculo de la elipse afín invariante en dos imágenes relacionadas por una transformación afín.



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

- Fuera de estas dos direcciones, una elipse se calcula al mismo tamaño que el de escala calculada con el operador LoG.
- La región dentro de la elipse se normaliza a una circular.
- La elipse inicial detectada y la forma circular resultante normalizada se muestran en la figura 10.73.

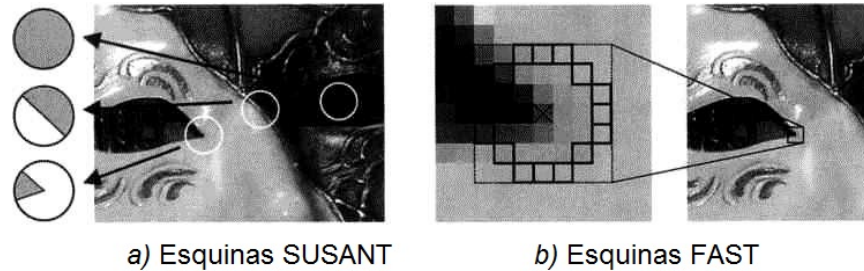
### Otros detectores de esquina.

**El detector de esquina Shi-Tomasi.** Este también se conoce a veces como el detector de esquina Kanade-Tomasi (*Good features to track, IEEE, 1994*). Este detector está fuertemente basado en el detector de esquina de Harris. Los autores muestran que para parches de imagen que se someten a transformaciones afines,  $\min(\lambda_1, \lambda_2)$  es una medida mejor que la función cornerness  $C$  (10.111).

**El detector esquina SUSAN.** SUSAN significa (Smallest Univalued Segment Assimilating Nucleus) núcleo más pequeño segmento UNIVALUED Asimilar y, además de ser utilizado para la detección de la esquina, también se utiliza para la detección de bordes y la supresión de ruido. El detector de esquina SUSAN ha sido introducido por Smith y Brady (*SUSAN: A new approach to low level image processing, 1997*). Su principio de funcionamiento es diferente del detector de Harris. Como hemos visto, Harris se basa en gradientes locales de la imagen, que son computacionalmente costosos de calcular. A la inversa, SUSAN se basa en un enfoque morfológico, que es computacionalmente mucho más eficiente que Harris.

El principio de funcionamiento de Susan es muy simple (véase también la figura 10.74a). Para cada píxel de la imagen, SUSAN considera una ventana circular de radio fijo centrado en él. Entonces, todos los píxeles dentro de esta ventana se dividen en dos categorías, dependiendo de si tienen valores de intensidad “similares” o “diferentes” como

Figura 10.74: (a) Detector SUSAN compara píxeles dentro de una región circular, mientras que FAST (b) los compara sólo en un círculo.



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

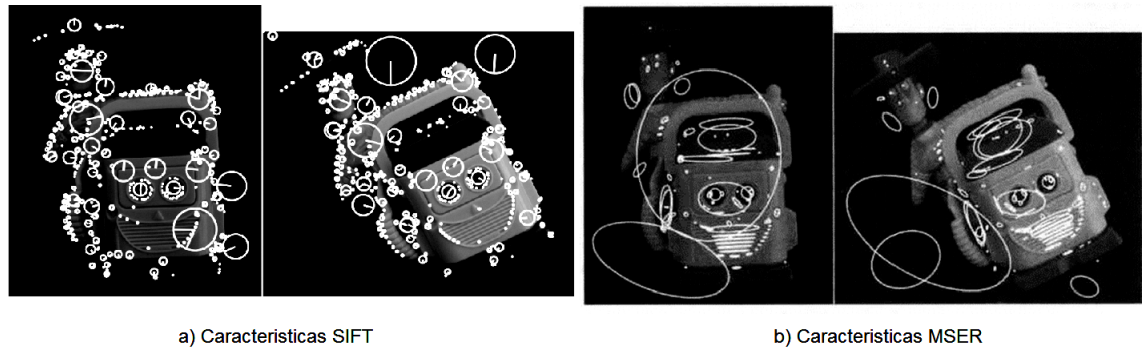
el píxel central. En consecuencia, en las regiones de intensidad uniforme de la imagen, la mayoría de los píxeles dentro de la ventana tendrán un brillo similar como el píxel central. Cerca de los bordes, la fracción de píxeles con intensidad similar se reducirá un 50 %, mientras que cerca de las esquinas se reducirá a más del 25 % aproximadamente. Así, las esquinas SUSAN son identificadas como lugares de imagen donde el número de píxeles con un brillo similar en un sector local alcanza un mínimo local y está por debajo de un umbral especificado. Como paso final, la “nonmaxima supression” se utiliza para identificar mínimos locales.

**El detector de esquina FAST.** El detector FAST (Features from accelerated Segment Test), fue presentado por Rosten y Drummond, este detector se basa en el detector SUSAN (*Fusing points and lines for high performance tracking, 2005* y *Machine learning for high-speed corner detection, 2006*). SUSAN calcula la fracción de píxeles dentro de una ventana circular, que tienen una intensidad similar como el píxel central. A la inversa, FAST compara píxeles sólo en un círculo de 16 píxeles alrededor de la esquina candidata (figura 10.74b). Esto se traduce en un detector muy eficiente, que es de hasta treinta veces más rápido que Harris: FAST tarda sólo de 1 a 2 milisegundos en un ordenador portátil de doble núcleo de 2 GHz y es actualmente el detector de características computacionalmente más eficiente disponible. Sin embargo, como SUSAN, no es robusto a niveles altos de ruido.

### 10.5.5. Detectores de mancha (Blob).

Una mancha (blob) es un patrón de imagen que difiere de su vecindad inmediata en términos de intensidad, color y textura. No es un borde, ni una esquina. La precisión

Figura 10.75: La extracción de características SIFT y MSER de la imagen ejemplo utilizada para el detector de Harris en la figura 10.64



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

de localización de una mancha es normalmente menor que la de una esquina, pero su amplitud y forma se definen mejor. Para ser más claro, una esquina puede ser localizada por un solo punto (por ejemplo, la intersección de dos bordes), mientras que una mancha sólo puede ser localizada por su límite. Por otro lado, una esquina es menos precisamente localizada sobre la escala, porque, como se ha señalado antes, una esquina se encuentra en la intersección de los bordes y por lo tanto su aspecto cambia muy poco a escalas adyacentes. A la inversa, una mancha es más exactamente localizada sobre la escala, porque el límite de una mancha define inmediatamente su tamaño y por lo tanto su escala.

Utilizando la nueva terminología, los detectores de mancha también pueden ser conocidos como operadores de puntos de interés, o alternativamente como operadores de región de interés. Algunos ejemplos de características de tipo mancha se muestran en la figura 10.75. En esta figura se pueden observar dos tipos de características que se describen en esta sección, es decir SIFT y MSER. Como se observa, MSER da privilegios a las regiones con intensidad uniforme, mientras que SIFT no. **Ver Anexo C página 627 para el taller en Matlab correspondiente.**

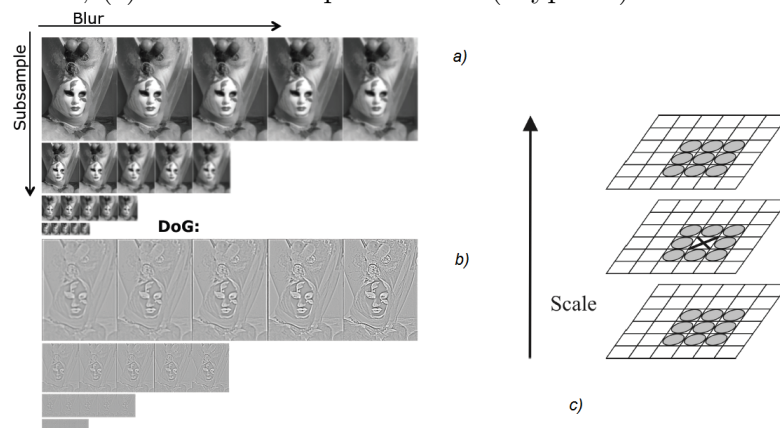
### **Características SIFT.**

SIFT significa (Scale Invariant Feature Transform) y es un método para detectar y hacer coincidir puntos clave robustos, el cual fue inventado en 1999 por Lowe (*Object recognition from local scale-invariant features*). La singularidad de SIFT es que estas características son muy distintivas y pueden hacerse coincidir con éxito entre imágenes

con cambios de iluminación, rotación, punto de vista y la escala muy diferentes. Su alta capacidad de repetición y alta tasa de coincidencia en condiciones muy difíciles han hecho que SIFT sea el mejor detector de característica hasta ahora. Se ha encontrado muchas aplicaciones en el reconocimiento de objetos, la cartografía y la navegación robótica, composición de la imagen (por ejemplo, panoramas, mosaicos), modelado 3D, reconocimiento de gestos, seguimiento por cámara, y el reconocimiento de rostros.

La principal ventaja de las características SIFT en comparación con todos los métodos explicados anteriormente es que un “descriptor” se calcula a partir de la región alrededor del punto de interés, el cual distintivamente se describe la información que lleva la característica. Como se verá, este descriptor es un vector que representa la distribución local de los gradientes de la imagen alrededor del punto de interés. Como ha demostrado por su inventor, en realidad este descriptor lo que hace robusto a SIFT a la rotación y pequeños cambios de iluminación, escala y puntos de vista.

Figura 10.76: (a) Imágenes gaussianas borrosas en diferentes escalas, (b) Imágenes de Diferencia de Gauss, (c) Selección de punto clave (keypoint)



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

A continuación, se analizarán los pasos principales del algoritmo SIFT, que son:

- Identificación de la ubicación del punto clave y escala
- Asignación de orientación
- Generación del descriptor del punto clave

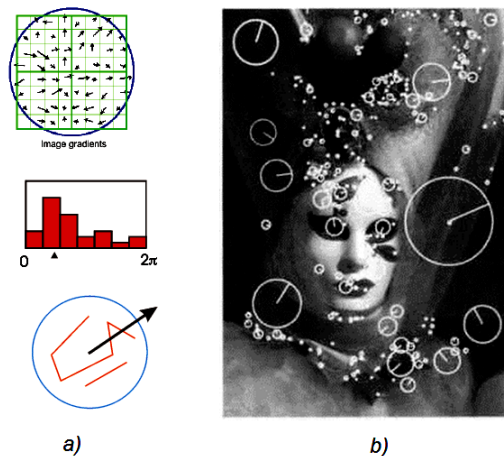
**Identificación de la ubicación punto clave y escala.** El primer paso hacia la identificación de puntos clave SIFT es la generación de las llamadas imágenes de

diferencia de Gauss (DoG). Esto se realiza primero borrando la imagen original con filtros gaussianos a escalas diferentes (es decir, sigma diferente) y después tomando la diferencia de las sucesivas imágenes borrosas gaussianas. Este proceso se muestra en la figura 10.76a: la imagen original (parte superior izquierda) está enmascarada con cuatro filtros gaussianos con diferente sigma, y esto se repite después de reducción de la resolución de la imagen de un factor de 2. Finalmente, las imágenes DoG son calculadas simplemente tomando la diferencia entre las sucesivas imágenes borrosas 10.76b.

El segundo paso es la selección de los puntos clave. Los puntos clave SIFT son identificados como locales máximos o mínimos de las imágenes DoG a través de escalas. En particular, cada píxel de la imagen DoG es comparada con sus ocho vecinos en la misma escala, además de los nueve vecinos a escalas adyacentes (figura 10.76c). Si el píxel es un máximo o mínimo local, se selecciona como un punto clave candidato.

El tercer paso consiste en refinar la localización, tanto en el espacio como en la escala, de los puntos clave interpolando los datos cercanos. Finalmente, puntos clave con poco contraste o largo de los bordes se eliminan debido a su bajo carácter distintivo y debido a su inestabilidad al ruido de la imagen.

Figura 10.77: (a) Asignación de Orientación, (b) Algunas características SIFT con orientación y escala detectada.



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

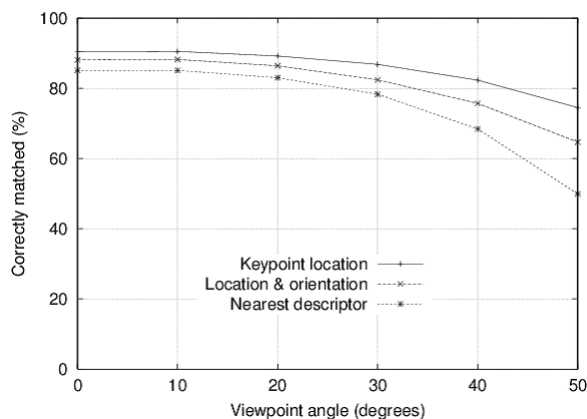
**Asignación de orientación.** Esta etapa consiste en asignar a cada punto clave una orientación específica para hacerlo invariante a la rotación de la imagen. Para determinar la orientación de punto clave, un histograma de gradiente de orientación

se calcula en la vecindad del punto clave. En otras palabras, para cada píxel de ese sector de los puntos clave, el gradiente de intensidad (magnitud y dirección) se calcula. Entonces un histograma de orientaciones se construye de tal manera que la contribución de cada píxel es ponderada por la magnitud del gradiente.

Los picos en el histograma corresponden a las orientaciones dominantes (figura 10.77a). Una vez que el histograma se llena, la orientación correspondiente al pico más alto se asigna al punto clave. Los puntos clave finales con la orientación y la escala seleccionada se muestran en la figura 10.77b. Finalmente, el descriptor se construye mediante el apilamiento de todas las entradas de histograma de orientación. Por lo tanto, la longitud final del vector descriptor es  $4 \times 4 \times 8 = 128$  elementos. Para lograr invariancia de iluminación parcial, el vector descriptor finalmente se normaliza para tener norma unitaria.

En el trabajo de Lowe (*Distinctive image features from scale-invariant keypoints, 2004*), se ha demostrado también que la precisión de coincidencia de característica está por encima del 50 % para los cambios punto de vista de hasta 50 grados (figura 10.78). Por lo tanto los descriptores SIFT son invariantes a los cambios de punto de vista de menor importancia.

Figura 10.78: Fracción de puntos clave SIFT correctamente emparejados como una función del ángulo de punto de vista.



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

Debido a su alta capacidad de repetición y distintividad, SIFT ha demostrado en los últimos diez años ser uno de los mejores detectores de características en una amplia gama de aplicaciones, aunque es superado en rendimiento por SURF. Excelentes resultados se

han logrado en la navegación de robots, reconocimiento de objetos 3D, reconocimiento de lugares, SLAM, composición panorámica, recuperación de imágenes, y muchos otros.

### Otros detectores de mancha.

**El detector MSER.** Regiones Extremas Máximamente Estables (MSER - Maximally Stable Extremal Regions) ha sido propuesto por Matas et al., para hacer coincidir las características que son robustas ante cambios de punto de vista de gran tamaño (*Robust wide-line stereo from maximally stable extremal regions, 2002*). Una región extrema máximamente estable es un componente conectado de píxeles que tiene intensidad o más alta o más baja que todos los píxeles en su contorno exterior (figura 10.75b). Estas regiones extremas son seleccionadas usando un umbral de intensidad adecuado y tienen una serie de propiedades deseables. Primero, son completamente invariante a los cambios de intensidad monotónicas. En segundo lugar, son invariantes a transformaciones de imagen afines.

**El detector SURF.** SURF significa Speeded Up Robust Features en español características robustas aceleradas y ha sido propuesto por Bay et al (2008). Este detector de característica invariante en escala está fuertemente inspirado en SIFT pero es varias veces más rápido. Básicamente, utiliza “Haar wavelets” (Un haar wavelet es una función constante por tramos) para aproximar filtros DoG e imágenes integrales para convolución, que hacen que el proceso de filtración mucho más eficiente a expensas de una menor robustez con respecto a SIFT.

### Resumen de los detectores de características.

La Tabla 10.2 proporciona una visión general de las propiedades más importantes de los detectores de características descritos en los apartados anteriores. La más alta repetibilidad y precisión de localización se obtiene por el detector de Harris y sus versiones invariable afín y de escala y. Los detectores SUSAN y FAST evitan el cálculo de derivadas de la imagen y por tanto son más eficientes que los de Harris pero la ausencia de suavizado los hace más sensibles al ruido. Los originales Harris, Shi-Tomasi, SUSAN, y FAST no son invariante en escala, sin embargo un poco de literatura que existe sobre cómo lograr la invarianza de escala mediante el método descrito anteriormente.

A diferencia del método original Harris, Harris-Laplace alcanza invariancia de escala, sin embargo, su estimación de escala es menos precisa que SIFT, MSER, o SURF por la naturaleza multiescalar de las esquinas. Por último, el detector SURF muestra una alta repetibilidad, escala e invariancia de punto de vista. Sin embargo, se ha ideado para la eficiencia, y por lo tanto no funcionan tan bien como SIFT.

Cuadro 10.2: Comparación de los detectores de características: propiedades y rendimiento.

	Detector de Esquina	Detector de Mancha	Rotación invariante	Escala invariante	Afinidad invariante	Repetitividad	Precisión Localización	Robustez	Eficiencia
Harris	X		X			+++	+++	++	++
Shi-Tomasi	X		X			+++	+++	++	++
Harris-Laplacian	X	X	X	X		+++	+++	++	+
Harris-Affine	X	X	X	X	X	+++	+++	++	++
SUSAN	X		X			++	++	++	+++
FAST	X		X			++	++	++	++++
SIFT		X	X	X	X	+++	++	+++	+
MSER		X	X	X	X	+++	+	+++	+++
SURF		X	X	X	X	++	++	++	++

Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.

## 10.6. RECONOCIMIENTO DE LUGAR

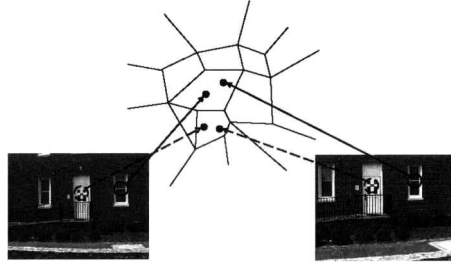
### 10.6.1. Introducción

El reconocimiento de ubicación (o reconocimiento de lugar) describe la capacidad de nombrar los lugares discretos en el mundo. El proceso de reconocimiento de ubicación, funciona mediante el cálculo de una representación a partir de las mediciones de sensor actuales del robot y la búsqueda de la base de datos para la representación más similar almacenada. La representación recuperada luego nos dice la localización del robot.

### 10.6.2. De la bolsa de características a las palabras visuales

Una representación de una imagen por sólo un conjunto de puntos de interés se llama normalmente una bolsa de características. Para cada interés, un descriptor es computado en una forma que es invariante al cambio de rotación, escala, intensidad, y punto de vista (sección 10.5.4). Una manera popular es el uso de histogramas de gradientes, por ejemplo, SIFT o SURF. Este conjunto de descriptores que es la nueva representación de la imagen.

Figura 10.79: Partición del espacio descriptor de características: Cada celda representa una palabra visual



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

Se llama bolsa de características porque la relación espacial original entre los puntos de interés se elimina y sólo los descriptores se recuerdan. La similitud entre dos conjuntos de descriptores puede calcularse contando el número de descriptores de características comunes. Para ello, es necesario definir una función de coincidencia, lo que nos permite determinar si dos descriptores de características son las mismas. Esta función de coincidencia por lo general depende del tipo de descriptor de entidad. Pero, en general, un descriptor de función es un vector de alta dimensión, y características de coincidencia se pueden encontrar mediante el cálculo de la distancia utilizando la norma  $L_2$ . Las palabras visuales son una representación de 1 dimensión del descriptor de característica de alta dimensión. Esto significa que la palabra visual para un descriptor SIFT de 128 dimensiones es sólo un número entero único. La conversión de las palabras visuales crea una bolsa de palabras visuales en lugar de una bolsa de características. Para esta conversión, el espacio descriptor de alta dimensional está dividido en celdas que no se superponen. Esta división se calcula por la agrupación de “k-medias” (K-means Clustering). Para la agrupación, un gran número de descriptores de característica es necesario. El grupo de fronteras calculadas forman las divisiones de la célula del espacio de características. Cada una de las células ahora se le asigna un número que se asigna a cualquier descriptor de función dentro de la celda. Este número se refiere como palabra visual. Descriptores similares característica se clasifican después en la misma célula y por lo tanto, obtendrán la misma palabra visual asignada. Esto se ilustra en la figura 10.79, el cual es un método muy eficaz para encontrar descriptores de coincidencia de características. Las palabras visuales creadas por la división se denominan vocabulario visual.

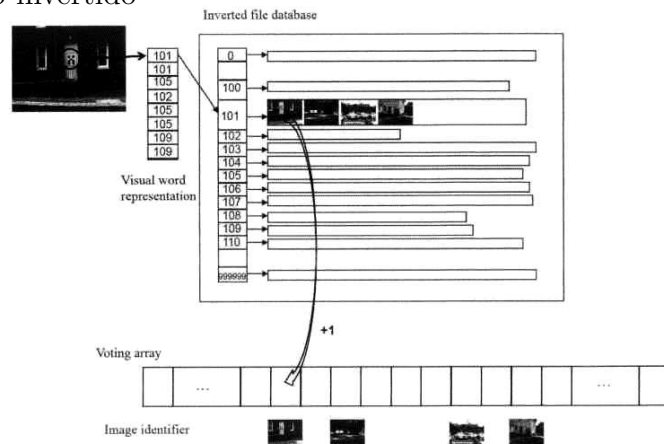
Para la cuantificación, se almacena un vector prototipo para cada celda, que es el

vector descriptor medio de todos los descriptores de formación de la célula.

### 10.6.3. Reconocimiento de ubicación eficiente usando un archivo invertido

La cuantificación de funciones con palabras visuales es un ingrediente clave para el reconocimiento de localización eficiente. Otro es el uso de un archivo invertido para la base de datos y un esquema de votación para los cálculos de similitud. La base de datos organizada como un archivo invertido consiste en una lista de todas las palabras visuales posibles. Cada elemento de esta lista apunta a otra lista que contiene todos los identificadores de imágenes en las cuales esta palabra visual particular apareció. Esto se ilustra en la figura 10.80.

Figura 10.80: Reconocimiento de ubicación basado en palabras visuales mediante un sistema de archivo invertido



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

Este algoritmo no sólo da la imagen más similar en la base de datos sino que también crea una clasificación de todas las imágenes en la base de datos por similitud sin ningún coste computacional adicional. Esto se puede utilizar para darle robustez al reconocimiento de lugar.

#### **10.6.4. Verificación geométrica para el reconocimiento de lugar robusto**

El conjunto de palabras visuales no contiene más relaciones espaciales, por lo tanto una imagen que tenga las mismas palabras visuales pero en una disposición espacial diferente también tienen una alta similitud. Las relaciones espaciales sin embargo pueden ser ejecutadas de nuevo por una verificación geométrica final. Para esto, las imágenes  $k$  más similares en una búsqueda se analizan para determinar la consistencia geométrica. La prueba de consistencia geométrica calcula transformaciones geométricas utilizando las coordenadas  $x$  y  $y$  de la imagen de palabras visuales coincidentes. Las transformaciones utilizadas son transformaciones afines, homografías, o la matriz esencial entre las imágenes.

El cálculo se realiza de una manera robusta mediante RANSAC, y el Número de Inliers a la transformación se cuenta. La imagen que logra el mayor número de inliers con la imagen de búsqueda se reporta entonces como la coincidencia final. Esto devuelve la ubicación deseada en el lugar de reconocimiento.

#### **10.6.5. Otras representaciones de imágenes para el reconocimiento lugar**

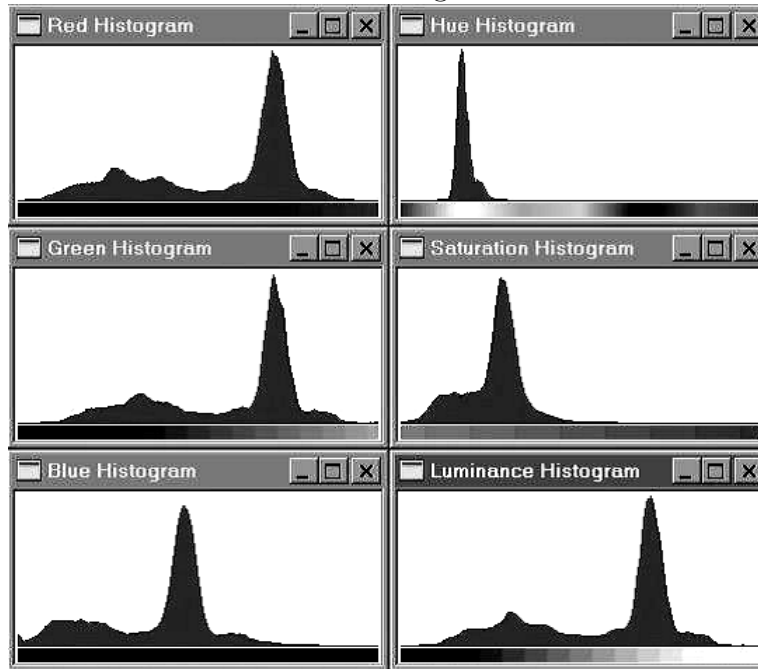
Se revisan dos enfoques. El primer método utiliza histogramas de las imágenes, mientras que el segundo usa las huellas digitales de imagen.

##### **Histogramas de imagen.**

Una imagen visual única proporciona información tanto sobre los alrededores inmediatos de un robot que una alternativa a la búsqueda de la imagen para las características localizadas espacialmente es hacer uso de la información capturada por toda la imagen para extraer una característica de imagen completa o característica de imagen global.

Las características de imagen entera no están diseñadas para identificar determinadas estructuras espaciales, como los obstáculos o la posición de los lugares específicos. Por el contrario, sirven como representaciones compactas de la región local entera. En otras palabras, pequeños cambios en la posición del robot deberían causar sólo pequeños cambios en las características de imágenes completa, mientras que grandes cambios en la posición del robot deberían causar correspondientemente cambios grandes a las características de imagen completa. Un primer paso lógico en el diseño de un sensor de visión para este propósito es el de maximizar el campo de visión de la cámara. Un sistema de

Figura 10.81: Seis histogramas de ID de la imagen de arriba. A 5 x 5 filtro de suavizado se convolucionna con cada banda antes de histografía.



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

cámara catadióptrica, hoy en día muy popular en robótica móvil, ofrece un campo de visión extremadamente amplio (sección 10.2.3).

Una imagen catadióptrica es una imagen de 360 grados deformada sobre una superficie de imagen 2D. Debido a esto, se ofrece otra ventaja crítica en términos de sensibilidad al movimiento del robot de pequeña escala. Si la cámara está montada verticalmente en el robot de modo que la imagen representa el entorno circundante del robot (es decir, su horizonte; figura 10.33a), entonces la rotación de la cámara y el robot simplemente resulta en rotación de la imagen. En resumen, la cámara catadióptrica puede ser invariante a la rotación del campo de visión. Por supuesto, la rotación del robot móvil todavía va a cambiar la imagen, es decir, las posiciones de pixel cambiarán, pero la nueva imagen será simplemente una rotación de la imagen original. Pero tenemos la intención de extraer características de la imagen a través de histografía.

Una imagen de salida de una cámara de color generalmente contiene información útil a lo largo de varias bandas: los valores  $r$ ,  $g$ , y  $b$ , así como los valores de tono, saturación y luminancia. La estrategia más simple de extracción de histograma es la construcción

de diferentes histogramas 1D que caracterizan a cada banda. Dada una imagen de la cámara de color,  $G$ , el primer paso es crear mapeos a partir de  $G$  a cada una de las  $n$  bandas disponibles. Usamos  $G_i$ , para referirse a una matriz que almacena los valores en la banda  $i$  para todos los píxeles de  $G$ . Cada histograma de banda específica  $H_i$  se calcula como antes:

- Como preprocesamiento, se suaviza  $G_i$ , mediante un operador de suavizado de Gauss.
- Inicializar  $H_i$  con  $n$  niveles:  $H[j] = 0$  para  $j = 1, \dots, n$ .
- Por cada pixel  $(x, y)$  en  $G_i$ , incrementar el histograma:  $H_i[G_i[x, y]] + 1$ .

Teniendo en cuenta la imagen que se muestra en la figura 10.33a, la técnica de histograma de imagen extrae seis histogramas (para cada  $r$ ,  $g$ ,  $b$ , matiz, saturación y luminancia) como se muestra en la figura 10.81. Con el fin de hacer uso de histogramas como características de imagen completa, se necesitan medios para comparar a los histogramas para cuantificar la probabilidad de que los histogramas mapeen en las posiciones cercanas del robot. El problema de definir una métrica de distancia útil de histograma es en sí mismo un subcampo importante dentro del campo de recuperación de imágenes. Una de las más exitosas métricas de distancia encontrada en localización del robot móvil es la divergencia Jeffrey. Dadas dos histogramas  $H$  y  $K$ , con  $h_i$  y  $k_i$  denotando las entradas de histograma, la divergencia Jeffrey  $d(H, K)$  se define como

$$d(H, K) = \sum_i \left( h_i \log \frac{2h_i}{h_i + k_i} + k_i \log \frac{2k_i}{h_i + k_i} \right) \quad (10.112)$$

Usando este método de extracción de imagen completa, un robot puede recuperar fácilmente el pasillo particular o habitación particular en la que se encuentra.

### Huellas digitales de imagen.

Este método es similar al enfoque de la palabra visual con la diferencia de que aquí las características no son puntos de interés, sino más bien características morfológicas, líneas y manchas de color. Aunque superados por los nuevos métodos de reconocimiento de lugar basados en palabra visual, este enfoque está siendo muy utilizado en muchas aplicaciones de la robótica móvil, tanto en interiores como en exteriores.

Se describe una aplicación particular de este método, llamado huella digital de la imagen, que se desarrolló primero en (*Deriving and matching image fingerprint sequences for mobile robot localization, 2001*). Tal como el método anterior, el sistema hace uso

Figura 10.82: Dos imágenes panorámicas y sus secuencias de huellas digitales asociadas.



(Fuente: Deriving and matching image fingerprint sequences for mobile robot localization, Roland Siegwart, Illah Nourbakhsh, IEEE, 2001.)

de una imagen panorámica de 360-grados. El primer nivel de extracción busca en la imagen panorámica características espacialmente localizadas: bordes verticales y dieciséis tonalidades de color diferenciadas. El detector de borde vertical es un enfoque de gradiente simple aplicando un operador de diferencia horizontal. Los bordes verticales se “votación” por cada píxel de borde como en una transformada de Hough de borde vertical. Un umbral adaptativo se utiliza para reducir el número de bordes. Supongamos la conteos de la tabla de Hough para cada línea vertical tienen un candidato  $\mu$  media y una desviación estándar  $\sigma$  una. El umbral elegido es simplemente  $\mu + \sigma$ .

Las bandas verticales de color se identifican en gran medida de la misma manera, la identificación de las estadísticas sobre la ocurrencia de cada color, a continuación, se filtran todos los parches de color candidatos excepto aquellos más grandes que  $\mu + \sigma$ . La figura 10.82 muestra dos ejemplos de imágenes panorámicas y sus huellas digitales asociadas.

Los algoritmos de correspondencia de cadenas son otro gran campo de estudio, en particular con interesantes aplicaciones actuales en las áreas de genética. Por ejemplo, la figura 10.83 muestra tres secuencias reales generadas utilizando el algoritmo anterior. La cadena superior debe coincidir con la parte “Place 1”, pero hay eliminaciones e inserciones entre las dos cadenas. La técnica utilizada en el enfoque de huellas para diferenciación de cadena es conocida como un algoritmo de energía mínimo. Tomado de la comunidad de la visión estéreo, este algoritmo de optimización encontrará la energía mínima necesaria para “transformar” una secuencia en otra secuencia. El resultado es una métrica de distancia que es relativamente insensible a la adición o resta individuales de características locales, mientras que todavía es capaz de identificar robustamente la cadena coincidente correcta en una variedad de circunstancias.

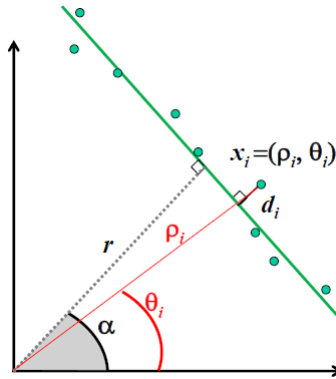


### Ajuste probabilístico de línea a partir de datos de alcance inciertos del sensor.

El objetivo es ajustar una recta a un conjunto de mediciones de sensor, como se muestra en la figura 10.84. Hay incertidumbre asociada con cada una de las mediciones ruidosas del sensor, y así que no hay una sola línea que pase a través del conjunto. En su lugar, se desea seleccionar la mejor coincidencia posible, teniendo en cuenta algún criterio de optimización.

Más formalmente, se supone que  $n$  puntos de medición de alcance en coordenadas pola-

Figura 10.84: Estimación de una línea en el sentido de mínimos cuadrados. Los parámetros del modelo  $r$  (longitud de la perpendicular) y  $\alpha$  (su ángulo con el eje de abscisas) describen unicamente una línea.



(Fuente: Deriving and matching image fingerprint sequences for mobile robot localization, Roland Siegwart, Illah Nourbakhsh, IEEE, 2001.)

res  $x_i = (\rho_i, \theta_i)$  son producidos por los sensores del robot. Hay incertidumbre asociados con cada medición, por lo que podemos modelar cada medida utilizando dos variables aleatorias  $X_i = (P_i, Q_i)$ . En este análisis se supone que la incertidumbre con respecto al valor real de  $P$  y  $Q$  es independiente. Se puede afirmar que:

$$E[P_i \cdot P_j] = E[P_i]E[P_j] \quad \forall i, j = 1, \dots, n \mid i \neq j \quad (10.113)$$

$$E[Q_i \cdot Q_j] = E[Q_i]E[Q_j] \quad \forall i, j = 1, \dots, n \mid i \neq j \quad (10.114)$$

$$E[P_i \cdot Q_j] = E[P_i]E[Q_j] \quad \forall i, j = 1, \dots, n \quad (10.115)$$

Además, se supone que cada variable aleatoria está sujeta a una curva de densidad de probabilidad gaussiana con una media en el valor verdadero y con alguna diferencia

especificada:

$$P_i = -N(\rho_i, \sigma_{\rho_i}^2) \quad (10.116)$$

$$Q_i = -N(\theta_i, \sigma_{\theta_i}^2) \quad (10.117)$$

Dado algún punto de medición  $(\rho, \theta)$ , se puede calcular las coordenadas euclidianas correspondientes como  $x = \rho \cos \theta$  y  $y = \rho \sin \theta$ . Si no hay error, nos gustaría encontrar una línea para la que todas las mediciones se encuentran en esa línea:

$$\rho \cos \theta \cos \alpha + \rho \sin \theta \sin \alpha - r = \rho \cos(\theta - \alpha) - r = 0 \quad (10.118)$$

Por supuesto que hay un error de medición, por lo que esta cantidad no será cero. Cuando no es cero, esto es una medida del error entre el punto de medición  $(\rho, \theta)$  y la línea, específicamente en términos de la distancia mínima ortogonal entre el punto y la línea. Para cada específico  $(\rho_i, \theta_i)$  se puede escribir la distancia ortogonal  $d_i$  entre  $(\rho_i, \theta_i)$  y la línea como:

$$\rho \cos(\theta_i - \alpha) - r = d_i \quad (10.119)$$

Si se considera cada medición, para ser igualmente incierta, podemos resumir el cuadrado de todos los errores juntos, para todos los puntos de medición, para cuantificar un ajuste global entre la línea y todas las mediciones:

$$S = \sum_i d_i^2 = \sum_i (\rho_i \cos(\theta_i - \alpha) - r)^2 \quad (10.120)$$

El objetivo es minimizar  $S$  al seleccionar los parámetros de la línea  $(\alpha, r)$ . Podemos hacerlo mediante la resolución del sistema de ecuaciones no lineales

$$\frac{\partial S}{\partial \alpha} = 0 \quad \frac{\partial S}{\partial r} = 0 \quad (10.121)$$

Para hacer uso de la varianza  $\sigma_i^2$  que modela la incertidumbre con respecto a la distancia  $\rho_i$  de una medición del sensor particular, se calcula un peso  $w_i$  individual para cada medición según la fórmula:

$$w_i = \frac{1}{\sigma_i^2} \quad (10.122)$$

Entonces, la ecuación 10.120 se convierte en:

$$S = \sum w_i d_i^2 = \sum w_i (\rho_i \cos(\theta_i - \alpha) - r)^2 \quad (10.123)$$

Se puede demostrar que la solución de la ecuación 10.121 en el sentido de mínimos cuadrados ponderado es

$$\alpha = \frac{1}{2} \text{atan} \left( \frac{\sum w_i \rho_i^2 \sin 2\theta_i - \frac{2}{\sum w_i} \sum \sum w_i w_j \rho_i \rho_j \cos \theta_i \sin \theta_j}{\sum w_i \rho_i^2 \sin 2\theta_i - \frac{1}{\sum w_i} \sum \sum w_i w_j \rho_i \rho_j \cos(\theta_i + \theta_j)} \right) \quad (10.124)$$

$$r = \frac{\sum w_i \rho_i \cos(\theta_i + \alpha)}{\sum w_i} \quad (10.125)$$

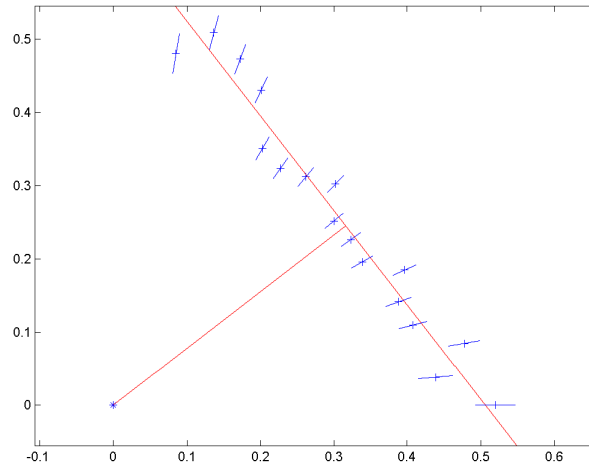
En la práctica, la ecuación 10.124 utiliza el arco tangente de cuatro cuadrantes ( $\text{atan2}$ ). Vamos a demostrar las ecuaciones 10.124 y 10.125, con un ejemplo concreto. Las diecisiete medidas  $(\rho_i, \theta_i)$  en la tabla 10.3 se han tomado con un sensor de láser instalado en un robot móvil. Las mediciones se muestran en la figura 10.85. La aplicación directa de estas ecuaciones genera la solución de la línea definida por  $\alpha = 37,36$  y  $r = 0,4$ . Esta línea representa el mejor ajuste en un sentido de mínimos cuadrados y se muestra visualmente en la figura 10.85.

Cuadro 10.3: Valores de Medición

<b>ÁNGULO DE PUNTERIA DEL SENSOR <math>\theta_i</math> (Grad)</b>	<b>Rango <math>\rho_i</math> [M]</b>
0	0.5
5	0.4
10	0.4
15	0.4
20	0.4
25	0.4
30	0.3
35	0.3
40	0.3
45	0.4
50	0.4
55	0.3
60	0.4
65	0.4
70	0.5
75	0.5
80	0.4

Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.

Figura 10.85: Línea extraída a partir de mediciones de distancias por láser (+)



(Fuente: Deriving and matching image fingerprint sequences for mobile robot localization, Roland Siegwart, Illah Nourbakhsh, IEEE, 2001.)

### Propagación de la incertidumbre durante la conexión de la línea.

Se quiere entender cómo las incertidumbres de las mediciones específicas del sensor de alcance se propagan para gobernar la incertidumbre de la línea de extraída. En otras palabras, ¿cómo la incertidumbre en  $\rho_i$  y  $\theta_i$  se propaga en las ecuaciones 10.124 y 10.125 para afectar a la incertidumbre de  $\alpha$  y  $r$ ?  $A$  y  $R$  que representan las variables de salida aleatorias de  $\alpha$  y  $r$ , respectivamente. El objetivo es obtener la matriz  $2 \times 2$  de covarianza de salida:

$$C_{AR} = \begin{bmatrix} \sigma_A^2 & \sigma_{AR} \\ \sigma_{AR} & \sigma_A^2 \end{bmatrix} \quad (10.126)$$

Dada la matriz de covarianza de entrada  $2n \times 2n$

$$C_X = \begin{bmatrix} C_P & 0 \\ 0 & C_Q \end{bmatrix} = \begin{bmatrix} \text{diag}(\sigma_{\rho_i}^2) & 0 \\ 0 & \text{diag}(\sigma_{\theta_i}^2) \end{bmatrix} \quad (10.127)$$

Y las relaciones del sistema (ecuaciones 10.124 y 10.125). Entonces, mediante el cálculo de la Jacobiana,

$$F_{PQ} = \begin{bmatrix} \frac{\partial \alpha}{\partial P_1} & \frac{\partial \alpha}{\partial P_1} & \cdots & \frac{\partial \alpha}{\partial P_n} & \frac{\partial \alpha}{\partial Q_1} & \frac{\partial \alpha}{\partial Q_2} & \cdots & \frac{\partial \alpha}{\partial Q_n} \\ \frac{\partial r}{\partial P_1} & \frac{\partial r}{\partial P_2} & \cdots & \frac{\partial r}{\partial P_n} & \frac{\partial r}{\partial Q_1} & \frac{\partial r}{\partial Q_2} & \cdots & \frac{\partial r}{\partial Q_n} \end{bmatrix} \quad (10.128)$$

Podemos instanciar la ecuación de propagación incertidumbre para generar  $C_{AR}$ :

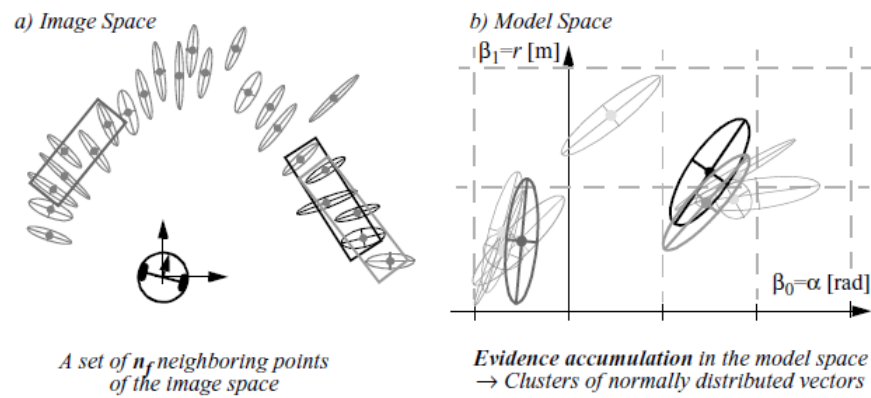
$$C_{AR} = F_{PQ} C_x F_{PQ}^T \quad (10.129)$$

Así, se ha calculado la probabilidad de  $C_{AR}$  de la línea extraída  $(\alpha, r)$  sobre la base de las probabilidades de los puntos de medición.

### 10.7.2. Seis algoritmos de extracción de línea

La sección anterior describe cómo extraer una entidad lineal dado un conjunto de mediciones de distancia. Desafortunadamente, el proceso de extracción de características es significativamente más complejo que eso. Un robot móvil en efecto, adquiere un conjunto de mediciones de alcance, pero, en general, las mediciones de alcance no son parte de una sola línea. Más bien, sólo algunas de las mediciones de alcance deberían desempeñar un papel en la línea de extracción y, además, puede haber más de una característica de línea representado en el conjunto de medición. Este escenario más realista se muestra en la figura 10.86.

Figura 10.86: Clustering: encontrar segmentos vecinos de una línea común.



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

El proceso de dividir un conjunto de mediciones en subconjuntos que puedan ser interpretados uno por uno se denomina segmentación y es el paso más importante de la extracción de la línea. A continuación, se describen seis algoritmos populares de extracción de línea (segmentación).

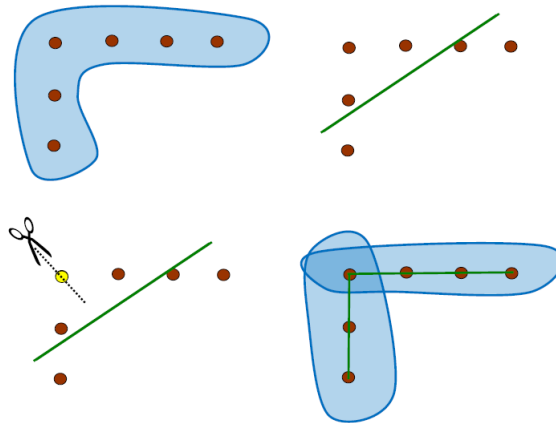
### Algoritmo 1: División y fusión (Split-and-Merge).

Split-and-Merge es el algoritmo de extracción de línea más popular. Este algoritmo tiene su origen en la visión por computador y se ha estudiado y utilizado en muchos trabajos. El algoritmo se describe en el algoritmo 1 y su procedimiento se ilustra en la figura 10.87. Finalmente, una aplicación de división y fusión a un escaneo láser 2D se muestra en la figura 10.88.

#### Algoritmo 1: Division y fusión.

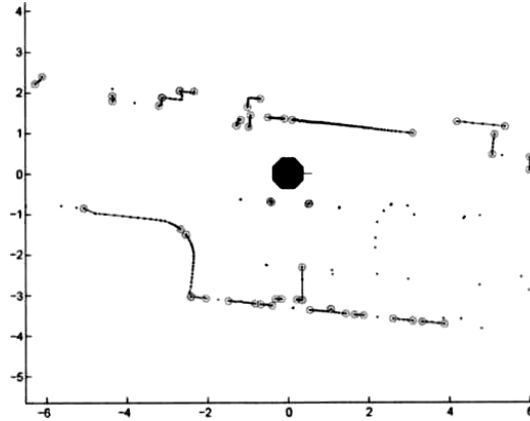
1. Inicial: El conjunto  $s$ , consta de  $N$  puntos. Dejar  $S_1$ , en una lista  $L$
2. Ajustar una línea al siguiente conjunto en  $L$
3. Detectar el punto  $P$  con la máxima distancia  $d_p$  a la línea
4. Si  $d_p$  es inferior a un umbral, continua (ver paso 2)
5. De lo contrario, dividir  $S_1$  en  $P$  en  $S_{i1}$  y  $S_{i2}$ , reemplace  $S_i$  en  $L$  por  $S_{i1}$  y  $S_{i2}$ , continua (ir a 2)
6. Cuando todos los grupos (segmentos) en  $L$  han sido verificados, fusionar segmentos colineales.

Figura 10.87: División y fusión implementada de la manera Iterativa ajuste-punto-extremo



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

Figura 10.88: División y fusión aplicado a una tomografía láser 2D (cortesía de B Jensen).



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

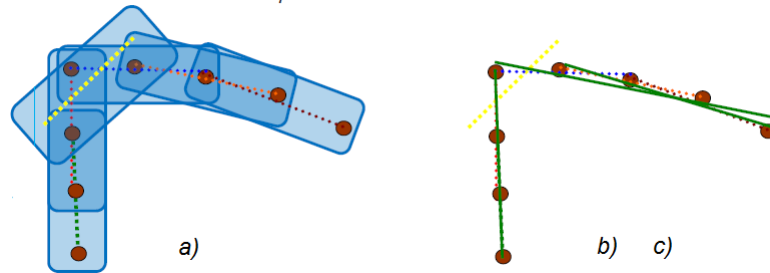
### Algoritmo 2: Regresión de línea.

Este utiliza una ventana deslizante de tamaño  $N_f$ . En cada paso, una línea se ajusta a los puntos  $N_f$  dentro de la ventana. La ventana se desplaza un punto adelante (por eso se llama ventana deslizante), y la operación de la línea de ajuste se repite de nuevo. Este algoritmo se describe en el algoritmo 2, mientras que los pasos principales se representan en la figura 10.89, donde (a) es una línea se ajusta a cada conjunto de tres puntos consecutivos, (b) segmentos similares consecutivos se fusionan entonces, y (c) es el conjunto de todos los segmentos a continuación se comprueba de nuevo, y los segmentos similares consecutivos restantes se fusionan con los generados en el paso anterior.

### Algoritmo 2: Regresión de línea.

1. Inicializar el tamaño ventana deslizante  $N_f$
2. Ajustar una línea por cada  $N_f$  puntos consecutivos
3. Calcule una amplia línea de fidelidad. Cada elemento de la matriz contiene la suma de las distancias de Mahalanobis entre cada tres ventanas adyacentes.
4. Construir segmentos de línea mediante el escaneo de la matriz de fidelidad para elementos consecutivos que tienen valores inferiores a un umbral

Figura 10.89: Una ventana deslizante con Tamaño  $N_f = 3$  se utiliza en este ejemplo



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition. )

5. Combinar segmentos de línea superpuestos y volver a calcular los parámetros de línea para cada segmento.

### Algoritmo 3: Incremental.

Este algoritmo es sencillo de implementar y se ha utilizado en muchas aplicaciones. El algoritmo es explicado en el algoritmo 3.

#### Algoritmo 3: Incremental.

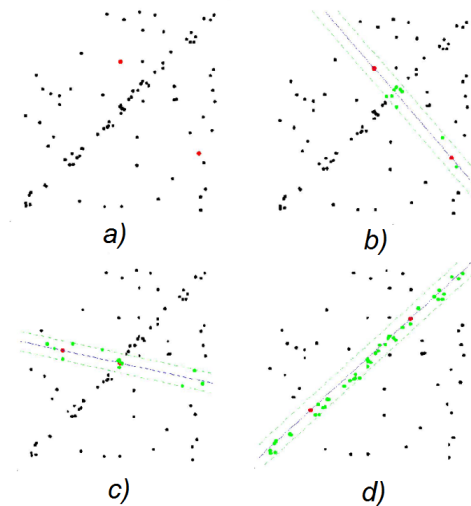
1. Empezar por las 2 primeros puntos, construcción de una línea
2. Añadir el siguiente punto al modelo de línea actual
3. Recalcular los parámetros de la línea mediante ajuste de línea
4. Si se satisface la condición de la línea, continuar (ver paso 2)
5. En caso contrario, volver a poner el último punto, volver a calcular los parámetros de la línea, retornar la línea
6. Continúe con los siguientes dos puntos, vaya al paso 2

### Algoritmo 4: RANSAC.

RANSAC (Random Sample Consensus), Consenso de muestra aleatorio, es un algoritmo para estimar la robustez entre los parámetros de un modelo a partir de unos datos dados en la presencia de valores atípicos (outliers). Los valores atípicos (outliers)

son datos que no encajan en el modelo. Tales valores atípicos pueden ser debido al alto ruido en los datos, mediciones erróneas, o pueden ser simplemente puntos que provienen de otros objetos para los que nuestro modelo matemático no se aplica. RANSAC es un método iterativo y es no determinista en el que la probabilidad de encontrar una línea libre de valores atípicos aumenta a medida que se utilizan más iteraciones.

Figura 10.90: Principio de funcionamiento de RANSAC



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

El algoritmo se describe en el algoritmo 4, mientras que la figura 10.90 ilustra su principio de funcionamiento: (a) Conjunto de datos de  $N$  puntos, (b) Dos puntos son seleccionados al azar, una línea se ajusta a través de ellos, y los puntos dentro de una distancia predefinida a esta se identifican, (c) El procedimiento se repetido (iterado) varias veces, (d) El conjunto con el número máximo de inliers es elegido como una solución al problema.

#### Algoritmo 4: RANSAC

1. Inicial: sea  $A$  un conjunto de  $N$  puntos
2. Repetir
3. Seleccione al azar una muestra de 2 puntos de  $A$
4. Coloque una línea a través de los 2 puntos

5. Calcule las distancias de todos los demás puntos de esta línea
6. Construya el conjunto inlier (es decir, contar el número de puntos con distancia a la línea  $< d$ )
7. Guarde estos inliers
8. Hasta alcanzar el número máximo de iteraciones  $k$
9. El conjunto con el número máximo de inliers es elegido como una solución al problema

Sea  $p$  la probabilidad de encontrar un conjunto de puntos libres de valores atípicos. Sea  $w$  la probabilidad de seleccionar un inlier de nuestra base de datos de  $N$  puntos. Por lo tanto,  $w$  expresa la fracción de inliers en los datos, es decir,  $w = \text{número de inliers} / N$ . Si asumimos que los dos puntos necesarios para la estimación de una línea se seleccionan de forma independiente,  $w^2$  es la probabilidad de que ambos puntos sean inliers y  $1 - w^2$  es la probabilidad de que al menos uno de estos dos puntos sea un valor atípico. Ahora, sea  $k$  el número de iteraciones de RANSAC ejecutadas hasta ahora, entonces  $(1 - w^2)^k$  será la probabilidad que RANSAC nunca seleccione dos puntos que sean ambos inliers. Esta probabilidad debe ser igual a  $1 - p$ . Por consiguiente,

$$1 - p = (1 - w^2)^k \quad (10.130)$$

Y por lo tanto

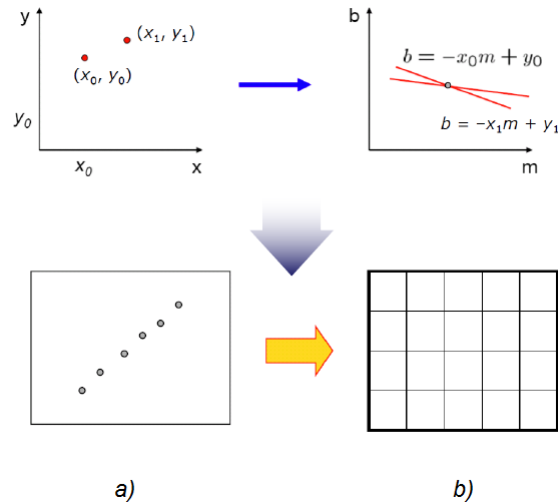
$$k = \frac{\log(1 - p)}{\log(1 - w^2)} \quad (10.131)$$

La principal ventaja de RANSAC es que es un método de extracción genérico y se puede utilizar con muchos tipos de funciones una vez que tenemos el modelo de características.

**Algoritmo 5: Transformada de Hough.**

1. Inicial: sea  $A$  un conjunto de  $N$  puntos
2. Inicializar la matriz acumulador estableciendo todos los elementos a 0
3. Calcular los valores para la matriz
4. Elegir el elemento con más votos  $V_{max}$
5. Si  $V_{max}$  es menor que un umbral, terminar

Figura 10.91: Transformada de Hough: a) Espacio de imagen. b) Parámetro de espacio de Hough.



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

6. De lo contrario, determinar los inliers
7. Colocar una línea a través de los inliers y almacenar la línea
8. Retire los inliers del conjunto, vaya al paso 2

#### Algoritmo 5: Transformada de Hough (HT).

Este algoritmo se ha descrito ya para la detección de borde recto en imágenes de intensidad, pero puede ser aplicado sin ninguna modificación para imágenes de rango 2D. El algoritmo se describe en el algoritmo 5. En las figuras 10.91 y 10.92 se puede encontrar la explicación del funcionamiento del algoritmo, y una aplicación básica.

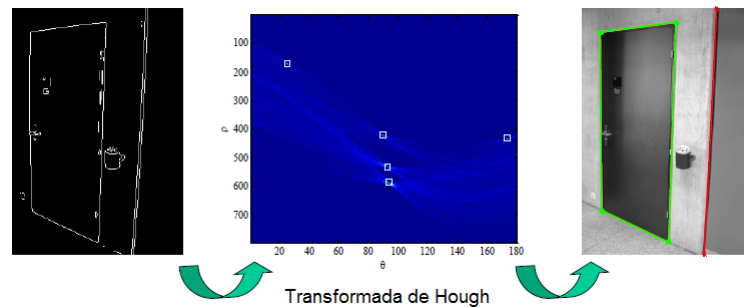
#### Algoritmo 6: Maximización de Expectativa (EM).

Maximización de expectativa (EM), es un método probabilístico comúnmente utilizado en falta de problemas variables. El algoritmo se describe en el algoritmo 6.

#### Algoritmo 6: Maximización de expectativa.

1. Inicial: sea A un conjunto de N puntos

Figura 10.92: Detección de puerta usando la transformada de Hough.



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

2. **repetir**
3. Generar aleatoriamente parámetros para una línea
4. Inicializar pesos para los puntos restantes
5. **repetir**
6. Paso-E: Calcular los pesos de los puntos del modelo de línea
7. Paso-M: Recalcular los parámetros del modelo de línea
8. **hasta** el número máximo de pasos alcanzados o la convergencia
9. **hasta** el número máximo de intentos alcanzados o encontrar una línea
10. Si la encuentra, almacenar la línea, quite los inliers, vaya al paso 2
11. De lo contrario, terminar

### Los detalles de aplicación.

**El agrupamiento (Clustering).** En la mayoría de los casos, los escáneres láser 2D presentan algunas aglomeraciones de algunos puntos dispersos (Figura 10.88). Básicamente, este algoritmo escanea grandes saltos en diferencias radiales de puntos consecutivos y pone puntos de interrupción en esas posiciones. Como resultado, el escaneo es segmentado en grupos contiguos de puntos.

**Fusión.** Debido a las oclusiones, una línea puede ser observada y extraída como varios segmentos. Cuando esto sucede, es probable fusionar segmentos colineales de

línea en un solo segmento de línea. Esta rutina de fusión debe ser aplicada en el extremo de salida de cada algoritmo visto anteriormente, después de que los segmentos han sido extraídos.

### Una comparación de los algoritmos de extracción de línea.

Estos seis algoritmos se pueden dividir en dos categorías: métodos deterministas y no deterministas:

1. Determinista: División y fusión (Split-and-Merge), Incremental, Regresión, transformada de Hough.
2. No deterministas: RANSAC, EM.

Cuadro 10.4: Comparación de los algoritmos para la extracción de la línea de datos de láser 2D.

	Complejidad	Velocidad [Hz]	Falsos positivos	Precisión
<b>Split-and-Merge</b>	$N \cdot \log N$	1500	10%	+++
<b>Incremental</b>	$S \cdot N^2$	600	6%	+++
<b>Regresión de Línea</b>	$N \cdot N_f$	400	10%	+++
<b>RANSAC</b>	$S \cdot N \cdot N_{trials}$	30	30%	++++
<b>Transformada de Hough</b>	$S \cdot N \cdot N_C + S \cdot N_R \cdot N_C$	10	30%	++++
<b>Maximización de Expectación</b>	$S \cdot N_1 \cdot N_2 \cdot N$	1	50%	++++

Fuente: Autores del proyecto

Una comparación entre los seis algoritmos que se ha hecho por Nguyen et al. Se evaluaron cuatro medidas de calidad: la complejidad, la velocidad, la exactitud (falsos positivos), y precisión. Los resultados de este estudio se muestran en la tabla 10.4. La terminología utilizada se explica de la siguiente manera:

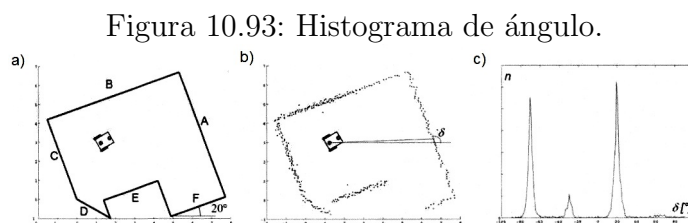
- $N$ : número de puntos en el escaneo de entrada (por ejemplo, 722)
- $S$ : Número de segmentos de línea extraídos (por ejemplo, 7 en promedio dependiendo del algoritmo)

- $N_f$ : Ventana corredera tamaño de la Línea-Regresión (por ejemplo, 9)
- $N_{trials}$ : Número de pruebas para RANSAC (por ejemplo, 1000)
- $N_C, N_R$ : Número de columnas, filas, respectivamente, para la matriz acumulador de Hough ( $N_C = 401, N_R = 671$  para resoluciones de 1 *cm* y grados 0.9)
- $N_1, N_2$ : Número de pruebas e iteraciones de convergencia, respectivamente, para EM (por ejemplo,  $N_1 = 50, N_2 = 200$ ).

Como se muestra en la tercera columna (velocidad) de la tabla 10.4, división y fusión, incremental y la línea de regresión, funcionan mucho más rápido que los demás. El algoritmo incremental parece funcionar mejor en términos de corrección. Por el contrario, RANSAC, HT y EM parecen producir muchos más falsos positivos. A pesar de su mala exactitud, como se observa en la cuarta columna de la tabla 10.4, RANSAC, HT, y EM producen más líneas precisas que los otros algoritmos. En conclusión, la división y fusión e incremental son la mejor opción en términos de exactitud y eficiencia y, por tanto, los mejores candidatos para la localización del robot basado en láser 2D y mapeo. Sin embargo, la elección correcta depende en gran medida del tipo de aplicación y la precisión deseada.

### 10.7.3. Características de histograma de alcance

Un histograma es una forma sencilla de combinar los elementos característicos de una imagen. Un histograma de ángulo, tal como se presenta en la figura 10.93, grafica las estadísticas de líneas extraídas por dos mediciones de alcance adyacentes. En primer



(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

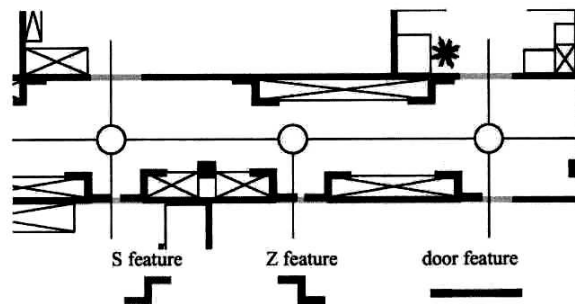
lugar, una exploración de 360 grados de la habitación se toma con el escáner de alcance, y los “golpes” resultantes se registran en un mapa. A continuación, el algoritmo mide

el ángulo relativo entre dos golpes adyacentes (véase la figura 10.93b). Después de la compensación de ruido en las lecturas (causada por las imprecisiones en posición entre golpes adyacentes), se puede construir el histograma de ángulo que se muestra en la figura 10.93c.

#### 10.7.4. Extracción de otras características geométricas

Las funciones de línea son de particular valor para los robots móviles que operan en ambientes artificiales, en donde, por ejemplo, la construcción de muros y paredes de los pasillos suelen ser recta. En general, un robot móvil hace uso de funciones múltiples simultáneamente, que comprende un conjunto de características que es más apropiado para su entorno operativo. Para robots móviles de trabajo en interiores, la función de línea es sin duda un miembro del conjunto de características óptimas.

Figura 10.94: Varias características geométricas en un pasillo único, incluyendo puertas y discontinuidades en la anchura del pasillo.

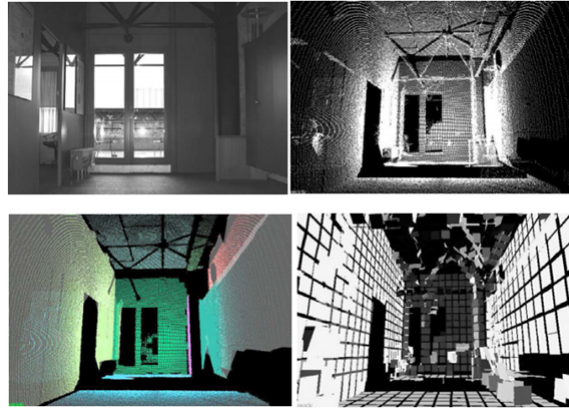


(Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.)

Además, otros núcleos (kernels) geométricas aparecen constantemente en todo el entorno artificial interior. Las características de la esquina se definen como una característica de punto con una orientación. Las discontinuidades de paso, definido como un cambio de paso perpendicular a la dirección de desplazamiento del pasillo, se caracterizan por su forma (convexa o cóncava) y tamaño de paso. Las puertas, definidas como aberturas de dimensiones adecuadas en las paredes, se caracterizan por su anchura. Por lo tanto, el problema de segmentación estándar no es tan simple como decidir sobre un mapeo de lecturas de sensores a segmentos de línea, sino más bien es un proceso en el que las características de los diferentes tipos se extraen basándose en las mediciones de los sensores disponibles. La figura 10.92 muestra un modelo de un entorno de pasillo

interior junto con ambas características de hendidura (es decir, las discontinuidades de paso) y las puertas.

Figura 10.95: Proceso de extracción de características planas



(Fuente: 3D SLAM using planar segments, Roland Siegwart, J. Weingarten, 2006.)

Los diferentes tipos de características pueden proporcionar información cuantitativa diferente para la localización del robot móvil. La característica de línea, por ejemplo, proporciona dos grados de información, el ángulo y la distancia. Pero la característica de paso proporciona información 2D posición relativa, así como el ángulo. Las características del plano son valiosas en entornos creados por el hombre, debido a las paredes lisas, suelos y techos de los ambientes interiores. Así son prometedores como otra característica muy informativa para los robots móviles a utilizar para el mapeo y localización. Algunos experimentos con características planas se han hecho en el ASL (ETH Zurich) (*3D SLAM using planar segments, Roland Siegwart, J. Weingarten, 2006*), el proceso de extracción de característica de plano de se ilustra en la figura 10.95: (Izquierda superior) Fotografía del entorno original. (Superior derecha) Escaneo bruto 3D. (Abajo a la derecha) Segmentación y montaje de característica de plano. (Parte inferior izquierda) Resultado de la segmentación de plano final.

---

---

# CAPÍTULO 11

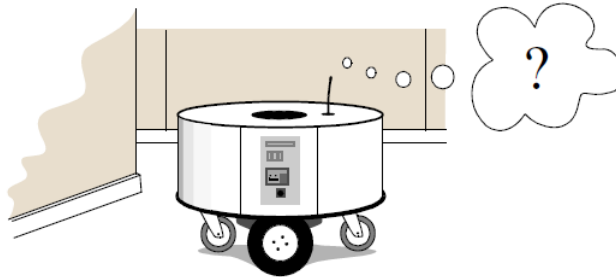
---

## LOCALIZACIÓN EN LOS ROBOTS MÓVILES

### 11.1. GENERALIDADES

El éxito en la navegación requiere el éxito en los cuatro componentes básicos: la percepción, el robot debe interpretar sus sensores para extraer datos significativos, la localización, el robot debe determinar su posición en el medio ambiente (figura 11.1), la cognición, el robot debe decidir cómo actuar para alcanzar sus objetivos, y el control de movimiento, el robot debe modular sus salidas de motores para obtener la trayectoria deseada.

Figura 11.1: ¿Dónde estoy?, determinación de la posición en el medio ambiente.

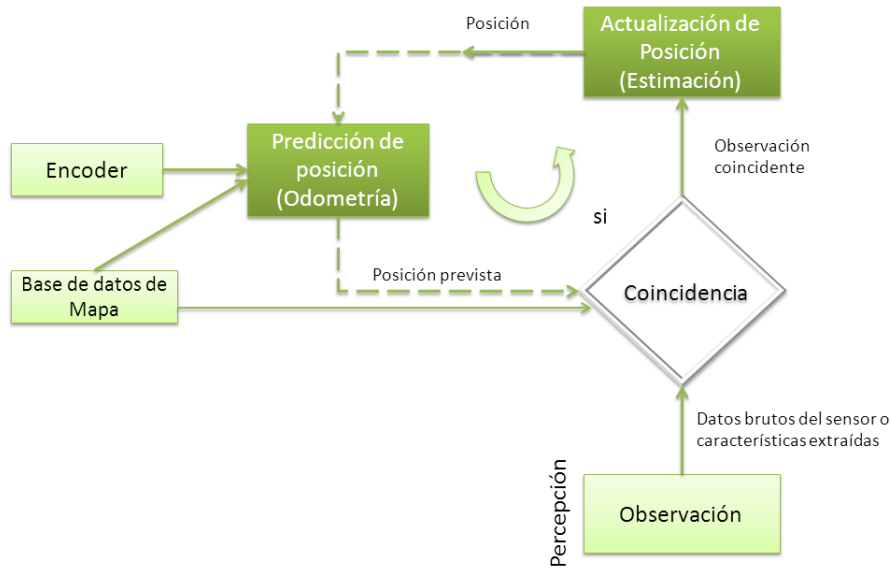


Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition.

De estos cuatro componentes (figura 11.2), la localización ha recibido la atención

mayor investigación en la última década y, como consecuencia, importantes avances se han hecho en este frente.

Figura 11.2: Esquema general para la localización del robot móvil.



Fuente: Autores del proyecto.

## 11.2. EL DESAFÍO DE LA LOCALIZACIÓN: EL RUIDO Y ALIASING

Si se pudiera conectar un sensor GPS preciso (sistema de posicionamiento global) para un robot móvil, gran parte del problema de localización serían obviados. El GPS informaría al robot de su posición exacta, en interiores y al aire libre. Por desgracia, dicho sensor no está en práctica. La actual red de GPS proporciona una precisión de unos pocos metros, lo cual es inaceptable para la localización de robots móviles a escala humana, así como para robots móviles en miniatura tales como robots de escritorio y los nanorobots que navegan el cuerpo del futuro. Además, las tecnologías GPS no pueden funcionar en interiores o en áreas obstruidas y por lo tanto son limitadas en ciertos espacios de trabajo.

La localización significa más que simplemente la determinación de una postura absoluta en el espacio, sino también la construcción de un mapa, y luego, la identificación de

la posición relativa a ese mapa del robot. Claramente, los sensores del robot y efectores juegan un papel integral en todas las formas anteriores de localización.

### **11.2.1. Ruido del sensor**

Los sensores son la entrada de robot fundamental para el proceso de percepción, y por lo tanto el grado en el que los sensores pueden discriminar el estado del mundo es crítico. El ruido del sensor induce una limitación de la consistencia de las lecturas del sensor en el estado del medio ambiente mismo y, por lo tanto, sobre el número de bits útiles disponibles de cada lectura del sensor. A menudo, la fuente de los problemas del ruido del sensor es que algunas de las características ambientales no son capturadas por la representación del robot y en consecuencia se pasan por alto. La dependencia de la iluminación es sólo un ejemplo del aparente ruido en un sistema de sensor basado en visión. Imagen Jitter, ganancia de señal, brillo, y borrosidad, son fuentes adicionales de ruido, reduciendo potencialmente el contenido útil de una imagen de vídeo a color.

El ruido del sensor reduce el contenido de información útil de las lecturas del sensor. Evidentemente, la solución es tomar múltiples lecturas en cuenta, empleando fusión temporal o fusión multisensor para aumentar el contenido de información general de las entradas del robot.

### **11.2.2. Aliasing del sensor**

El problema que se plantea a la navegación a causa del sensor aliasing es que, incluso con sensores sin ruido, la cantidad de información es en general insuficiente para identificar la posición del robot a partir de una lectura de una sola percepción. Así, las técnicas deben ser empleadas por el programador del robot que base la localización del robot en una serie de lecturas y, por lo tanto, información suficiente para recuperar la posición del robot en el tiempo.

### **11.2.3. Ruido del actuador Effector**

Así como sensores del robot son ruidosos, limitando el contenido de información de la señal, así los actuadores de los robots son también ruidosos. En particular, una sola acción tomada por un robot móvil puede tener varios resultados posibles diferentes, a pesar de que desde el punto de vista del robot, el estado inicial que fue asumido antes de la acción sea bien conocido. Los efectores de robots móviles introducen incertidumbre acerca de su futuro estado.

Así, la posición tiene que ser actualizada de vez en cuando por otros mecanismos de localización. Hay muchas fuentes odométricas de error, de los factores ambientales para la resolución:

- Resolución limitada durante la integración (incrementos de tiempo, resolución de medición, etc);
- La desalineación de las ruedas (determinista);
- La incertidumbre en el diámetro de la rueda y de diámetro desigual entre ruedas (determinista);
- La variación en el punto de contacto de la rueda;
- Contacto con el piso desigual (deslizamiento, superficie no plana, etc.)

Algunos de los errores pueden ser determinísticos (sistemáticos), por lo que pueden ser eliminados por la calibración adecuada del sistema. Sin embargo, todavía hay un número de errores no determinísticos (aleatorios) que permanecen, lo que conduce a incertidumbres en la estimación de la posición en el tiempo. Desde un punto de vista geométrico se puede clasificar los errores en tres tipos:

1. Error de rango: longitud del camino integrado (distancia) del movimiento del robot → Suma de los movimientos de las ruedas
2. Error de giro: similar al error de rango, pero por turnos  
→ Diferencia de los movimientos de las ruedas
3. Error de desvío: diferencia en el error de las ruedas que conduce a un error en la orientación angular del robot

Durante largos períodos de tiempo, los errores de giro y desvío son muy superiores a los errores de rango, ya que su contribución al error de posición es en general no lineal.

#### **11.2.4. Un modelo de error de estimación de la posición odométrica**

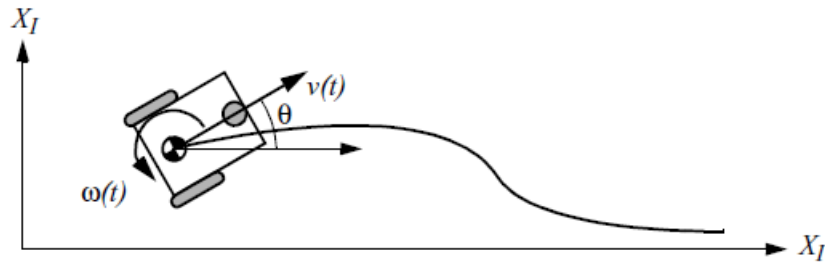
Generalmente la postura (posición) de un robot se representa por el vector

$$p = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad (11.1)$$

Para un robot de accionamiento diferencial la posición se puede estimar a partir de una posición conocida por la integración del movimiento (sumando las distancias de recorrido incrementales). Para un sistema discreto con un intervalo fijo de muestreo  $\Delta t$  los recorridos incrementales  $(\Delta x; \Delta y; \Delta \theta)$  son

$$\Delta x = \Delta s \cos \left( \theta + \frac{\Delta \theta}{2} \right) \quad (11.2)$$

Figura 11.3: El movimiento de un robot diferencial-drive.



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition.

$$\Delta y = \Delta s \sin \left( \theta + \frac{\Delta \theta}{2} \right) \quad (11.3)$$

$$\Delta \theta = \frac{\Delta s_r - \Delta s_l}{b} \quad (11.4)$$

$$\Delta s = \frac{\Delta s_r + \Delta s_l}{2} \quad (11.5)$$

Dónde:

$(\Delta x; \Delta y; \Delta \theta)$  = trayecto recorrido en el último intervalo de muestreo;

$\Delta s_r; \Delta s_l$  = distancias recorridas para la rueda derecha e izquierda respectivamente;

$b$  = distancia entre las dos ruedas del robot diferencial-drive;

Así se obtiene la posición actualizada  $p'$

$$p' = \begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = p \begin{bmatrix} (\Delta s \cos (\theta + \frac{\Delta \theta}{2})) \\ \Delta s \sin (\theta + \frac{\Delta \theta}{2}) \\ \Delta \theta \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \Delta s \cos (\theta + \frac{\Delta \theta}{2}) \\ \Delta s \sin (\theta + \frac{\Delta \theta}{2}) \\ \Delta \theta \end{bmatrix} \quad (11.6)$$

Mediante el uso de la relación de  $(\Delta s, \Delta \theta)$  de las ecuaciones (11.4) y (11.5) también se obtiene la ecuación básica para la actualización de la posición odométrica (para

robots de accionamiento diferencial):

$$p' = (x, y, \theta, \Delta s_r, \Delta s_l) = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \frac{\Delta s_r + \Delta s_l}{2} \cos \left( \frac{\theta + (\Delta s_r - \Delta s_l)}{2b} \right) \\ \frac{\Delta s_r + \Delta s_l}{2} \sin \left( \frac{\theta + (\Delta s_r - \Delta s_l)}{2b} \right) \\ \frac{\theta + (\Delta s_r - \Delta s_l)}{2b} \end{bmatrix} \quad (11.7)$$

Debido a errores de integración de las incertidumbres de  $p$  y los errores de movimiento durante el movimiento incremental  $(\Delta s_r; \Delta s_l)$  el error de posición basado en la integración odometría crece con el tiempo. En el siguiente paso se establece un modelo de error para la posición  $p'$  integrada para obtener la matriz de covarianza  $\sum_{p'}$  de la estimación de la posición odométrica. Para ello, se supone que en el punto de partida inicial de la matriz de covarianza de  $\sum_{p'}$  es conocida. Para el incremento de movimiento  $(\Delta s_r; \Delta s_l)$  se asume la siguiente matriz de covarianza  $\sum_{\Delta}$  :

$$\sum_{\Delta} = covar(\Delta s_r; \Delta s_l) = \begin{bmatrix} k_r | \Delta s_r | & 0 \\ 0 & k_l | \Delta s_l | \end{bmatrix} \quad (11.8)$$

Donde  $\Delta s_r, \Delta s_l$  , son las distancias recorridas por cada rueda, y  $k_r$  y  $k_l$  son constantes de error que representan los parámetros no determinísticos del motor conductor y la interacción entre las ruedas y el piso. Como se puede ver, en la ecuación (11.8) realizamos los siguientes supuestos:

- Los dos errores de las ruedas individualmente impulsadas son independientes;
- La varianza de los errores (ruedas izquierda y derecha) son proporcionales al valor absoluto de las distancias recorridas  $(\Delta s_r; \Delta s_l)$ .

Estos supuestos, aunque no son perfectos, son adecuados y por lo tanto se utilizarán para el desarrollo del modelo de error. Los errores de movimiento son debido al movimiento impreciso debido a la deformación de la rueda, el deslizamiento, suelo desigual, los errores en codificadores (encoders), y así sucesivamente. Los valores para las constantes de error  $k_r$  y  $k_l$  dependen del robot y el entorno y deben ser establecidas experimentalmente mediante la ejecución y análisis de movimientos representativos. Si se supone que  $p$  y  $\Delta r_l = (\Delta s_r; \Delta s_l)$  no están correlacionados y la derivación de la ecuación (11.7) es razonablemente aproximada por la expansión Taylor de primer orden (linealización), se llega a la conclusión, usando la ley de propagación de errores:

$$\sum_{p'} = \nabla_p f \cdot \sum_p \cdot \nabla_p f^T + \nabla_{(\Delta r_l)} f \cdot \sum_{\Delta} \cdot \nabla_{(\Delta r_l)} f^T \quad (11.9)$$

La matriz de covarianza  $\Sigma_p$  es por supuesto, siempre dada por la  $\Sigma_{p'}$  de el paso anterior, y por lo tanto se puede calcular después de especificar un valor inicial (por ejemplo, 0). Usando la ecuación (11.7) podemos usar los dos jacobianos  $F_p = \nabla_p f$  y  $F_{(\Delta_{rl})} = \nabla_{(\Delta_{rl})} f$

$$F_p = \nabla_p f = \nabla_p (f^T) = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} & \frac{\partial f}{\partial \theta} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\Delta s \sin(\theta + \frac{\Delta\theta}{2}) \\ 0 & 1 & \Delta s \cos(\theta + \frac{\Delta\theta}{2}) \\ 0 & 0 & 1 \end{bmatrix} \quad (11.10)$$

$$F_{(\Delta_{rl})} = \begin{bmatrix} \frac{1}{2} \cos(\theta + \frac{\Delta\theta}{2}) - \frac{\Delta s}{2b} \sin(\theta + \frac{\Delta\theta}{2}) & \frac{1}{2} \cos(\theta + \frac{\Delta\theta}{2}) + \frac{\Delta s}{2b} \sin(\theta + \frac{\Delta\theta}{2}) \\ \frac{1}{2} \sin(\theta + \frac{\Delta\theta}{2}) + \frac{\Delta s}{2b} \cos(\theta + \frac{\Delta\theta}{2}) & \frac{1}{2} \sin(\theta + \frac{\Delta\theta}{2}) - \frac{\Delta s}{2b} \cos(\theta + \frac{\Delta\theta}{2}) \\ \frac{1}{b} & -\frac{1}{b} \end{bmatrix} \quad (11.11)$$

Los detalles para llegar a la ecuación 11.11 son:

$$F_{\Delta_{rl}} = \nabla_{\Delta_{rl}} f = \begin{bmatrix} \frac{\partial f}{\partial \Delta s_r} & \frac{\partial f}{\partial \Delta s_l} \end{bmatrix} = \dots \quad (11.12)$$

$$\begin{bmatrix} \frac{\partial \Delta s}{\partial \Delta s_r} \cos(\theta + \frac{\Delta\theta}{2}) + \frac{\Delta s}{2} - \sin(\theta + \frac{\Delta\theta}{2}) \frac{\partial \Delta\theta}{\partial \Delta s_r} & \frac{\partial \Delta s}{\partial \Delta s_l} \cos(\theta + \frac{\Delta\theta}{2}) + \frac{\Delta s}{2} - \sin(\theta + \frac{\Delta\theta}{2}) \frac{\partial \Delta\theta}{\partial \Delta s_l} \\ \frac{\partial \Delta s}{\partial \Delta s_r} \sin(\theta + \frac{\Delta\theta}{2}) + \frac{\Delta s}{2} \cos(\theta + \frac{\Delta\theta}{2}) \frac{\partial \Delta\theta}{\partial \Delta s_r} & \frac{\partial \Delta s}{\partial \Delta s_l} \sin(\theta + \frac{\Delta\theta}{2}) + \frac{\Delta s}{2} \cos(\theta + \frac{\Delta\theta}{2}) \frac{\partial \Delta\theta}{\partial \Delta s_l} \\ \frac{\partial \Delta\theta}{\partial \Delta s_r} & \frac{\partial \Delta\theta}{\partial \Delta s_l} \end{bmatrix} \quad (11.13)$$

Y con:

$$\Delta s = \frac{\Delta s_r + \Delta s_l}{2}; \Delta\theta = \frac{\Delta s_r - \Delta s_l}{b} \quad (11.14)$$

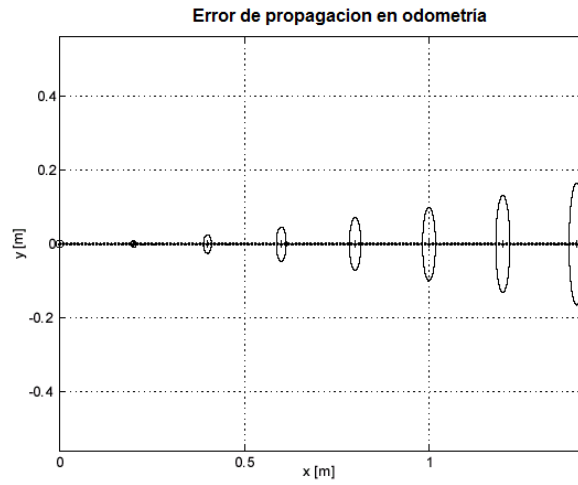
$$\frac{\partial \Delta s}{\partial \Delta s_r} = \frac{1}{2}; \quad \frac{\partial \Delta s}{\partial \Delta s_l} = \frac{1}{2}; \quad \frac{\partial \Delta\theta}{\partial \Delta s_r} = \frac{1}{b}; \quad \frac{\partial \Delta\theta}{\partial \Delta s_l} = -\frac{1}{b} \quad (11.15)$$

Se obtiene la ecuación 11.11. Las figuras 11.4 y 11.5 muestran ejemplos típicos de cómo los errores de posición crecer con el tiempo. Los resultados se han calculado utilizando el modelo de error presentado anteriormente. Una vez que el modelo de error se ha establecido, los parámetros de error deben ser especificados.

### 11.3. LOCALIZAR O NO LOCALIZAR: LOCALIZACIÓN BASADA EN NAVEGACIÓN VERSUS SOLUCIONES PROGRAMADAS

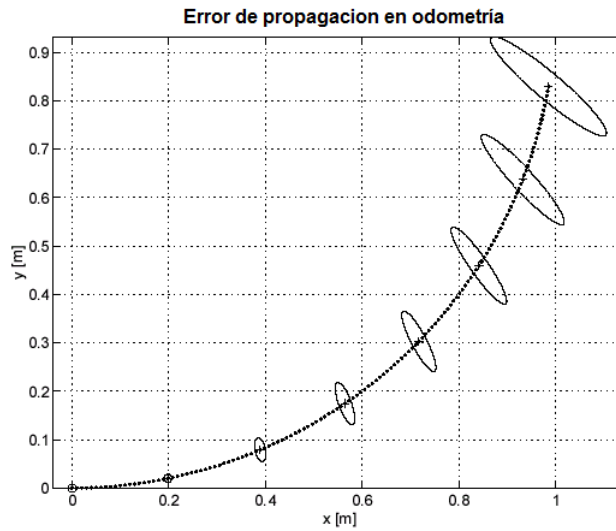
La figura 11.6 muestra un entorno interior estándar que un robot móvil navega. Se supone que el robot móvil en cuestión debe entregar mensajes entre dos habitaciones

Figura 11.4: Crecimiento de la incertidumbre de posición para movimiento en línea recta.



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition.

Figura 11.5: Crecimiento de la incertidumbre de posición para movimiento circular ( $r = \text{const}$ ).

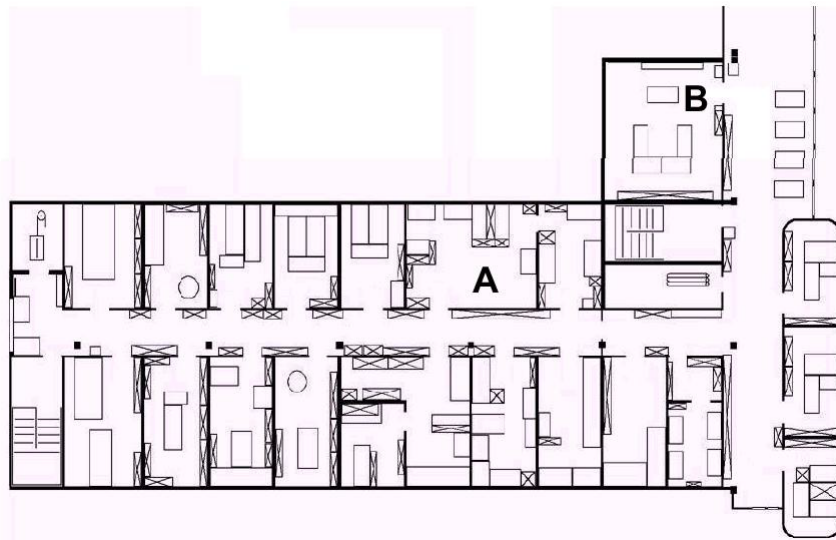


Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition.

específicas en este entorno: salas A y B. En la creación de un sistema de navegación, es

evidente que el robot móvil necesitará sensores y un sistema de control de movimiento. Una alternativa, adoptada por la comunidad basada en el comportamiento, sugiere que, dado que los sensores y actuadores son ruidosos y de limitada información, se debe evitar la creación de un mapa geométrico para la localización. En su lugar, esta comunidad sugiere el diseño de un conjunto de comportamientos que juntos dan como resultado el movimiento que se desea del robot. Fundamentalmente, este enfoque evita razonamiento explícito acerca de la localización y posición, por lo que generalmente evita la planificación de ruta explícita también.

Figura 11.6: Una muestra del entorno.



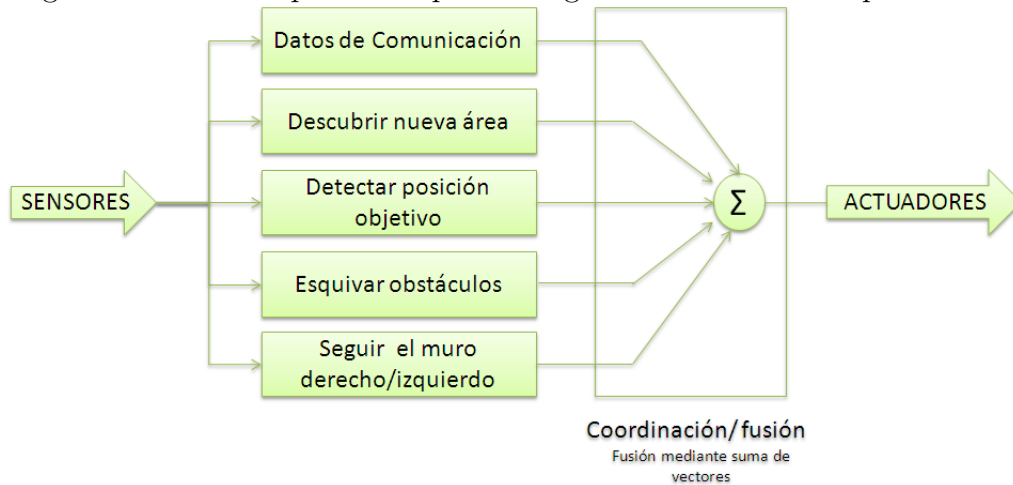
Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition.

Esta técnica se basa en la creencia de que existe un procedimiento solución para el problema de navegación particular en cuestión. Por ejemplo, en la figura 11.6, el enfoque de comportamiento para navegar de la habitación A a la habitación B podría ser diseñar un comportamiento de seguimiento del muro izquierdo y un detector para la habitación B que es activado por algún característica única en B ambiente, tal como el color de los la alfombra. A continuación, el robot puede alcanzar la sala B mediante la activación del seguidor izquierda de la pared con el detector de la habitación B como la condición de terminación para el programa.

La arquitectura de esta solución a un problema específico de navegación se muestra en la figura 11.7. La ventaja clave de este método es que, cuando sea posible, puede aplicarse muy rápidamente para un solo entorno con un pequeño número de posicio-

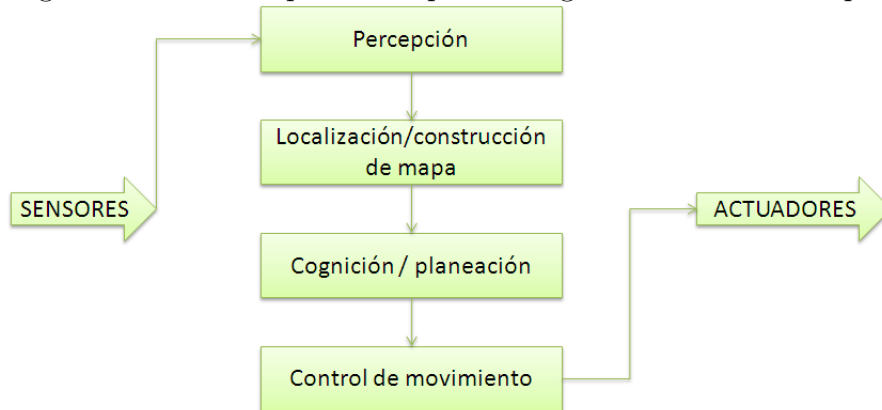
nes objetivo. A menudo, la adición de cada nuevo comportamiento incremental fuerza al diseñador de robots para volver a sintonizar todos los comportamientos existentes de nuevo para garantizar que las nuevas interacciones con el comportamiento recién introducido son estables.

Figura 11.7: Una arquitectura para navegación basada en comportamiento.



Fuente: Autores del proyecto

Figura 11.8: Una arquitectura para navegación basada en mapas.



Fuente: Autores del proyecto

En contraste con el enfoque basado en comportamiento, el enfoque basado en mapas incluye tanto la localización y módulos de cognición (véase la figura 11.8). En la navegación basada en mapas, el robot explícitamente intenta localizar mediante la recopilación de datos de los sensores, luego actualiza alguna creencia acerca de su posición

con respecto a un mapa del entorno. Las ventajas clave de este enfoque basado en el mapa para la navegación son las siguientes:

- El concepto de posición explícito basado en mapas hace las creencias del sistema acerca de su posición de transparentemente disponibles a los operadores humanos.
- La existencia del mapa en sí representa un medio para la comunicación entre el ser humano y robot: el ser humano sólo tiene que darle al robot un mapa nuevo si el robot va a un nuevo entorno.
- El mapa, si es creado por el robot, se puede utilizar por los seres humanos, así, de dos usos.

## 11.4. REPRESENTACIÓN DE LA CREENCIA

Se empieza hablando de la representación de la creencia. La primera rama principal en una taxonomía de los sistemas de representación de creencias diferencia entre los sistemas de creencias de una sola hipótesis y múltiples hipótesis. El primero cubre soluciones en las que el robot postula su posición única, mientras que la segunda permite que un robot móvil para describir el grado en el que no se sabe acerca de su posición.

### 11.4.1. Creencia de hipótesis única

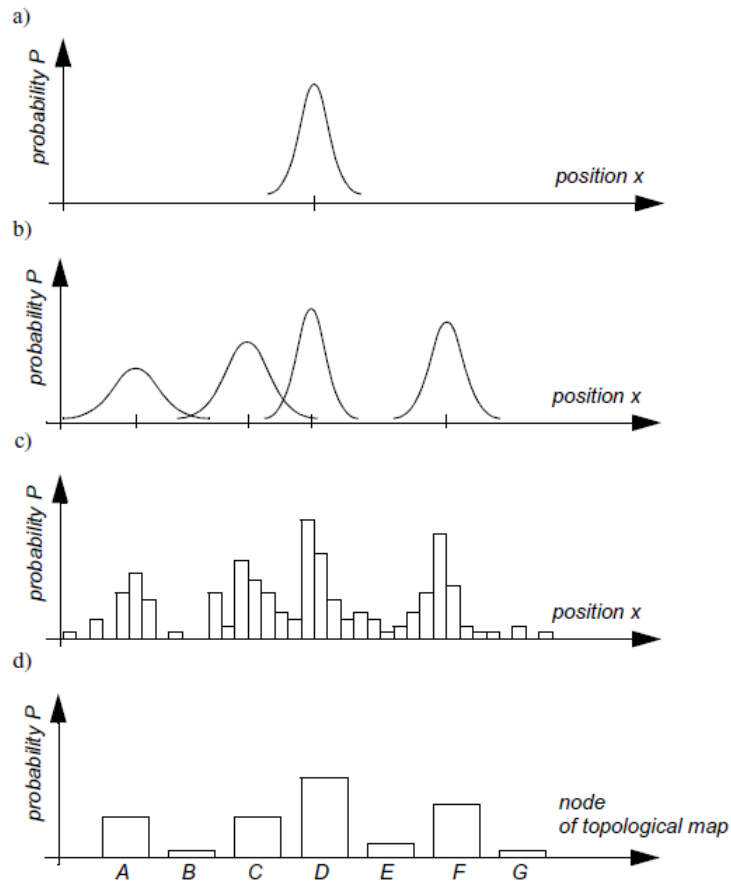
Dados algunos mapas del entorno, la creencia del robot sobre la posición se expresa como un único punto único en el mapa. En la figura 11.10, se muestran tres ejemplos de una creencia de hipótesis única con tres representaciones distintas para los mapas del mismo entorno real (figura 11.10a). En la figura 11.10b, un único punto es anotado geoméricamente como la posición del robot en un mapa continuo geométrico 2D. En la figura 11.10c, el mapa es un mapa teselado y discreto, y la posición se observó en el mismo nivel de fidelidad como el tamaño de celda del mapa. En la figura 11.10d, el mapa no es geométrico en absoluto, sino abstracto y topológico.

La ventaja principal de la representación de un solo hipótesis de posición se deriva del hecho de que, dada una creencia única, no hay ambigüedad de posición.

### 11.4.2. Creencia de múltiples hipótesis

En el caso de las creencias de múltiples hipótesis respecto a la posición, el robot rastrea no sólo una única posición posible, sino posiblemente un conjunto infinito de

Figura 11.9: La representación de la creencia con respecto a la posición del robot (1D) en mapas continuos y discretos (teselados).



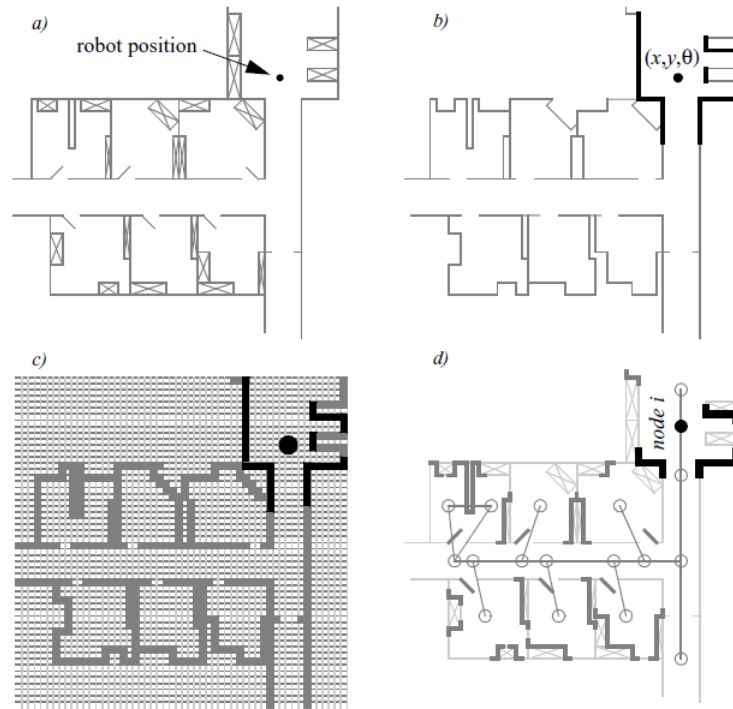
Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition.

posiciones.

En un ejemplo sencillo, partiendo de la obra de Jean-Claude Latombe, *“Motion Planning with Uncertainty: A Landmark Approach”*, y *“Robot Motion Planning”*, la posición del robot se describe en términos de un polígono convexo colocado en un mapa 2D del entorno. Esta representación es particularmente apta para las funciones de seguimiento definidas matemáticamente, tales como el filtro de Kalman, que están diseñadas para operar de manera eficiente en las distribuciones gaussianas.

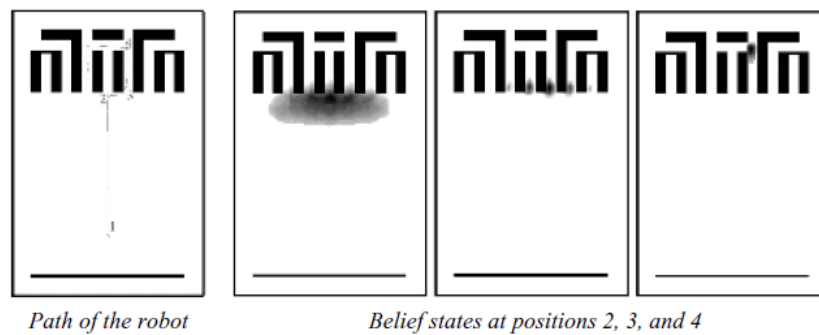
La ventaja clave de la representación de múltiples hipótesis es que el robot puede mantener explícitamente la incertidumbre con respecto a su posición. Una de las desventajas fundamentales de los enfoques de múltiples hipótesis implica la toma de

Figura 11.10: Tres ejemplos de hipótesis única de posición utilizando diferentes representaciones del mapa.



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition.

Figura 11.11: Ejemplo de seguimiento de múltiples hipótesis



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition.

decisiones. La figura 11.11 muestra un ejemplo de múltiple hipótesis. En la posición 3,

el estado del robot creencia se distribuye entre cinco pasillos separado. El beneficio más crítico del estado creencia de múltiples hipótesis es la capacidad de mantener un sentido de la posición, mientras se anota explícitamente la incertidumbre del robot sobre su propia posición.

## 11.5. REPRESENTACIÓN DEL MAPA

Las decisiones tomadas con respecto a la representación del medio ambiente pueden tener un impacto sobre las opciones disponibles para la representación de posición del robot. Tres relaciones fundamentales deben entenderse a la hora de elegir una representación de mapa en particular:

1. La precisión del mapa debe coincidir apropiadamente con la precisión con la que el robot necesita lograr sus objetivos.
2. La precisión del mapa y el tipo de características representadas deben coincidir con los tipos de precisión y los datos devueltos por los sensores del robot.
3. La complejidad de la representación del mapa tiene un impacto directo sobre la complejidad computacional del razonamiento acerca del mapeo, localización y navegación.

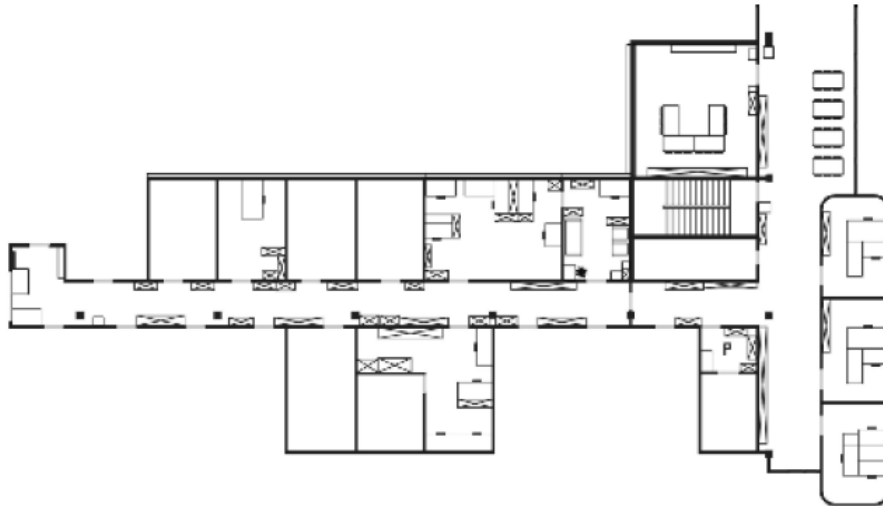
### 11.5.1. Representaciones continuas

La ventaja clave de una representación continua de mapa es el potencial para una alta precisión y expresividad con respecto a la configuración del entorno, así como la posición del robot dentro de ese entorno. El peligro de una representación continua es que el mapa puede ser computacionalmente costoso. Sin embargo, este peligro puede ser atenuado mediante el empleo de abstracción y capturar sólo las características ambientales más relevantes.

Un ejemplo de tal representación, que se muestra en la figura 11.12, es una representación 2D en la que los polígonos representan todos los obstáculos en un espacio de coordenadas con valores continuos. Todos los objetos se puede aproximar utilizando polígonos convexos muy simples, sacrificando la estética del mapa en aras de la velocidad de cálculo.

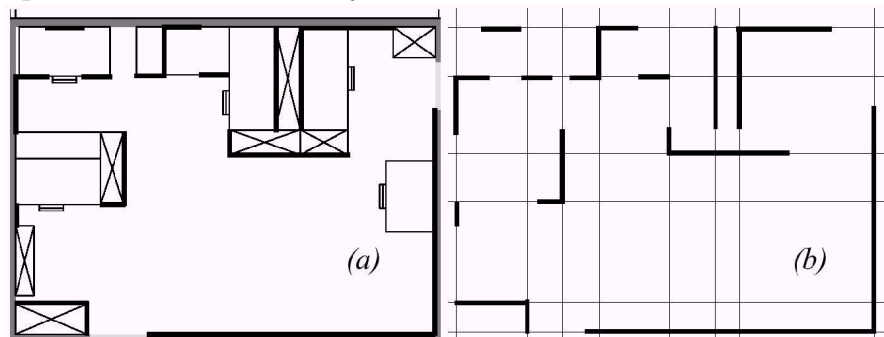
La figura 11.13 muestra un mapa de un ambiente interior de la EPFL (Swiss Federal Institute of Technology at Lausanne) utilizando una representación de línea continua. Las únicas características ambientales capturadas por el mapa son líneas rectas, tales

Figura 11.12: Una representación continua utilizando polígonos como obstáculos ambientales



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition.

Figura 11.13: Ejemplo de una representación de línea valor continuo de EPFL. (a) Mapa real. (b) Representación con un conjunto de líneas infinitas.

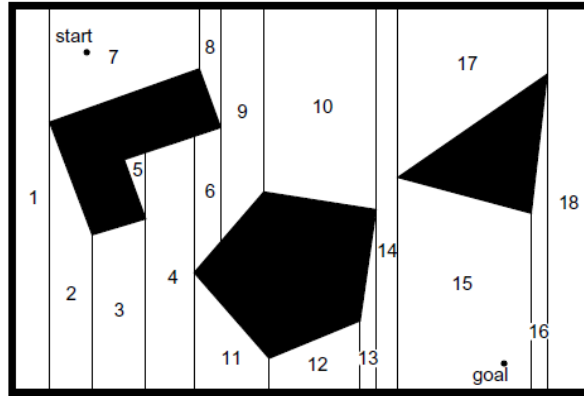


Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition.

como las que se encuentran en las esquinas y a lo largo de las paredes. En un caso, la posible posición del robot puede ser representada como una forma geométrica en el hiperplano. Esto se muestra en la figura 11.27, en la que se representa la posición del robot por un área oval delimitada.

## 11.5.2. Estrategias de descomposición

Figura 11.14: Ejemplo de descomposición de celda exacta.



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition.

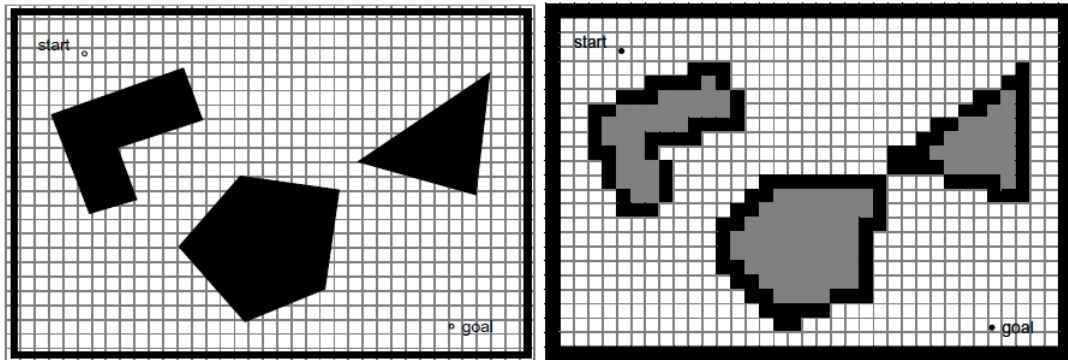
La figura 11.14 muestra una descomposición exacta de un área de trabajo plana poblada de obstáculos poligonales. La representación de mapa tesela el espacio en áreas de espacio libre donde cada área afectada se almacena como un único nodo, lo que resulta en un total de sólo dieciocho nodos en este ejemplo. Aquí la posición particular de un robot dentro de cada área de espacio libre no importa. Lo que importa es la capacidad del robot para recorrer desde cada área de espacio libre hacia las áreas adyacentes. Tal descomposición exacta no siempre es apropiada. La descomposición exacta es una función de los obstáculos del entorno particular y el espacio libre. Si esta información es difícil de recoger o incluso desconocida, entonces, tal enfoque no es factible.

Una alternativa es la descomposición fija, en la que el mundo es teselado, transformando el entorno real continuo en una aproximación discreta para el mapa. Esta transformación se demuestra en la figura 11.15. La desventaja principal de este enfoque se deriva de su naturaleza inexacta. Sin embargo, otro enfoque es la descomposición celular adaptativa, tal como se presenta en la figura 11.16.

El concepto de descomposición fija es muy popular en la robótica móvil, es quizás la técnica de representación de mapa común más utilizada actualmente.

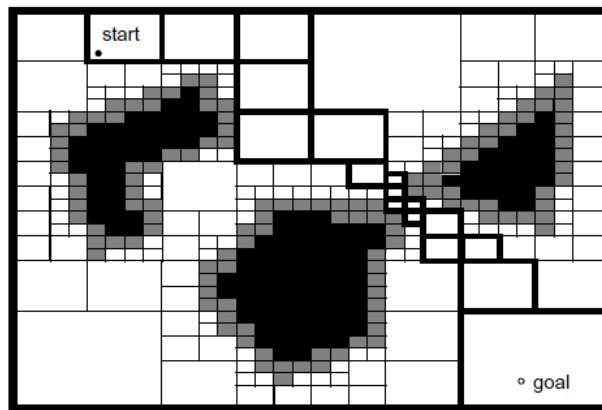
Por estas razones, una alternativa, llamada descomposición topológica, ha sido objeto de alguna exploración en robótica móvil. Los enfoques topológicos evitan la medición directa de las calidades ambientales geométricas, concentrándose en las características del entorno que son más relevantes para el robot para la localización. La figura 11.17

Figura 11.15: Descomposición fija del mismo espacio, el pasaje estrecho desaparece.



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition.

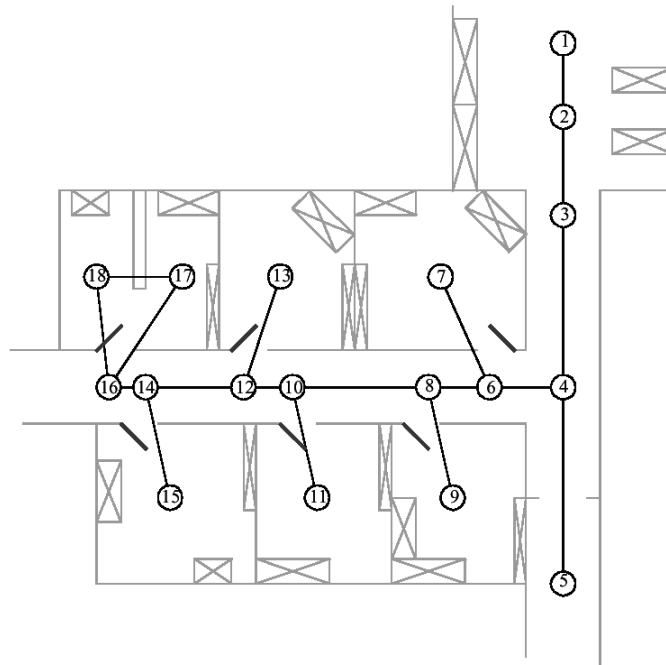
Figura 11.16: Ejemplo de descomposición adaptada (celda variable aproximada) de un entorno.



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition.

muestra una representación topológica de una serie de pasillos y oficinas en un ambiente interior. En este caso, el robot se supone que tiene un detector de cruce, tal vez utilizando un sonar y visión para encontrar intersecciones entre pasillos y entre pasillos y habitaciones.

Figura 11.17: Una representación topológica de una zona interior de oficinas.



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition.

## 11.6. LOCALIZACIÓN PROBABILÍSTICA BASADA EN MAPAS

### 11.6.1. Generalidades

Las técnicas de localización probabilísticas basados en mapas identifican explícitamente probabilidades con las posiciones del robot posibles, y por esta razón, estos métodos han sido el foco de las investigaciones recientes. Esta nueva área de investigación se llama “Robótica probabilística”. La idea clave en la robótica probabilística es representar la incertidumbre utilizando la teoría de la probabilidad.

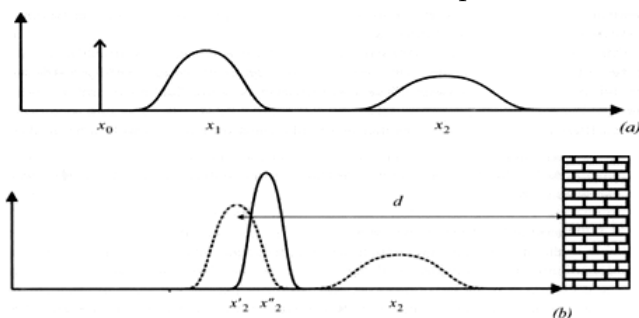
### 11.6.2. El problema de localización de robot

Los procesos de actualización basados en valores de los sensores propioceptivos y valores de los sensores exteroceptivos son a menudo separados lógicamente, conduciendo a un proceso general para la actualización de la posición del robot que comprende dos etapas, que se llaman actualización de predicción (o acción) y la actualización de

percepción (o medición, o corrección).

- Durante la actualización de predicción (o acción) el robot utiliza sus sensores propioceptivos para estimar su configuración, por ejemplo, el robot estima su movimiento utilizando los encoders. En esta fase, la incertidumbre acerca de la configuración del robot aumenta debido a la integración del error de odométrica en el tiempo. En la figura 11.18a, se ilustra este proceso para un robot móvil en un entorno de una sola dimensión.
- Durante la actualización de percepción (o medición, o corrección) el robot usa la información de sus sensores exteroceptivos para corregir la posición estimada durante la fase de predicción, por ejemplo, el robot utiliza un telémetro para medir su distancia actual de una pared y la corrige en de acuerdo a la posición estimada durante la fase de predicción. Durante la fase de percepción, la incertidumbre de la configuración del robot disminuye (figura 11.18b).

Figura 11.18: En la robótica probabilística, las creencias acerca de la configuración del robot se representan como funciones de densidad de probabilidad.



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.

La diferencia fundamental en la representación del estado de creencia conduce a las siguientes ventajas y desventajas de los dos métodos:

- La localización de Markov permite a la localización empezar a partir de cualquier posición desconocida y por lo tanto puede recuperarse de situaciones ambiguas ya que el robot puede seguir múltiples posiciones posibles, completamente diferentes. Sin embargo, para actualizar la probabilidad de que todas las posiciones dentro del espacio de estado completo en cualquier momento necesita una representación discreta del espacio, tal como una red geométrica o un gráfico topológico.

- La localización por filtro de Kalman rastrea el robot desde una posición inicialmente conocida y es de por sí a la vez precisa y eficiente. En particular, la localización del filtro de Kalman se puede utilizar en las representaciones mundo continuo. Sin embargo, si la incertidumbre del robot se hace demasiado grande (por ejemplo, debido a una colisión del robot con un objeto) y por lo tanto no es unimodal realmente, el filtro de Kalman puede no captar la multitud de posibles posiciones del robot y puede perderse irrevocablemente.

### 11.6.3. Conceptos básicos de la teoría de la probabilidad

Sea  $X$  una variable aleatoria y  $x$  un valor específico que  $X$  podría suponer. Un ejemplo típico es tirar los dados, donde  $X$  puede tomar cualquier valor entre 1 y 6. Se denota con:

$$p(X = x) \quad (11.16)$$

La probabilidad de que la variable aleatoria  $X$  tenga el valor  $x$ . Por ejemplo, el resultado de un rodillo de matriz se caracteriza por:

$$p(X = 1) = P(X = 2) = p(X = 3) = p(X = 4) = p(X = 5) = p(X = 6) = \frac{1}{6} \quad (11.17)$$

A partir de ahora, para simplificar la notación se omitará la mención explícita de la variable aleatoria, y en su lugar se utilizará la abreviatura simple  $p(x)$ . En espacios continuos, las variables aleatorias pueden tomarse en un continuo de valores y en este caso se hablará acerca de las funciones de densidad de probabilidad (FDP). Ambas probabilidades discretas y continuas integran a uno, por lo tanto:

$$\sum_x p(x) = 1 \text{ (Para probabilidades discretas)} \quad (11.18)$$

$$\int_x p(x), dx = 1 \text{ (Para probabilidades continuas)} \quad (11.19)$$

Además, las probabilidades son siempre no negativas, es decir,  $p(x) > 0$

**La distribución Gaussiana.** Como se ha mencionado anteriormente, una función de densidad de probabilidad común es la distribución gaussiana:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right) \quad (11.20)$$

También llamada distribución normal, que es comúnmente abreviada con:

$$p(x) = N(x, a^2) \quad (11.21)$$

Donde  $\mu$  y  $\sigma^2$  especifican la media y la varianza de la variable aleatoria  $x$ . Tenga en cuenta que en este caso la variable aleatoria  $x$  es un escalar. Sin embargo, cuando  $x$  es un vector de  $k$  dimensiones, se tiene una distribución normal multivariante caracterizada por una función de densidad de la siguiente forma:

$$p(x) = \frac{1}{(2\pi)^{k/2} \det(\Sigma)^{1/2}} \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)\right) \quad (11.22)$$

Donde  $\mu$  es el vector medio y  $\Sigma$  es una matriz semidefinida positiva y simétrica denominada matriz de covarianza.

**Distribución conjunta.** La distribución conjunta de dos variables aleatorias  $X$  y  $Y$  está dada por  $p(x, y)$ , la cual describe la probabilidad de que la variable aleatoria  $X$  tome el valor  $x$  y que  $Y$  tome el valor  $y$ . Si  $X$  y  $Y$  son independientes, se tiene:

$$p(x, y) = p(x)p(y) \quad (11.23)$$

**Probabilidad condicional.** La probabilidad condicional describe la probabilidad de que la variable aleatoria  $X$  tome el valor  $x$  condicionado en el conocimiento de que  $Y$  que tome el valor de  $y$ . La probabilidad condicional se denota con  $p(x | y)$  y, si  $p(y) > 0$ , se define como

$$p(x | y) = \frac{p(x, y)}{p(y)} \quad (11.24)$$

Si  $X$  y  $Y$  son independientes, tenemos:

$$p(x | y) = \frac{p(x)p(y)}{p(y)} = p(x) \quad (11.25)$$

**Teorema de la probabilidad total.** El teorema de la probabilidad total se origina a partir de los axiomas de la teoría de la probabilidad y se escribe como:

$$\sum_y p(x | y)p(y) \quad (\text{Para probabilidades discretas}) \quad (11.26)$$

$$\int_x p(x | y)p(y)dy \quad (\text{Para probabilidades continuas}) \quad (11.27)$$

El teorema de la probabilidad total es utilizado tanto por los algoritmos de localización de los filtros de Markov y Kalman, durante la actualización de la predicción.

**La regla de Bayes.** La regla de Bayes relaciona la probabilidad condicional  $p(x | y)$  a su inversa  $p(y | x)$ . Bajo la condición de que  $p(y) > 0$ , la regla de Bayes se escribe como:

$$p(x | y) = \frac{p(y | x)p(x)}{p(y)} \quad (11.28)$$

La regla de Bayes es utilizada tanto por los algoritmos de localización de los filtros de Markov y Kalman, de durante la actualización de la medición.

**Probabilidad previa y posterior.** Una distribución de probabilidad previa de una variable aleatoria  $x$  es la distribución de probabilidad  $p(x)$  que se debe incorporar antes de los datos  $y$ . Como se muestra en la ecuación (11.28), la regla de Bayes proporciona una forma conveniente para calcular la probabilidad posterior  $p(x | y)$ , utilizando la probabilidad condicional “inversa”  $p(y | x)$  y la probabilidad previa  $p(x)$ . En la regla de Bayes el factor  $p(y)^{-1}$  se escribe generalmente como un factor de normalización, genéricamente designado  $\eta$ , que se puede determinar sencillamente, recordando que la integral de una distribución de probabilidad es siempre 1. De este modo, la regla de Bayes puede ser escrita como:

$$p(x | y) = \eta p(y | x) p(x) \quad (11.29)$$

#### 11.6.4. Terminología

Aquí se introduce la terminología que se utiliza en las siguientes secciones. Sea  $t$  el tiempo y  $x_t$  denota la localización del robot. Para movimiento plano,  $x_t = [x, y, \theta]^T$  es un vector tridimensional que consiste en la posición y la orientación del robot. La trayectoria del robot se da como:

$$X_T = x_0, x_1, x_2, \dots, x_T \quad (11.30)$$

Donde  $T$  también puede ser infinito. El término  $u_t$  denota las lecturas de los sensores propioceptivos en el tiempo  $t$ . Si se supone que el robot recibe exactamente unos datos en cada punto en el tiempo, la secuencia de datos propioceptivos se puede escribir como:

$$U_T = u_0, u_1, u_2, \dots, u_T \quad (11.31)$$

Si se supone que el robot toma exactamente una medición en cada punto en el tiempo (sensores exteroceptivos), la secuencia de las mediciones se da como:

$$Z_T = z_0, z_1, z_2, \dots, z_T \quad (11.32)$$

Finalmente,  $M$  denota el verdadero mapa del entorno y supongamos que el entorno se compone de puntos de referencia de puntos en dos dimensiones (o líneas infinitas  $2D$ ). En este caso, el mapa es un vector de tamaño  $2n$ , donde  $n$  es el número de características en el mundo, por lo tanto,

$$M = m_0, m_1, m_2, \dots, m_{n-1} \quad (11.33)$$

Donde  $m_i$ ,  $i = 0 \cdots n - 1$  son vectores que representan las coordenadas  $2D$  de los puntos de referencia en el marco de referencia del mundo. Si se denota la creencia en un estado variable  $x_t$  por  $bel(x_t)$ , se escribe:

$$bel(x_t) = p(x_t | z_{1 \rightarrow t}, u_{1 \rightarrow t}) \quad (11.34)$$

Donde la posterior  $p(x_t | z_{1 \rightarrow t}, u_{1 \rightarrow t})$  representa la probabilidad de que el robot esté en  $x_t$  dado todas sus observaciones pasadas  $z_{1 \rightarrow t}$  y todas las últimas entradas de control  $u_{1 \rightarrow t}$ . En la localización de Markov y Kalman, también a menudo se refiere a la creencia calculada antes de la incorporación de la nueva observación  $z_t$  justo después de la entrada de control  $u_t$ . Tal posterior se denota como:

$$\overline{bel}(x_t) = p(x_t | z_{1 \rightarrow t-1}, u_{1 \rightarrow t}) \quad (11.35)$$

### 11.6.5. Los ingredientes de la localización basada en mapa probabilístico

Con el fin de resolver el problema de localización del robot, se requiere de la siguiente información.

1. **La distribución de probabilidad inicial  $bel(x_0)$ .** En el caso de la ubicación inicial del robot es desconocida, la creencia inicial  $bel(x_0)$  es una distribución uniforme en todas las posiciones. A la inversa, si la ubicación es perfectamente conocida la creencia inicial es una función delta de Dirac.
2. **Mapa del medio ambiente.** El mapa del entorno  $M = m_0, m_1, m_2, \dots, m_n$  debe ser conocido. Si el mapa no es conocido a priori, entonces, el robot tiene que construir un mapa del entorno.
3. **Datos.** Para localizar el robot, obviamente, tiene que utilizar los datos de sus sensores propioceptivos y exteroceptivos. La terminología se definió anteriormente.
4. **El modelo movimiento probabilístico.** El modelo de movimiento probabilístico se deriva de la cinemática del robot. En el caso libre de ruido, la ubicación actual del robot  $x_t$  se puede calcular como una función  $f$  de la  $x_{t-1}$  ubicación anterior, y las lecturas del codificador  $u_t$ , esto es:

$$x_t = f(x_{t-1}, u_t) \quad (11.36)$$

5. **El modelo de medición probabilística.** Esto se deriva directamente del modelo de sensor exteroceptivo, por ejemplo, el modelo de error del láser, del sonar, o de la cámara. La función de medición  $h$  depende claramente del mapa del entorno  $M$  y en la ubicación del robot  $x_t$ , por lo tanto, podemos escribir:

$$z_t = h(x_t, M) \quad (11.37)$$

En el ejemplo mostrado en la figura 11.18 el robot utiliza un telémetro para medir su distancia  $d$  de la pared de la derecha. Aquí,  $d$  es la observación, y por lo tanto  $z_t = d$ . El mapa  $M$  se representa mediante una única función  $m$  (es decir, la pared), que suponemos está en la coordenada  $m = 10$ . La función de medición en este ejemplo simple es la siguiente:

$$h(x_t, M) = 10 - x_t \quad (11.38)$$

Para obtener el modelo de medición probabilística, sólo tenemos que añadir un término de ruido a la función de medición de tal manera que la distribución de probabilidad  $p(z_t | x_t, M)$  llegue al máximo en el valor sin ruido  $h(x_t, M)$ . Por ejemplo, si suponemos ruido gaussiano podemos escribir:

$$p(z_t | x_t, M) = N(h(x_t, M), R_t) \quad (11.39)$$

Donde, más generalmente,  $N$  denota una distribución normal multivariante con media  $h(x_t, M)$  y la matriz de covarianza de ruido  $R_t$ .

### 11.6.6. Clasificación de los problemas de localización

Antes de proceder a la localización del filtro de Markov y Kalman, se debe entender la diferencia entre los tres tipos de problemas de localización, que son: el seguimiento de posición, la localización global, y el problema robot secuestrado.

**Seguimiento de posición.** En el seguimiento de posición, la ubicación actual del robot se actualiza basándose en el conocimiento de su posición anterior (tracking). Esto implica que la ubicación inicial del robot se supone que es conocida.

**Localización global.** La localización global, por el contrario, se supone que la ubicación inicial del robot es desconocida. Esto significa que el robot se puede colocar en cualquier lugar en el medio ambiente sin conocimiento sobre él y es capaz de localizar globalmente dentro de él. En la localización global, la creencia robot inicial es usualmente una distribución uniforme.

**El problema del robot secuestrado.** El problema del robot secuestrado aborda el caso en el que el robot es secuestrado y se traslada a otro lugar. El problema del robot secuestrado es similar al problema de localización global sólo si el robot se da cuenta de haber sido secuestrado. La dificultad surge cuando el robot no sabe que se ha movido a otro lugar y que cree que sabe dónde está, pero en realidad no lo hace.

## 11.6.7. Localización de Markov

Localización de Markov sigue el estado de creencia del robot mediante una función de densidad de probabilidad arbitraria para representar la posición del robot. En la práctica, todos los sistemas de localización de Markov conocidos implementan esta representación de creencia genérica primero teselando el espacio de configuración robot  $(x, y, \theta)$  en un número discreto y finito de posible de robot plantea en el mapa. En aplicaciones reales, el número de posibles plantea puede oscilar entre varios cientos a millones de posiciones y orientaciones. La localización de Markov aborda el problema de localización global, el problema posición de seguimiento, y el problema de robot secuestrado.

### Actualizaciones de predicción y medición.

**Actualización de predicción (acción).** Recordemos que en esta fase el robot calcula su posición actual (es decir, la creencia) basada en el conocimiento de la posición anterior (es decir, la creencia) y la entrada odométrica. El teorema de la probabilidad total se utiliza para calcular la creencia actual del robot  $\overline{bel}(x_t)$  como una función de la creencia anterior de  $bel(x_t)$  y los datos propioceptivos (por ejemplo, la medición del codificador o la entrada de control)  $u_t$ :

$$\overline{bel}(x_t) = \int p(x_t | u_t, x_{t-1}) bel(x_t) dx_{t-1} (\text{caso continuo}) \quad (11.40)$$

$$\overline{bel}(x_t) = \sum_{x_{t-1}} p(x_t | u_t, x_{t-1}) bel(x_{t-1}) (\text{caso discreto}) \quad (11.41)$$

La integral (suma) en (11.40) y (11.41) debe calcularse sobre todas las posibles posiciones del robot  $x_t$ . Esto significa que, en situaciones reales, donde el número de células utilizadas para representar las posiciones del robot es de varios millones, el cómputo (11.40) y (11.41) puede llegar a ser poco práctico y, por lo tanto, impedir la operación en tiempo real. Finalmente, (11.40) y (11.41) pueden ser vistas como una convolución entre la creencia anterior  $bel(x_{t-1})$  y el modelo de movimiento probabilístico  $p(x_t | u_t, x_{t-1})$ .

**Actualización de percepción (medición).** La regla de Bayes, mencionada anteriormente se utiliza para calcular el nuevo estado de creencia robot  $bel(x_t)$  como una función de su medición de datos  $z_t$ , y su ex estado de creencia  $\overline{bel}(x_t)$ :

$$bel(x_t) = \eta p(z_t | x_t, M) \overline{bel}(x_t) \quad (11.42)$$

Donde  $p(z_t | x_t, M)$  es el modelo de medición probabilística, es decir, la probabilidad de observar los datos medidos  $z_t$ , dado el conocimiento del mapa  $M$  y la posición del robot  $x_t$ . Por lo tanto, el nuevo estado de creencia no es más que el producto entre el modelo de medición probabilística y el estado creencia anterior. Obsérvese que (11.42) no se actualiza sólo una posición, sino todas las posibles posiciones del robot  $x_t$ .

El algoritmo de localización de Markov. La figura 11.19 muestra el algoritmo de localización de Markov en la forma pseudo-algortmica. El desafío más importante en la localización de Markov es el cálculo de  $p(z_t | x_t, M)$ . El modelo de sensor debe calcular la probabilidad de una medición perceptual específica dada la ubicación del robot y el mapa del entorno.

Figura 11.19: Algoritmo general para la localización de Markov.

*For all  $x_t$  do:*

$$\overline{bel}(x_t) = \int p(x_t | u_t, x_{t-1}) bel(x_{t-1}) dx_{t-1} \quad (\text{actualización de predicción})$$

$$bel(x_t) = \eta p(z_t | x_t, M) \overline{bel}(x_t) \quad (\text{actualización de medición})$$

*Endfor*

*Return  $bel(x_t)$*

Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.

### La suposición de Markov.

Las ecuaciones (11.40) y (11.42) forman la base de la localización de Markov e incorporan el supuesto de Markov. Formalmente, esto significa que su salida  $x_t$  es una función sólo del estado anterior del robot  $x_{t-1}$ , y sus acciones más recientes (odometría)  $u_t$  y percepción  $z_t$ . La suposición de Markov no es, por supuesto, una suposición válida, pues no tiene en cuenta diversos factores que pueden afectar las mediciones de los sensores: desgaste de una rueda mayor que la otra, choques del robot, etc.

### Ilustración de localización de Markov.

En la figura 11.20, se ilustra el principio de funcionamiento de la localización de Markov en el caso continuo. Para simplificar, el medio ambiente es un pasillo unidimensional con tres pilares idénticos. En este ejemplo, se supone que en un principio el robot no conoce su ubicación inicial y tiene por lo tanto que localizar a partir de cero. Según el marco probabilista descrito antes, la creencia inicial del robot  $(x_0)$  es una distribución uniforme en todos los lugares tal como se ilustra en la figura 11.20a.

Se supone ahora que el robot utiliza sus sensores exteroceptivos y sentidos que están al lado de un pilar. Este es claramente la actualización de percepción de la localización de Markov. Entonces, de acuerdo a la regla de Bayes su creencia  $(x_0)$  tiene que ser multiplicada por  $p(z_t | x_t, M)$  como se indica en la ecuación (11.42). Debido a que los tres pilares son exactamente idénticos, el robot no sabe a cuál de los pilares se enfrenta. Por lo tanto, la probabilidad  $p(z_t | x_t, M)$  de la observación de un pilar se caracteriza por tres picos, cada uno correspondiente a uno de los pilares indistinguibles en el entorno. El gráfico superior en (b) visualiza  $p(z_t | x_t, M)$ . Después de esta actualización de percepción el robot todavía no sabe dónde está. De hecho, ahora tiene tres hipótesis distintas, que son todos igualmente plausibles. El gráfico inferior en (b) visualiza el resultado de la multiplicación. Debido a que es el resultado de una multiplicación con una función constante, el resultado todavía se caracteriza por los tres picos exactamente idénticos.

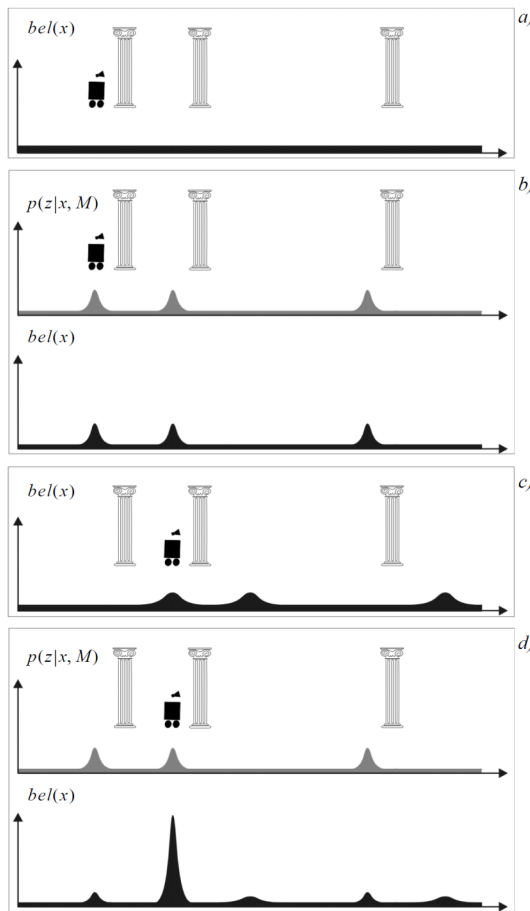
Ahora estamos en la actualización de acción de la localización de Markov. La figura 11.20c muestra el efecto en la creencia del robot. Como resultado de la convolución de la creencia anterior del robot con el modelo de movimiento  $p(x_t | u_t, x_{t-1})$ , la nueva creencia se ha desplazado en la dirección del movimiento y también aplanado. Los tres picos son ahora más grandes, lo cual refleja la incertidumbre que se introduce por el movimiento del robot.

La figura 11.20d muestra la creencia después de observar otro de los pilares. Estamos otra vez en la actualización de percepción. En este caso, el algoritmo de localización de Markov multiplica de nuevo la creencia actual con la probabilidad perceptual  $p(z_t | x_t, M)$ . Como se observa, esta vez el resultado de la multiplicación es un solo pico distinguible cerca de una de las columnas, y el robot está ahora bastante seguro de donde está.

### Estudio de caso 1: Markov localización utilizando un mapa de red.

Localización de Markov se implementa en la práctica mediante una representación espacial de malla del entorno. Por lo general, una descomposición fija se utiliza. Pa-

Figura 11.20: Ilustración del algoritmo de localización de Markov.



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.

ra movimiento plano, la configuración del robot se expresa mediante tres parámetros  $(x, y, \theta)$ . El espacio de estado final es por lo tanto almacenado en la memoria del robot, como una matriz tridimensional (figura 11.22). Se va a teselar el entorno en diez celdas equidistantes (figura 11.21). Supongamos que la creencia inicial del robot  $bel(x_0)$  es una distribución uniforme de 0 a 3 como se muestra en la figura 11.21a. Observe que todos los elementos se normalizan de tal manera que su suma sea 1.

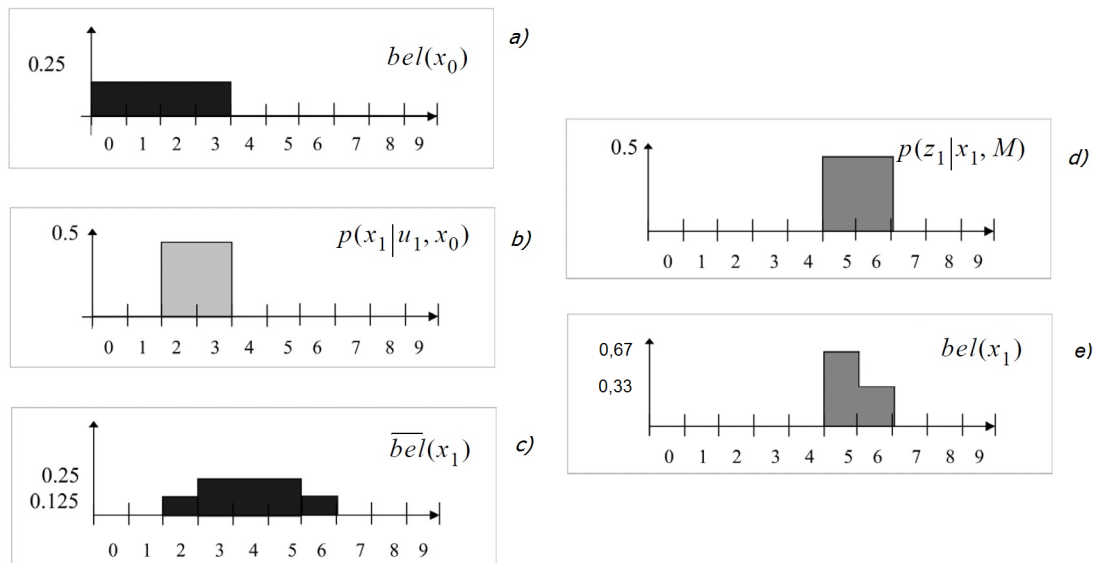
**Actualización de predicción.** Se supone que el modelo de movimiento probabilístico de la odometría  $p(x_1 | u_1, x_0)$  es aquel representado en la figura 11.21b. Este modelo debe interpretarse de esta manera: entre el tiempo  $t = 0$  y tiempo  $t = 1$ , el robot puede haberse movido o bien dos o tres unidades a la derecha. En este ejemplo, ambos movimientos tienen la misma probabilidad de ocurrir. La ecuación de predic-

ción de fase es (11.41), es decir, la creencia final  $\overline{bel}(x_1)$  está dada por el teorema de la probabilidad total, el cual convolucionna la creencia inicial  $bel(x_0)$  con el modelo de movimiento  $p(x_1 | u_1, x_0)$ . Usando (11.41), se obtiene:

$$\overline{bel}(x_1) = \sum_{x_0=0}^3 p(x_1 | u_1, x_0) bel(x_0). \quad (11.43)$$

Con el fin de calcular la probabilidad de la posición  $x_1$ , en el nuevo estado de creencia, hay que sumar general las posibles formas en que el robot puede alcanzar  $x_1$ , de acuerdo con las posiciones potenciales expresadas en (Informador creencia de estado  $x_0$ ) y la entrada potencial expresada por  $u_1$ . Observe que debido a que está limitado entre 0 y 3, el robot sólo puede alcanzar los estados entre 2 a 6, por lo tanto:

Figura 11.21: Localización de Markov usando un mapa de red.



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, First edition.

$$p(x_1 = 2) = p(x_0 = 0)p(u_1 = 2) = 0,125 \quad (11.44)$$

$$p(x_1 = 3) = p(x_0 = 0)p(u_1 = 3) + p(x_0 = 1)p(u_1 = 2) = 0,25 \quad (11.45)$$

$$p(x_1 = 4) = p(x_0 = 1)p(u_1 = 3) + p(x_0 = 2)p(u_1 = 2) = 0,25 \quad (11.46)$$

$$p(x_1 = 5) = p(x_0 = 2)p(u_1 = 3) + p(x_0 = 3)p(u_1 = 2) = 0,25 \quad (11.47)$$

$$p(x_1 = 6) = p(x_0 = 3)p(u_1 = 3) = 0,125 \quad (11.48)$$

La expresión (11.44) resulta del hecho de que el estado  $x_1 = 2$  sólo es alcanzado con la combinación  $(x_0 = 0, u_1 = 2)$ . La expresión (11.45) viene del hecho de que el estado  $x_1 = 3$  sólo se puede alcanzar con las combinaciones de  $(x_0 = 0, u_1 = 3)$  o  $(x_0 = 1, u_1 = 2)$ . Las otras expresiones siguen una forma similar. Las ecuaciones (11.44) - (11.48) están llevando a cabo nada más que el teorema de la probabilidad total (o convolución) enunciados en (11.43). El resultado de la aplicación de este teorema se muestra en la figura 11.21c.

**Medición de actualización.** Se supone ahora que el robot utiliza su telémetro a bordo y mide la distancia  $z$  desde el origen. Entonces se supone que el modelo de error estadístico del sensor de alcance es el que se muestra en la figura 11.21d. Esta gráfica dice que la distancia del robot desde el origen puede ser igualmente 5 o 6 unidades. La ecuación de medición de actualización es (11.42). La creencia final  $bel(x_1)$  se calcula de acuerdo a la regla de Bayes. Es el producto entre la creencia actual del robot  $\overline{bel}(x_1)$  y el modelo de error de medición  $p(z_1 | x_1, M)$ , donde, en este caso, el mapa  $M$  es simplemente el origen de los ejes. Por lo tanto:

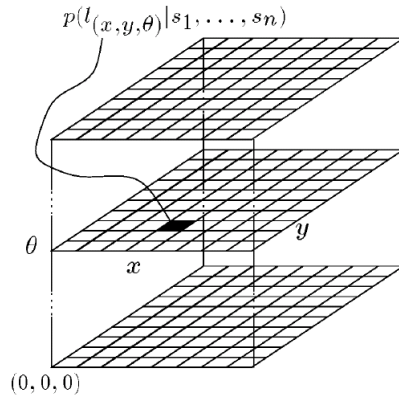
$$bel(x_1) = \eta p(z_1 | x_1, M) \overline{bel}(x_1). \quad (11.49)$$

Se puede verificar que se tiene  $\eta = 1/0,1875 \cong 5,33$  para hacer que el resultado final  $bel(x_1)$  sea normalizado a uno. La creencia final se muestra en la figura 11.21e.

**Mapas de red 3D.** En el caso de movimiento plano más general el mapa de red es una matriz tridimensional en la que cada celda contiene la probabilidad de que el robot esté en esa celda (figura 11.22). Durante cada uno de los pasos de predicción y medición, todas las células se actualizan. Si el número de células en el mapa es demasiado grande, el cálculo puede llegar a ser demasiado pesado para operaciones en tiempo real. La convolución en un espacio 3D es claramente el paso computacionalmente más costoso.

**Reducción de la complejidad computacional:** La idea básica, que se llama método de muestreo aleatorio, es conocida alternativamente como algoritmos de filtro de partículas, algoritmos de condensación y algoritmos de Monte Carlo. Independientemente de la técnica específica, el algoritmo básico es el mismo en todos estos casos. En lugar de representar cada posición posible del robot representando el estado creencia completa y correcta, se construye un estado de creencia aproximada representando sólo un subconjunto del conjunto completo de posibles localizaciones que deben ser consideradas.

Figura 11.22: La matriz 3D de representación del estado creencia utilizada en la localización de Markov



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.

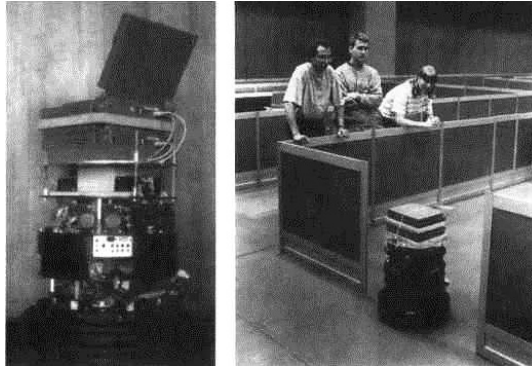
## Estudio de caso 2: Localización de Markov usando un mapa topológico

Una aplicación directa de la localización de Markov es posible cuando la representación del entorno del robot ya proporciona una descomposición adecuada. Este es el caso cuando la representación del entorno es puramente topológica. Dervish, empleó localización probabilística de Markov y utiliza un estado de creencia de múltiples hipótesis sobre una representación topológica del medio ambiente. Describimos ahora Dervish como un ejemplo de un robot con una representación discreta, topológica y un algoritmo de localización probabilística.

Dervish, que se muestra en la figura 11.23, incluye un arreglo de sonar de diseño personalizado para el Concurso Nacional de Robots AAI de 1994. El entorno en este concurso consistía en un espacio de oficina interior lineal interior lleno de muebles de oficina real como obstáculos. Los sonares tradicionales fueron dispuestos radialmente alrededor del robot en un anillo. Los robots con tales configuraciones de sensores están sujetos tanto a tropezarse con objetos cortos por debajo del anillo como a decapitarse por los objetos altos (tales como repisas, estantes y mesas) que están por encima del anillo.

Cuando el robot detecta una hendidura de 7-17 cm en la anchura del pasillo durante más de 1 segundo continuamente, un evento sensorial de puerta cerrada se activa. Si los picos de sonar saltan más allá de 17 cm durante más de 1 segundo, un evento sensorial de puertas abiertas se activa. La representación ambiental Dervish era un mapa topológico discreto. La figura 11.24 muestra una representación geométrica de un ambiente típico

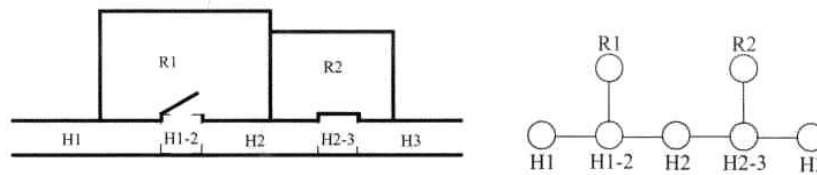
Figura 11.23: Dervish explorando su entorno.



Fuente:Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.

de oficina superpone con el mapa topológico para el entorno de oficina misma. Como se muestra en el lado izquierdo en la figura 11.24, Dervish utiliza una topología en la que los límites de nodo están marcados en primer lugar por las puertas (y los pasillos y vestíbulos).

Figura 11.24: Un entorno de oficina geométrica (izquierda) y su análogo topológico (derecha).



Fuente:Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.

Los arcos en este modelo son de longitud cero, mientras que los nodos tienen capacidad de expansión espacial y en conjunto cubren todo el espacio. Esta representación topológica en particular es especialmente apta para Dervish, dada su tarea de navegar por los pasillos en un ambiente específico y su capacidad perceptiva de reconocer discontinuidades en las paredes de los pasillos. Con el fin de representar un estado de creencia específica, Dervish se asocia con cada nodo topológico  $n$  una probabilidad de que el robot se encuentre en una posición física dentro de los límites de  $p(x_t = n)$ . Como se verá, la actualización probabilística utilizada por Dervish era aproximada, por

lo tanto, técnicamente hay que referirse a los valores resultantes como posibilidades en lugar de probabilidades.

Cuadro 11.1: Tabla de certeza de Dervish.

	Muro	Puerta cerrada	Puerta abierta	Pasillo Abierto	Vestíbulo
Nada detectado	0.70	0.40	0.05	0.001	0.30
Puerta cerrada detectada	0.30	0.60	0	0	0.05
Puerta abierta detectada	0	0	0.90	0.10	0.15
Pasillo abierto detectado	0	0	0.001	0.90	0.50

El proceso de actualización de percepción de Dervish funciona precisamente como en la ecuación (11.42). Dado un par específico perceptivo  $z$ , la ecuación (11.42) permite a la probabilidad de cada posible posición  $n$  para ser actualizado mediante la fórmula:

$$p(n | z) = \eta p(z | n) p(n) \quad (11.50)$$

El valor de  $p(n)$  ya está disponible en el estado actual creencia de Derviche, por lo que el reto está en la computación  $p(z | n)$ . La simplificación clave para Dervish se basa en la constatación de que, debido a que el sistema de extracción de características sólo extrae cuatro características totales y porque un nodo contiene (en un solo lado) una de las cinco características totales, cada combinación posibles de tipo de nodo y característica extraída puede ser representado en una tabla 4 x 5. La matriz certeza de Dervish (que se muestra en la tabla 11.6.7) es sólo esta tabla de búsqueda. Dervish hace la suposición simplificadora de que el rendimiento del detector de característica es sólo una función de la característica extraída y la función real en el nodo. Con esta suposición en la mano, se puede rellenar la matriz certeza con estimaciones de confianza para cada par posible de percepción y tipo de nodo.

Utilizando los valores específicos en el cuadro 11.6.7, si Dervish está al lado de un pasillo abierto, la probabilidad de error de reconocerlo como una puerta abierta es 0,10. Esto significa que para cualquier nodo  $n$  que es de tipo pasillo abierto y para el valor del sensor  $z = \text{puertaabierta}$ ,  $p(z | n) = 0,10$ . Junto con un mapa topológico específico, la matriz certeza permite el cálculo directo de  $p(z | n)$  durante el proceso de actualización de la percepción. Dervish no tiene ninguna etapa de actualización de predicción tal como se representa por la ecuación (11.41). Cuando el robot detecta un evento perceptual, múltiples pasos de actualización de percepción necesitan ser realizados para actualizar la probabilidad de cada posible posición del robot dado el estado creencia

anterior de Dervish. Formalmente, la fórmula de actualización percepción para Dervish es en realidad una combinación de la forma general de actualización de predicción y actualización de la medida. La probabilidad de la posición  $n$  dado un evento perceptual  $i$  se calcula como en la ecuación 11.51:

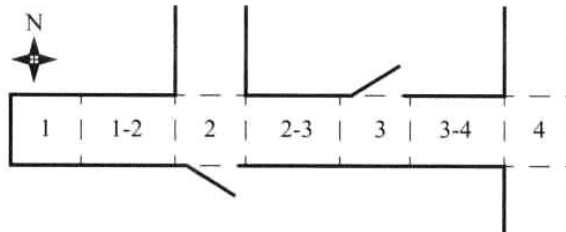
$$p(n_t | z_t) = \sum p(n_t | n'_{t-i}, z_t) p(n'_{t-i}) \quad (11.51)$$

El valor de  $p(n'_{t-i})$  denota la probabilidad de que Dervish esté en una posición  $n'$ , como es representado por el estado creencia previo de Dervish. El subíndice temporal  $t-i$  se utiliza en lugar de  $t-1$  porque para cada posición posible  $n'$  la distancia discreta topológica de  $n'$  a  $n$  puede variar en función del mapa topológico específico. El cálculo de  $p(n_t | n'_{t-i}, z_t)$  se realiza multiplicando la probabilidad de generar perceptual evento  $z$  en la posición  $n$  por la probabilidad de haber fallado al generar eventos de percepción en todos los nodos entre  $n'$  y  $n$ :

$$p(n_t | n'_{t-i}, z_t) = p(z_t, n_t) \cdot p(\emptyset, n_{t-1}) \cdot p(\emptyset, n_{t-2}) \cdot \dots \cdot p(\emptyset, n_{t-i+1}) \quad (11.52)$$

Por ejemplo (figura 11.25), se supone que el robot tiene sólo dos nodos distintos de cero en su estado de creencia,  $\{1-2, 2-3\}$ , con probabilidades asociadas con cada posición posible:  $p(1-2) = 1,0$  y  $p(2-3) = 0,2$ . Las probabilidades de los nodos 1-2 y 2-3 no suman 1.0. Estos valores no son probabilidades formales, y por tanto el esfuerzo computacional se reduce al mínimo en Dervish evitando la normalización del todo. Ahora bien el robot detecta un pasillo abierto a su izquierda y una puerta abierta a su derecha al mismo tiempo.

Figura 11.25: Un entorno realista topológico interior.



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.

El estado 2-3 progresará potencialmente a los estados 3, 3-4, y 4. Pero los estados 3 y 3-4 se pueden eliminar porque la probabilidad de detectar una puerta abierta cuando sólo hay pared es cero. La probabilidad de alcanzar el estado 4 es el producto de la

probabilidad inicial para el estado de 2-3, 0.2, la probabilidad de no detectar nada en el nodo 3, (a), y la probabilidad de detectar un pasillo de la izquierda y una puerta en la derecha en el nodo 4, (b). La probabilidad de no detectar nada en el nodo 3-4 es de 1.0.

(a) sólo se produce si Dervish no detecta la puerta a su izquierda en el nodo 3 (cerrado o abierto),  $[0,6 * 0,4(1 - 0,6) * 0,05]$ , y correctamente no detecta nada en su derecha, 0.7.

(b) si se produce Dervish identifica correctamente el pasillo abierto a su izquierda en el nodo 4, 0.90, y no detecta el pasillo a la derecha por una puerta abierta, 0.10.

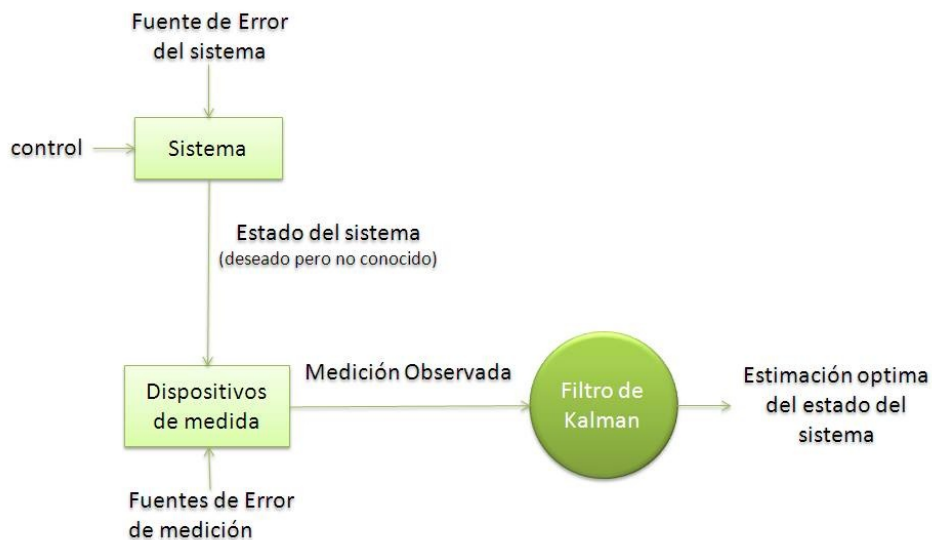
La fórmula final,  $0,2 * [0,6 * 0,4 + 0,4 * 0,05] * 0,7[0,9 * 0,1]$ , da una probabilidad de 0.003 para el estado 4. Este es un resultado parcial para  $p(4)$  siguiendo desde el nodo de estado de creencia previo de 2-3.

## 11.6.8. Localización del filtro de Kalman

### Introducción.

El modelo de localización de Markov puede representar cualquier función de densidad de probabilidad arbitraria sobre la posición del robot. Este enfoque es muy general, pero, debido a esto, ineficiente.

Figura 11.26: Típica aplicación del filtro de Kalman



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition

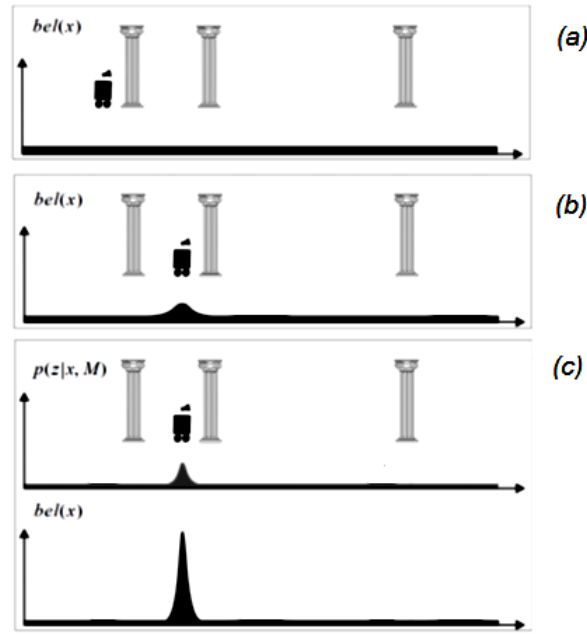
La figura 11.26 representa el esquema general de la estimación del filtro de Kalman, donde un sistema tiene una señal de control y fuentes de error del sistema como entradas. Un dispositivo de medición permite la medición de algunos estados del sistema con errores. El filtro de Kalman es un mecanismo matemático para producir una estimación óptima del estado del sistema basado en el conocimiento del sistema y el dispositivo de medición, la descripción del ruido del sistema y los errores de medición y de la incertidumbre en los modelos de dinámica. Así, el filtro de Kalman fusiona señales de los sensores y conocimiento del sistema de una manera óptima.

### **Ilustración de la localización del filtro de Kalman**

El algoritmo de localización del filtro de Kalman, o Localización KF, es un caso especial de la localización de Markov. En lugar de utilizar una función de densidad arbitraria, el filtro de Kalman utiliza Gaussianas para representar la creencia de robot  $bel(x_t)$ , el modelo de movimiento, y el modelo de medición. Las suposiciones realizadas por el filtro de Kalman limitan la elección de la creencia inicial  $bel(x_0)$  a una gaussiana, lo que significa que la ubicación inicial del robot debe ser conocida con una cierta aproximación. Por lo tanto, el robot no puede recuperar su posición si se pierde. El filtro de Kalman por tanto, aborda el problema de seguimiento de posición, pero no la localización global o el problema del robot secuestrado.

La figura 11.27 ilustra el algoritmo de localización del filtro de Kalman utilizando de nuevo el ejemplo de un robot móvil en un entorno de una sola dimensión. Como se muestra en la figura 11.27a, se supone que al principio el robot se encuentra cerca del primer pilar. A medida que el robot se mueve hacia la derecha, la incertidumbre aumenta como resultado de la convolución con el modelo de movimiento. La creencia resultante es por lo tanto un aumento Gaussiano de ancho incrementado, figura 11.27b. Ahora, el robot utiliza sus sensores exteroceptivos y mide que se encuentra cerca del segundo pilar. La posterior probabilidad  $p(z_t | x_t, M)$  de la observación se muestra en la figura 11.27c. Esta densidad de probabilidad es de nuevo una gaussiana. Con el fin de calcular la creencia actual de robot, hay que fusionar esta probabilidad de medición con la creencia del robot antes de la observación utilizando la regla de Bayes. El resultado de esta fusión es de nuevo una gaussiana mostrada en la parte inferior de la figura 11.27c. La varianza de la creencia resultante es más pequeña que las varianzas de tanto de la probabilidad de medición como de la creencia anterior del robot. Este resultado es obvio, ya que la fusión de las dos estimaciones independientes debe hacer el robot más certero que cada estimación individual.

Figura 11.27: Aplicación del algoritmo de filtro de Kalman para localización de robots móviles.



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.

### Introducción a la teoría del filtro de Kalman

Una distribución gaussiana está representada sólo por su primer y segundo momento, que son la media  $\mu_t$ , y la varianza  $\sigma_t^2$  (ecuación 11.20). Cuando la configuración del robot es un vector la distribución es una Gaussiana multivariante representada por un vector  $\mu_t$ , y una matriz de covarianza  $\Sigma_t$ , (ecuación 11.22). Durante las actualizaciones de predicción y medición sólo la media  $\mu_t$ , y la covarianza  $\Sigma_t$ , se actualizan. Por lo tanto, el filtro de Kalman se basa en cuatro ecuaciones: dos para actualizar  $\mu_t$  y  $\Sigma_t$ , en la actualización de predicción, y otras dos en la actualización de medida.

**Aplicando el teorema de la probabilidad total.** Sean  $x_1, x_2$  dos variables aleatorias que son independientes y normalmente distribuidas:

$$x_1 = N(\mu_1, \sigma_1^2) \quad (11.53)$$

$$x_2 = N(\mu_2, \sigma_2^2) \quad (11.54)$$

Que también y sea una función de  $x_1$  y  $x_2$ , es decir,

$$y = f(x_1, x_2) \quad (11.55)$$

Saber la distribución de  $y$  simple cuando  $f$  es una función lineal de las entradas, es decir, cuando:

$$y = Ax_1 + Bx_2 \quad (11.56)$$

En este caso, si las entradas son independientes, se puede demostrar que  $y$  es también una distribución normal con media y varianza dadas por las siguientes expresiones:

$$(y) = A\mu_1 + B\mu_2 \quad (11.57)$$

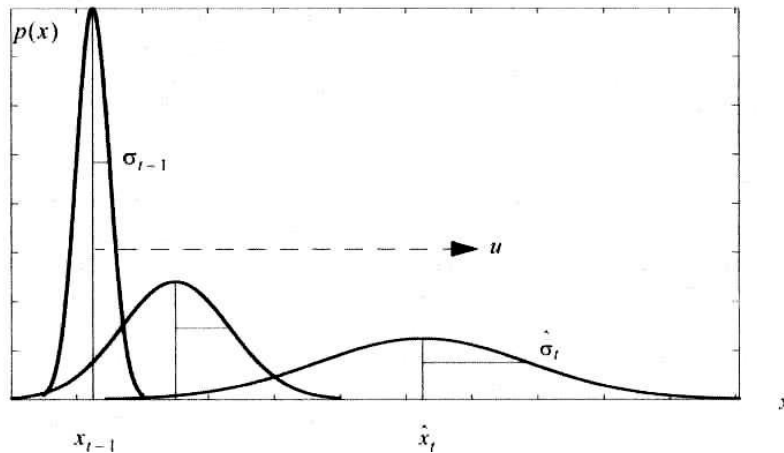
$$\sigma_y^2 = A^2\sigma_1^2 + B^2\sigma_2^2 \quad (11.58)$$

Si  $x_1$  y  $x_2$  son vectores con covarianzas  $\Sigma_1$  y  $\Sigma_2$  respectivamente, entonces:

$$(y) = A\mu_1 + B\mu_2 \quad (11.59)$$

$$\Sigma_y = A \Sigma_1 A^T + B \Sigma_2 B^T \quad (11.60)$$

Figura 11.28: Propagación de la densidad de probabilidad de un robot móvil.



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.

Este resultado se deduce directamente de la aplicación del teorema de la probabilidad total. En el caso  $f$  es no lineal, y no es normalmente distribuida. Sin embargo, es

una práctica común considerar una aproximación de primer orden linealizando  $f$  sobre  $(\mu_1, \mu_2)$ :

$$y \cong f(\mu_1, \mu_2) + F_{x_1}(x_1 - \mu_1) + F_{x_2}(x_2 - \mu_2) \quad (11.61)$$

Donde  $F_{x_1}$  y  $F_{x_2}$  son los jacobianos de  $f$ . De esta manera se obtiene:

$$(y) = f(\mu_1, \mu_2) \quad (11.62)$$

$$\sum_y = F_{x_1} \sum_1 F_{x_1}^T + F_{x_2} \sum_2 F_{x_2}^T \quad (11.63)$$

Las ecuaciones (11.62) y (11.63) se utilizarán en la sección (Aplicación a robots móviles: Localización del filtro de Kalman) para implementar la actualización de predicción de la localización del filtro extendido de Kalman (EKF). En la localización de Kalman  $f$  se utiliza para representar la actualización de la posición odométrica y sus entradas son la posición anterior del robot  $x_{t-1}$  y la entrada de control  $u$ . En el caso simple de un entorno de una dimensión, la actualización de la posición odométrica se describe mediante una simple suma, por lo tanto,  $f(x_{t-1}, u) = x_{t-1} + u$  y la actualización de la incertidumbre sobre el tiempo expresada en (11.63) se ilustra en la figura 11.28. La incertidumbre de la posición del robot después de la aplicación de (11.63) es mayor.

**Aplicando la regla de Bayes.** Sea  $q$  denotando la posición del robot,  $p_1(q)$  la creencia del robot como resultado de la actualización de predicción, y  $p_2(q)$  la creencia del robot como resultado de alguna medición de sensor exteroceptivo. Las densidades de probabilidad en el filtrado de Kalman se suponen que se distribuyen normalmente, por lo tanto:

$$p_1(q) = N(\hat{q}_1, \sigma_1^2) \quad (11.64)$$

$$p_2(q) = N(\hat{q}_2, \sigma_2^2) \quad (11.65)$$

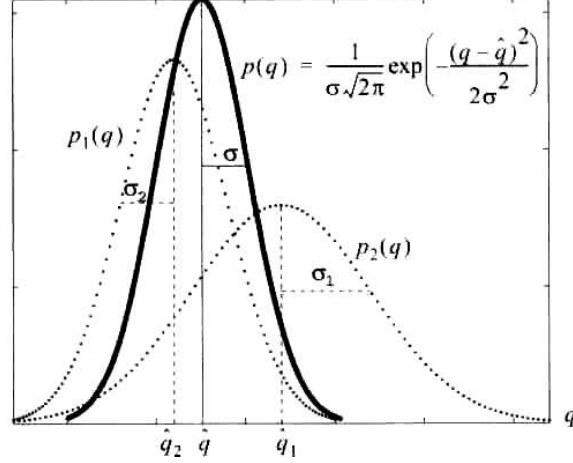
De acuerdo con la regla de Bayes, la distribución final  $p(q)$ , después de la medición, es proporcional al producto  $p_1(q) * p_2(q)$  (figura 11.29). A partir del producto de las dos funciones de densidad (11.64) y (11.65), se obtiene:

$$\frac{1}{\sigma_1 \sqrt{2\pi}} \exp\left(-\frac{(q - \hat{q}_1)^2}{2\sigma_1^2}\right) \cdot \frac{1}{\sigma_2 \sqrt{2\pi}} \exp\left(-\frac{(q - \hat{q}_2)^2}{2\sigma_2^2}\right) = \frac{1}{\sigma_1 \sigma_2 \sqrt{2\pi}} \exp\left(-\frac{(q - \hat{q}_1)^2}{2\sigma_1^2} - \frac{(q - \hat{q}_2)^2}{2\sigma_2^2}\right) \quad (11.66)$$

Como se puede ver, el argumento de este exponencial es cuadrática en  $q$ , por lo tanto,  $p(q)$  es una gaussiana. Ahora se tiene que determinar su valor  $\hat{q}$  media y la varianza  $\sigma$  de una que nos permiten volver a escribir (11.66) en la forma:

$$\Omega \exp\left(-\frac{(q - \hat{q})^2}{2\sigma^2}\right) \quad (11.67)$$

Figura 11.29: La fusión de la densidad de probabilidad de dos estimaciones.



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.

Reordenando la exponencial en (11.66):

$$\begin{aligned}
 & \exp\left(-\frac{(q - \hat{q}_1)^2}{2\sigma_1^2} - \frac{(q - \hat{q}_2)^2}{2\sigma_2^2}\right) \\
 = & \exp\left(-\frac{1}{2} \left( \frac{q^2(\sigma_1^2 + \sigma_2^2) - 2q(\hat{q}_1\sigma_2^2 + \hat{q}_2\sigma_1^2) + (\hat{q}_1^2\sigma_2^2 + \hat{q}_2^2\sigma_1^2)}{\sigma_1^2\sigma_2^2} \right)\right) \\
 = & \exp\left(-\frac{1}{2} \left( \frac{q^2 - 2q(\frac{\hat{q}_1\sigma_2^2 + \hat{q}_2\sigma_1^2}{\sigma_1^2 + \sigma_2^2}) + \frac{\hat{q}_1^2\sigma_2^2 + \hat{q}_2^2\sigma_1^2}{\sigma_1^2 + \sigma_2^2}}{\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2}} \right)\right) \\
 = & \exp\left(-\frac{1}{2} \frac{\left(q - \frac{\hat{q}_1^2\sigma_2^2 + \hat{q}_2^2\sigma_1^2}{\sigma_1^2 + \sigma_2^2}\right)^2}{\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2}}\right) \cdot \exp\left(-\frac{1}{2} \frac{\frac{\hat{q}_1^2\sigma_2^2 + \hat{q}_2^2\sigma_1^2}{\sigma_1^2 + \sigma_2^2} - \left(\frac{\hat{q}_1^2\sigma_2^2 + \hat{q}_2^2\sigma_1^2}{\sigma_1^2 + \sigma_2^2}\right)^2}{\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2}}\right) \quad (11.68)
 \end{aligned}$$

Se puede notar que el segundo término de este producto depende sólo de  $\hat{q}_1$  y  $\hat{q}_2$  y, por lo tanto, es constante. Por lo tanto, podemos reescribir (11.68) como:

$$\Omega \exp\left(-\frac{1}{2} \frac{\left(q - \frac{\hat{q}_1^2\sigma_2^2 + \hat{q}_2^2\sigma_1^2}{\sigma_1^2 + \sigma_2^2}\right)^2}{\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2}}\right) = \Omega \exp\left(-\frac{(q - \hat{q})^2}{2\sigma^2}\right) \quad (11.69)$$

Dónde:

$$\hat{q} = \frac{\hat{q}_1^2 \sigma_2^2 + \hat{q}_2^2 \sigma_1^2}{\sigma_1^2 + \sigma_2^2}, \quad \text{o alternativamente} \quad \hat{q} = \frac{\frac{1}{\sigma_1^2} \hat{q}_1 + \frac{1}{\sigma_2^2} \hat{q}_2}{\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}}, \quad (11.70)$$

Y

$$\sigma^2 = \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}, \quad \text{o alternativamente} \quad \frac{1}{\sigma^2} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} = \frac{\sigma_1^2 + \sigma_2^2}{\sigma_1^2 \sigma_2^2} \quad (11.71)$$

Se puede observar que 11.70 y 11.71 pueden también ser escritas como:

$$\hat{q} = \hat{q}_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} (\hat{q}_2 - \hat{q}_1) \quad (11.72)$$

$$\sigma^2 = \sigma_1^2 - \frac{\sigma_1^4}{\sigma_1^2 + \sigma_2^2} \quad (11.73)$$

Estas dos últimas expresiones serán valiosas en la aplicación del filtro de Kalman. En el filtrado de Kalman el factor  $\frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}$  se denomina comúnmente ganancia de Kalman. De la ecuación (11.73) se puede ver claramente que la varianza resultante  $\sigma^2$  es menor que tanto  $\sigma_1^2$  como  $\sigma_2^2$ . Por lo tanto, la incertidumbre de la estimación de la posición se ha reducido mediante la combinación de las dos medidas, es decir, la anterior creencia del robot y la medición del sensor exteroceptivo. Este es un resultado que se espera sobre la base de la teoría de la información, la curva sólida de probabilidad de densidad en la figura 11.31 representa el resultado de la fusión operado por el filtro de Kalman.

Las (11.72) y (11.73) son válidas sólo para el caso unidimensional. Para vectores n-dimensionales, la media final  $\hat{q}$  y la covarianza  $\hat{P}$  después de la fusión se puede escribir, respectivamente, como:

$$\hat{q} = q_1 + P(P + R)^{-1}(q_2 - q_1) \quad (11.74)$$

$$\hat{P} = P - P(P + R)^{-1}P \quad (11.75)$$

Donde  $P$  y  $R$  son las covarianzas de  $q_1$  y  $q_2$  respectivamente.

Las ecuaciones (11.74) y (11.75) se utilizarán en la sección (Aplicación a robots móviles: Localización del filtro de Kalman) para aplicar luego actualización de medida de la localización EKF. En el filtrado de Kalman estas ecuaciones se escriben normalmente como:

$$\hat{q} = q_1 + K(q_2 - q_1) \quad (11.76)$$

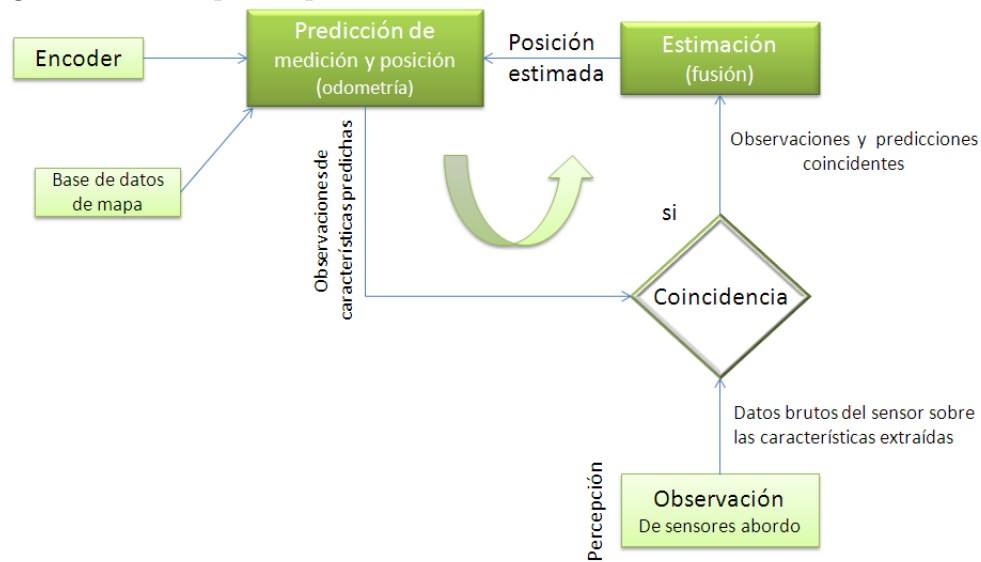
$$\hat{P} = P - K * \sum_{IN} K^T \quad (11.77)$$

Donde  $K = P(P + R)^{-1}$  es la ganancia de Kalman,  $(q_2 - q_1)$  es la innovación y  $\sum_{IN} = (P + R)$  es la covarianza de innovación.

### Aplicación a robots móviles: Localización del filtro de Kalman.

La aplicación del filtro de Kalman para localización requiere presentar el problema de localización de robot como un problema de fusión de sensores. La actualización probabilística básica del estado de creencia del robot puede ser segmentada en dos fases, actualización de predicción y actualización de medición. La diferencia fundamental entre el enfoque del filtro de Kalman y el enfoque anterior de localización de Markov se encuentra en el proceso de actualización de medida. En la localización de Markov, la percepción entera, se utiliza para actualizar cada posible posición del robot en el estado creencia individualmente. Por el contrario, la actualización de la medida usando un filtro de Kalman es un proceso de múltiples pasos.

Figura 11.30: Esquema para la localización del robot móvil de filtro Kalman.



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.

La figura 11.30 representa el esquema particular para la localización de filtro de Kalman. La actualización de medida, como se acaba de mencionar, se compone de

múltiples etapas que se resumen aquí:

1. En la etapa de la observación, el robot recoge datos reales de sensores y características apropiadas extraídas (por ejemplo, líneas, puertas, o incluso el valor de un sensor específico).
2. Al mismo tiempo, sobre la base de su posición predicha en el mapa, el robot genera una predicción de medición que consiste en las características que el robot espera observar a partir de la posición en la que se piensa que está (por ejemplo, la posición estimada en el paso de predicción).
3. En la etapa de coincidencia el robot calcula el mejor ajuste entre las características extraídas durante la observación y las características esperadas seleccionadas durante la predicción de medición.
4. Finalmente, el filtro de Kalman fusiona la información proporcionada por todas estas coincidencias para actualizar el estado de creencia robot en la etapa de estimación.

**Actualización de predicción: Aplicando el teorema de la probabilidad total.** La posición del robot  $\hat{x}_t$  en un paso de tiempo  $t$  se predice en base a su ubicación anterior en el tiempo  $t - 1$  y su movimiento debido al control de entrada  $u_t$ :

$$\hat{x}_t = f(x_{t-1}, u_t) \quad (11.78)$$

Para el robot diferencial,  $f(x_{t-1}, u_t)$  es dada por (11.7), la cual describe la estimación de la posición odométrica. Conociendo el modelo de planta y de error, también puede calcular la varianza  $P_{t-1}$  asociado con esta predicción utilizando la ecuación derivada del teorema de la probabilidad total aplicado a las distribuciones gaussianas (ecuación 11.63):

$$\hat{P}_t = F_x * P_{t-1} * F_x^T + F_u * Q_t * F_u^T \quad (11.79)$$

Donde  $P_{t-1}$  es la covarianza del estado previo del robot  $x(t - 1)$  y  $Q_t$  es la covarianza del ruido asociado al modelo de movimiento. Esta ecuación es la aplicación de la ley de propagación de errores. Las ecuaciones (11.78) y (11.79) son las dos ecuaciones fundamentales de la actualización de predicción en la localización EKF. Nos permiten predecir la posición del robot y su incertidumbre después de un movimiento especificado por la entrada de control  $u_t$ .

**Actualización de medición.** Como se ha dicho antes, esta fase consta de cuatro pasos:

1. **Observación.** El primer paso consiste en obtener las mediciones del sensor  $z_t$  desde el robot en el tiempo  $t$ . En general, la observación  $z_t$  consta de un conjunto de  $n$  observaciones individuales  $z_t^i (i = 0 \dots n)$  extraídas del sensor. Se necesitan representar las observaciones y las predicciones de medición en el mismo marco  $\{S\}$ . Se van a transformar las predicciones de medición desde el marco de coordenadas global  $\{W\}$  hacia el sensor de fotograma  $\{S\}$ .
2. **Predicción de medición.** Se usa la posición predicha el robot  $x_t$  y el mapa  $M$  para generar múltiples observaciones de características predichas  $\hat{z}_t^j$ , las observaciones previstas son lo que el robot espera ver si estaba en esa posición particular. Con el fin de calcular la observación predicha, el robot debe transformar todas las características  $m^j$  en el mapa  $M$  en el marco local de coordenadas del sensor. Si se define la transformación de la función  $j$  a través de la función  $h^j$ , entonces se puede escribir:

$$\hat{z}_t^j = h^j(\hat{x}_t, m^j) \quad (11.80)$$

La cual obviamente depende de la posición de cada característica en el mapa (representada por  $m^j$ ) y la posición actual del robot  $\hat{x}_t$ .

3. **Coincidencia (Matching).** Formalmente, el objetivo del procedimiento de coincidencia es producir una asignación de la observación  $z_t^i$  para las observaciones predichas  $\hat{z}_t^j$ . Para cada predicción de medición para la cual se encuentra una observación correspondiente, se calcula la innovación  $v_t^{ij}$ . La innovación es una medida de la diferencia entre la medición predicha y lo observado:

$$v_t^{ij} = [z_t^i - \hat{z}_t^j] = [z_t^i - h^j(\hat{x}_t, m^j)] \quad (11.81)$$

La covarianza de innovación  $\sum_{IN_t}^{ij}$  se puede encontrar mediante la aplicación de la ley de propagación, sección *Error de propagación* del capítulo anterior:

$$\sum_{IN_t}^{ij} = H^j * \hat{P}_t * H^{jT} * R_t^i \quad (11.82)$$

Donde  $H^j$  es el jacobiano de  $h^j$  y  $R_t^i$  representa la covarianza (ruido) de la observación real  $z_t^i$ . Para determinar la validez de la correspondencia entre la predicción de medición y observación, una puerta de validación  $g$  tiene que ser especificada. Una posible opción para la puerta de validación es la distancia de Mahalanobis:

$$v_t^{ijT} * \left( \sum_{IN_t}^{ij} \right)^{-1} * v_t^{ij} \leq g^2 \quad (11.83)$$

4. **Estimación.** Aplicando la regla de Bayes. En este paso, se calcula la mejor estimación  $x_t$  de la posición del robot basada en la predicción de posición  $\hat{x}_t$  y todas las observaciones  $z_t^i$  en el tiempo  $t$ . Para hacer esta actualización de posición, primero se apilan las observaciones validadas  $z_t^i$  en un único vector para formar  $z_t$  y designar la innovación compuesta  $v_t$ . Luego, se apilan los jacobianos de medición  $H^j$  para cada medición validada juntos para formar el Jacobiano compuesto  $H$  y el vector de error (ruido) de medición  $R_t = \text{diag} [R_t^i]$ . De estos, se puede entonces calcular la covarianza de innovación compuesta  $\sum_{IN_t}$ , utilizando la ecuación (11.82). Finalmente, usando los resultados de la aplicación de la regla de Bayes para distribuciones gaussianas, ecuaciones (11.74) y (11.75), se puede actualizar la estimación de posición del robot  $x_t$  y su covarianza asociada  $P_t$  como:

$$x_t = \hat{x}_t + K_t v_t \quad (11.84)$$

$$P_t = \hat{P}_t - K_t * \sum_{IN_t} * K_t^T \quad (11.85)$$

Dónde:

$$K_t = \hat{P}_t * H_t^T * \left( \sum_{IN_t} \right)^{-1} \quad (11.86)$$

Es la ganancia de Kalman. Mediante la imposición de igual a la matriz identidad  $H$ , la ecuación (11.84) se simplifica a

$$x_t = \hat{x}_t + \hat{P}_t (\hat{P}_t + R_t)^{-1} (z_t - \hat{x}_t) \quad (11.87)$$

$$P_t = \hat{P}_t - \hat{P}_t (\hat{P}_t + R_t)^{-1} \hat{P}_t \quad (11.88)$$

Que corresponden respectivamente a (11.74) y (11.75). La ecuación (11.84), dice que la mejor estimación  $x_t$  del estado robot en el tiempo  $t$  es igual a la mejor predicción del valor  $\hat{x}_t$  antes de que la nueva medición  $z_t$  se tome, más un término de corrección de un valor de ponderación óptima  $K_t$  la diferencia entre tiempos  $z_t$  y la mejor predicción  $\hat{z}_t$  en el tiempo  $t$ .

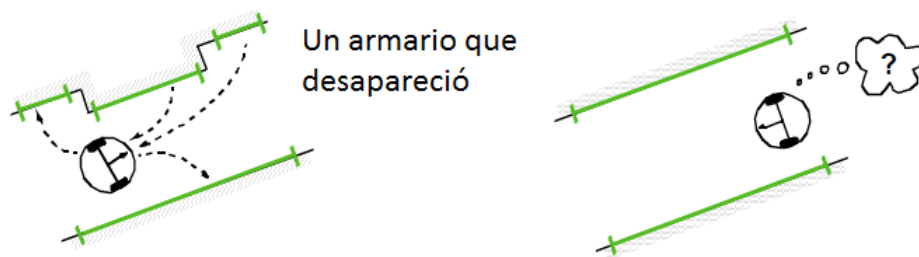
## 11.7. CONSTRUCCIÓN AUTÓNOMA DE MAPAS

### 11.7.1. Generalidades

Un robot debe ser capaz de detectar cambios en el mapa debido a la reordenación de los muebles (Figura 11.31). La alternativa a la construcción de mapas hechos a mano

es por lo tanto “la construcción automática del mapa” de hecho, un robot que localice exitosamente tiene los sensores adecuados para detectar el medio ambiente, por lo que el robot debe construir su propio mapa. Esta ambición va al corazón de la robótica móvil autónoma. En prosa, partiendo de un punto inicial arbitrario, el robot móvil debe ser capaz de explorar el entorno de forma autónoma con sus sensores a bordo, obtener conocimiento sobre él, interpretar la escena, construir un mapa adecuado y localizarse sí mismo en relación con este mapa.

Figura 11.31: Un robot autónomo debería ser capaz de seguir cambios en los ambientes para la localización.



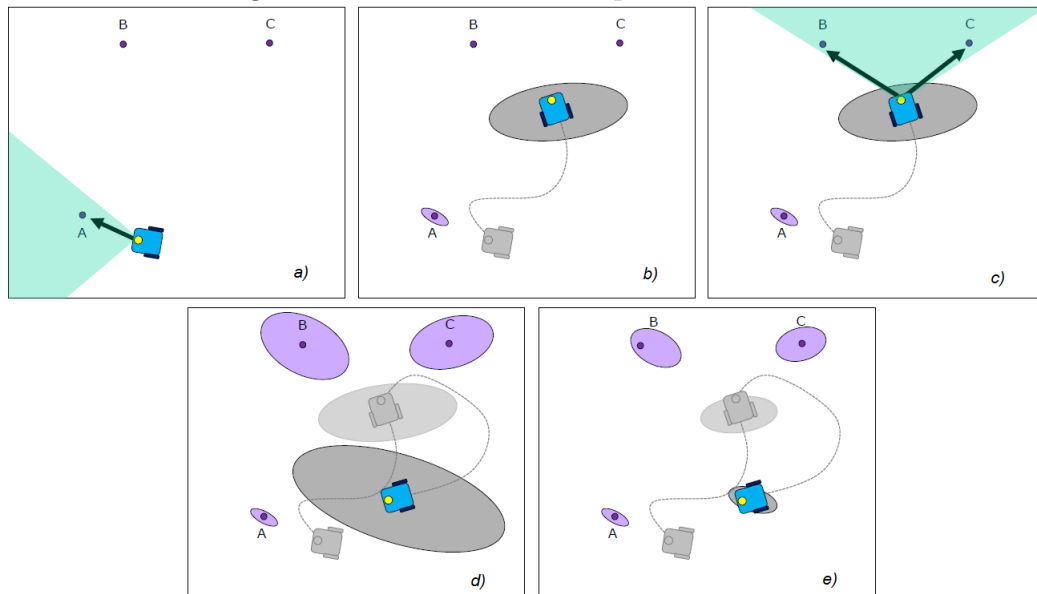
Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.

### 11.7.2. SLAM: La localización simultánea y el problema de asignación

El objetivo del SLAM es tanto recuperar la trayectoria del robot como el mapa de entorno utilizando sólo los datos recogidos por los sensores propioceptivos y exteroceptivos. SLAM es difícil porque tanto la trayectoria estimada y las características extraídas están corrompidas por ruido. El problema se ilustra en la figura 11.32. Se supone que la incertidumbre del robot en su posición inicial es cero. Desde esta posición, el robot observa una característica la cual es mapeada con una incertidumbre relacionada con el modelo de error del sensor exteroceptivo (a). A medida que el robot se mueve, su incertidumbre de posición aumenta bajo el efecto de los errores introducidos por la odometría (b). En este punto, el robot observa dos características y las mapea con una incertidumbre que resulta de la combinación del error de medición con la incertidumbre de posición del robot (c). De esto, se puede notar que el mapa se correlaciona con la estimación de posición del robot. Con el fin de reducir su incertidumbre, el robot debe

observar las características cuya ubicación es relativamente bien conocida. Estas características pueden ser, por ejemplo, puntos de referencia que el robot ya ha observado antes. En este caso, la observación se llama detección de cierre de lazo (bucle). Cuando un cierre de lazo es detectado, la incertidumbre de posición del robot se reduce. Al mismo tiempo, el mapa se actualiza y se plantea la incertidumbre de otras características observadas y todas las posiciones previas del robot también se reducen (figura 11.32e)

Figura 11.32: Ilustración del problema SLAM.



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.

### 11.7.3. Definición matemática de SLAM

La terminología utilizada en esta sección es la misma introducida en las secciones 11.6.4 y 11.6.5 para localización probabilística basada en mapas. Se definió la posición del robot en el tiempo  $t$  como  $x_t$ . La trayectoria del robot es dada por:

$$X_T = \{x_0, x_1, x_2, \dots, x_T\} \quad (11.89)$$

Donde  $T$  también podría ser infinito. En SLAM, la ubicación inicial del robot  $x_0$  se supone que se conoce, mientras que las otras ubicaciones no lo son. El término  $u_T$  denota el movimiento del robot entre el tiempo  $t - 1$  y el tiempo  $t$ . La secuencia de los

movimientos relativos del robot puede entonces ser escrita como:

$$U_T = \{u_0, u_1, u_2, \dots, u_T\} \quad (11.90)$$

Sea  $M$  lo que denota el verdadero mapa del medio ambiente.

$$M = \{m_0, m_1, m_2, \dots, m_{(n-1)}\} \quad (11.91)$$

Entonces  $m_i, i = 0 \dots n-1$ , son vectores que representan las posiciones de los puntos de referencia que, de nuevo, podrían ser puntos, líneas, planos, o cualquier tipo de característica de alto nivel (por ejemplo, puertas). Esta Comunicación, para simplificar, se supone que el mapa es estático. Por último, si se supone que el robot toma una medición en cada momento, podemos denotar

$$Z_T = \{z_0, z_1, z_2, \dots, z_T\} \quad (11.92)$$

De acuerdo con esta terminología, se puede definir SLAM como el problema de recuperar un modelo del mapa  $M$  y la trayectoria del robot  $X_T$  de la odometría  $U_T$  y las observaciones  $Z_T$ . En la literatura, se distinguen entre el problema de SLAM completo y el problema de SLAM en línea. El problema de SLAM completo consiste en estimar la probabilidad posterior de conjunto sobre  $X_T$  y  $M$  a partir de los datos, es decir:

$$p(X_T, M \mid Z_T, U_T) \quad (11.93)$$

El problema SLAM en línea, por el contrario, consiste en estimar el conjunto posterior sobre  $x_T$  y  $M$  a partir de los datos, es decir:

$$p(x_t, M \mid Z_T, U_T) \quad (11.94)$$

Por lo tanto, el problema de SLAM completo intenta recuperar toda la trayectoria del robot  $X_T$ , mientras que el problema de SLAM en línea trata de estimar sólo la posición actual del robot  $x_T$ . Con el fin de resolver el problema de SLAM, se necesita saber el modelo de movimiento probabilístico y modelo de medición probabilístico. Estos modelos se han introducido en la sección 11.6.5. En particular:

$$p(x_t \mid x_{t-1}, u_T) \quad (11.95)$$

Representa la probabilidad de que la posición del robot es  $x_t$  dada la posición anterior del robot  $x_{t-1}$ , y los datos propioceptivos (o entrada de control)  $u_t$ . Del mismo modo:

$$p(z_t \mid x_t, M) \quad (11.96)$$

Es la probabilidad de medir  $z_t$  dado el  $M$  mapa conocido y suponiendo que el robot toma la observación en la ubicación  $x_t$ .

En las siguientes secciones, se describen los tres principales paradigmas para resolver el problema de SLAM, que son EKF SLAM, gráfico basados filtro SLAM y SLAM partículas.

#### 11.7.4. SLAM del Filtro de Kalman Extendido (EKF)

La SLAM del EKF procede exactamente igual que el EKF estándar que hemos visto para la localización del robot (sección 11.6.8), con la única diferencia de que utiliza un vector de estado extendido  $y_t$  el cual comprende tanto la posición del robot  $x_t$  y la posición de todas las características  $m_i$  en el mapa, es decir:

$$y_t = [x_t, m_0, \dots, m_{n-1}]^T \quad (11.97)$$

En el ejemplo de localización basada en funciones de línea, la dimensión de  $y_t$  sería  $3 + 2n$ , ya que se necesitan tres variables para representar la pose del robot  $(x, y, \theta)$  y  $2n$  variables para los  $n$  puntos de referencia de línea que tienen componentes de vectores  $(\alpha^i, r^i)$ . Por lo tanto, el vector de estado se escribiría como:

$$y_t = [x_t, y_t, \theta_t, \alpha_0, r_0, \alpha_1, r_1, \dots, \alpha_{n-1}, r_{n-1}]^T \quad (11.98)$$

A medida que el robot se mueve y realiza las mediciones, el vector de estado y la matriz de covarianza se actualizan utilizando las ecuaciones estándar del filtro de Kalman extendido. Los mapas en el SLAM de EKF se supone que son basados en características (es decir, puntos, líneas, planos). A medida que nuevas características se observan, se añaden al vector de estado. Por lo tanto, la matriz de covarianza de ruido crece cuadráticamente, con tamaño de  $(3 + 2n) \times (3 + 2n)$ . Por razones de computación, el tamaño del mapa se limita generalmente a menos de un millar de características.

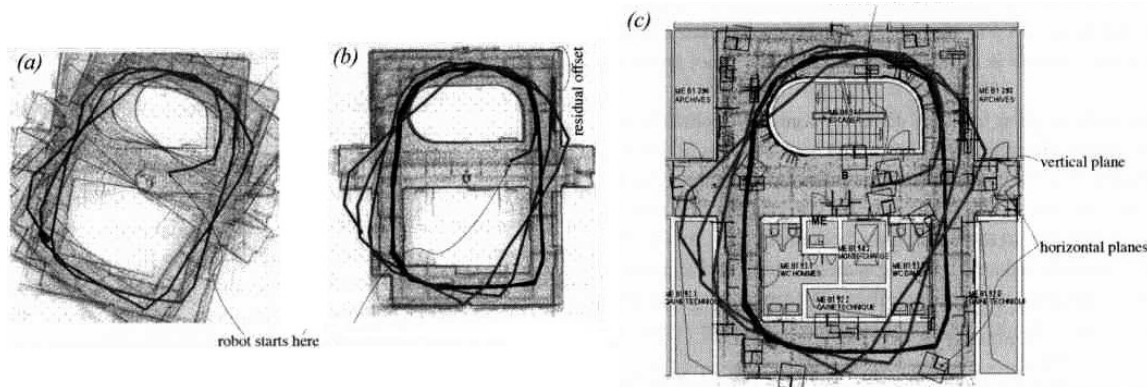
La aplicación de la SLAM de EKF es más que la aplicación directa de las ecuaciones de SLAM de EKF para el problema en línea, es decir, las ecuaciones (11.78) - (11.79) y (11.84) - (11.85). Para esto, se tienen que especificar las funciones que caracterizan el modelo de predicción y de medición. El modelo de predicción, tiene que tener en cuenta que el movimiento sólo se actualizará la posición del robot, mientras que las características permanecen sin cambios. Por lo tanto, se puede escribir el modelo de

predicción de la SLAM de EKF como:

$$\hat{y}_t = y_t + \begin{bmatrix} \frac{\Delta s_r + \Delta s_l}{2} \cos \left( \theta_{t-1} + \frac{\Delta s_r + \Delta s_l}{2b} \right) \\ \frac{\Delta s_r + \Delta s_l}{2} \sin \left( \theta_{t-1} + \frac{\Delta s_r + \Delta s_l}{2b} \right) \\ \frac{\Delta s_r - \Delta s_l}{b} \\ 0 \\ 0 \\ \dots \\ 0 \\ 0 \end{bmatrix} \quad (11.99)$$

La SLAM de EKF se ha aplicado con éxito en muchos campos diferentes, incluyendo ambientes en el aire, bajo el agua, al aire libre y bajo techo. La figura 11.33 muestra los resultados de una SLAM de 6 DOF usando un telémetro láser 3D. El robot comienza en el centro y hace tres rondas. La figura 11.33a muestra el mapa resultante usando sólo odometría. Como se puede ver, el mapa es inconsistente debido a la desvió de odometría acumulado. En (b), el error de odometría acumulado se reduce drásticamente por el uso de técnicas de coincidencias de escaneo y de alineación. Por último, en (c), la desviación acumulada y el error de desplazamiento ya no están presentes después de la aplicación de SLAM de EKF. **Ver en la página 644 del Anexo C taller en Matlab.**

Figura 11.33: SLAM de EKF con escáner láser 3D.



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.

### 11.7.5. SLAM Visual con una sola cámara

El término SLAM Visual (SLAM-V) fue inventado en 2003 por Davison [“Real time simultaneous localization and mapping with a single camera”,2003], quien presentó el

primer sistema SLAM de EKF en tiempo real con una sola cámara de mano. La V-SLAM puede ser vista como una estructura-de-Movimiento multivista (SfM Structure from motion). La principal diferencia entre V-SLAM y SfM es que V-SLAM tiene en cuenta la incertidumbre de característica utilizando un marco probabilístico. Otra diferencia es que V-SLAM necesita procesar las imágenes cronológicamente mientras que el SfM trabaja para conjuntos de datos desordenados. La V-SLAM también se conoce como SLAM de sólo orientación, y hace hincapié en el hecho de que utiliza sólo observaciones de ángulo.

El primer problema en V-SLAM monocular es la estimación de la posición de las características en el momento en que se inicia el sistema. Usando un telémetro, obviamente esto no es un problema, pero para V-SLAM esto no es posible. Para superar este problema, en su implementación original Davison utiliza un patrón plano de la geometría conocida donde se conoce la posición relativa de al menos cuatro esquinas fronterizas. Desde las cuatro esquinas de posición conocida, la posición de la cámara de 6 DOF con respecto a estos puntos se puede determinar de forma única. Mientras la cámara se desplaza en frente del patrón, la posición de la cámara puede estimarse a partir de imágenes individuales. A medida que la cámara comienza a alejarse del patrón, se deben triangular y agregar nuevas características al mapa. En este punto, es que el proceso V-SLAM EKF comienza.

Al igual que se hizo para el estándar EKF SLAM, el vector de estado  $y$  contiene tanto la posición de la cámara y la posición de característica, pero esta vez también la velocidad de la cámara. La dimensión del vector de estado en la V-SLAM EKF es  $13+3n$ , de hecho, se necesitan tres parámetros para la posición  $r$ , cuatro para la orientación quaternion  $q$ , tres para la velocidad de traslación  $v$ , otros tres para la velocidad angular  $w$ , y  $3n$  para las posición de característica  $m_i$ . El vector de estado en el tiempo  $t$  puede ser escrito como:

$$y_t = [x_t, m_0, m_1, \dots, m_{(n-1)}]^T \quad (11.100)$$

Donde

$$x_t = [r_t, q_t, v_t, w_t]^T \quad (11.101)$$

**Etapas de predicción.** En V-SLAM no se usa odometría para predecir la siguiente posición de la cámara. Para superar este problema, Davison propuso el uso de un modelo de velocidad constante. Esto significa que entre cuadros consecutivos la velocidad se asume que es constante, y por lo tanto la posición de la cámara en el tiempo  $t$  se calcula mediante la integración del movimiento de partida en el tiempo  $t - 1$ , suponiendo que la velocidad inicial es la estimada en el momento  $t - 1$ . Teniendo esto en mente, en

realidad se puede escribir la función de predicción de movimiento  $f$  como:

$$\hat{x}_t = f(x_{t-1}, u_t) = \begin{bmatrix} r + (v + V)\Delta t \\ q \times q((w + \Omega)\Delta t) \\ v + V \\ w + \Omega \end{bmatrix} \quad (11.102)$$

Donde las intenciones desconocidas (en términos de velocidad y aceleración) del portador de la cámara se tienen en cuenta en el modelo de velocidad constante como  $V$  y  $\Omega$ , que se calcula como:

$$V = a\Delta t \quad y \quad \Omega = \alpha\Delta t \quad (11.103)$$

Donde  $a$  y  $\alpha$  son las aceleraciones desconocidas de traslación y angular que se modelan como distribuciones gaussianas medias cero. La ecuación de actualización de predicción del EKF por lo tanto, se puede escribir como:

$$\begin{bmatrix} \hat{x}_t \\ \hat{\alpha}_t^0 \\ \hat{r}_t^0 \\ \dots \\ \hat{\alpha}_t^{n-1} \\ \hat{r}_t^{n-1} \end{bmatrix} = \begin{bmatrix} f(x_{t-1}, u_t) \\ \alpha_{t-1}^0 \\ r_{t-1}^0 \\ \dots \\ \alpha_{t-1}^{n-1} \\ r_{t-1}^{n-1} \end{bmatrix} \quad (11.104)$$

Donde  $(\hat{\alpha}_t^i, \hat{r}_t^i)$  y  $(\alpha_{t-1}^i, r_{t-1}^i)$  denotan la posición de la  $i$ -ésima característica a los tiempos  $t$  y  $t - 1$  respectivamente

**Actualización de medición.** En la actualización de medida, la posición de la cámara se corrige basada en la reobservación de características. Además, las nuevas características se inicializan y se añaden al mapa. En V-SLAM, las características son puntos de interés (figura 11.34) extraídos usando uno de los detectores de punto de interés descritos en el capítulo de percepción. Por lo tanto, las características se expresan en coordenadas de píxel de imagen.

Los pasos principales de V-SLAM de Davison se ilustran en la figura 11.35 donde: (a) La cámara comienza a moverse con seis características conocidas en un patrón. b) características desconocidas cercanas se inicializan y se añaden al mapa. c) A medida que la cámara se mueve, la incertidumbre de las características estimadas en el mapa aumenta, d) Dado que la cámara vuelve a visitar algunas de las características que se observan en un principio, se reduce su incertidumbre.

Figura 11.34: Parches de características de imagen y regiones de búsqueda predicha a partir del marco anterior utilizando un modelo de velocidad constante movimiento.



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.

### 11.7.6. Discusión sobre EKF SLAM

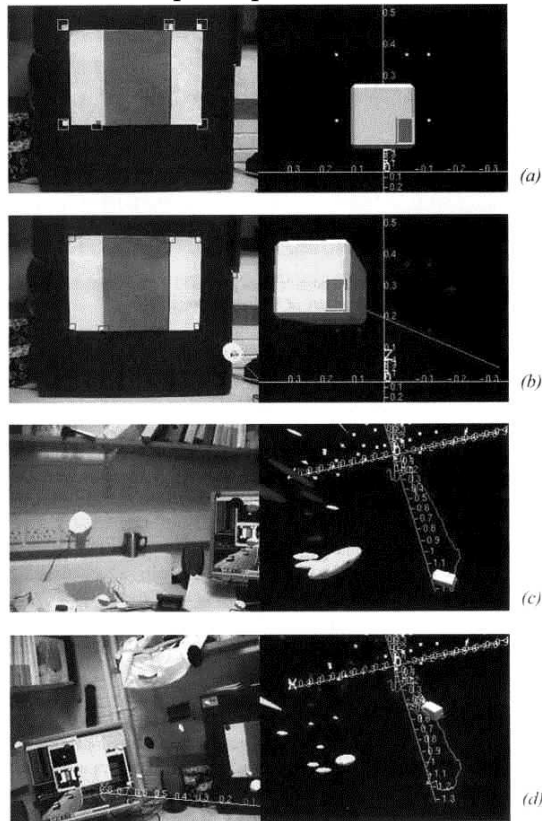
La SLAM de EKF no es más que la aplicación de la filtro de Kalman extendido con un estado conjunto compuesto por la posición del robot y las ubicaciones de características. En cada iteración tanto el estado y la matriz de covarianza conjunta se actualizan, lo que significa que el cálculo crece cuadráticamente con el número de características. Con el fin de superar estas limitaciones, se han propuesto implementaciones en tiempo real eficientes de EKF SLAM en los últimos años, lo que puede hacer frente a miles de características. La idea principal consiste en descomponer el mapa en pequeños submapas para que las covarianzas se actualicen por separado.

### 11.7.7. SLAM basada en Gráficos

La SLAM basada en gráfico nace de la intuición de que el problema SLAM puede ser interpretado como un gráfico escaso de nodos y restricciones entre nodos. Los nodos del gráfico son las posiciones del robot  $x_0, x_1, \dots, x_T$  y las  $n$  características en el mapa  $m_0, m_1, \dots, m_{n-1}$ . Las restricciones son la posición relativa entre el posiciones consecutivas del robot  $x_{t-1}, x_t$ , (dado por la entrada de odometría  $u_t$ ) y la posición relativa entre las ubicaciones del robot y las características observadas de esos ubicaciones.

La propiedad clave que hay que recordar acerca de la SLAM basada en gráficos es que las restricciones no deben ser consideradas como limitaciones rígidas, sino como restricciones blandas (figura 11.36). La SLAM basada en gráfico representa ubicaciones del robot y características como nodos de una red elástica. La solución SLAM entonces se puede encontrar mediante el cálculo del estado de mínima energía de esta red. Las

Figura 11.35: Pasos principales de V-SLAM de Davison.

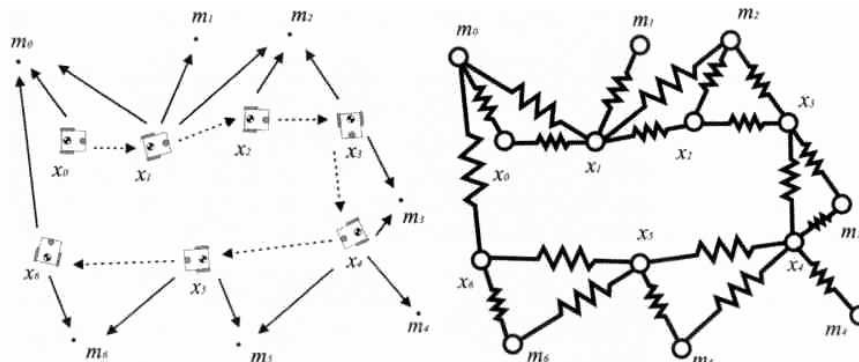


Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.

técnicas comunes de optimización para encontrar la solución se basan en el gradiente de descenso, y otros similares.

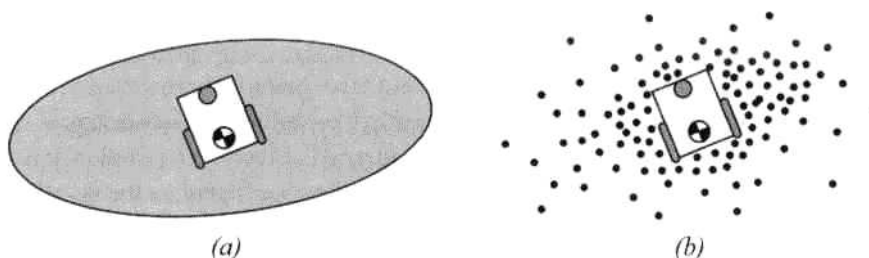
Hay una ventaja significativa de las técnicas de SLAM basada en gráfico sobre la SLAM de EKF. En la SLAM de EKF la cantidad de cálculos y requisitos de memoria para actualizar y almacenar la matriz de covarianza crece cuadráticamente con el número de características. A la inversa, en el SLAM basado en gráfico el tiempo de actualización de la gráfica es constante y la memoria requerida es lineal con el número de características. Sin embargo, la optimización gráfica final puede ser computacionalmente costosa si la trayectoria del robot es larga. Los algoritmos de la SLAM basada en gráficos han mostrado resultados impresionantes y exitosos incluso con cien millones de característica. Pero, estos algoritmos intentan optimizar la trayectoria completa del robot y fueron implementados por lo tanto para trabajar fuera de línea. Algunos de las implementaciones fuera línea utilizaron enfoques de submapa.

Figura 11.36: Ilustración evocativa de la construcción gráfica.



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.

Figura 11.37: SLAM de EKF estándar



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition.

### 11.7.8. Filtro de partículas SLAM

Esta solución particular al problema SLAM se basa en el muestreo aleatorio de la distribución de creencia. El término filtro de partículas nace del hecho de que representa la distribución creencia del robot no en una forma paramétrica (como una gaussiana), sino más bien como un conjunto de muestras (es decir, partículas) extraídas al azar a partir de esta distribución. Este concepto está gráficamente ilustrado en la figura 11.37. La SLAM de EKF estándar representa la distribución de probabilidad de la ubicación robot es una forma paramétrica, que es una gaussiana bidimensional (a). A la inversa, el filtro de partículas SLAM representan esta distribución de probabilidad como un conjunto de partículas extraídas al azar de la distribución paramétrica (b). El poder de esta representación está en su capacidad para modelar cualquier tipo de distribución

(por ejemplo, una no gaussiana) y también transformaciones no lineales.

En cada paso de tiempo, el filtro de partículas mantiene siempre el mismo número de partículas  $K$  (por ejemplo,  $K = 1000$ ). Cada partícula contiene una estimación de la trayectoria del robot  $X_t^{[k]}$  y las estimaciones de la posición de cada elemento en el mapa, las cuales se representan como Gaussianas de dos dimensiones con valores medios  $\mu_{t,i}^{[k]}$  y matrices de covarianza  $\sum_{(t,i)}^{[k]}$ . Por lo tanto, una partícula se caracteriza por:

$$X_t^{[k]}; \left( \mu_{t,0}^{[k]}, \sum_{t,0}^{[k]} \right); \left( \mu_{t,1}^{[k]}, \sum_{t,1}^{[k]} \right); \cdots; \left( \mu_{t,1n-1}^{[k]}, \sum_{t,n-1}^{[k]} \right) \quad (11.105)$$

Donde  $k$  denota el índice de la partícula y  $n$  el número de características en el mapa. Se debe saber que en filtro de partículas SLAM, la media y la covarianza de cada característica se actualizan utilizando distintos filtros de Kalman, uno para cada característica en el mapa. Cuando el robot se mueve, el modelo de movimiento especificado por la lectura de odometría  $u_t$  se aplica a cada partícula  $x_{t-1}^{[k]}$  para generar la nueva ubicación  $x_t^{[k]}$ .

Cuando el robot hace una observación  $z_t$ , se calcula para cada partícula el llamado factor de importancia  $w_t^{[k]}$ , que se determina como la probabilidad de observar  $z_t$  dada la partícula  $x_t^{[k]}$  y todas las observaciones anteriores  $z(0 \rightarrow t-1)$ , es decir,

$$w_t^{[k]} = p(z_t | x_t^{[k]}, z(0 \rightarrow t-1)) \quad (11.106)$$

El cálculo del factor de importancia para cada partícula es como el muestreo de la distribución de probabilidad  $p(z_t | x_t, z(0 \rightarrow t-1))$ .

El paso final en filtro de partículas SLAM se llama remuestreo. Este paso reemplaza el conjunto actual de partículas con otro conjunto de acuerdo con el factor de importancia determinado anteriormente. Por último, la media y la covarianza de cada característica se actualizan de acuerdo con la norma de actualización de EKF estándar.

---

---

# CAPÍTULO 12

---

## PLANIFICACIÓN Y NAVEGACIÓN EN LOS ROBOTS MÓVILES

Las tareas involucradas en la navegación de un robot móvil son: la percepción del entorno a través de sus sensores, de modo que le permita crear una abstracción del mundo; la planificación de una trayectoria libre de obstáculos, para alcanzar el punto destino seleccionado; y el guiado del vehículo a través de la referencia construida. De forma simultánea, el vehículo puede interactuar con ciertos elementos del entorno. Así, se define el concepto de operación como la programación de las herramientas de a bordo que le permiten realizar la tarea especificada. Un ejemplo de esta última noción es el transporte automático de materiales y herramientas dentro de una célula de manufactura flexible (FMS), lo que implica no sólo el movimiento físico de dichos elementos desde la estación de entrada de material hasta la máquina que lo requiera, sino que además pueda realizar operaciones como el cambio automático de la herramienta o la descarga automática del material en la máquina que lo haya solicitado.

### 12.1. GENERALIDADES

Esta investigación se ha centrado en los elementos de un robot móvil que son críticos para la movilidad robusta: la cinemática de la locomoción, los sensores para la determi-

nación del contexto ambiental del robot, y las técnicas para la localización con respecto a su mapa. Dado el conocimiento parcial sobre su entorno y una posición objetivo o una serie de posiciones, la navegación abarca la capacidad del robot para actuar en base a su conocimiento y valores de los sensores a fin de llegar a sus posiciones objetivo de la forma más eficiente y fiable posible. El objetivo de este capítulo es cómo las herramientas de los capítulos anteriores se pueden combinar para resolver este problema de navegación.

En primer lugar, sin embargo, se discuten dos competencias clave adicionales requeridas para la navegación de un robot móvil. La planificación de ruta es una competencia estratégica para la resolución del problema, ya que el robot debe decidir qué hacer a largo plazo para lograr sus objetivos. La segunda competencia es igual de importante, la evasión de obstáculos significa modulación de la trayectoria del robot con el fin de evitar colisiones.

## 12.2. COMPETENCIAS PARA LA NAVEGACIÓN: PLANIFICACIÓN Y REACCIÓN

La planificación y la reacción son un fuerte complemento, cada uno siendo crítico para el éxito del otro. Sin la reacción, el esfuerzo de planificación no valdrá la pena porque el robot físicamente nunca alcanzará su objetivo. Sin planificación, el esfuerzo de reacción no puede guiar el comportamiento del robot en general para llegar a una meta distante, y de nuevo, el robot nunca alcanzará su objetivo. Supongamos que un robot  $R$  en el instante  $i$  tiene un mapa  $M_i$  y un estado de creencia inicial  $b_i$ . El objetivo del robot es alcanzar una posición  $p$  al tiempo que satisface algunas restricciones temporales:  $loc_g(R) = p; (g \leq n)$ . Por lo tanto, el robot debe estar en la ubicación  $p$  en o antes del paso de tiempo  $n$ .

Aunque el objetivo del robot es netamente físico, el robot sólo puede realmente sentir su estado de creencia, no su ubicación física, y por lo tanto mapeamos el objetivo de la posición  $p$  a alcanzar para obtener un estado de creencia  $b_g$ , correspondiente a la creencia de que  $loc_g(R) = p$ . Con esta formulación, un plan  $q$  no es nada más que una o más trayectorias de  $b_i$  a  $b_g$  si el plan se ejecuta a partir de un estado de mundo coherente tanto con  $b_i$  y  $M_i$ . Por supuesto, el problema es que esta última condición no puede satisfacerse. Es muy posible que la posición del robot no sea muy consistente con  $b_i$ , e incluso es más probable que  $M_i$  sea incompleta o incorrecta. Además, el entorno

del mundo real es dinámico. Aun cuando  $M_i$  sea correcta como una sola foto en el tiempo, el modelo del planificador en cuanto a cómo cambia  $M$  con el tiempo suele ser imperfecto.

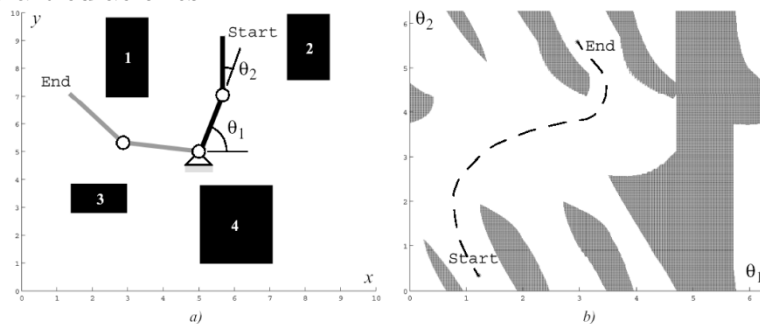
Para alcanzar su objetivo, sin embargo, el robot debe incorporar nueva información obtenida durante la ejecución del plan. Este extremo, en cuyo punto el concepto de la planificación y el concepto de la reacción se fusionan, se llama planificación integrada y la ejecución se describe en la sección *Arquitecturas de navegación*.

**Complejidad.** Un concepto útil en toda esta discusión de la arquitectura del robot consiste en determinar si las decisiones particulares de diseño sacrifican la capacidad del sistema para lograr un objetivo deseado cuando existe una solución. Este concepto se denomina *complejidad*. A menudo, la complejidad se sacrifica por la complejidad computacional en el nivel de representación o razonamiento.

## 12.3. PLANIFICACIÓN DE RUTA

**Configuración del espacio.** La planificación de ruta para los robots más móviles, se hace formalmente en una representación llamada *espacio de configuración*. Supongamos que un brazo robot (por ejemplo, robot SCARA) tiene  $k$  grados de libertad. Cada estado o configuración del robot se puede describir con valores de  $k$  reales:  $q_1, \dots, q_k$ . Los valores  $k$  pueden ser considerados como un punto  $p$  en un espacio  $k$  – *dimensional* llamado el espacio  $C$  de configuración del robot. Esta descripción es conveniente, ya que permite describir la forma compleja en 3D del robot con un solo punto  $k$  – *dimensional*.

Figura 12.1: El espacio físico (a) y la configuración del espacio (b) para un brazo robótico planar con dos articulaciones



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

El objetivo de la planificación de ruta es encontrar un camino en el espacio físico de la posición inicial del brazo a la posición objetivo, evitando las colisiones con los obstáculos. Este es un problema difícil de visualizar y resolver en el espacio físico, en particular a medida que  $k$  aumenta de tamaño. Pero en la configuración de espacio el problema es sencillo. Si se define la *configuración del espacio obstáculo*  $O$  como el subespacio de  $C$ , donde el brazo robótico golpea en algo, podemos calcular el espacio libre  $F = C - O$  en el que el robot puede moverse con seguridad. La figura 12.1 muestra una imagen del espacio físico y la configuración de espacio para un brazo robótico planar con dos articulaciones. El objetivo del robot es mover su efector final desde la posición de principio a fin (start to end).

En robótica móvil, el método más común es asumir con fines de planificación de trayectoria que el robot es de hecho holonómico, lo que simplifica enormemente el proceso.

**Visión general de la planificación de ruta.** El primer paso de cualquier sistema de planificación de ruta es por tanto transformar el modelo ambiental posiblemente continuo en un mapa discreto adecuado para el algoritmo de planificación de ruta escogido. Los planificadores de trayectoria difieren en cómo utilizar esta descomposición discreta. Se describen dos estrategias generales:

1. **Búsqueda Gráfica:** Un gráfico de conectividad en el espacio libre es construido primero y luego buscado. El proceso de construcción gráfica se realiza a menudo fuera de línea.
2. **Planificación de campo potencial:** Una función matemática se impone directamente en el espacio libre. El gradiente de esta función puede ser seguido al objetivo.

### 12.3.1. Búsqueda gráfica

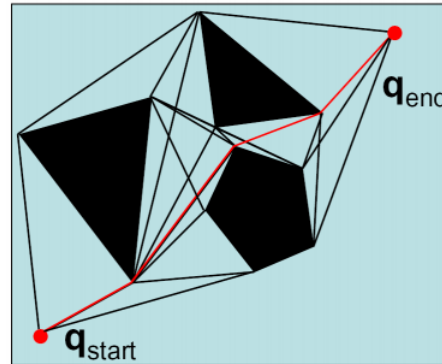
Para la mayoría de estos métodos se distinguen dos etapas principales: construcción de gráfico, donde los nodos son colocados y conectados a través de los bordes, y la búsqueda gráfica, donde el cálculo de una solución (óptima) se realiza.

#### Construcción de gráfico

El reto está en construir un conjunto de nodos y aristas que permitan que el robot vaya a cualquier lugar en su espacio libre al tiempo que limita el tamaño total de la gráfica. En primer lugar, se describen dos enfoques de mapa de ruta que logran este resultado con tipos radicalmente diferentes de caminos. En el caso gráfico de visibilidad, las rutas se acercan lo más posible a los obstáculos y resultan en soluciones de

caminos óptimos de mínima longitud. En el caso del diagrama de Voronoi, los caminos se mantienen lo más alejados posibles de obstáculos.

Figura 12.2: Gráfico de visibilidad.



Fuente: Robot Motion Planning. Norwood, Latombe, J.C., MA, Kluwer Academic

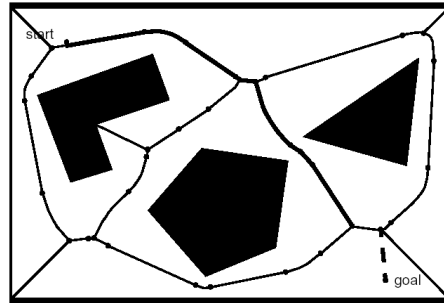
**Gráfico de Visibilidad.** El gráfico de visibilidad para una configuración del espacio poligonal  $C$  consta de aristas que unen todos los pares de vértices que se pueden ver unos a otros (incluyendo tanto las posiciones iniciales y finales, así como vértices) como se ve en la figura 12.2. Los gráficos de visibilidad son bastante populares en la robótica móvil, en parte debido a que su aplicación es bastante simple. El método es extremadamente rápido y eficiente en entornos dispersos, pero puede ser lento e ineficaz en comparación con otras técnicas cuando se usa en entornos densamente poblados. También, las vías de solución encontradas por búsqueda de gráfico tienden a tomar el robot lo más cerca posible a los obstáculos en el camino hacia la meta.

**Diagrama de Voronoi.** Es un método completo de mapa de trayectoria que tiende a maximizar la distancia entre el robot y los obstáculos en el mapa. Para cada punto en el espacio libre, se calcula su distancia al obstáculo más cercano. Si traza esa distancia como la altura saliente de la página, esta aumenta a medida que uno se aleja de un obstáculo (figura 12.3). El diagrama de Voronoi tiene una debilidad importante en el caso de los sensores de localización de rango limitado. Ya que sus bordes maximizan la distancia a los obstáculos, cualquier sensor de corto alcance en el robot estará en peligro de no percibir su entorno. **Taller en Matlab, Anexo C página 646.**

Hay, sin embargo, una ventaja sutil importante que el método de diagrama de Voronoi tiene sobre la mayoría de otros gráficos: *ejecutabilidad*, ya que Voronoi puede mitigar la inexactitud del encoder.

**Descomposición de celda exacta.** La figura 12.4 representa la descomposición

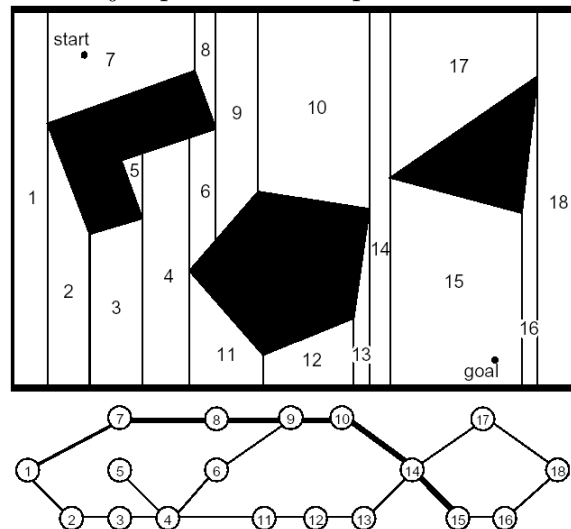
Figura 12.3: Diagrama de Voronoi



Fuente: Robot Motion Planning. Norwood, Latombe, J.C., MA, Kluwer Academic

de celda exacta. La abstracción básica detrás de esta descomposición es que la posición particular del robot dentro de cada celda de espacio libre no importa, lo que importa es más bien la capacidad del robot para recorrer desde cada celda libre a las celdas adyacentes libres. La desventaja fundamental de este método es que el número de celdas y, por lo tanto, la eficiencia global de planificación de cálculo depende de la densidad y la complejidad de los objetos en el medio ambiente, al igual que con los sistemas basados en mapeo de trayectorias. La ventaja clave es el resultado de esta misma correlación.

Figura 12.4: Ejemplo de descomposición de celda exacta.

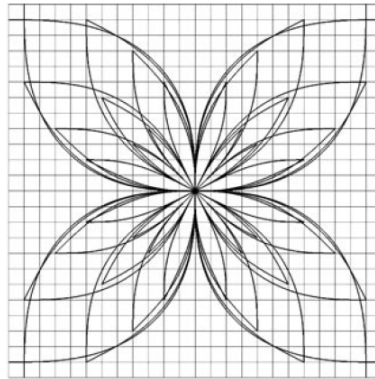


Fuente: Robot Motion Planning. Norwood, Latombe, J.C., MA, Kluwer Academic

**Descomposición celular aproximada.** Por el contrario, la descomposición de celda aproximada es una de las técnicas de construcción de gráfico más populares en

la robótica móvil. Esto es en parte debido a la popularidad de las representaciones de rejilla del entorno. Estas representaciones de cuadrícula son en sí mismas descomposiciones de tamaño fijo de rejilla y por tanto son idénticas a una descomposición de celda aproximada del medio ambiente. El tamaño de la celda no es dependiente de los objetos particulares de un entorno, y los pasillos de paso estrechos se pueden perder debido a la naturaleza inexacta de la teselación. El método se muestra en la figura 11.16.

Figura 12.5: Estado de enrejado de 16 direcciones construido para un rover de exploración planetaria.



Fuente: Differentially constrained mobile robot motion planning in state lattices, Pivtoraiko, M., Knepper, R., A., Kelly, A., Journal of filed robotics 29, 2009

**Gráfico de celosía (enrejado).** Las estructuras de celosía sólo recientemente se han adaptado a la búsqueda gráfica. Se forman primero, construyendo un conjunto básico de bordes (tal como el representado en la figura 12.5) y después repitiéndolo sobre la totalidad de la configuración de espacio para formar un gráfico. El principal beneficio con respecto a otros métodos de construcción de gráficos está en la libertad de diseño en la creación de bordes viables, es decir, los bordes pueden ser inherentemente ejecutados por una plataforma robótica.

### Búsqueda de gráfico determinista

Sea cualquier representación de mapa escogida, el objetivo de la planificación de ruta es encontrar el mejor camino en el gráfico del mapa de conectividad entre el inicio y la meta, donde “mejor” se refiere a los criterios de optimización seleccionados (por ejemplo, el camino más corto). En esta sección se presentan varios algoritmos de búsqueda que se han vuelto populares en robótica móvil.

**Discriminadores.** Debido a la similitud entre muchos algoritmos de búsqueda de gráficos, comenzamos esta sección con una elaboración de sus respectivas diferencias. Para ello, es benéfico introducir los conceptos de costo total previsto  $f(n)$ , costo de trayectoria  $g(n)$ , costo del borde transversal  $c(n, n')$ , y costo heurístico  $h(n)$ , las cuales todas son funciones del nodo  $n$  (y un nodo adyacente  $n'$ ). En particular, se denota el costo acumulado desde el nodo de partida para cualquier nodo  $n$  dado con  $g(n)$ . El costo de un nodo  $n$  a un nodo adyacente  $n'$  se convierte en  $c(n, n')$ , y el costo esperado (costo heurístico) desde un nodo  $n$  al nodo objetivo se describe con  $h(n)$ . El costo total previsto de principio a fin a través de estado  $n$ , entonces se puede escribir como:

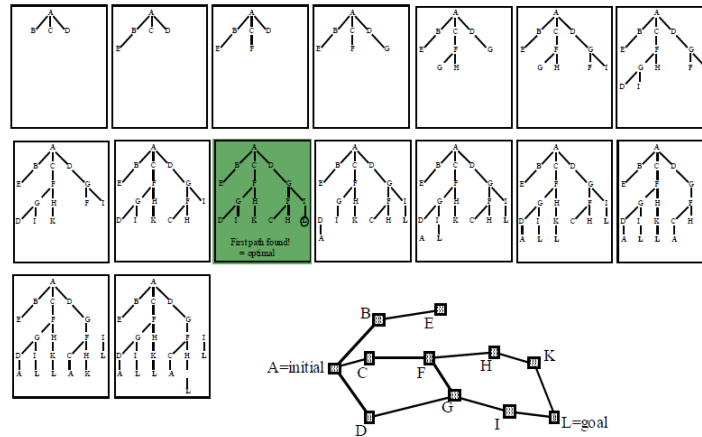
$$f(n) = g(n) + \varepsilon \cdot h(n) \quad (12.1)$$

Donde  $\varepsilon$  es un parámetro que asume valores dependen del algoritmo. En el caso especial de que cada borde individual en el gráfico se asume el mismo costo de recorrido (tal como en una rejilla de ocupación), las implementaciones óptimas se pueden desarrollar en una forma más simple y obtener una mayor velocidad de ejecución en comparación con el ejemplo general. En el caso de  $\varepsilon \neq 0$ , se emplea una función heurística  $h(n)$ , que en esencia incorpora información adicional sobre el conjunto de problemas y por lo tanto a menudo permite una convergencia más rápida de la consulta de búsqueda. En esta investigación, se restringe la atención a heurísticos que son a la vez consistentes y subestiman el costo real. Los heurísticos más prácticos cumplen estos requisitos. Para  $\varepsilon = 1$ , el algoritmo óptimo  $A^*$  resulta, mientras que para  $\varepsilon > 1$  se obtienen variantes subóptimas o codiciosas de  $A^*$ .

**Búsqueda de primero amplitud.** Este algoritmo de búsqueda gráfica comienza con el nodo de inicio (indicado por  $A$  en la figura 12.6) y explora la totalidad de sus nodos vecinos. Entonces, para cada uno de estos nodos, este explora todos sus vecinos inexplorados y así sucesivamente. Este proceso (es decir, marcando un nodo “activo”, explorando cada uno de sus vecinos y marcándolos “abierto” y, finalmente, marcando el nodo familiar “visitado”) se denomina expansión del nodo. En la búsqueda de primero-amplitud, los nodos se expanden en orden de proximidad al nodo de inicio con la proximidad definida como el menor número de transiciones de borde. El algoritmo procede hasta que llega al nodo objetivo en donde termina. El cálculo de una solución es rápido, ya que el reordenamiento de los nodos esperando para expansión no es necesario. **Ver taller en Matlab página 648 del Anexo C.**

Un ejemplo de algoritmo de búsqueda de primero amplitud en el contexto de la robó-

Figura 12.6: Principio de funcionamiento de la búsqueda de primero amplitud

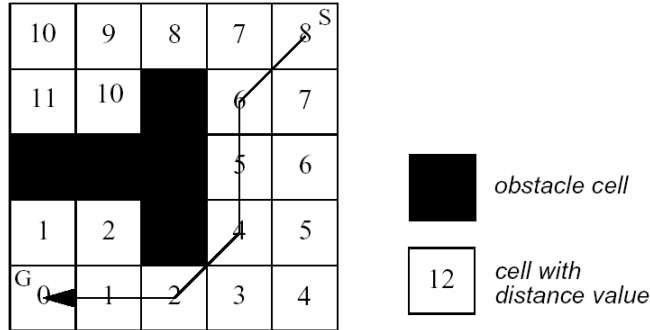


Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

tica es el algoritmo de expansión de frente de onda, que también se conoce como NF1 o “Grassfire” (Latombe, J.C., Barraquand, J., “Robot Motion Planning: A distribution presentation approach.”. *International Journal of Robotics Research*, 1991). Este algoritmo es una técnica eficiente y fácil de implementar para la búsqueda de rutas en tamaño fijo arreglos de celdas (figura 12.7).  $S$  indica el principio,  $G$  el objetivo. Los vecinos de cada celda  $i$  se definen como las cuatro celdas adyacentes que comparten un borde con  $i$  (4 vecindades). Este proceso continúa hasta que la celda célula correspondiente a la posición inicial del robot se alcanza. En este punto, el planificador de trayectoria puede estimar la distancia del robot a la posición objetivo, así como recuperar una trayectoria de solución específica simplemente uniendo celdas que son adyacentes y siempre cerca de la meta.

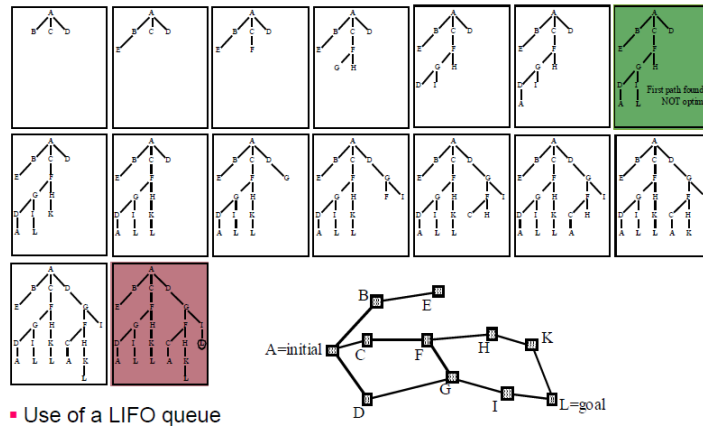
**La búsqueda primero en profundidad.** El principio de funcionamiento del algoritmo de búsqueda primero en profundidad se muestra en la figura 12.8. En contraste con la búsqueda primero en amplitud, ésta expande cada nodo hasta el nivel más profundo de la gráfica (hasta que el nodo no tiene más sucesores). A medida que los nodos se expanden, su rama es removida del gráfico y la búsqueda da marcha atrás mediante la expansión del siguiente nodo vecino del nodo inicial hasta su nivel más profundo y así sucesivamente. Un inconveniente de este algoritmo es que puede visitar nodos previamente visitados o introducir rutas redundantes. Sin embargo, estas situaciones pueden ser evitadas fácilmente a través de una implementación eficiente.

Figura 12.7: Un ejemplo de la transformada de la distancia y la trayectoria resultante como es generada por la función NF1.



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

Figura 12.8: Principio de funcionamiento de la búsqueda primero en profundidad



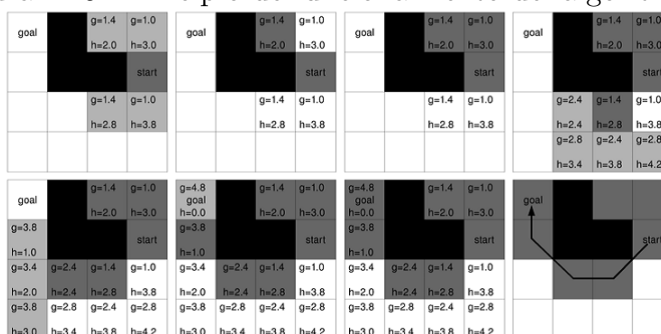
Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

**Algoritmo de Dijkstra.** Llamado así por su inventor, E.W. Dijkstra, este algoritmo es similar al de búsqueda de primero amplitud, mientras que los gastos de borde pueden asumir cualquier valor positivo y la búsqueda aún garantiza una solución óptima (*Dijkstra, E.W. “A note on two problems in conexión with graphs”, 1959*). Esto introduce una complejidad adicional en el algoritmo para el que es necesario introducir el concepto de cúmulo o pila, un árbol especializado basado en estructura de datos. Sus elementos (que comprenden nodos del gráfico a ser expandidos) se ordenan de acuerdo

a una clave, que en nuestro caso equivale al costo total de ruta esperado  $f(n)$  en ese nodo  $n$  dado. El algoritmo de Dijkstra expande los nodos iniciales desde el principio, similar a la búsqueda de primero amplitud, con la excepción de que los vecinos del nodo expandido se colocan en la pila y se reordenan de acuerdo con su  $f(n)$ , que corresponde a  $g(n)$  ya que el ningún heurístico se utiliza.

**Algoritmo A\*.** Para heurística consistente, el algoritmo  $A^*$  (que se pronuncia “A estrella”) (Hart, P. E, “A formal basis of the heuristic determination of minimum cost paths”, 1968), es similar al algoritmo de Dijkstra. Sin embargo, la inclusión de una función heurística  $h(n)$ , la cual codifica el conocimiento adicional acerca de la gráfica, hace que este algoritmo sea especialmente eficiente en un solo nodo para las consultas de nodo único. Con el fin de garantizar una solución óptima, la heurística se requiere para ser una función de desestimación del costo. Para un ejemplo, ver figura 12.9. La complejidad de tiempo de  $A^*$  depende en gran medida de la heurística elegida  $h(n)$ . En promedio, sin embargo, se puede esperar un rendimiento mucho mejor que con el algoritmo de Dijkstra.

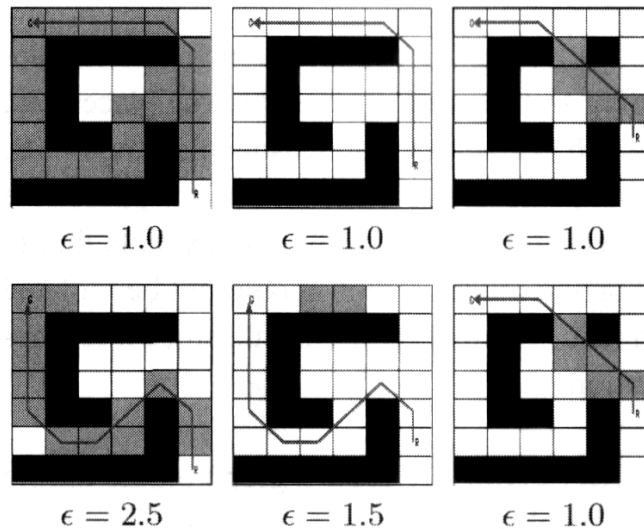
Figura 12.9: Principio de funcionamiento del algoritmo  $A^*$ .



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

**Algoritmo D\*.** El algoritmo  $D^*$  (Koenig, S., Likhachev, M., “Fast replanning for navigation in unknown terrain”, 2005), (Stentz, A., “Optimal and efficient path planning for partially-known environments”, 1994), representa una versión nueva de planificación incremental de  $A^*$ , donde el término incremental se refiere a la reutilización del algoritmo de esfuerzo de búsqueda anterior en iteraciones de búsqueda posteriores. Se ilustra esto con un ejemplo (figura 12.10): nuestro robot está inicialmente provisto de un mapa crudo del medio ambiente (es decir, que se obtiene a partir de una imagen aérea). En este mapa, el módulo de navegación planea una ruta de acceso inicial mediante el empleo

Figura 12.10: Comparación del número de celdas expandidas para  $D^*$  y en cualquier momento  $D^*$



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

de  $A^*$ . Después de ejecutar este camino por un tiempo, el robot observa algunos cambios en el medio ambiente con sus sensores a bordo.

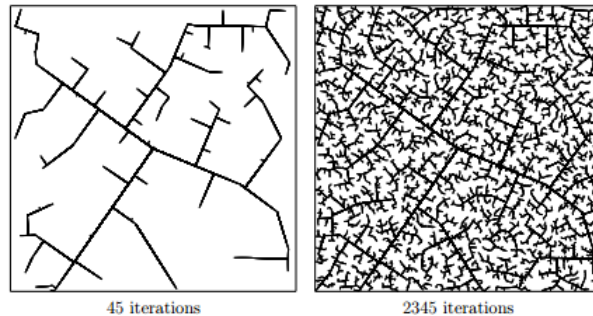
### Búsqueda gráfica aleatoria

Cuando se encuentran problemas de planificación de ruta complejos de alta dimensiones (por ejemplo, en tareas de manipulación de brazos robóticos, o molécula plegables y consultas de acoplamiento para la colocación de drogas, y así sucesivamente) se vuelve inviable resolverlos de manera exhaustiva dentro de un tiempo límite razonable. En tales situaciones, la búsqueda aleatoria llega a ser útil, ya que se renuncia a una solución óptima por un cálculo más rápido de la solución.

**Arboles aleatorios de exploración rápida (Rapidly Exploring Random Trees-RRTs).** Los RRTs suelen desarrollar un gráfico en línea durante el proceso de búsqueda y, por tanto, a priori, sólo requieren un mapa del obstáculo pero no descomposición gráfica. El algoritmo comienza con un árbol inicial (que puede estar vacío) y luego sucesivamente añade nodos, conectados a través de los bordes, hasta que una condición de terminación se activa. Específicamente, durante cada paso una configuración aleatoria  $q_{rand}$  en el espacio libre se selecciona. El nodo de árbol que está más

cerca a  $q_{rand}$ , denotado como  $q_{near}$ , se calcula entonces. A partir de  $q_{near}$ , un borde (con longitud fija) crece hacia  $q_{rand}$  utilizando un modelo de movimiento del robot apropiado. La configuración  $q_{new}$  al extremo de este borde se añade entonces al árbol, si el borde de conexión es libre de colisión, (figura 12.11).

Figura 12.11: La evolución de un RRT. (a) 45 iteraciones. (b) 2345 iteraciones



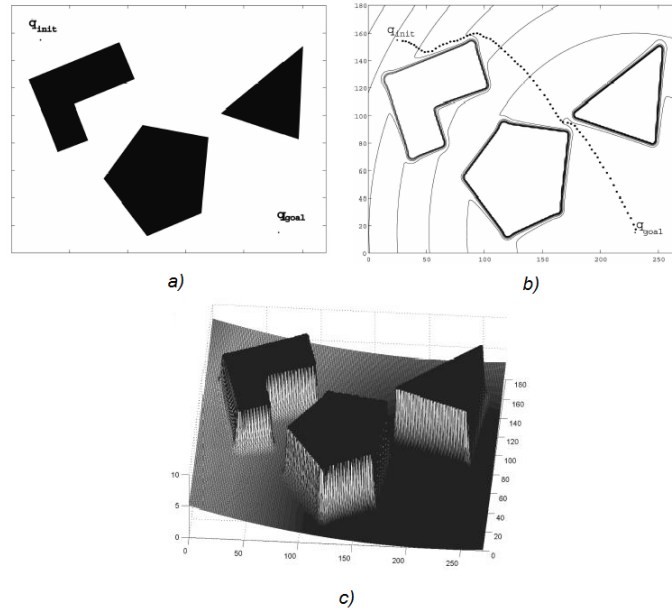
Fuente: Planning Algorithms, LaValle, S.M., Cambridge, UK, Cambridge University Press, 2006

### 12.3.2. Planificación de trayectorias de campo potencial

La planificación de trayectorias de campo potencial crea un campo, o gradiente, a través del mapa del robot que dirige el robot a la posición objetivo desde múltiples posiciones anteriores. Este método fue inventado originalmente para la planificación de trayectoria de robot manipulador y se utiliza a menudo y con muchas variantes en la comunidad robótica móvil. El método de campo potencial trata el robot como un punto bajo la influencia de un campo artificial potencial  $U(q)$ . El robot se mueve siguiendo el campo, al igual que una bola rodaría cuesta abajo. El objetivo actúa como una fuerza de atracción sobre el robot, y los obstáculos actúan como picos, o fuerzas repulsivas (figura 12.12). Este campo potencial artificial suavemente guía al robot hacia la meta y al mismo tiempo evita los obstáculos conocidos.

La idea básica detrás de todos los enfoques posibles sobre el terreno es que el robot es atraído hacia la meta, mientras es repelido por los obstáculos que se conocen de antemano. Si aparecen nuevos obstáculos durante el movimiento del robot, se puede actualizar el campo potencial para integrar esta nueva información. En el caso más simple, se supone que el robot es un punto, por lo que la orientación  $\theta$  del robot se descuida, y el campo potencial resultante es sólo 2D  $(x, y)$ . Si se asume una función de

Figura 12.12: Típico campo potencial generado por el objetivo atrayente y dos obstáculos



Fuente: Robot Motion Planning, Latombe, J.C., 1991

campo potencial diferenciable  $U(q)$ , se encuentra la fuerza artificial relacionada  $F(q)$  que actúa en la posición  $q = (x, y)$ :

$$F(q) = -\nabla U(q) \quad (12.2)$$

Donde  $\nabla U(q)$ , denota el vector gradiente de  $U$  en la posición  $q$ .

$$\nabla U = \begin{bmatrix} \frac{\delta U}{\delta x} \\ \frac{\delta U}{\delta y} \end{bmatrix} \quad (12.3)$$

El campo potencial que actúa sobre el robot se calcula como la suma del campo atractivo del objetivo y los campos de repulsión de los obstáculos:

$$U(q) = U_{att}(q) + U_{rep}(q) \quad (12.4)$$

Del mismo modo, las fuerzas se pueden separar también en una parte atrayente y repulsiva:

$$F(q) = F_{att}(q) - F_{rep}(q) = -\nabla U_{att}(q) - \nabla U_{att}(q) \quad (12.5)$$

**Potencial atractivo.** Un potencial atractivo puede, por ejemplo, definirse como una función parabólica.

$$U_{att}(q) = \frac{1}{2}k_{att} \cdot \rho_{att}^2(q) \quad (12.6)$$

Donde  $k_{att}$  es un factor de escala positivo y  $\rho_{goal}(q)$  denota la distancia  $\|q - q_{goal}\|$ . Este potencial atractivo es diferenciable, conduciendo a la fuerza de atracción  $F_{att}$ :

$$F_{att}(q) = -\nabla U_{att}(q) \quad (12.7)$$

$$F_{att}(q) = -k_{att} \cdot \rho_{goal}(q) \nabla \rho_{goal}(q) \quad (12.8)$$

$$F_{att}(q) = -k_{att} \cdot (q - q_{goal}) \quad (12.9)$$

Que converge linealmente hacia 0 cuando el robot alcanza el objetivo.

**Potencial repulsivo.** La idea detrás del potencial repulsivo es generar una fuerza lejos de todos los obstáculos conocidos. Este potencial repulsivo debe ser muy fuerte cuando el robot se encuentra cerca del objeto, pero no debe influir en su movimiento cuando el robot está lejos del objeto. Un ejemplo de un campo de repulsión es:

$$U_{rep}(q) = \begin{cases} \frac{1}{2}k_{rep} \left( \frac{1}{\rho(q)} - \frac{1}{\rho_0} \right)^2 & \text{si } \rho(q) \leq \rho_0 \\ 0 & \text{si } \rho(q) \geq \rho_0 \end{cases} \quad (12.10)$$

Cuando  $K_{rep}$  es de nuevo un factor de escala,  $\rho(q)$  es la distancia mínima desde  $q$  al objeto y  $\rho_0$  la distancia de influencia del objeto. La función de potencial repulsivo  $U_{rep}$  es positivo o cero y tiende a infinito cuando  $q$  se acerca más al objeto. Si la frontera del objeto es convexa y diferenciable a trozos,  $\rho(q)$  es diferenciable en todas partes en el espacio de configuración libre. Esto lleva a la fuerza de repulsión  $F_{rep}$ :

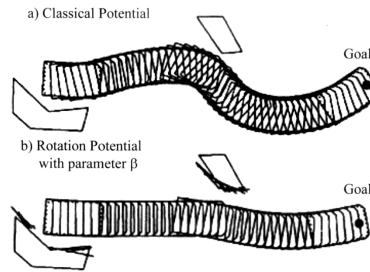
$$F_{rep}(q) = \nabla U_{rep}(q) = \begin{cases} \frac{1}{2}k_{rep} \left( \frac{1}{\rho(q)} - \frac{1}{\rho_0} \right) \frac{1}{\rho^2(q)} \frac{q - q_{obstacle}}{\rho(q)} & \text{si } \rho(q) \leq \rho_0 \\ 0 & \text{si } \rho(q) \geq \rho_0 \end{cases} \quad (12.11)$$

La fuerza resultante  $F(q) = F_{att}(q) + F_{rep}(q)$  que actúa sobre un punto expuesto del robot a las fuerzas de atracción y de repulsión mueve el robot lejos de los obstáculos y hacia el objetivo (figura 12.12).

**El método extendido de campo potencial.** Khatib y Chatila propusieron el enfoque extendido de campo de potencial (*Khatib, M., Chatila, R., “An extended potential field approach for mobile robot sensor motions”, 1995*). Como todos los métodos potenciales de campo, este enfoque hace uso de las fuerzas de atracción y repulsión que se originan a partir de un campo potencial artificial. Sin embargo, dos adiciones al campo potencial básico se realizan: el campo potencial de rotación y el campo potencial de trabajo.

El *campo potencial de giro* supone que la fuerza de repulsión es una función de la distancia desde el obstáculo y la orientación del robot en relación con el obstáculo. El *campo potencial de trabajo* considera la velocidad actual del robot, y filtra esos obstáculos que no deberían afectar el potencial a corto plazo basado en la velocidad del robot. Un ejemplo clásico de comparación de un campo potencial y un campo de potencial extendido se muestra en la figura 12.13.

Figura 12.13: Comparación entre un campo potencial clásico y un campo potencial extendido



Fuente: An extended potential field approach for mobile robot sensor motions, Khatib, M., Chatila, R.,1995

## 12.4. EVASIÓN DE OBSTÁCULOS

La evasión de obstáculos locales se centra en el cambio de trayectoria del robot según lo informado por los sensores durante el movimiento del robot. El movimiento del robot resultante es una función tanto de las lecturas actuales o recientes del sensor

del robot como de su posición objetivo y la ubicación relativa a la posición objetivo. Los algoritmos de evasión de obstáculos que se presentan aquí dependen en varios grados de la existencia de un mapa global y el conocimiento preciso del robot de su ubicación con respecto al mapa. A pesar de sus diferencias, todos los algoritmos pueden ser denominados algoritmos de evasión de obstáculos porque las lecturas locales del sensor del robot desempeñan un papel importante en la trayectoria futura del robot.

### 12.4.1. Algoritmo de Bug

El algoritmo Bug (*Lumensky, V., "Incorporating range sensing in the robot navigation function", 1990; "Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape"*) es tal vez el algoritmo de evasión de obstáculos más simple que se podría imaginar. La idea básica es seguir el contorno de cada obstáculo en el camino del robot y por lo tanto circunnavegarlo.

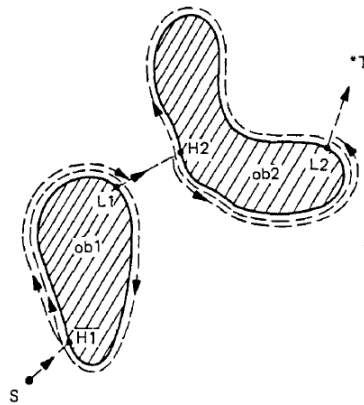
Con *Bug1*, el robot da completamente la vuelta al primer objeto, luego se aparta del punto con la distancia más corta hacia el objetivo (figura 12.14). Este enfoque es, por supuesto, muy ineficiente pero garantiza que el robot logre cualquier objetivo alcanzable. Con *Bug2*, el robot comienza a seguir el contorno del objeto, pero se aparta inmediatamente cuando es capaz de moverse directamente hacia el objetivo como se muestra en la figura 12.15.

Un número de variaciones y extensiones del algoritmo Bug existen. Se menciona uno más, el Bug Tangente (*Kamon, I., "A new range-sensor based globally convergent navigation algorithm for mobile robots", 1996*), el cual aumenta el rango de detección y una representación del medio ambiente local llamada el gráfico tangente local (LTG). No sólo puede el robot moverse de manera más eficiente a la meta con el LTG, sino que también puede ir a lo largo de atajos cuando rodea los obstáculos y volver a la búsqueda anterior del objetivo. En muchos ambientes simples, Bug tangente se acerca a los caminos óptimos globales. **Ver taller en la página 652 del Anexo C.**

### 12.4.2. Histograma de campo vectorial

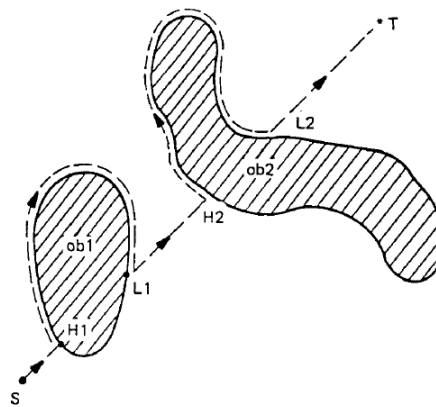
Borenstein, junto con Koren, desarrolló el Histograma de campo vectorial. (VFH) (*Borenstein, J., koren., "The vector field histogram-fast obstacle avoidance for mobile robots", 1991*). Su trabajo anterior, que se concentró en los campos potenciales (High speed obstacle avoidance for mobile robots), fue abandonado debido a la inestabilidad del método y la imposibilidad de pasar a través de pasajes estrechos. Más tarde, Borenstein, junto con Ulrich, amplió el algoritmo VFF para producir VFH+ (VFH+: Reliable

Figura 12.14: Algoritmo *Bug1* con H1, H2, puntos de alcance, y L1, L2, puntos de salida



Fuente: Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape, Lumensky, V

Figura 12.15: Algoritmo *Bug2* con H1, H2, puntos de alcance, y L1, L2, puntos de salida

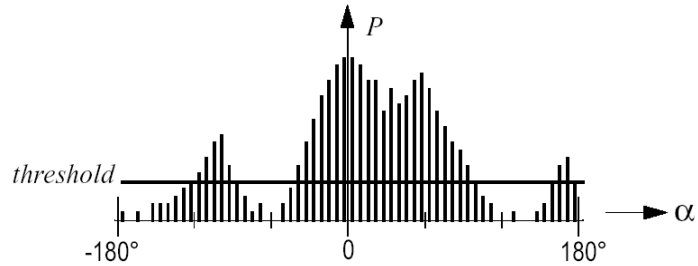


Fuente: Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape, Lumensky, V

obstacle avoidance for fast mobile robots) y *VFH\** (*VFH\**: Local obstacle avoidance with look-ahead verification).

Para la evasión de obstáculos, el VFH genera un histograma polar como se muestra en la figura 12.16. El eje  $x$  representa el ángulo  $\alpha$  en el cual se encuentra el obstáculo, y el eje  $y$  representa la probabilidad  $P$  de que realmente haya un obstáculo en esa dirección basada en los valores de celda de la cuadrícula de ocupación.

Figura 12.16: Histograma polar



Fuente: Real-time obstacle avoidance for mobile robots in cluttered environments,  
Koren Y and Borenstein J

De este histograma un sentido de la dirección se calcula. En primer lugar, se identifican todas las aberturas lo suficientemente grandes como para que el vehículo pase a través. A continuación, una función de costo se aplica a cada abertura opcionada. El pasaje con el menor costo es elegido. La función de costo  $G$  tiene tres términos:

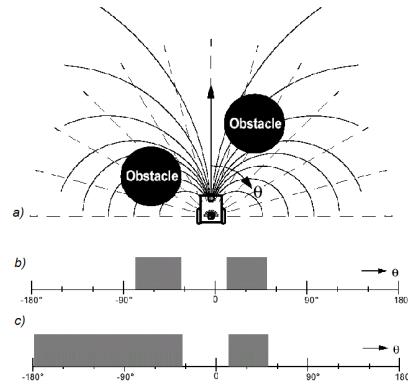
$$G = a \cdot target_{direction} + b \cdot wheel_{orientation} + c \cdot previous_{direction} \quad (12.12)$$

Donde:

- $target_{direction}$  = alineación de la trayectoria del robot con el objetivo.
- $wheel_{orientation}$  = diferencia entre la nueva dirección y la orientación de la rueda actual.
- $previous_{direction}$  = diferencia entre la dirección previamente seleccionada y la nueva dirección.

Los términos se calculan de tal manera que una gran desviación de la dirección objetivo conduce a un gran costo en el término “dirección objetivo”. Los parámetros  $a, b, c$  en la función de costo  $G$  ajustan el comportamiento del robot. Por ejemplo, una fuerte tendencia de objetivo sería expresada con un valor grande para  $a$ . En la mejora de  $VFH+$ , una de las etapas de reducción tiene en cuenta un modelo simplificado de las trayectorias posibles del robot móvil, basadas en sus limitaciones cinemáticas (por ejemplo, radio de giro para un vehículo Ackerman). El robot se modela para moverse en arcos o líneas rectas. Un obstáculo por lo tanto bloquea todas las trayectorias permitidas del robot que pasan a través del obstáculo (figura 12.17a). Esto se traduce en un histograma polar enmascarado donde los obstáculos se amplían para que todas las trayectorias cinemáticamente bloqueadas se tengan debidamente en cuenta (figura 12.17c).

Figura 12.17: Ejemplo de direcciones bloqueadas e histogramas polares resultantes: (a) Robot y los obstáculos que lo bloquean, (b) Histograma polar, (c) histograma polar enmascarado

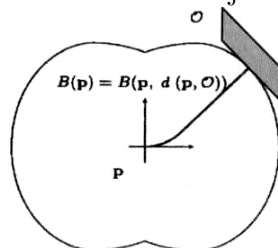


Fuente: Bio-inspired flying robots: Experimental synthesis of autonomous indoor flyers, Zufferey, J.C

### 12.4.3. Técnica de banda de burbuja

Esta idea es una extensión para los vehículos no holonómicos del concepto de banda elástica sugerido por Khatib y Quinlan (*Elastic bands: connecting, path planning and control, 1993*). Una burbuja se define como el subconjunto máximo local del espacio libre alrededor de una determinada configuración del robot que puede ser recorrida en cualquier dirección sin riesgo de colisión. La burbuja se genera utilizando un modelo simplificado del robot en relación con la información de alcance disponible en el mapa del robot. Incluso con un modelo simplificado de la geometría del robot, es posible tener en cuenta la forma real del robot al calcular el tamaño de la burbuja (figura 12.18).

Figura 12.18: La forma de las burbujas alrededor del vehículo

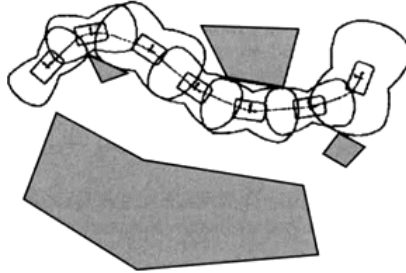


Fuente: Dynamic path modification for car-like nonholonomic mobile robots

Dadas estas burbujas, una banda o una cadena de burbujas puede ser utilizada a lo

largo de la trayectoria de la posición inicial del robot hasta su posición objetivo para mostrar al robot el espacio libre previsto a lo largo de su trayectoria (figura 12.19).

Figura 12.19: Una banda de burbuja típica



Fuente: Dynamic path modification for car-like nonholonomic mobile robots

Es evidente que el cálculo de la banda burbuja requiere un mapa global y un planificador de ruta global. Una vez que la trayectoria inicial del planificador de ruta ha sido calculada y la banda de burbuja se calcula, entonces se produce la modificación de la trayectoria prevista. La banda de burbuja tiene en cuenta las fuerzas de los objetos modelados y las fuerzas internas. Estas fuerzas internas tratan de minimizar el “slack” (energía) entre las burbujas adyacentes. Este proceso, además de una operación de suavizado final, hace que la trayectoria sea suave en el sentido del espacio libre del robot que cambiará tan suavemente como sea posible durante la ejecución del camino.

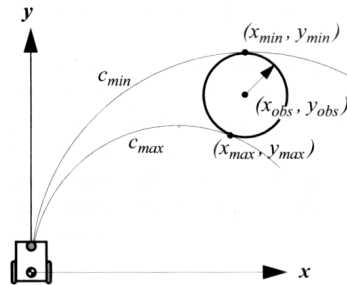
#### 12.4.4. Técnicas de velocidad de curvatura

El enfoque básico de velocidad de curvatura. El enfoque básico de velocidad de curvatura (CVM) de Simmons (*The curvature velocity method for local obstacle avoidance, 1996*), permite las restricciones cinemáticas reales e incluso algunas restricciones dinámicas del robot que deben tenerse en cuenta durante la evasión de obstáculos, lo cual es una ventaja sobre las técnicas más primitivas. CVM se inicia mediante la adición de restricciones físicas del robot y el entorno a un espacio de velocidades. La velocidad espacial consiste de velocidad de rotación  $\omega$  y velocidad de traslación  $v$ , asumiendo de esta manera que el robot se desplaza solamente a lo largo de arcos de círculos con curvatura  $c = \omega/v$ .

Dos tipos de restricciones se identifican: las limitaciones derivadas en aceleración y velocidad del robot, típicamente  $-v_{max} < v < v_{max}$ , y, segundo, las limitaciones de los obstáculos que bloquean ciertos valores de  $v$  y  $\omega$  debido a sus posiciones. Los obstáculos comienzan como objetos en una rejilla cartesiana, pero luego se transforman

en el espacio de velocidades mediante el cálculo de la distancia desde la posición del robot al obstáculo siguiendo alguna trayectoria del robot de curvatura constante, como se muestra en la figura 12.20.

Figura 12.20: Curvaturas tangentes para un obstáculo



Fuente: The curvature velocity method for local obstacle avoidance, Simmons

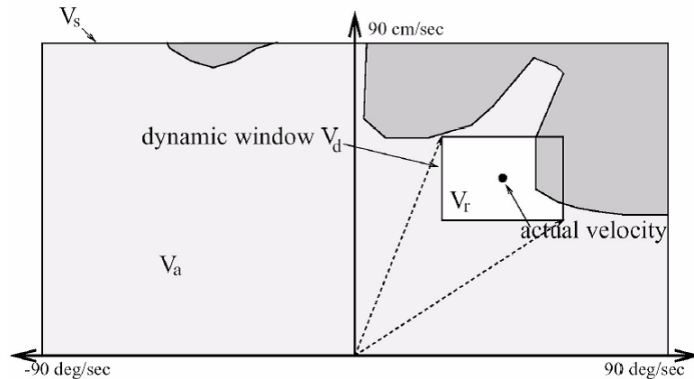
### 12.4.5. Enfoques dinámicos de ventana

Otra técnica para tener en cuenta las limitaciones cinemáticas del robot es el método de evasión de obstáculos de ventana dinámica. Un modelo dinámico simple pero muy efectivo da su nombre a este enfoque. Dos enfoques son representados en la literatura. El enfoque dinámico de ventana de Fox, Burgard, y Thrun (*The dynamic window approach to collision avoidance 1997*), y el enfoque dinámico global de ventana de Brock y Khatib (*High-speed navigation using the global dynamic window approach, 1999*).

**El enfoque de ventana dinámica local.** En este enfoque, la cinemática del robot se tiene en cuenta mediante la búsqueda en un espacio de velocidades bien elegido. El espacio de velocidad es todos los posibles conjuntos de tuplas  $(v, \omega)$  donde  $v$  es la velocidad y  $\omega$  es la velocidad angular. El enfoque asume que los robots se mueven sólo en arcos circulares que representan a cada tupla, al menos durante un lapso de tiempo (timestamp).

Dada la velocidad actual del robot, el algoritmo selecciona primero una ventana dinámica de todas las tuplas  $(v, \omega)$  que pueden ser alcanzadas dentro del siguiente período de muestra, teniendo en cuenta las capacidades de aceleración del robot y el tiempo de ciclo. El siguiente paso es el de reducir la ventana dinámica, manteniendo sólo

Figura 12.21: El enfoque dinámico de ventana



Fuente: The dynamic window approach to collision avoidance

aquellas tuplas que aseguran que el vehículo puede llegar a una parada antes de chocar contra un obstáculo. Las velocidades restantes se denominan velocidades admisibles. En la figura 12.21, una ventana dinámica típica está representada. La función objetivo  $O$  tiene la forma:

$$O = a \cdot heading(v, \omega) + b \cdot velocity(v, \omega) + c \cdot dist(v, \omega) \quad (12.13)$$

Donde:

- *heading* = Medida del progreso hacia la ubicación de meta.
- *velocity* = Velocidad de avance del robot  $\rightarrow$  provocando movimientos rápidos.
- *dist* = Distancia más cercana al obstáculo en la trayectoria.

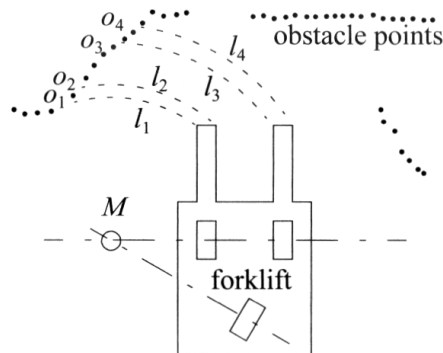
**El enfoque de ventana dinámica global.** Este enfoque agrega, como su nombre indica, el pensamiento global para el algoritmo presentado anteriormente. Esto se hace mediante la adición de NF1, o Grassfire, a la función objetivo  $O$  presentada anteriormente (figura 12.7). La NF1 etiqueta las celdas de la cuadrícula de ocupación con la distancia total al objetivo  $L$ . Para hacer esto más rápido, el enfoque calcula la NF1 sólo en una región rectangular seleccionada que se dirige desde el robot hacia la meta. El ancho de la región se amplía y se vuelve a calcular si el objetivo no puede ser alcanzado dentro de las limitaciones de esta región elegida. Esto permite que el enfoque de ventana dinámica global logre algunas de las ventajas de la planificación de la trayectoria global sin completar el conocimiento a priori.

### 12.4.6. El enfoque de Schlegel para evasión de obstáculos

Schlegel presenta un enfoque que considera la dinámica, así como la forma real del robot (*Fast local obstacle under kinematic and dynamic constraints, 1998*). El enfoque es adoptado para mediciones toscas de datos de láser y fusión de sensores utilizando una cuadrícula cartesiana para representar los obstáculos en el entorno. El rendimiento en tiempo real se consigue mediante el uso de tablas de consulta precalculadas.

Al igual que con los métodos anteriores descritos, la suposición básica es que un robot se mueve en las trayectorias construidas por arcos circulares, definidas como curvaturas  $i_c$ . Dada una cierta curvatura  $i_c$ , Schlegel calcula la distancia  $l_i$  a la colisión entre un punto de obstáculo único  $[x, y]$  en la cuadrícula cartesiana y el robot, representado en la figura 12.22. Puesto que al robot se le permite ser de cualquier forma, este cálculo consume mucho tiempo, y el resultado es por lo tanto precalculado y almacenado en una tabla de búsqueda.

Figura 12.22: Distancias  $l_i$  resultantes de la curvatura  $i$ , donde el robot gira alrededor de  $M$



Fuente: Fast local obstacle under kinematic and dynamic constraints, Schlegel

### 12.4.7. Diagrama de proximidad

Intentando cerrar una brecha de fidelidad del modelo en los métodos de evasión de obstáculos, el diagrama de proximidad (ND) se puede considerar que tiene alguna similitud con un VFH pero resuelve varias de sus deficiencias, especialmente en espacios muy saturados. También fue utilizado en (*Minguez, J, "Nearness diagram navigation, ND: A new real-time collision avoidance approach, 2000"*). También fue utilizado en (*Robot navigation in very complex, dense, and cluttered indoor / outdoor ambients, 2002*) para tener en cuenta restricciones geométricas, cinemática y dinámica más precisas. Esto se

logró por la descomposición del problema en la generación de la dirección más prometedora de camino con la única restricción de un robot circular, entonces adaptando esto a las restricciones cinemáticas y dinámicas del robot, seguida de una corrección de la forma del robot si no es circular (sólo formas rectangulares fueron apoyadas en la publicación original). El razonamiento Global se añadió a enfoque y el denominado diagrama de proximidad global (GND) en (*Global nearness diagram navigation, GND, 2001*) en cierto modo similar a la extensión GDWA a la DWA, pero en base a una representación del espacio de trabajo (en lugar de espacio de configuración) y la actualización de espacio libre además de la información de obstáculo.

#### **12.4.8. Método de gradiente**

Al darse cuenta de que la tecnología informática actual permite un cálculo rápido de las técnicas de propagación del frente de onda, el método del gradiente formula una de planificación de ruta global de red que tiene en cuenta la proximidad de obstáculos y permite la generación de interpolaciones continuas de la dirección del gradiente en cualquier punto dado en la red (*A gradient method for realtime robot control, Konolige K, 2000*). El *NF1* es un caso especial del algoritmo propuesto, que calcula una función de navegación en cada paso de tiempo y utiliza la información resultante del gradiente para conducir el robot hacia la meta en un camino liso y sin obstáculos de pasto a menos que sea necesario.

#### **12.4.9. Adición de restricciones dinámicas**

Intentando solucionar la falta de modelos dinámicos en la mayor parte de los enfoques de evasión de obstáculos discutidos anteriormente, un nuevo tipo de representación espacial fue propuesto por Minguez, Montano, y Khatib en (*Reactive collision avoidance for navigation with dynamic constraints, 2002*). El espacio ego-dinámico es igualmente aplicable a los métodos de área de trabajo y espacio de configuración. Se transforman obstáculos en distancias que dependen de las restricciones de frenado y el tiempo de muestreo del método de evasión de obstáculos subyacente. En combinación con la ventana espacial propuesta (PF) para representar las capacidades de aceleración, el enfoque se probó en combinación con los métodos de ND y PF y dio resultados satisfactorios para robots holonómicos circulares, con planes de extenderlo a arquitecturas no holonómicas, no circulares.

## 12.5. ARQUITECTURAS DE NAVEGACIÓN

El estudio de las arquitecturas de navegación es el estudio de los diseños de principio para los módulos de software que constituyen un sistema de navegación de robot móvil. El uso de una arquitectura de navegación bien diseñada tiene una serie de ventajas concretas.

### 12.5.1. Modularidad para la reutilización e intercambio de código

Por ejemplo, se puede introducir un telémetro de láser Sick a un robot que previamente usó solamente telémetros ultrasónicos. O se puede probar un robot navegador existente en un nuevo ambiente donde hay obstáculos que los sensores no pueden detectar, por lo que exigen una nueva representación de la planificación de ruta. Sería increíble cambiar parte de la competencia del robot sin causar una serie de efectos secundarios que obligan a revisar el funcionamiento de otras competencias del robot. Por ejemplo, sería genial conservar el módulo de evasión de obstáculos intacto, incluso cuando el juego del sensor particular de alcance cambie. En un ejemplo más extremo, sería ideal si el módulo de evasión de obstáculos no holonómico pudiera permanecer intacto incluso cuando la estructura cinemática del robot cambia de un chasis de tres ruedas a un chasis de tracción diferencial.

### 12.5.2. Localización de control

La localización del control de robot es una cuestión aún más crítica en la navegación de un robot móvil. La razón básica es que una arquitectura de robot incluye varios tipos de funcionalidad de control (por ejemplo, evasión de obstáculos, planificación de la trayectoria, ejecución de trayectoria, etc.) Al localizar cada funcionalidad para una unidad específica en la arquitectura, se hace posible la comprobación individual, así como una estrategia basada en principios para la composición de control. Por ejemplo, se considera la evasión colisiones. Para la estabilidad en el frente del software del robot cambiante, así como para la verificación enfocada que el sistema de evasión de obstáculos se aplica correctamente, es valioso localizar todo el software relacionado con el proceso de evasión de obstáculos del robot. En el otro extremo, un alto nivel de planificación y toma de decisiones de tareas se requieren para que los robots realicen un papel útil en su entorno. También es útil localizar tal software para la toma de decisiones de alto nivel, lo que le permite ser probado exhaustivamente en la simulación y verificado por tanto, incluso sin una conexión directa con el robot físico.

Cuadro 12.1: Enfoques de evasión de obstáculos (Parte 1)

Histograma de campo vectorial (VFH)			Bug			Método			
VFH*	VFH+	VFH	Tangent Bug	Bug2	Bug1	Forma	Fidelidad del modelo.	Vista	Otros requisitos
Circulo	Circulo	Simplicista	Punto	Punto	Punto		Cinemática		Mapa local
Básica	Básica						Dinámica		Mapa Global
									Planeador de ruta
Esencialmente Local	Local	Local	Local	Local	Local				Sensores
Cuadrícula histograma	Cuadrícula histograma	Cuadrícula histograma	Grafica tangente local						Robots probados
									Tiempo de ciclo
									Arquitectura
Sonar	Sonar	De alcance	De alcance		Táctil				Observaciones
No holonómico (GuideCane)	No holonómico (GuideCane)	Synchro-drive (Hexagonal)							
6 ... 242 ms	6 ms	27 ms							
66 MHz, 486 PC	66 MHz, 486 PC	20 MHz, 386 AT							
Pocos mínimos locales,	Mínimos locales,	Mínimos locales, Trayectorias oscilantes	Robusto, eficiente en muchos casos	Robusto, ineficiente	Robusto, Muy eficiente				

Fuente: Autores de proyecto

Cuadro 12.2: Enfoques de evasión de obstáculos (Parte 2)

Ventana dinámica		Velocidad de curvatura.		Banda de burbuja		Método	Fidelidad del modelo.	Vista	Otros requisitos			Rendimiento			
Ventana dinámica global	Enfoque de ventana dinámica	Metodice curvatura de carril	Método de velocidad de curvatura	Banda de burbuja	Banda elástica				Forma	Mapa local	Mapa Global	Planeador de ruta	Sensores	Robots probados	Tiempo de ciclo
Circulo	Circulo	Circulo	Circulo	C-Espacio	C-Espacio	Forma									
(Holonómico)	Exacto	Exacto	Exacto	Exacto	Exacto	Cinemática									
Básica						Dinámica									
Global	Local	Local	Local	Local	Local	Vista									
Rejilla de espacio-C	Cuadrícula histograma	Cuadrícula histograma	Cuadrícula histograma				Mapa local								
NF1							Mapa Global								
Escáner laser FOV SCK 180°	Anillo de sonar 24, Anillo infrarrojo 56 , cámara estéreo	Anillo de sonar 24, Laser FOV 30°	Anillo de sonar 24, Laser FOV 30°				Planeador de ruta								
Holonómico (circular)	Synchro-drive (Circular)	Synchro-drive (circular)	Synchro-drive (circular)												
6,7 ms	250 ms	125 ms	125 ms												
450 MHz, 486 PC	486 PC	200 MHz, Pentium	66 Mhz, 486 PC												
Giro en pasillos	Mínimos locales	Mínimos locales	Mínimos locales, giro en pasillos												

Fuente: Autores de proyecto

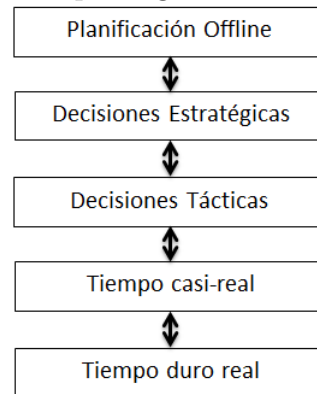
Cuadro 12.3: Enfoques de evasión de obstáculos (Parte 3)

Otro				Método			Fidelidad del modelo.			Otros requisitos			Rendimiento		
Método de Gradiente	Diagrama de cercanía global	Diagrama de cercanía	Schlegel	Forma	Cinemática	Dinámica	Vista	Mapa local	Mapa Global	Planeador de ruta	Sensores	Robots probados	Tiempo de ciclo	Arquitectura	Observaciones
Círculo	Círculo (pero formulación general)	Círculo (pero formulación general)	Polígono		Exacta	Básica	Global								
Exacto	(Holonómica)	(Holonómicos)													
Básica															
Global	Global	Local													
	Cuadrícula														
Espacia perceptual local	NF1		Cuadrícula												
			Frente de onda												
Sensor de distancia 180°	Escáner laser FOV SCK 180°	Escáner laser FOV SCK 180°	Escáner laser FOV 360°												
No holonómico (Aprox. circular)	Holonómico (circular)	Holonómico circular	Synchrodrive (Circular), Triciclo												
100 ms. Core algorithm: 10ms															
266 MHz, Pentium															
		Mínimos locales	Permite cambio de forma												

Fuente: Autores de proyecto

A continuación se definen estos dos tipos de descomposición y se presenta una introducción a los comportamientos, que son una herramienta general para la aplicación de la descomposición control.

Figura 12.23: Descomposición temporal genérica de una arquitectura de navegación



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

### 12.5.3. Técnicas para la descomposición

Las descomposiciones identifican los ejes a lo largo de los cuales se puede justificar la discriminación de software del robot en distintos módulos. Las descomposiciones también sirven como una manera de clasificar varios robots móviles en una taxonomía más cuantitativa.

#### Descomposición temporal

Una descomposición temporal de software del robot distingue entre los procesos que tienen diferentes demandas en tiempo real y no real. La figura 12.23 muestra una descomposición temporal genérica para navegación. En esta figura, la mayoría de los procesos en tiempo real se muestran en el fondo de la pila (montón), con la categoría más alta siendo ocupada por procesos sin demandas en tiempo real. El nivel más bajo en este ejemplo captura la funcionalidad que debe proceder con un tiempo de ciclo rápido garantizado, tal como un ancho de banda de 40 Hz. En contraste, una capa de tiempo casi real, puede capturar procesos que requieren, por ejemplo, 0.1 segundos de tiempo de respuesta, con grandes ciclos de tiempo individuales permisibles en el peor

de los casos. Cuatro tendencias importantes y relacionadas entre sí se correlacionan con la descomposición temporal. Estas no son inamovibles, hay excepciones. Sin embargo, estas propiedades generales de descomposiciones temporales son esclarecedoras.

**Tiempo de respuesta del sensor.** El tiempo de respuesta del sensor de un módulo particular, se puede definir como la suma de tiempo entre la adquisición de un evento de sensor y un cambio correspondiente en la salida del módulo. A medida que se avanza en la pila en la figura 12.23, el tiempo de respuesta del sensor tiende a aumentar. Para los módulos de nivel más bajo, el tiempo de respuesta del sensor a menudo está limitado sólo por el procesador en bruto y las velocidades del sensor. En los módulos de más alto nivel, la respuesta del sensor puede ser limitada por unos procesos de toma de decisiones lentos y deliberados.

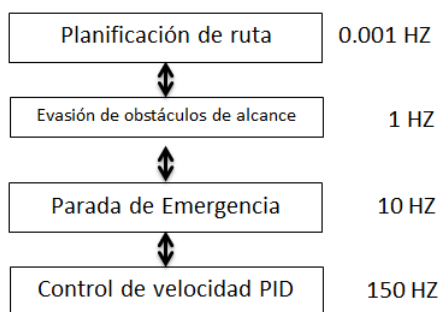
**Profundidad temporal.** La profundidad temporal es un concepto útil para la aplicación de la ventana temporal que afecta a la salida del módulo, tanto hacia atrás y hacia adelante en el tiempo. El *Horizonte temporal* describe la cantidad de mirada hacia el futuro utilizado por el módulo durante el proceso de selección de una salida. La *Memoria temporal* describe el período histórico de la entrada del sensor que se utiliza por el módulo para determinar la siguiente salida. Los módulos de más bajo nivel tienden a tener muy poca profundidad temporal en ambas direcciones, mientras que los procesos deliberativos de módulos de más alto nivel hacen uso de una memoria temporal grande y consideran las acciones basadas en sus consecuencias a largo plazo, tomando nota de grandes horizontes temporales.

**Localidad espacial.** De la mano con el lapso temporal, el impacto espacial de las capas aumenta dramáticamente a medida que se pasa de módulos de bajo nivel a los módulos de alto nivel. Los módulos en tiempo real tienden a controlar la velocidad de la rueda y la orientación, controlando el comportamiento localizado espacialmente. Una toma de decisiones estratégicas de alto nivel tiene poco o ningún efecto en la posición local, pero informa la posición global en el futuro.

**Contexto de especificidad.** Un módulo toma decisiones en función no sólo de sus entradas inmediatas sino también como una función del contexto del robot como capturado por otras variables, tales como la representación del medio ambiente del robot. Los módulos de más bajo nivel tienden a producir salidas directamente como resultado de entradas inmediatas del sensor, utilizando un pequeño contexto y por lo tanto siendo un contexto relativamente insensible. Los módulos de más alto nivel tienden

a exhibir especificidad de contexto muy alta. Para la toma de decisiones estratégicas, teniendo en cuenta los valores de los sensores mismos, los resultados son totalmente diferentes, sin embargo concebibles en función de otros parámetros contextuales.

Figura 12.24: Muestra de descomposición temporal de cuatro niveles de una navegación simple de un robot móvil.



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

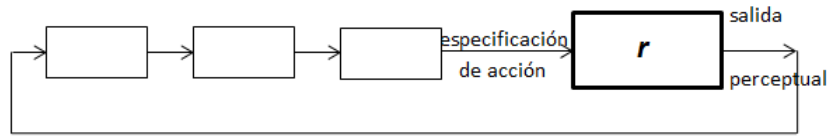
Un ejemplo que demuestra estas tendencias es representado en la figura 12.24, que muestra una descomposición temporal de una arquitectura simplista de navegación en cuatro módulos. En el nivel inferior, el bucle de control PID proporciona realimentación para controlar las velocidades del motor. Un módulo de parada de emergencia de corto alcance utiliza sensores ópticos y parachoques para cortar la corriente a los motores cuando se predice una colisión inminente.

### Descomposición de control

Mientras que la descomposición temporal discrimina basándose en el comportamiento de tiempo de los módulos del software, la descomposición de control identifica la forma en que la salida de cada módulo contribuye a las salidas de control global del robot. La presentación de la descomposición de control requiere que el evaluador comprenda los principios básicos de los sistemas de representación y análisis discreto.

La descomposición de control discrimina entre los diferentes tipos de vías de control a través de la porción de este sistema que comprende el algoritmo del robot. En un extremo, representado en la figura 12.24 se puede considerar una vía de control perfectamente lineal o secuencial. Tal sistema serial utiliza el estado interno de todos los módulos asociados y el valor de la percepción del robot  $I$  de una manera secuencial para calcular la siguiente acción del robot  $O$ . Una arquitectura serial pura tiene ventajas

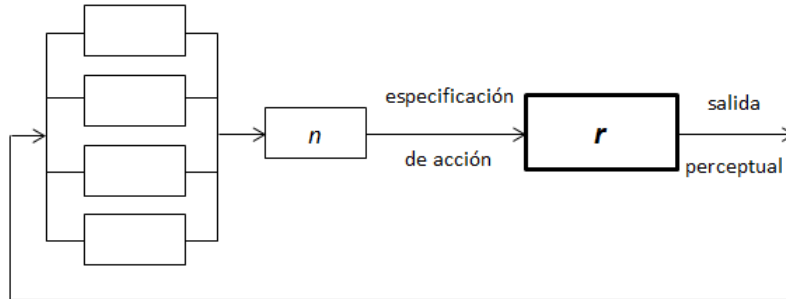
Figura 12.25: Ejemplo de una descomposición serial pura



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

relativas a la previsibilidad y la verificabilidad.

Figura 12.26: Ejemplo de una descomposición paralela pura



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

La figura 12.26 representa el extremo opuesto de control serial puro, una arquitectura de control completamente paralela. Debido a que se define  $r$  como un módulo, con precisamente una entrada, este sistema en paralelo incluye un módulo especial  $n$  que proporciona una única salida para el consumo de  $r$ . Intuitivamente, el sistema completamente paralelo distribuye la responsabilidad para la salida del sistema de control  $O$  a través de varios módulos, posiblemente al mismo tiempo. En un sistema secuencial puro, el flujo de control es una secuencia lineal a través de una cadena de módulos. Aquí, el control de flujo contiene una combinación de pasos en cuyo punto el resultado de múltiples módulos puede impactar  $O$  de manera arbitraria. Así la paralelización de control lleva a una pregunta importante: ¿cómo la salida de cada componente del módulo informará la decisión general concerniente al valor de  $O$ ? Una simple combinación es la conmutación temporal. En este caso, llamada paralela conmutada, el sistema tiene una descomposición paralela pero en cualquier instante particular en el tiempo la salida

$O$  puede ser atribuida a un módulo específico. El valor de  $O$  puede por supuesto depende de un módulo diferente en cada instante de tiempo sucesivo, pero el valor instantáneo de  $O$  siempre se puede determinar basado en las funciones de un solo módulo. Por ejemplo, supongamos que un robot tiene un módulo de evasión de obstáculos y un módulo de seguimiento de trayectoria. Una aplicación de control conmutada puede implicar la ejecución de la recomendación del seguimiento de trayectoria cada vez que el robot está a más de 50 cm de todos los obstáculos detectados y ejecutar la recomendación de evasión de obstáculos cuando cualquier sensor informa de una distancia cercana de 50 cm.

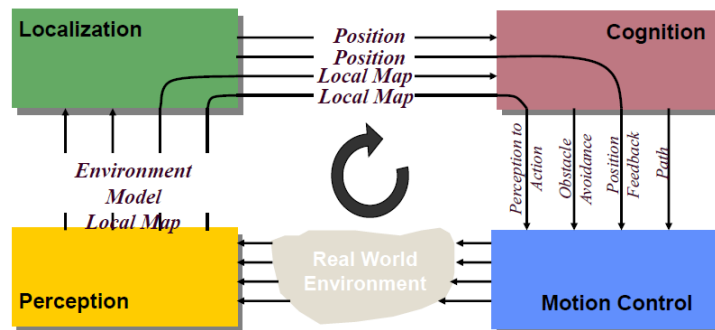
La ventaja de tal control conmutado es particularmente clara si la conmutación es relativamente excepcional. En contraste, el modelo paralelo mixto más complejo permite el control en cualquier momento dado para ser compartido entre varios módulos. Por ejemplo, el mismo robot podría tomar la salida del módulo de evasión de obstáculos en todo momento, convirtiéndolo en un vector de velocidad, y lo combina con la salida del módulo de seguimiento de ruta, utilizando la suma de vectores. Entonces, la salida del robot nunca se debería a un módulo único, sino que resultaría de la combinación matemática de las dos salidas de los módulos. El control paralelo mixto es más general que el control de conmutación, pero eso también muestra que es una técnica más difícil de usar también. Mientras que con el control de conmutación la mayoría del comportamiento pobre surge de un cambio inoportuno de comportamiento, en control mixto el comportamiento del robot puede ser muy pobre, incluso con más facilidad. La combinación de múltiples recomendaciones matemáticamente no garantiza un resultado que es globalmente superior, así como la combinación de múltiples vectores cuando se decide una dirección de viraje para evitar un obstáculo puede resultar en la decisión muy pobre al seguir hacia adelante. Por lo tanto, gran cuidado se debe tener en implementaciones mixtas de control paralelo a las fórmulas de mezcla y especificaciones individuales de módulos que conducen a la eficacia de resultados mixtos.

#### **12.5.4. Estudios de casos: arquitecturas escalonadas de robot**

Se han descrito descomposiciones temporales y de control de la arquitectura de robot, con el tema común de que la robótica está siempre compuesta de varios módulos juntos para compensar esa arquitectura. Es evidente que los comportamientos de robots juegan un papel importante en los niveles de control del robot en tiempo real, por ejemplo, seguimiento de ruta y evasión de obstáculos. A mayores niveles temporales,

más tareas tácticas necesitan modular la activación de comportamientos, o módulos, a fin de lograr el movimiento del robot a lo largo de la trayectoria deseada. Aún más arriba, un planificador global podría generar caminos para proporcionar tareas tácticas con previsión global. En esta representación, los arcos representan aspectos de la competencia en tiempo real y no real. Por ejemplo, la evasión de obstáculos requiere poca entrada del módulo de localización y consiste en tomar decisiones rápidas en el nivel de cognición seguidos de ejecución en el control del movimiento. En contraste los ciclos de retroalimentación de posición del PID evitan todo procesamiento de alto nivel, vinculando la percepción de los valores de encoder directamente a los ciclos de control PID de más bajo nivel en el control del movimiento. La trayectoria de arcos a través de los cuatro módulos de software proporciona la información temporal en una representación.

Figura 12.27: El ejemplo básico arquitectónico utilizado a lo largo de este estudio



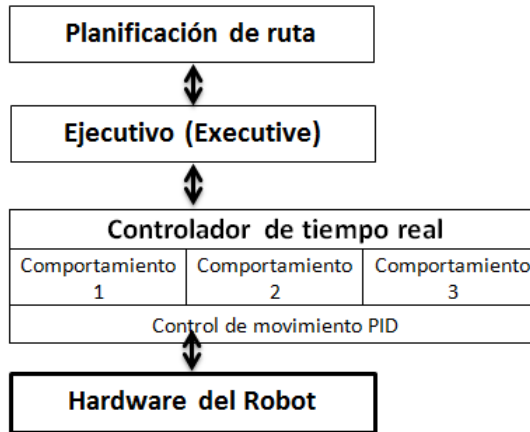
Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

La figura 12.28 muestra una arquitectura genérica por niveles basada en el enfoque de Pell y sus colegas utilizado en el diseño de una nave espacial autónoma, Deep Space One (*An autonomous spacecraft agent prototype, 1998*). Esta figura es similar a la figura 12.24 en la presentación de una descomposición temporal de la competencia de robot. Sin embargo, los límites que separan cada módulo de los módulos adyacentes son específicos de la navegación del robot.

La planificación de trayectoria encarna una toma de decisiones a nivel estratégico para el robot móvil. La planificación de ruta utiliza toda la información global disponible en tiempo no real para identificar la secuencia correcta de acciones locales para el robot. En el otro extremo, el control en tiempo real representa las competencias que requieren gran ancho de banda y ciclos de control de sensor/efector ajustados. En su nivel más bajo, esto incluye ciclos PID de velocidad del motor. Por encima de aquellos, el control

en tiempo real incluye también un bajo nivel de comportamientos que pueden formar un conmutador o una arquitectura paralela mixta.

Figura 12.28: Una arquitectura de navegación escalonada general basada en una descomposición temporal



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

Entre el planificador de ruta y los niveles de control en tiempo real se encuentra el *Ejecutivo (executive)*, que se encarga de mediar la relación entre la planificación y la ejecución. El ejecutivo es responsable de gestionar la activación de comportamientos basados en la información que recibe del planificador. El ejecutivo también es responsable del reconocimiento de falla, ahorrando (colocando el robot en un estado estable), e incluso reiniciando el planificador cuando sea necesario. Es el ejecutivo en esta arquitectura el que contiene toda toma de decisiones tácticas, así como actualizaciones frecuentes de memoria a corto plazo del robot, como es el caso de la localización y mapeo.

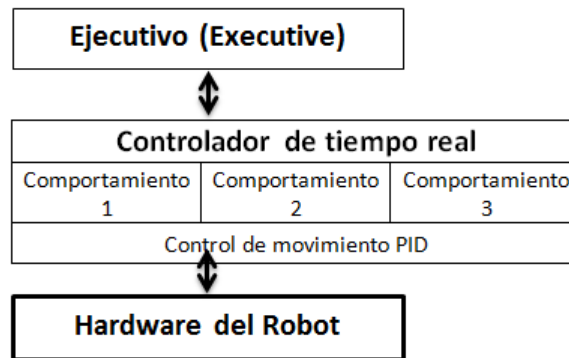
Aunque la arquitectura general que se muestra en la figura 12.28 es útil como un modelo para la navegación del robot, las implementaciones variantes en la comunidad robótica pueden ser bastante diferentes. A continuación, se presentan tres versiones particulares de la arquitectura por niveles generales, describiendo para cada versión por lo menos una aplicación de robot móvil en el mundo real.

### Planificación offline

Se considera la figura 12.29, en la que sólo hay dos niveles de software. En tales arquitecturas de navegación, el ejecutivo (executive) no tiene un planificador a su dis-

posición, sino que deben contener a priori todos los esquemas pertinentes para viajar a los destinos deseados. La estrategia de dejar fuera a un planificador en conjunto, es por supuesto, muy limitante. Mover un robot a un nuevo entorno exige una nueva instancia del sistema de navegación, y por ello este método no es útil como una solución general al problema de navegación. Sin embargo, tales sistemas robóticos sí existen, y este método puede ser útil en dos casos.

Figura 12.29: Una arquitectura de dos niveles para la planificación offline



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

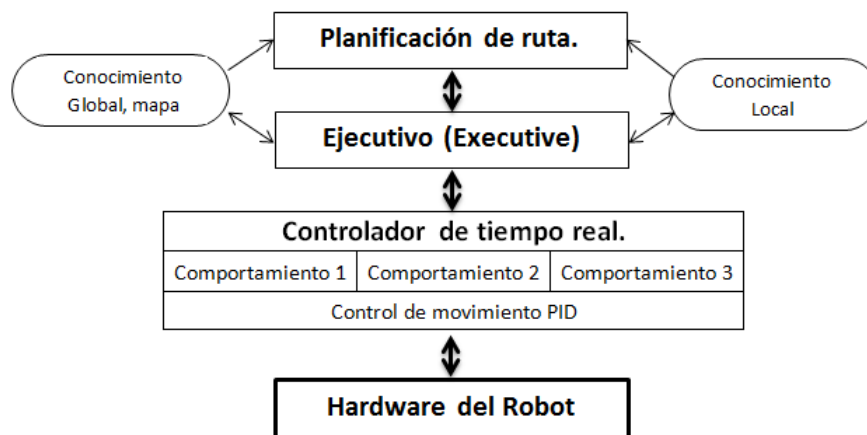
**Aplicaciones de ruta estática.** En las aplicaciones de robots móviles en las que el robot funciona en un entorno completamente estático mediante un sistema de navegación de ruta, es concebible que el número de posiciones objetivo discretas es tan pequeño que la representación ambiental directamente puede contener rutas a todos los puntos objetivos deseados. Por ejemplo, en la configuración de una fábrica o un almacén, un robot puede recorrer una ruta de bucle único, siguiendo un hilo guía enterrado. En tales aplicaciones industriales, los sistemas de planificación de trayectoria son a veces totalmente innecesarios cuando un conjunto precompilado de soluciones de ruta puede ser fácilmente generado por los programadores del robot.

**Demandas de extrema fiabilidad.** Como era de esperarse, otra razón para evitar planificación en línea es maximizar la fiabilidad del sistema. Dado que el software de planificación puede ser la parte más sofisticada del sistema de software de un robot móvil, y dado que en teoría al menos, la planificación puede tomar un tiempo exponencial de la complejidad del problema, imponer fuertes restricciones temporales sobre la planificación exitosa no es difícil, si no imposible.

## Planificación episódica

La planificación episódica es el método más popular en navegación de robots móviles hoy en día. Como se muestra en la figura 12.30, la estructura es de tres niveles, como es la arquitectura general de la figura 12.28. La intuición detrás del papel del planificador es como sigue. La planificación es computacionalmente intensiva, por lo que con demasiada frecuencia la planificación tendría serias desventajas. Pero el ejecutivo se encuentra en una excelente posición para identificar cuándo se ha encontrado suficiente información (por ejemplo, a través de la extracción de características) para garantizar un cambio significativo en la dirección estratégica. En estos puntos, el ejecutivo invocará el planificador para generar, por ejemplo, un nuevo camino a la meta.

Figura 12.30: Una arquitectura de planificación episódica de tres niveles



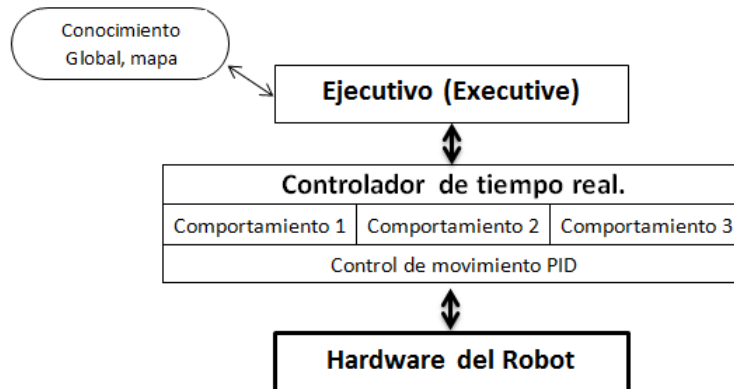
Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

En general, las arquitecturas de planificación episódicas son muy populares en la comunidad de investigación de la robótica móvil. Ellas combinan la versatilidad de dar respuesta a cambios ambientales y nuevos objetivos con respuesta rápida de un nivel táctico ejecutivo y los comportamientos que controlan el movimiento del robot en tiempo real. Como se muestra en la figura 12.30, es común en dichos sistemas para tener tanto un mapa local a corto plazo como un mapa global más estratégico. Parte del trabajo del ejecutivo en tales representaciones duales es decidir cuándo y si la nueva información integrada en el mapa local es lo suficientemente no transitoria para ser copiada en la base de conocimiento global.

## Planificación y ejecución integrada

La arquitectura que se muestra en la figura 12.31 puede ser similar a la arquitectura de la planificación offline de la figura 12.29, pero en realidad es mucho más avanzada. En este caso, el nivel planificador ha desaparecido porque ya no hay una descomposición temporal entre el ejecutivo y el planificador. La planificación es simplemente una pequeña parte del ciclo nominal de actividades del ejecutivo, cuando las representaciones locales y globales son las mismas. La ventaja de este enfoque es que las acciones del robot en cada ciclo son guiadas por un planificador de trayectoria global, y son por lo tanto óptimas en vista de toda la información que el robot ha reunido.

Figura 12.31: Una arquitectura integrada de planificación y ejecución en el que la planificación no es más que un paso de ejecución en tiempo real



Fuente: Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah R. Nourbakhsh, Second edition, 2011

El reciente éxito de los métodos de planificación y ejecución integrada pone de relieve el hecho de que el diseñador de una arquitectura de navegación del robot debe tener en cuenta no sólo todos los aspectos del robot y su tarea ambiental, sino también el estado del procesador, la GPU y la tecnología de memoria. Se espera que el diseño de la arquitectura de un robot móvil siga siendo un área activa de la innovación para los próximos años. Todas las formas de progreso tecnológico, desde invenciones de sensores de robot hasta el aumento de la velocidad del procesador, y haciendo un paralelo adicional es posible catalizar más innovaciones en la arquitectura de robots móviles, así como en años pasados algunas tácticas inimaginables se han vuelto realidad.

---

---

## CAPÍTULO 13

---

# FUTURO DE LA ROBÓTICA MÓVIL AUTÓNOMA

En este capítulo se mostrarán cuáles son las tendencias actuales de la robótica móvil autónoma, y de acuerdo a la evolución de ella en los últimos años, se hará una visualización de lo que a futuro se puede evolucionar, y cuál será el camino a seguir de ahora en adelante.

### 13.1. TENDENCIAS ACTUALES Y FUTURAS

En la actualidad hay varios campos que componen la robótica avanzada. Todos ellos tocan ramas muy dispares de la ingeniería, tales como la mecánica, la informática, la eléctrica, la electrónica, y de la ciencia, tales como física, anatomía, psicología, zoología. El fundamento de todas estas investigaciones es la Ciencia Cognitiva Corporizada y la Nueva Inteligencia Artificial. La finalidad de todo esto es la creación de robots inteligentes y autónomos, capaces de razonar, comportarse, evolucionar y actuar como las personas.

#### 13.1.1. Robótica situada

La Robótica Situada se ocupa de los robots que están insertos en entornos complejos y dinámicamente cambiantes. Se basa sobre dos ideas centrales, en las que se supone que los robots:

- “están corporizados” (embodiment), su cuerpo físico es apto para experimentar el entorno de manera directa y sus acciones tienen una realimentación inmediata sobre sus propias percepciones.
- “están situados” (situatedness), inmersos dentro de un entorno; interactúan con el mundo, el cual influye de forma directa sobre su comportamiento.

Existen tres tipos básicos de arquitectura de control, que dependen de las necesidades de la situación y de las características del entorno:

- Control reactivo: “*no pienses, actúa*”.
- Control deliberativo: “*primero piensa, y después actúa*”.
- Control híbrido: “*piensa y actúa independientemente en paralelo*”.

Aspectos como representación, memoria y aprendizaje toman un enfoque diferente al tradicional:

- La memoria no es un lugar donde se almacenen las cosas, sino una construcción dinámica de coordinaciones previas.
- Las representaciones son creadas por una interacción de procesos neurales y externos.
- El aprendizaje se da durante todo el tiempo, con cada acción, coordinación y movimiento, recompone todas las categorizaciones y secuencias de comportamientos previos.

### **Robótica situada**

- *Control reactivo*. Es una técnica que une fuertemente las entradas de los sensores con las salidas de los actuadores, de forma que el robot responda muy rápidamente a los cambios de su entorno.
- *Control deliberativo*. Usa la totalidad de la información sensorial disponible, y todos los conocimientos almacenados internamente, para decidir cómo actuar. El razonamiento es en forma de planificación, que es uno de los principales y más complejos componentes de la inteligencia artificial.
- *Control híbrido*. El control híbrido aúna las ventajas del control reactivo y del deliberativo: combinar la respuesta en tiempo real con la racionalidad y eficiencia de la deliberación. El sistema de control contiene ambos componentes, y éstos tienen que interactuar de forma que produzcan respuestas coherentes.

### 13.1.2. Robótica basada en el comportamiento de conducta

Se podría decir que es el cuarto tipo de arquitectura de control. Su lema es *“piensa según la forma en que actúes”*.

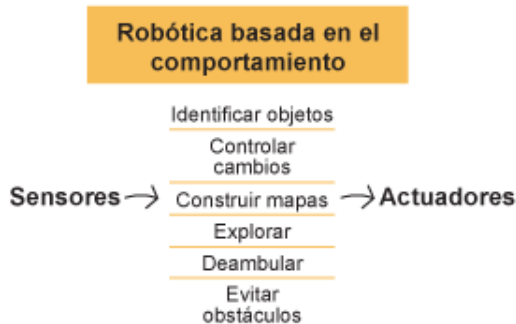
Figura 13.1: Robótica tradicional



Fuente: Robots inteligentes autónomos de nueva generación, Carolina Cebrecos del Castillo, Abel Lozano Prieto, Agustín Nieto Domínguez

Los robots basados en comportamiento son entidades físicas situadas en un entorno en el que los datos detectados por sus sensores actúan directamente en las respuestas de éste como se ve en la figura 13.2.

Figura 13.2: Robótica basada en el comportamiento



Fuente: Robots inteligentes autónomos de nueva generación, Carolina Cebrecos del Castillo, Abel Lozano Prieto, Agustín Nieto Domínguez

La inteligencia de este sistema surge de la interacción con su entorno debido a

la vinculación entre su percepción y sus acciones. Están basados en una arquitectura modular relativamente simple, que codifica diversas conductas para conseguir una determinada funcionalidad. Estas funciones pueden ser la de evitar obstáculos, caminar, levantarse, etc. Las funciones inteligentes del sistema, tales como percepción, planificación, modelado, aprendizaje, etc. emergen de la interacción entre los distintos módulos y el entorno físico en donde está inmerso el robot. Las pautas básicas de la robótica basada en comportamiento son:

1. Principio de modularidad diferente.
2. Desarrollo incremental de la inteligencia del sistema.
3. Descomposición comportamiento en un conjunto de comportamientos simples.
4. Cada comportamiento resolverá alguna tarea y podrá ser observado externamente.
5. Un conjunto de comportamientos proporcionaran al robot un nivel de competencia (prestaciones globales) determinado.
6. En el caso de que se sumen nuevos comportamientos, el nivel de competencia aumentará automáticamente.

### 13.1.3. Robótica cognitiva

La Robótica Cognitiva y el campo de la Consciencia Artificial son dos campos que están muy relacionados entre sí. En realidad, no hay una funcionalidad de la consciencia que no esté enmarcada dentro del ámbito de la arquitectura cognitiva. La Robótica Cognitiva se encarga de implementar robots capaces de percibir, razonar y actuar ante entornos dinámicos e imprevisibles. El principal objetivo es la implementación de máquinas que sean capaces de “saber lo que hacen”; máquinas más robustas, adaptativas y flexibles.

#### Algunos ejemplos de aplicaciones en robótica cognitiva

- **Robots de servicio** robots sociales y robots asistentes personales. Un ejemplo de estos robots es “Maggie”, desarrollado por Robotics Labs de la Universidad Carlos III de Madrid. El robot “Maggie” es un “humanoide” capaz de sentir cosquillas, de bailar, de dar las últimas informaciones de actualidad o de distinguir medicinas.
- **Robots autónomos de exploración espacial** robots asistentes para montaje y reparación orbital, como “Robonaut”, desarrollado por la NASA.

- **Jugadores de la Robocup** es un proyecto internacional para promover, a través de competiciones integradas por robots autónomos, la investigación y educación sobre inteligencia artificial.
- **Robots autónomos de rescate y emergencias** como “BEAR”, desarrollado por Vecna Robotics. Está diseñado para recoger o rescatar gente herida. “BEAR” puede levantar grandes pesos (hasta 135Kg) y cargar con ellos largas distancias. El objetivo principal es el rescate de humanos en zonas de difícil acceso. No es un robot autónomo, es controlado de forma remota por un operario humano.

### 13.1.4. Robótica Evolutiva

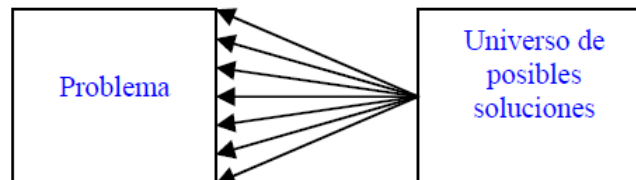
Basándose en los conocimientos obtenidos de las Ciencias Naturales (biología y etología) y de la Vida Artificial (redes neuronales, técnicas evolutivas y sistemas dinámicos) sobre robots reales, este acercamiento trata de que estos robots desarrollen sus propias habilidades en interacción íntima con el entorno y sin la intervención humana. Esta rama de la inteligencia artificial tiene sus raíces en tres desarrollos relacionados pero independientes entre sí:

- Algoritmos genéticos.
- Programación evolutiva.
- Estrategia evolutiva.

### Sistemas de programación tradicionales

Los sistemas de programación tradicionales se basan en probar exhaustivamente con todas las posibles soluciones y escoger la más adecuada. (Figura 13.3)

Figura 13.3: Sistemas de programación tradicionales

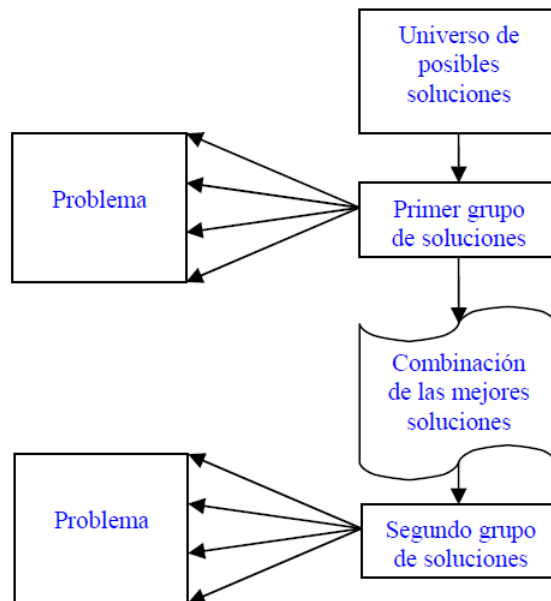


Fuente: Robots inteligentes autónomos de nueva generación, Carolina Cebrecos del Castillo, Abel Lozano Prieto, Agustín Nieto Domínguez

## Algoritmos genéticos

Estos algoritmos hacen evolucionar a una población inicial de individuos sometién-dola a acciones aleatorias semejantes a las que actúan en la evolución biológica, tales como mutaciones y recombinaciones genéticas, así como también a una selección de acuerdo con algún criterio, en función del cual se decide cuáles son los individuos mejor adaptados, que sobreviven y cuáles los menos aptos, que son descartados. El escenario de trabajo es el que se muestra en la figura 13.4. En líneas generales, todo algoritmo genético sigue los siguientes pasos:

Figura 13.4: Escenario de trabajo de los algoritmos genéticos

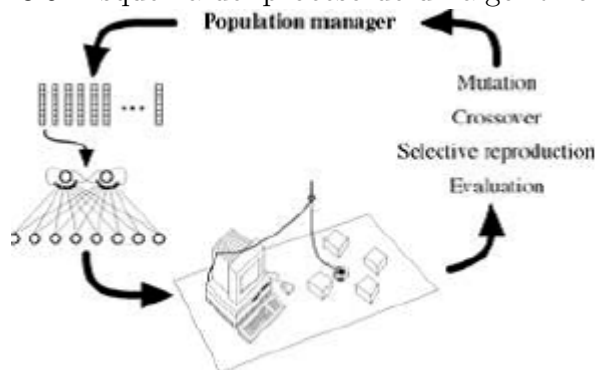


Fuente: Robots inteligentes autónomos de nueva generación, Carolina Cebrecos del Castillo, Abel Lozano Prieto, Agustín Nieto Domínguez

1. Inicializar aleatoriamente una población de soluciones a un problema, representadas por una estructura de datos adecuada.
2. Evaluar cada una de las soluciones, y asignarle una puntuación o fitness según lo bien que lo hayan hecho.
3. Escoger de la población la parte que tenga una puntuación mayor.
4. Mutar (cambiar) y entrecruzar (combinar) las diferentes soluciones de esa parte escogida, para obtener descendencia y reconstruir la población.

5. Repetir hasta que se haya encontrado la solución deseada.

Figura 13.5: Esquema del proceso de un algoritmo genético



Fuente: Robots inteligentes autónomos de nueva generación, Carolina Cebrecos del Castillo, Abel Lozano Prieto, Agustín Nieto Domínguez

El esquema se muestra en la figura 13.5.

### Programación evolutiva

Es prácticamente una variación de los algoritmos genéticos, donde lo que cambia es la representación de los individuos. En este caso, los individuos son tripletas (ternas) cuyos valores representan estados de un autómata finito. Cada terna está formada por:

- Estado actual.
- Cadena reconocida.
- Nuevo estado.

Estos valores se utilizan, como en un autómata finito, de la siguiente manera: cada terna es una regla que especifica las condiciones (cadena reconocida) que se deben cumplir para llegar a un estado (nuevo estado) suponiendo que la máquina se encuentra en un cierto estado (estado origen). Esta regla puede interpretarse de dos formas:

- $ESTADO + SUCESO \rightarrow NUEVO ESTADO$ : Esta regla tiene forma imperativa, y puede resumirse como “Si estoy realizando esta acción y veo esta información, debo obligatoriamente cambiar a realizar esta otra acción”.
- $ESTADO + ACCION \rightarrow NUEVO ESTADO$ : Esta regla tiene forma declarativa y puede resumirse como “Si veo esta información y realizo esta acción, estimo que como consecuencia de mi acción la próxima información será esta otra”.

## Estrategia evolutiva

Actualmente, se puede considerar como un tipo de algoritmo genético y viceversa. En robótica evolutiva, los robots funcionan con los programas de control creados y configurados siguiendo las estrategias anteriores.

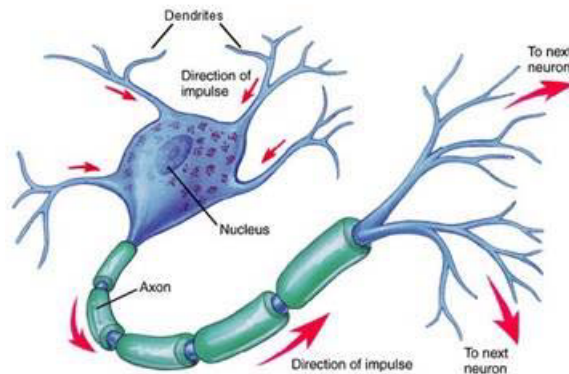
## Redes neuronales

Son sistemas que intentan simular el cerebro humano a nivel de neurona. El cerebro humano presenta tres características básicas:

- El conocimiento está distribuido en un gran número de neuronas en el cerebro.
- Las neuronas se comunican unas con otras.
- El cerebro es adaptable, es decir, puede aprender.

Todas las neuronas tienen una estructura similar, presentan unas conexiones de entrada llamadas dendritas y una conexión de salida llamada axón, como muestra la figura 13.6:

Figura 13.6: Estructura de una neurona

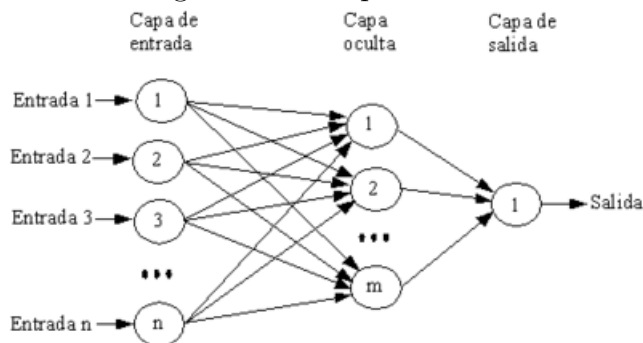


Fuente: Robots inteligentes autónomos de nueva generación, Carolina Cebrecos del Castillo, Abel Lozano Prieto, Agustín Nieto Domínguez

Ese será precisamente el esquema que seguirán las redes neuronales artificiales (RNA) mostrado en la figura 13.7.

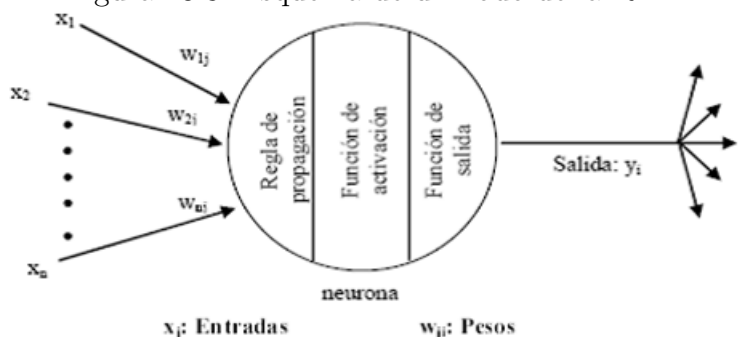
Cada nodo de esta red será el equivalente a una neurona en el cerebro y tendrá el siguiente esquema de la figura 13.8.

Figura 13.7: Esquema RNA



Fuente: Robots inteligentes autónomos de nueva generación, Carolina Cebrecos del Castillo, Abel Lozano Prieto, Agustín Nieto Domínguez

Figura 13.8: Esquema de un nodo de la RNA



Fuente: Robots inteligentes autónomos de nueva generación, Carolina Cebrecos del Castillo, Abel Lozano Prieto, Agustín Nieto Domínguez

Por el hecho de estar basadas en el cerebro humano, son muy usadas puesto que presentan grandes ventajas:

- *Aprendizaje:* Las RNA tienen la habilidad de aprender mediante una etapa que se llama etapa de aprendizaje.
- *Auto organización:* Una RNA crea su propia representación de la información en su interior, descargando al usuario de esto.
- *Tolerancia a fallos:* Debido a que una RNA almacena la información de forma redundante, ésta puede seguir respondiendo aceptablemente aún si se daña parcialmente.

- *Flexibilidad*: Una RNA puede manejar cambios no importantes en la información de entrada, como señales con ruido u otros cambios en la entrada.
- *Tiempo real*: La estructura de una RNA es paralela, por lo cual si esto es implementado con computadoras o en dispositivos electrónicos especiales, se pueden obtener respuestas en tiempo real.

### 13.1.5. Robótica Biomimética, Biorrobótica o Robótica Inspirada Biológicamente

Esta aproximación se ocupa de diseñar robots que funcionan como los sistemas biológicos, de ahí que se basen sobre las Ciencias Naturales (biología, zoología y etología) y la robótica. Dado que los sistemas biológicos realizan muchas tareas de procesamiento complejas con máxima eficiencia, constituyen una buena referencia para implementar sistemas artificiales que ejecuten tareas que los seres vivos realizan de forma natural (interpretación de la información sensorial, aprendizaje de movimientos, coordinación motora, etc.).

## 13.2. EVOLUCIÓN ARTIFICIAL

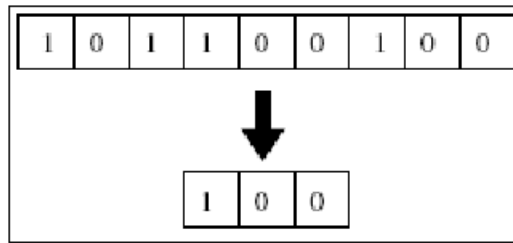
Esta investigación se enfoca en el uso de diversos procedimientos evolutivos: algoritmos genéticos, estrategias genéticas, programación genética, programación evolutiva y coevolución. Una de las aplicaciones más comunes de estos procedimientos es la optimización de redes neuronales artificiales utilizadas como arquitecturas de control. Aun siendo esta forma de aplicación la más usual, existen varias aplicaciones robóticas que no involucran redes neuronales artificiales como controladores.

### 13.2.1. Operadores Genéticos

Para un mejor entendimiento de la utilización de los operadores, primero se introducirán algunos conceptos básicos:

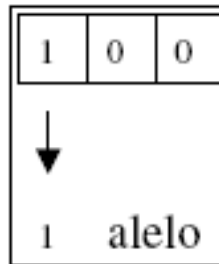
- *Cromosoma*. Dado un problema a resolver en evolución artificial, por ejemplo una evasión de obstáculos de un robot, un cromosoma corresponde a la representación de una estructura de datos que codifica los parámetros de una posible solución.
- *Gen*. Recibe el nombre de gen a una sección de un cromosoma, y este de forma individual codifica el valor de un solo parámetro que corresponde a una solución del problema.

Figura 13.9: Representación de cromosoma y gen.



Fuente: Coevolución para un esquema de estrategia presa-depredador, Tesis, Luis Miguel Pellegrin Zazueta, 2009

Figura 13.10: El valor 1 corresponde al alelo de esta posición del gen



Fuente: Coevolución para un esquema de estrategia presa-depredador, Tesis, Luis Miguel Pellegrin Zazueta, 2009

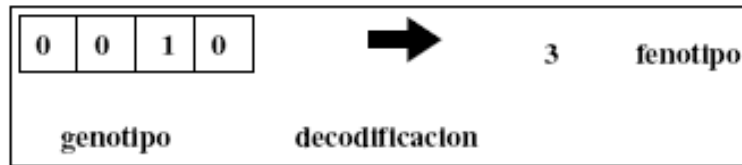
- *Alelo*. Corresponden a los valores individuales que puede tomar cada posición genética, ya sea en un gen o en un cromosoma. Por ejemplo, si la codificación es binaria un alelo toma el valor de 0 y 1.

La transformación de los parámetros de un problema a un cromosoma, recibe diferentes nombres de acuerdo a la etapa en que se encuentre dicha transformación, por ejemplo si se tiene una fórmula a representar ( $x + y = 5$ ), se necesitan transformar los datos (en este caso los operandos) a la representación que mejor convenga (ya sea binaria, entera o real), una vez que se elige la representación y se transforman los datos, esta transformación recibe el nombre de genotipo.

Las técnicas de selección pueden clasificarse en tres grandes grupos:

- *Selección proporcional*. Los individuos se eligen de acuerdo a su contribución de aptitud con respecto al total de la población.
- *Selección mediante torneo*. Los individuos se seleccionan con base en comparaciones directas de los individuos.

Figura 13.11: Representación de un genotipo binario y su posterior decodificación a fenotipo



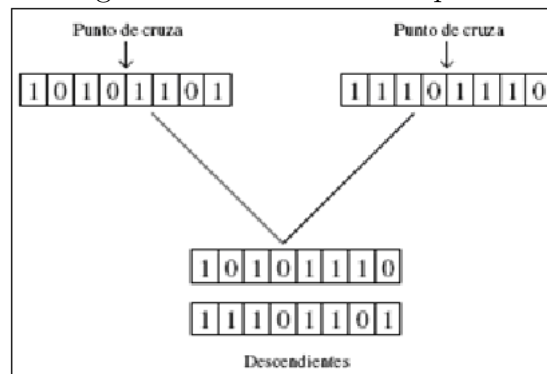
Fuente: Coevolución para un esquema de estrategia presa-depredador, Tesis, Luis Miguel Pellegrin Zazueta, 2009

- *Selección de estado uniforme.* Solo unos cuantos individuos son reemplazados en cada generación (los menos aptos).

Dentro de los métodos usuales para llevar a cabo este operador existen los siguientes:

- *Cruce de un punto:* Se selecciona una posición en los cromosomas de los individuos seleccionados (generalmente se trabaja con dos), y se intercambian los genes a la izquierda y derecha de esta posición, es decir, el punto de cruce divide el cromosoma en dos partes, del individuo uno tiene las partes A1 y B1, al igual el individuo dos forma las partes A2 y B2, entonces las diferentes partes se combinan creando dos nuevos individuos. (Figura 13.12).

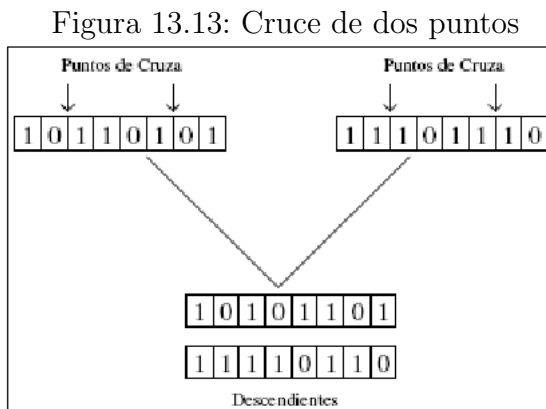
Figura 13.12: Cruce de un punto



Fuente: Coevolución para un esquema de estrategia presa-depredador, Tesis, Luis Miguel Pellegrin Zazueta, 2009

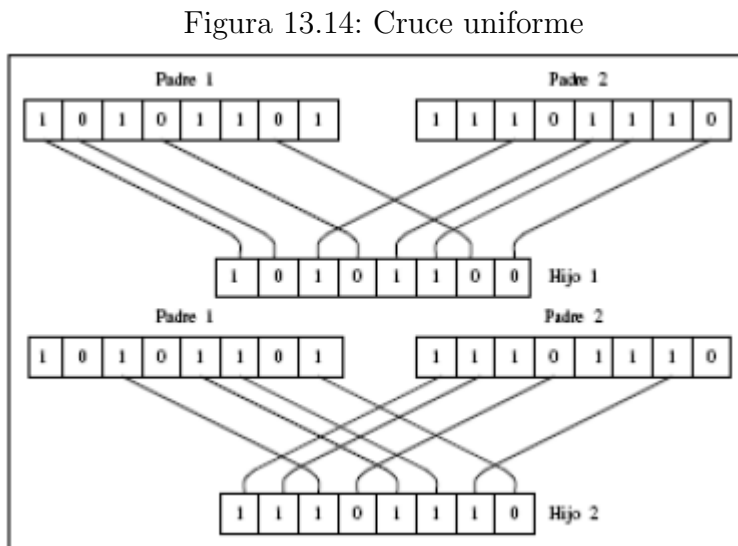
- *Cruce de n puntos:* Es una generalización del método anterior. Se seleccionan varias posiciones  $n$  en los cromosomas de los progenitores y se intercambian genes

de las partes diferentes entre los individuos, cuidando de generar descendientes con el mismo número de cromosomas que sus progenitores. (Figura 13.13).



Fuente: Coevolución para un esquema de estrategia presa-depredador, Tesis, Luis Miguel Pellegrin Zazueta, 2009

- *Cruce uniforme*: Se realiza una prueba aleatoria para decidir de cuál de los progenitores se toma cada posición del cromosoma como se muestra en la figura 13.14.



Fuente: Coevolución para un esquema de estrategia presa-depredador, Tesis, Luis Miguel Pellegrin Zazueta, 2009

- *Cruces para permutación*: Existe una familia de cruces específicas para los problemas de permutación, que incluyen cruces de mapeo parcial, cruces de orden y

cruces de ciclo.

### 13.2.2. Evaluación de Calidad

En robótica, la evaluación se realiza mediante una función de calidad que mide el rendimiento de un controlador para un robot, en este caso la calidad representa que tan bien desarrolla el robot su tarea en el entorno durante su tiempo de ejecución. En robótica evolutiva es posible aplicar la función de calidad a un cromosoma en dos perspectivas diferentes: una perspectiva local y una perspectiva global. Un ejemplo de aplicación de función de calidad con una perspectiva local, es la que realizan los investigadores Florean y Mondada [Florean and Mondada, 1998], en el Instituto Federal de Tecnología en Lausana, utilizan un robot Khepera para tareas de navegación, en las que no requiere el seguimiento de paredes, hacen uso de la siguiente fórmula:

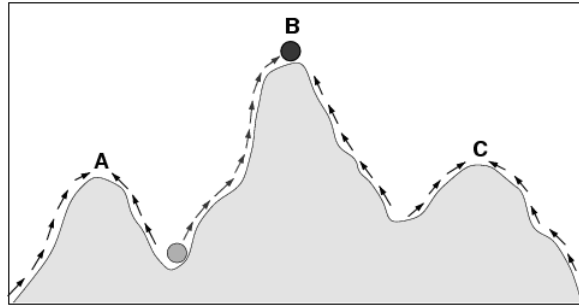
$$calidad = V \cdot (1 - \sqrt{DV}) \cdot (1 - I) \quad (13.1)$$

El término  $V$  representa la velocidad media de las dos ruedas del robot Khepera,  $DV$  el valor absoluto de la diferencia de velocidades de las ruedas, e  $I$  es el valor de detección normalizado devuelto por el sensor de infrarrojos con mayor nivel de activación. De esta forma el primer término maximiza la velocidad, el segundo el movimiento recto y el tercero la distancia a objetos. La fórmula utiliza producto y no sumas de los términos, con ello se tiende a optimizar los tres aspectos. El algoritmo que hace uso de esta fórmula llega a converger cuando se obtiene un equilibrio entre los tres términos.

### 13.2.3. Hipersuperficie o Paisaje de Calidad

Para generar un paisaje de calidad en un algoritmo genético primero se introduce una función de calidad que evalúa que tan satisfactoriamente se desempeña el individuo ante el problema que se busca resolver, después se realiza un mapeo geométrico de todos estos valores generados por la función de calidad, a esta representación gráfica se le llama paisaje de calidad. En la figura 13.15; el pico donde se encuentra el punto B representa al mejor individuo de la población, y los puntos A y C son los llamados máximos locales (individuos que pueden ser soluciones al problema, sin embargo no son las mejores), el punto que se localiza entre A y B representa un valle.

Figura 13.15: Representación hipotética de un paisaje de calidad



Fuente: Coevolución para un esquema de estrategia presa-depredador, Tesis, Luis Miguel Pellegrin Zazueta, 2009

### 13.2.4. Técnicas de Evolución Artificial

Hasta este apartado se han explicados los diferentes operadores aplicados en las técnicas de evolución artificial, así como la importancia y los tipos de aplicaciones de la función de calidad. El siguiente paso es la descripción de cada una de estas técnicas de evolución artificial relevantes.

#### Algoritmos Genéticos

En AG se genera una población de individuos de forma aleatoria, para cada generación se evalúa la población asignando la calidad a cada uno de sus miembros. Esta calidad es vista como un índice de adaptación al entorno, representando las posibilidades que tiene este individuo de procrear. Los individuos procrean de acuerdo a su calidad asignada, posteriormente la población sufre procesos de mutación y se genera una nueva población para continuar el mismo proceso en esta nueva generación. Resumiendo el funcionamiento en general de los AG, estos requieren de la aplicación de los siguientes cinco componentes básicos para llevar a cabo su aplicación:

1. Una representación de las soluciones potenciales del problema.
2. Una forma de crear una población inicial de posibles soluciones (normalmente un proceso aleatorio).
3. Una función de evaluación que juegue el papel del ambiente, clasificando las soluciones en términos de su aptitud.
4. Operadores genéticos (cruza y mutación) que alteran la composición de los descendientes que se producirán para las siguientes generaciones.

5. Valores para los diferentes parámetros que utiliza el AG (tamaño de la población, probabilidad de la cruce, probabilidad de la mutación, número máximo de generaciones).

### **Algoritmos Genéticos Paralelos.**

Un algoritmo puede considerarse paralelo si en cualquier momento de su ejecución puede ejecutarse instancias del mismo. Bajo este paradigma de programación existen tres enfoques diferentes para paralelizar un AG:

- **Grano grueso.** En este enfoque cada AG es ejecutado en un procesador diferente. Donde cada AG contiene una población aislada de individuos compartiendo los mejores individuos encontrados mediante el operador migración.
- **Grano fino.** En este caso, cada procesador hace una parte de cada paso de un AG (selección, cruce o mutación) sobre la población común.
- **Enfoque híbrido.** Utiliza una implementación híbrida de las dos anteriores.

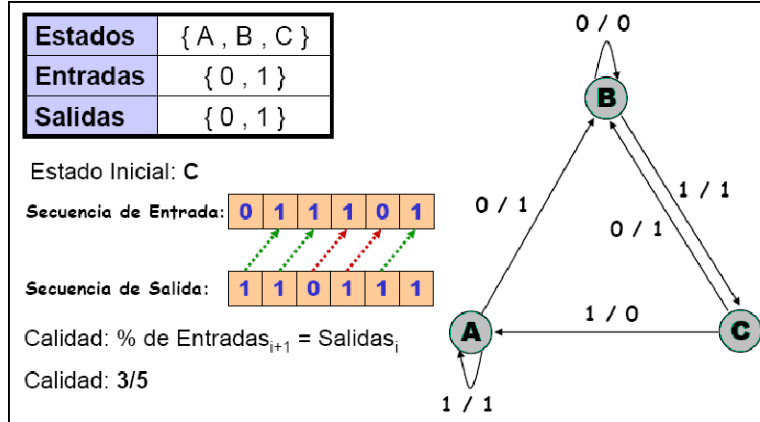
Después de  $k$  generaciones los procesadores tienen permitido compartir los mejores individuos con los otros procesadores. A este proceso de compartir individuos se le conoce como migración. Existen dos enfoques populares de migración:

- El modelo de la isla. En este tipo de modelo se permite enviar individuos a cualquier subpoblación (en cada uno de los procesadores participantes).
- El modelo de la pasarela (*Stepping Stone*). A diferencia del modelo anterior en este modelo la migración es limitada, ya que solo se permite la migración de individuos a las subpoblaciones vecinas.

### **Programación Evolutiva y Genética**

La programación evolutiva (PE) fue definida en los años 60, por Lawrence Fogel en la universidad de California, es basada en los algoritmos genéticos con variaciones en la representación de los individuos. En PE la evolución se realiza sobre un lenguaje de alto nivel, en donde los individuos son ternas (tripletas) cuyos valores representan estados de un autómata finito (figura 13.16). Cada terna está formada por: el valor del estado actual, un símbolo del alfabeto utilizado y el valor del nuevo estado.

Figura 13.16: Ejemplo de predicción de entradas con PE.



Fuente: Coevolución para un esquema de estrategia presa-depredador, Tesis, Luis Miguel Pellegrin Zazueta, 2009

## Coevolución

El término de coevolución se refiere a dos o más genotipos que evolucionan de forma simultánea y estos corresponden a problemas diferentes. Este proceso puede realizarse de dos tipos: cooperativa y competitiva.

### Coevolución Competitiva.

En el Instituto de Tecnología Federal de Lausanne en Suiza, y Stefano Nolfi, en el Departamento de sistemas Neuronales y Vida Artificial en Roma, se ha aplicado la coevolución competitiva para poblaciones de presas y depredadores que se utilizan como controladores en robots Khepera (figura 13.17).

En la figura 13.18, se muestra también el “efecto de la reina roja por el que la superficie de calidad de una de las poblaciones se está continuamente modificando por la evolución de la población contraria. En la gráfica de la izquierda se muestra la calidad medida en simulación (etiquetada con *py* para presa y *pr* para depredador), en la gráfica de la derecha la calidad de los mejores individuos en cada generación.

### Coevolución Cooperativa.

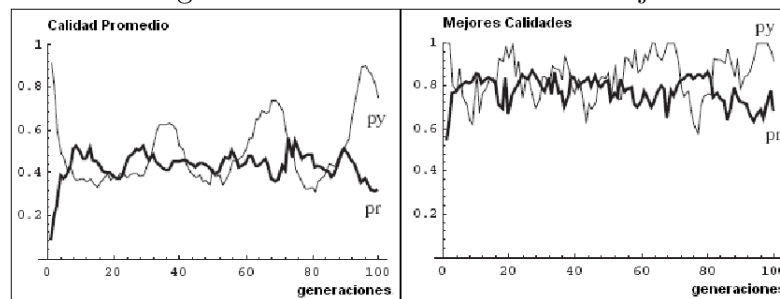
De esta forma se busca optimizar de modo automático la morfología adecuada para un robot y a su vez su controlador. En la arquitectura *Multi-Agent ESP* (figura 13.19), cada agente es construido mediante un ESP, esto es, emplea un grupo de subpoblaciones, en la que cada población contribuye con una neurona para construir la red neuronal

Figura 13.17: Robots Khepera utilizados por Nolfi y Floreano



Fuente: Coevolución para un esquema de estrategia presa-depredador, Tesis, Luis Miguel Pellegrin Zazueta, 2009

Figura 13.18: Efecto de la Reina Roja



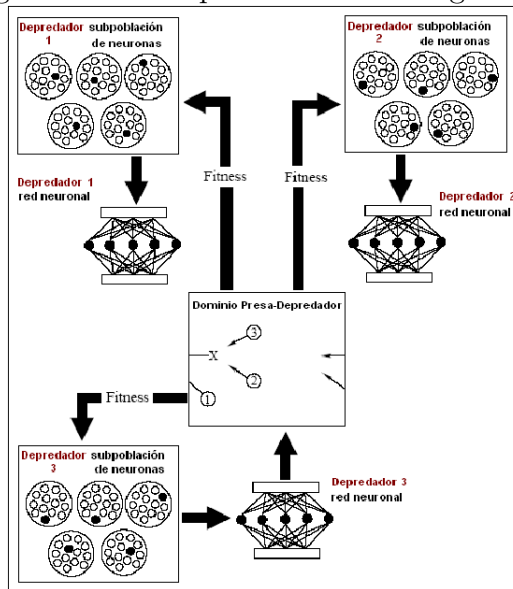
Fuente: Coevolución para un esquema de estrategia presa-depredador, Tesis, Luis Miguel Pellegrin Zazueta, 2009

que será evaluada en un dominio (el espacio de experimentación), así pues, se tiene que lograr un equilibrio entre este grupo de neuronas para constituir una buena calidad y es ahí donde se da la coevolución cooperativa.

### 13.2.5. Estructuras de Control para Robótica Evolutiva

Se tratarán estructuras de control en robótica que hacen uso de las técnicas de evolución artificial, ya sea para optimización o creación de estos controladores, con este breve análisis se busca seleccionar aquella estructura que muestre la mejor flexibilidad para el experimento.

Figura 13.19: Arquitectura Multi-Agent ESP



Fuente: Coevolución para un esquema de estrategia presa-depredador, Tesis, Luis Miguel Pellegrin Zazueta, 2009

## Sistemas Basados en Lógica Difusa

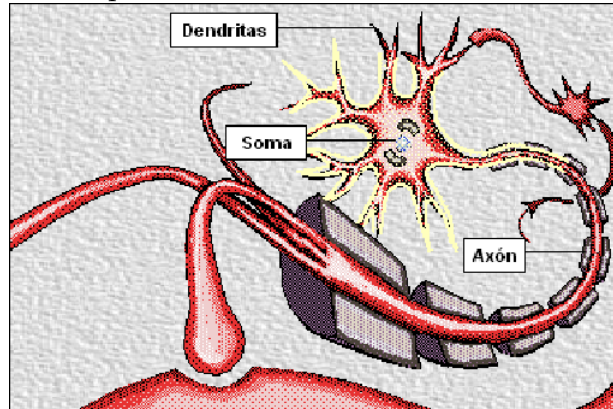
Los sistemas basados en lógica difusa emplean transiciones de pertenencia graduales que sirven para representar la incertidumbre o conceptos vagos, imprecisos o difusos.

## Redes Neuronales Artificiales

Una red neuronal artificial es básicamente definida como un sistema de procesamiento de información con ciertas características de rendimiento en común con las neuronas biológicas. Para entender el funcionamiento de las redes neuronales biológicas, se explicaran tres de sus componentes principales: sus dendritas, soma y axón (figura 13.20).

Así pues, una red neuronal artificial consiste de un variado número de elementos de procesamiento simples conocidos como neuronas. Cada neurona es conectada a otras mediante enlaces o vínculos de comunicación, cada uno de estos asociado a un peso. Estos pesos representan la información que utiliza la red para resolver el problema que se le plantea a la red. Además, cada neurona tiene un estado interno llamado activación o nivel de activación, el cual consiste en una función de la entrada que este recibe. A continuación en la figura 13.21, se muestran dos de las funciones de activación comúnmente utilizadas, y un esquema de funcionamiento de la neurona artificial es

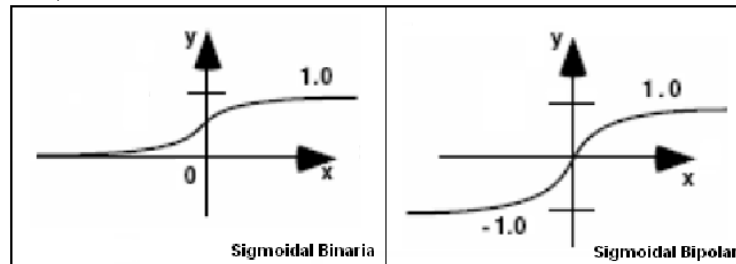
Figura 13.20: Esquema del funcionamiento una neurona biológica



Fuente: Coevolución para un esquema de estrategia presa-depredador, Tesis, Luis Miguel Pellegrin Zazueta, 2009

presentado en la figura 13.22.

Figura 13.21: Dos de las funciones de activación de mayor utilización, (sigmoideal binaria y sigmoideal bipolar)

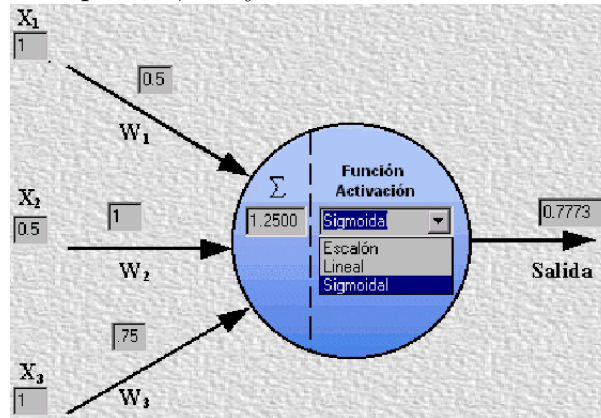


Fuente: Coevolución para un esquema de estrategia presa-depredador, Tesis, Luis Miguel Pellegrin Zazueta, 2009

### 13.3. ¿QUÉ SE PUEDE EVOLUCIONAR EN ROBÓTICA?

Se han estudiado las diferentes alternativas a la hora de utilizar la evolución simulada para automatizar el diseño de los controladores que gobiernan las acciones del robot. Sin embargo, la idea de automatización en el diseño se puede extender para incluir no sólo el controlador sino también la morfología adecuada del robot o del modelo de

Figura 13.22: Esquema de funcionamiento de una neurona artificial. Las entradas de esta neurona están dadas por  $X_1$ ,  $X_2$  y  $X_3$ .



Fuente: Coevolución para un esquema de estrategia presa-depredador, Tesis, Luis Miguel Pellegrin Zazueta, 2009

entorno en el que actúa. Estas opciones de qué puede ser evolucionado y, por tanto, automatizado su diseño, se pueden organizar en estas cuatro categorías:

- Evolución de módulos específicos para controladores tradicionales de robots.
- Evolución de controladores de comportamientos.
- Evolución del hardware o morfología del robot.
- Evolución del modelo de entorno a utilizar por el mecanismo cognitivo.

### 13.3.1. Evolución de los controladores de comportamientos.

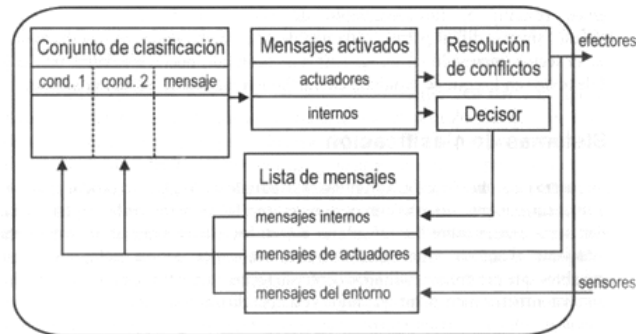
#### Tipos de estructuras de control

En esta sección se estudian qué estructuras soportan los mecanismos cognitivos o controladores de los robots y, por lo tanto, qué estructuras se quieren diseñar automáticamente mediante evolución. En este primer apartado se considerará la posibilidad de evolucionar únicamente módulos de comportamiento concreto, concentrándose en las posibles estructuras de control utilizadas para la realización de dichos módulos. Los ejemplos concretos de comportamientos irán surgiendo, progresivamente.

## Sistemas de clasificación

Un ejemplo de uso de sistemas de clasificación lo encontramos en los trabajos de Colombetti y col., y Dorigo y Colombetti. La figura 13.23 muestra los módulos para su sistema de clasificación, en el cual las reglas únicamente poseen dos condiciones a verificar y una acción o mensaje. Existen mensajes que actúan sobre los actuadores, mensajes provenientes del entorno a través de los sensores y mensajes internos del sistema. Un grupo de clasificadores activa mensajes para los actuadores, en tanto que el resto añaden mensajes internos al activarse.

Figura 13.23: Sistema de clasificación utilizado por Dorigo y Colombetti



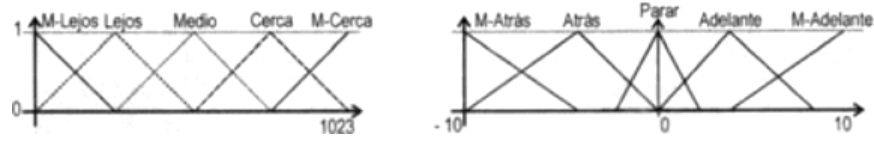
Fuente: Evolución artificial y robótica autónoma, José Santos, Richard J Duro

## Sistemas basados en lógica borrosa

En la figura 13.24 se muestra un ejemplo de categorización, realizado por los españoles Vicente Matellán y col, de los posibles valores devueltos por un sensor de infrarrojos (el valor cero en el sensor indica que no se percibe ningún objeto y un valor 1.023 corresponde al mayor nivel de detección o proximidad de objeto) de un robot Khepera en cinco conjuntos borrosos: objeto detectado muy lejano, lejano, a una distancia media, cercano y muy cercano. Cada lectura en un sensor pertenece en un determinado grado a un conjunto borroso, de forma que un valor de detección en el sensor puede pertenecer en un grado de 0.4 a objeto cercano y, al mismo tiempo, con un valor 0.6 a objeto muy cercano.

En la figura 13.25 se muestra un ejemplo completo con las diversas fases del proceso de inferencia, cuando se activan las dos reglas de control que se muestran en la misma figura y se utilizan los mismos conjuntos borrosos y funciones de pertenencia anteriores. De modo típico, los operadores *min* y *max* se utilizan para las operaciones *and* y *or*

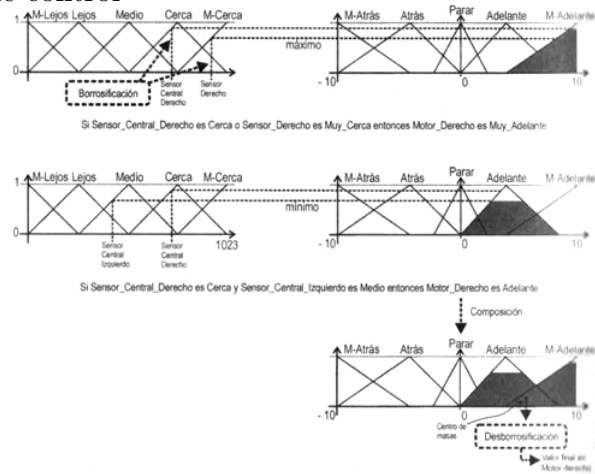
Figura 13.24: Funciones de pertenencia a cada uno de los conjuntos borrosos en los valores de los sensores (izquierda) y de los motores (derecha) de ambas ruedas de un Khepera.



Fuente: Evolución artificial y robótica autónoma, José Santos, Richard J Duro

(y/o lógicos) en las condiciones de las reglas, y producir un valor de certeza en cada regla.

Figura 13.25: Proceso de inferencia cuando se activan diferentes reglas que actúan sobre una misma variable de control



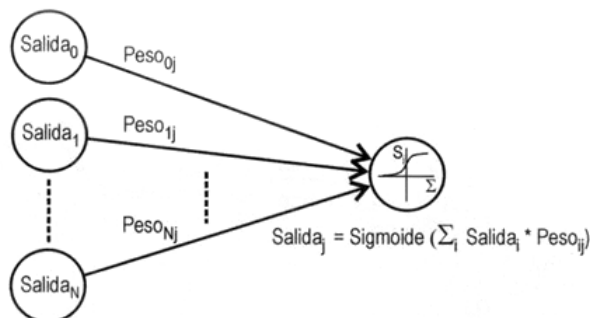
Fuente: Evolución artificial y robótica autónoma, José Santos, Richard J Duro

### Redes neuronales artificiales

En la figura 13.26 se muestra la estructura de conexión típica utilizada en las simulaciones entre las neuronas presinápticas de una capa precedente con otra neurona postsináptica. Se calcula primeramente una combinación (generalmente una suma) en la que la salida de cada nodo presináptico contribuye con una eficiencia definida por su peso de conexión. Ese valor combinado entra a una función de activación del nodo, generalmente no lineal, siendo una de las más utilizadas la función sigmoide de la figura. Estas estructuras dan origen a la rama de Inteligencia Artificial denominada

conexionista, en contraposición a la rama de IA simbólica, y presentan características muy interesantes desde el punto de vista computacional.

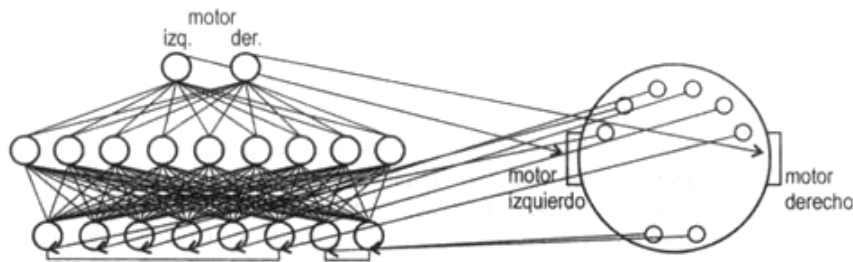
Figura 13.26: Organización en capas de los nodos de una RNA



Fuente: Evolución artificial y robótica autónoma, José Santos, Richard J Duro

La figura 13.27 muestra un ejemplo de red neuronal típica que actúa como controlador, en este caso, de un robot Khepera, que recibe las entradas directamente de los 8 sensores de infrarrojos, mientras que las salidas de la red actúan directamente sobre los motores de las dos ruedas. La red utilizada es la de varias capas de nodos, conocida como “perceptron multicapa”, en este ejemplo con una única capa de nodos “ocultos” y sin realimentaciones entre sus nodos.

Figura 13.27: Ejemplo de Lund y Hallam de conexión directa de los valores de los seis sensores frontales y de los dos traseros de infrarrojos del robot Khepera con los nodos de entrada del controlador neuronal.

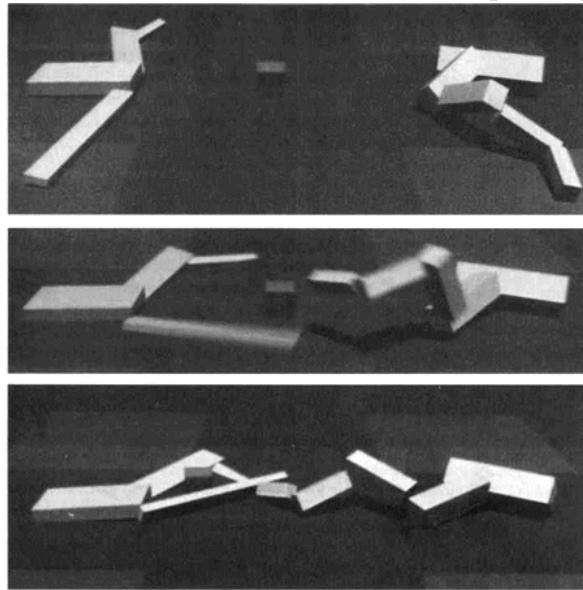


Fuente: Evolución artificial y robótica autónoma, José Santos, Richard J Duro

### 13.3.2. Evolución del hardware o morfología del robot

En el campo de robótica es a menudo deseable no sólo determinar buenos mecanismos de control sino también la morfología óptima del robot y el número y tipo de sensores para cada tarea en particular. Una configuración inadecuada de la morfología puede hacer difícil o incluso imposible conseguir el controlador para una tarea. Esta simbiosis controlador-morfología queda perfectamente definida por los investigadores de la Universidad de Edimburgo Wei-Po Lee y col. en la frase *la calidad de un agente se mide por lo bien que se consigue su tarea ejecutando el cerebro en su correspondiente cuerpo*. De este modo se introduce el concepto de evolución de hardware EHW, Evolvable Hardware. Las últimas criaturas de Sims son tridimensionales y poseen masa y propiedades inerciales, como se muestra en la figura 13.28. Sims destaca que las criaturas a menudo producen comportamientos interesantes en sus enfrentamientos con sus oponentes para atrapar la pieza cúbica.

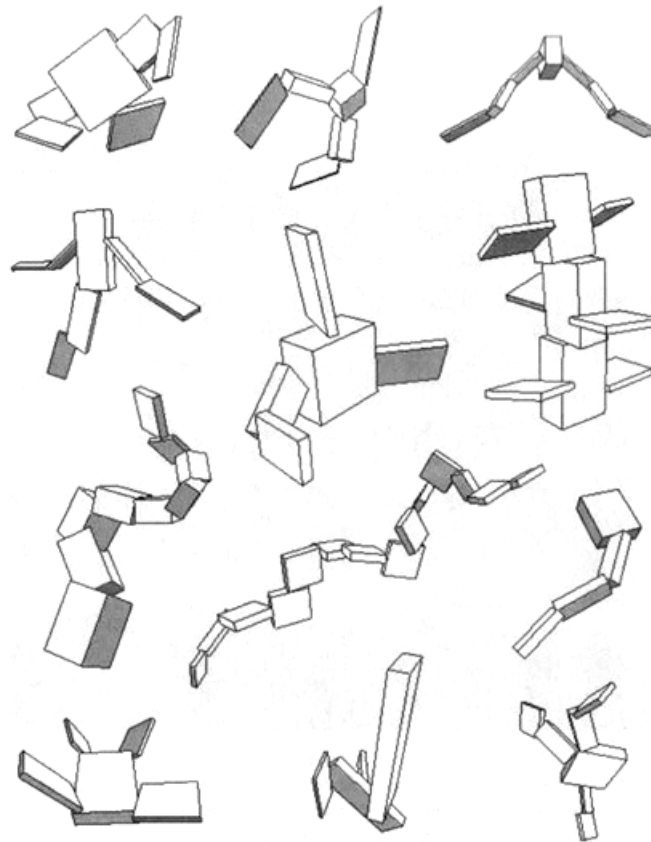
Figura 13.28: Dos criaturas virtuales de Sims luchando por la posesión de un bloque



Fuente: Evolución artificial y robótica autónoma, José Santos, Richard J Duro

La figura 13.29 muestra otro ejemplo de criaturas virtuales de Sims, coevolucionando su morfología y control (de modo cooperativo), en este caso para “nadar” en un entorno virtual acuoso. La simulación incluye el efecto de resistencia de la viscosidad del agua y se utiliza como función de calidad únicamente la velocidad de desplazamiento en el medio acuoso del centro de masas de la criatura.

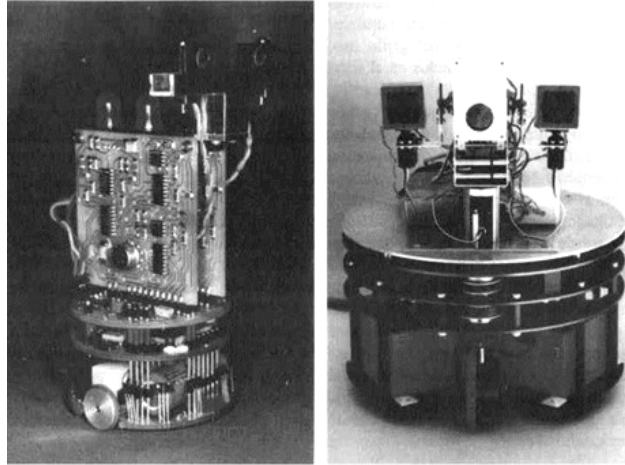
Figura 13.29: Ejemplos de criaturas virtuales de Sims evolucionadas para nadar en un entorno acuoso



Fuente: Evolución artificial y robótica autónoma, José Santos, Richard J Duro

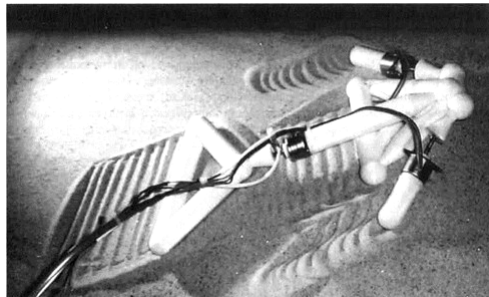
En la parte derecha de la figura 13.30 se muestra otro ejemplo de los investigadores de la Universidad de Edimburgo que corresponde a la simulación en hardware del sistema perceptivo de un murciélago, concretamente de la superficie de cartílago de las orejas. Al robot lo denominan RoBat, en el que se distingue un emisor central de sonar con dos receptores laterales orientables de forma independiente. Mediante un algoritmo genético determinan la forma adecuada de las “orejas” (pinnae) para obtener características deseables de direccionalidad en los receptores. En la figura 13.31 se debe destacar, como indican los autores, que con una función de calidad que sólo tiene en cuenta la distancia recorrida para un comportamiento de caminar, la evolución ha generado una estructura simétrica, lo que le confiere una mayor facilidad de moverse en línea recta. El movimiento se consigue gracias a que las barras se pueden alargar o contraer mediante pistones controlados por una red neuronal ejecutada en un microcontrolador, y obtenida de modo cooperativo en el proceso evolutivo.

Figura 13.30: Khepera con “oídos” que simula un grillo hembra y RoBat, una simulación de la morfología de recepción de los ecos de sonido emitidos por un murciélago



Fuente: Evolución artificial y robótica autónoma, José Santos, Richard J Duro

Figura 13.31: Ejemplo de estructura final en el trabajo de Lipson y Pollack (2000)



Fuente: Evolución artificial y robótica autónoma, José Santos, Richard J Duro

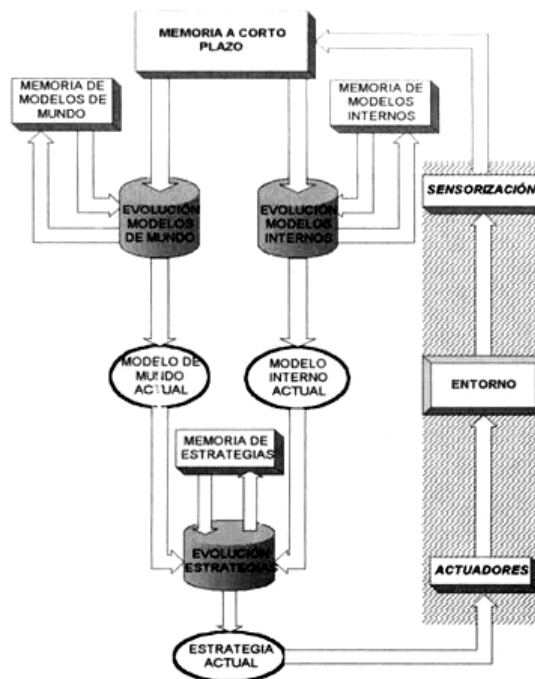
### 13.3.3. Evolución del modelo de entorno dentro del mecanismo cognitivo

Para la obtención del modelo de entorno, el robot necesita adquirirlo de modo automático. Como posibilidad existe una tendencia de “aprendizaje basado en evolución”, proveniente de una perspectiva biológica denominada *darwinismo neuronal*, introducida por el francés Jean-Pierre Changeux y col. (1984), y los americanos Gerald Maurice Edelman (1987), Michael Conrad (1974) y Willian Calvin (1987), entre otros, que indica que la evolución juega el mismo papel en el desarrollo del cerebro en tiempo somático que en los ecosistemas en tiempo filogenético.

Esta aproximación considera un modelo de dos niveles. Un nivel que podríamos llamar “inconsciente” donde el robot evoluciona poblaciones de modelos de mundo y de modelos internos en función de lo que ha ido sensando en el entorno y de sus propias acciones. Cada vez que existe una necesidad de realizar una acción o determinar una estrategia, el robot entra en un nivel que podríamos llamar “consciente”, en el cual utiliza el mejor modelo de mundo y el mejor modelo interno del que disponga en ese instante para evolucionar una base de posibles estrategias hasta que obtenga una estrategia que maximiza la consecución de sus motivaciones.

Esta aproximación se presenta en la figura 13.32. Básicamente, en el nivel consciente el robot evalúa una estrategia pasándola por el modelo de mundo, que a su salida le proporciona cuáles prevé que van a ser las percepciones del robot una vez ejecutada la estrategia, y este resultado, pasado por el modelo interno le indica cuán satisfactoria habría sido de ser aplicada.

Figura 13.32: Módulos de la arquitectura cognitiva con aprendizaje automático del entorno



Fuente: Evolución artificial y robótica autónoma, José Santos, Richard J Duro

En la figura 13.33 se puede observar cómo ha ido evolucionando el error existente entre las predicciones realizadas por el modelo de mundo generado por el robot y el mundo real. Según avanza la interacción con el mundo, este modelo va mejorando hasta

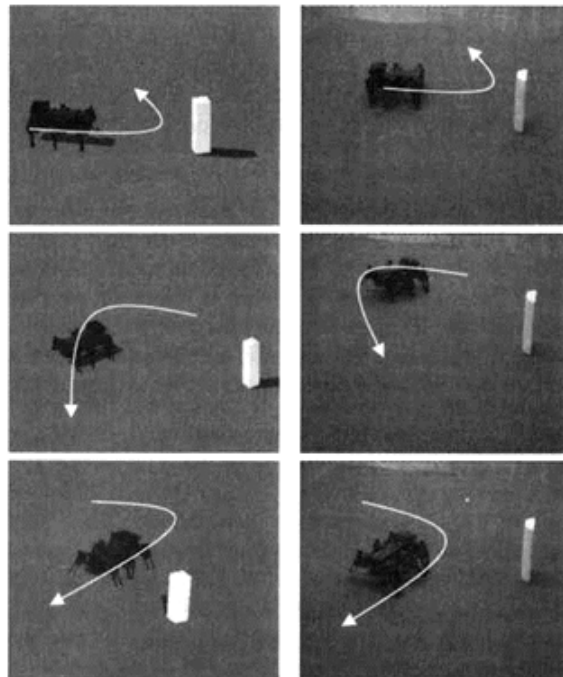
Figura 13.33: Evolución del error cuadrático medio del modelo de mundo respecto al mundo real según interactúa el robot con el mundo



Fuente: Evolución artificial y robótica autónoma, José Santos, Richard J Duro

que es suficientemente bueno para que el robot alcance su objetivo.

Figura 13.34: Camino seguido por el robot en simulación (izquierda) y en el mundo real (derecha) cuando aún no ha aprendido a alcanzar un objeto (iteraciones de la 53 a la 57)

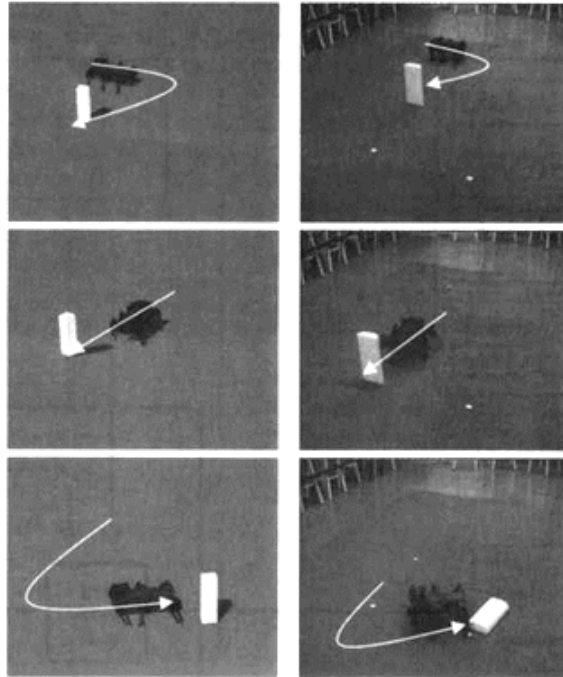


Fuente: Evolución artificial y robótica autónoma, José Santos, Richard J Duro

Un ejemplo se puede apreciar en las figuras 13.34 y 13.35. En la primera hay una serie de trayectorias seguidas por el robot entre las iteraciones 53 y 57 en un modelo simulado y en el propio mundo real. El modelo de mundo todavía es malo y, por lo

tanto, además de no caminar bien, el robot no alcanza el objetivo por medio de las estrategias que elige. En la figura 13.35 se muestra el mismo tipo de imágenes pero para las iteraciones 421-424. En este caso, a la hora de elegir sus estrategias el robot ya dispone de modelos de mundo e internos adecuados y los resultados son muy buenos.

Figura 13.35: Camino seguido por el robot en simulación y en el mundo real cuando ya ha aprendido a alcanzar un objeto (iteraciones a partir de la 420)

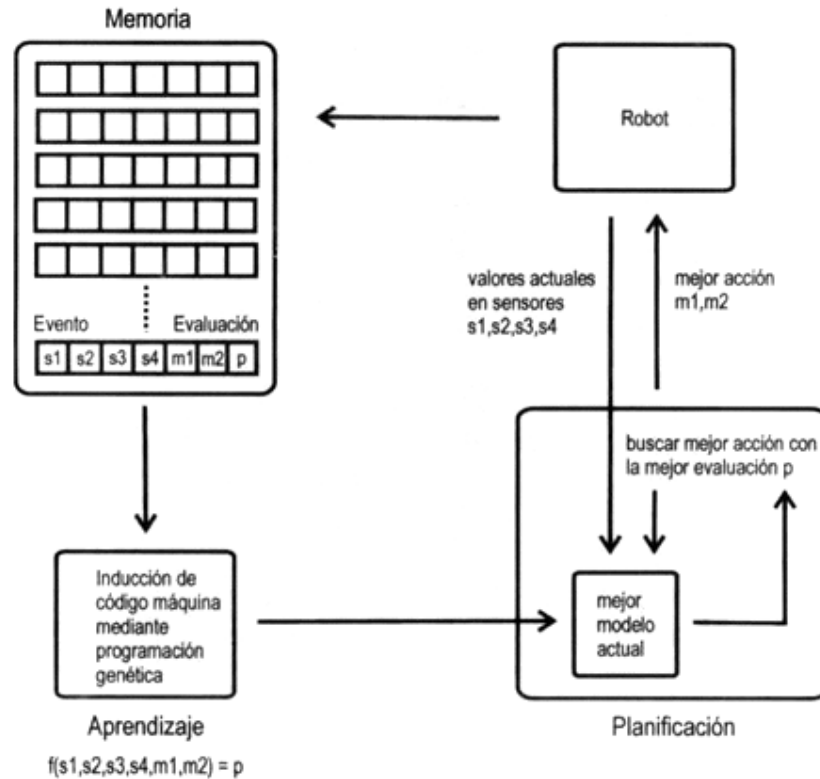


Fuente: Evolución artificial y robótica autónoma, José Santos, Richard J Duro

Un trabajo en la misma línea es el de los investigadores de la Universidad de Dortmund (Alemania) Peter Nordin y col, quienes emplearon un “mecanismo de memoria”, como ellos denominaron, mostrado en la figura 13.36, en el que guardan tripletas consistentes en los valores de entrada a los sensores de un robot Khepera (valores  $s_1$ ,  $s_2$ ,  $s_3$ ,  $s_4$ ), la acción realizada (valores  $m_1$  y  $m_2$  aplicados en los motores de las ruedas del Khepera) y el valor de calidad adquirido con esa acción ( $p$ ). Un “proceso de aprendizaje”, según la definición de los autores, encuentra mediante programación genética la función que predice la calidad en cada situación entradas-acción a partir de las tripletas guardadas en memoria. Esa función de predicción constituye el modelo de mundo o entorno. Posteriormente, un “proceso de planificación”, para cada situación de entrada en los sensores, determina la mejor acción que maximizará la calidad según el mejor modelo de entorno actual, realiza la acción, calcula la calidad resultante real, y guarda

la tripleta en memoria, repitiéndose el ciclo en cada movimiento del robot. Sin embargo, con esa definición de mundo, el modelo de entorno es dependiente de la tarea que tenga que llevar a cabo el robot.

Figura 13.36: Esquema del método de modelización utilizado por Nordiny

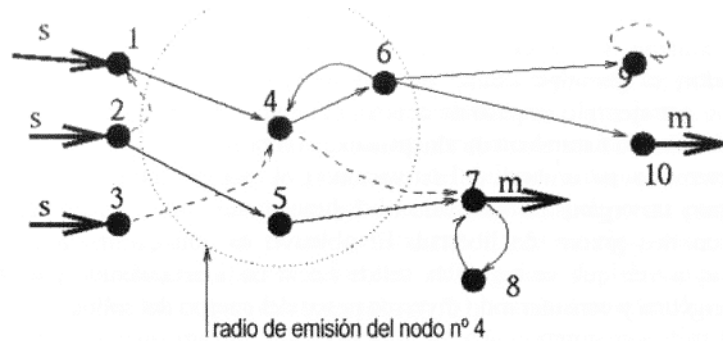


Fuente: Evolución artificial y robótica autónoma, José Santos, Richard J Duro

## 13.4. ¿CUÁL PUEDE SER EL CAMINO A SEGUIR EN LA ROBOTICA MOVIL AUTONOMA?

En lo que sigue se dará un poco de espacio a la especulación y a tratar de concretar lo que constituyen temas o líneas sobre las que, de modo lógico, debe seguir el trabajo de investigación. Estos puntos básicos que se tratarán, llevarán a la consideración de diversos aspectos en la aproximación de robótica evolutiva que deberán ser estudiados en mayor detalle.

Figura 13.37: Red de “gas” de Husbands. Cada nodo emite un “gas” que modula propiedades de los nodos afectados



Fuente: Evolución artificial y robótica autónoma, José Santos, Richard J Duro

### 13.4.1. Estructuras de control más complejas

No porque sea obvio hay que dejar de mencionar que uno de los primeros caminos a seguir en el desarrollo de la robótica evolutiva se basa en la utilización de estructuras de control más complejas y adaptadas al comportamiento a realizar. Las estructuras más utilizadas son, con diferencia, las redes neuronales artificiales, gracias a sus propiedades ya explicadas. Pero, la mayoría de los investigadores utilizan sólo un pequeño conjunto de tipos de redes neuronales artificiales, aquéllas que se consideran estándar o de propósito general. La naturaleza ha seguido un camino diferente y ha diseñado diversos tipos de circuitos neuronales o, incluso, de neuronas para determinadas tareas, y esto se tendrá que reflejar en una medida mayor que la actual al estudiar la evolución de sistemas artificiales.

Así, estructuras específicas de procesamiento temporal, de procesamiento espacial u otro tipo de estructuras en las que se incrementa la complejidad computacional mediante el aumento de la capacidad de procesado de los nodos o conexiones, y que se han denominado de modo general “redes de alto orden” en el campo conexionista, deben estar disponibles en los procesos evolutivos para obtener controladores de comportamientos mejor adaptados. La figura 13.37 muestra un ejemplo en el que el nodo número 4 de la red emite un gas con un determinado radio de acción (definido en el genotipo), siguiendo una concentración gaussiana en función de la distancia al punto emisor, concentración que además decae con el tiempo.

### 13.4.2. Mejora del proceso de evolución

Propiedades cruciales de un problema de optimización son:

- Es un problema específico.
- El espacio de búsqueda de posibles soluciones está típicamente bien definido en términos de un número finito de parámetros.

Pero esto no es lo que sucede en la evolución natural, en la cual los “problemas” con los que se encuentran los diferentes organismos no fueron predeterminados en el origen de la vida en la Tierra. La evolución natural se puede imaginar como un método para mejorar de modo incremental y adaptativo a organismos que se encuentran, en tiempo geológico, problemas que varían con el entorno que significativamente incluye a otros organismos variados.

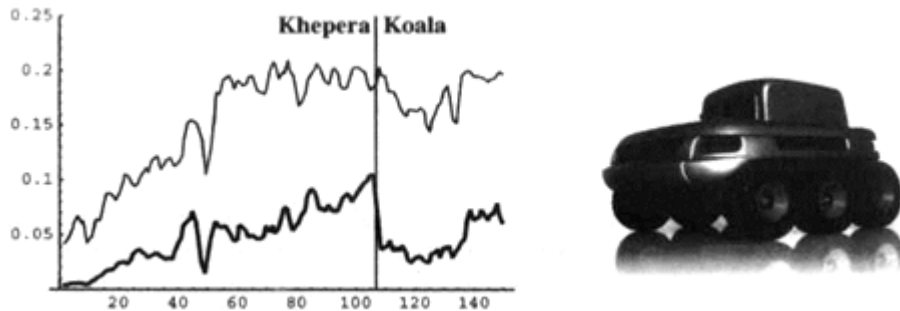
### 13.4.3. Independencia de la plataforma del robot, comunicación y cultura

Se comentarán dos posibilidades en este sentido: una primera en la cual la evolución reutiliza comportamientos previamente obtenidos en otra plataforma diferente, y una segunda en la que ya se define un nivel de separación entre la sensorización y actuación de cada robot del nivel propio de comportamientos.

Respecto a la primera posibilidad y como continuación a lo dicho en el apartado anterior, en la línea de no comenzar la evolución desde cero, se ha realizado poca investigación en la reusabilidad de los controladores de comportamientos cuando se exportan a otros sistemas. Un ejemplo es el trabajo de Floreano y Mondada, al evolucionar controladores para un robot Khepera y usarlos como “semilla” en la evolución de controladores en un robot Koala, similar al primero aunque de mayor tamaño (figura 13.38 derecha), y también desarrollado en el Laboratorio de Microcomputación del Instituto Federal de Tecnología de Lausanne. Como muestra la figura 13.38, de evolución de la calidad del mejor individuo y de la calidad media de la población, hasta la generación 106 se evoluciona el comportamiento para el robot Khepera. En ese momento los individuos de la población pasan a ser los controladores de un robot Koala, continuando la evolución hasta la generación 150.

Una posibilidad para solucionar el problema es a través de una hibridación con otras soluciones de Inteligencia Artificial. A estos módulos se les denomina “*sensores virtuales*” y se pueden considerar como una extensión de los sensores. Así, un sensor virtual no es más que una etapa que resume o concentra información, de manera que el módulo de

Figura 13.38: Evolución de calidades al considerar un cambio de plataforma entre un robot Khepera y un robot Koala



Fuente: Evolución artificial y robótica autónoma, José Santos, Richard J Duro

Figura 13.39: Niveles de sensores y actuadores virtuales



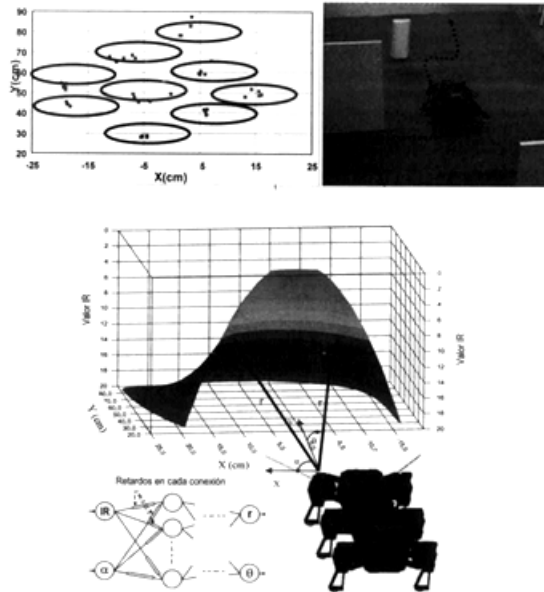
Fuente: Evolución artificial y robótica autónoma, José Santos, Richard J Duro

comportamiento no tenga que tratar con grandes cantidades de información sensorial atomizada, la mayoría de la cual podría ser irrelevante. Análogamente, un “actuador virtual” recibe las órdenes del controlador de comportamientos y genera la secuencia correcta de acciones en el actuador correspondiente. Ahora, en vez del acercamiento tradicional de la robótica basada en comportamientos donde se establece un esquema del tipo conexión directa *sensores* → *controlador* → *actuadores*, se tiene un esquema *sensores virtuales* → *controlador* → *actuadores virtuales*, como indica la figura 13.39.

Un ejemplo simple pero ilustrativo, el caso de los sensores de infrarrojos (IR) del robot tipo Hermes II. Estos sensores están montados encima de las patas del robot, como se muestra en la figura 13.40, y su campo sensorial se puede cambiar moviendo dicha pata. Como consecuencia de esto, aunque el robot esté parado, los sensores se pueden mover para barrer una determinada área.

La figura 13.41 muestra un ejemplo en el que se reutiliza un comportamiento de bajo nivel de seguir paredes para el robot Rug Warrior, que utiliza para ello únicamente sus dos sensores de infrarrojos binarios. Ese mismo módulo y el resto de módulos de la arquitectura jerárquica de comportamientos se usan para un robot totalmente diferente,

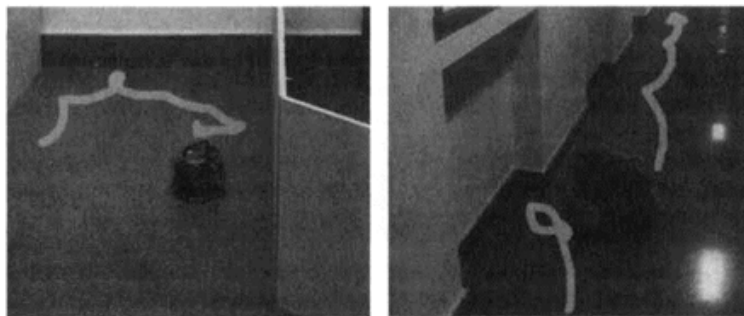
Figura 13.40: Robot Hermes en el proceso de detectar la posición exacta de un objeto con un único sensor de IR



Fuente: Evolución artificial y robótica autónoma, José Santos, Richard J Duro

un Pioneer 2-DX. Este robot posee un anillo de sonares que retoman una medida de distancia a la pared u objeto más cercano detectado, al contrario de los infrarrojos del Rug Warrior.

Figura 13.41: Reutilización de un comportamiento de seguir paredes en una arquitectura jerárquica



Fuente: Evolución artificial y robótica autónoma, José Santos, Richard J Duro

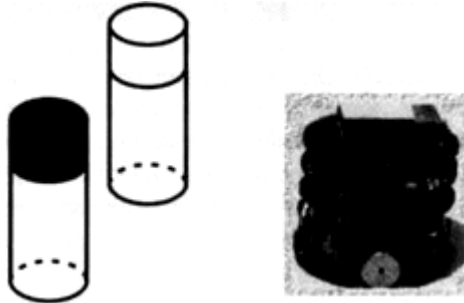
El ejemplo muestra claramente que el problema está en decidir cuáles son los sensores y actuadores virtuales adecuados, que se pueden considerar “estándar”, para tener esa

separación explícita, dado el amplio abanico de sensores y actuadores reales. Aunque se utilicen pocos tipos de sensores, sus diferentes rangos de detección en distancia y ángulo de apertura, y las diferentes localizaciones en el cuerpo del robot, hacen que el nivel de sensorización virtual pueda ser muy diferente en cada robot o se necesite un considerable tiempo de cómputo.

#### 13.4.4. Percepción activa y atención

Continuando con las diferentes líneas por las se cree, se encaminará la investigación en un corto plazo, se ve que, por ahora, en la línea de percepción pocas arquitecturas cognitivas soportan la idea de “percepción activa” o “percepción dirigida por objetivos”, es decir, realizar acciones por parte del robot para mejorar la percepción del entorno. Esto lleva a percepciones peores o ambiguas en situaciones en las cuales unos mecanismos de sensorización activa muy simples mejorarían en gran medida la imagen que el controlador tiene del entorno. Stefano Nolfi y Domenico Parisi, exponen un ejemplo ilustrativo como muestra la figura 13.42, sitúan un robot Khepera dotado con sus sensores de infrarrojos y con su pequeña cámara en un entorno en el cual hay dos tipos de cilindros, cuya parte superior está pintada de negro o de blanco.

Figura 13.42: Ejemplo de necesidad de percepción activa provocada por la existencia de aliasing en la percepción con los infrarrojos del Khepera



Fuente: Evolución artificial y robótica autónoma, José Santos, Richard J Duro

La figura 13.43 muestra otro ejemplo de percepción activa. Con un robot Rug Warrior y forzándolo a utilizar un único fotosensor para el seguimiento de una luz, éste es incapaz de reconocer la posición de la misma a no ser que realice un movimiento de zig-zag para poder comparar los niveles de intensidad de luz y poder determinar la localización espacial de la misma.

Figura 13.43: Trayectorias del robot Rug Warrior en una tarea de seguimiento de luz que utiliza un único fotosensor en simulación (izquierda) y en la realidad con diferentes niveles de luz ambiente (centro y derecha)



Fuente: Evolución artificial y robótica autónoma, José Santos, Richard J Duro

### 13.4.5. Hacia comportamientos más deliberativos. Tratamiento de información temporal

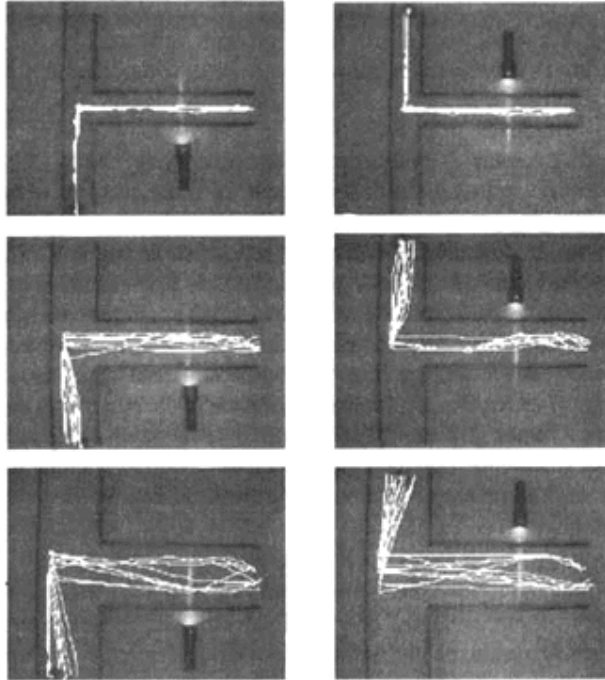
Una alternativa es utilizar redes neuronales artificiales recurrentes, que codifican un resumen de la información temporal recibida en sus entradas, es decir, el “estado”, codificado en los nodos que incorporan las recurrencias. Nick Jakobi (1997) evoluciona RNAs recurrentes de topología arbitraria, con únicamente el número de nodos previamente establecido, para negociar un laberinto con pasillos en forma de T con un robot Khepera y en la dirección previamente especificada por una luz. La figura 13.44 corresponde a resultados de este ejemplo, en ella las seis figuras muestran las trayectorias seguidas por el Khepera para 10 pruebas consecutivas, con diferentes anchos del pasillo y con dos posiciones diferentes del foco en cada una.

La figura 13.45 muestra un ejemplo en el que el robot central debe predecir las situaciones en las que el Khepera móvil va a chocar contra él, cambiando de modo adecuado la velocidad de una de las ruedas del móvil a fin de evitar el choque. Para esto el controlador neuronal con retardos del robot central debe realizar un comportamiento de seguimiento del robot móvil al mismo tiempo que realiza esa predicción, comparando los valores temporales de la información de sus sensores.

### 13.4.6. Consideraciones finales

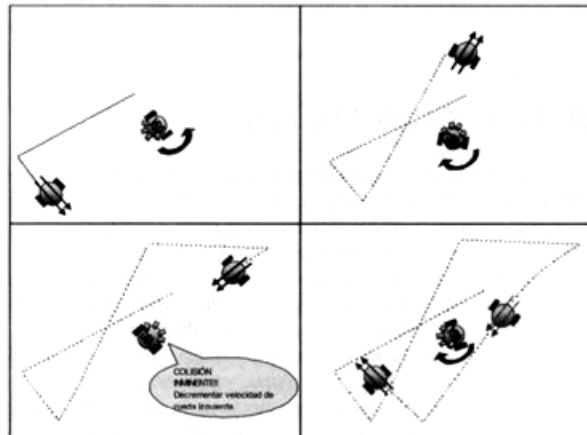
Día a día se puede ver la fuerte interdependencia que existe entre diferentes ramas de la Ciencia. El caso de la robótica es un ejemplo paradigmático de cómo depende de los resultados de otras disciplinas, aunque también genera desarrollos o mejoras en ellas, produciéndose así una sinergia mutua. De hecho, muchas de las investigaciones

Figura 13.44: Trayectorias del Khepera en 60 pruebas, en el giro de la unión T previamente indicado por el foco de luz



Fuente: Evolución artificial y robótica autónoma, José Santos, Richard J Duro

Figura 13.45: Trayectorias del Khepera móvil cuando se aproxima al robot central, evitando los choques



Fuente: Evolución artificial y robótica autónoma, José Santos, Richard J Duro

de robótica autónoma se presentan actualmente en congresos científicos de inteligencia artificial, sistemas autónomos o vida artificial, existiendo muy pocos dedicados en exclusiva a la disciplina. También, por esta dependencia, podemos decir que la robótica es una ciencia “dura”, en el sentido de que necesita integrar los desarrollos de varias disciplinas, ya que se debe conseguir interpretar la diversa información sensorial del robot, típicamente de la imagen de una cámara y de los diferentes sensores que han aparecido en los ejemplos, junto con los propios mecanismos cognitivos de éste y sin olvidar la limitación que impone la parte electrónica y mecánica.

La aproximación de robótica basada en comportamientos ha demostrado ser una alternativa muy prometedora en el desarrollo de sistemas autónomos que pueden actuar en entornos dinámicos parcialmente desconocidos. La aproximación todavía tiene muchos problemas por resolver, especialmente debido a la complejidad que surge en las interacciones del robot con el entorno y entre los mismos comportamientos. Una vía para superar estos problemas de complejidad es el uso de técnicas de evolución artificial, y muchos investigadores han recurrido a ella en estos últimos años. Actualmente existe un intenso esfuerzo investigador en este sentido. Día a día se están desarrollando nuevos sistemas y están surgiendo nuevos e interesantes subcampos.

## CONCLUSIONES

Como resultado del seminario de investigación en robótica móvil autónoma se concluye lo siguiente:

*Con respecto a la formación de los estudiantes*

- El seminario de investigación como modalidad de trabajo de grado forma a los participantes para la investigación científica mediante el desarrollo de habilidades específicas aplicadas al asumir los diferentes roles dentro del seminario.
- Fortaleció el aprendizaje como una experiencia grupal entre los integrantes y el director, sin ningún tipo de jerarquía en donde las cualidades del profesor se enfocaron hacia el liderazgo y no al autoritarismo.
- El hecho de que los participantes cambien de roles en cada sesión, permitió una formación integral y un aporte al desarrollo de capacidades dependiendo del papel que se desempeñó.
- Se mantuvo en todo momento la cohesión del grupo y un adecuado ambiente de trabajo, durante el proceso de recopilación de información, las sesiones, el desarrollo de talleres y la redacción del documento final, en un clima de participación activa, basado en el respeto y el apoyo mutuo.

*Con respecto a la temática tratada*

- En concordancia con el estudio del estado del arte de la robótica móvil autónoma, se puede concluir que actualmente ésta se considera un área de la tecnología avanzada manejadora de problemas de alta complejidad. Sus productos se constituyen en aplicaciones de las áreas de control, programación, inteligencia artificial, percepción e instrumentación, y sirven de base para el avance en diversos campos de la industria, aportando soluciones tecnológicas innovadoras orientadas al desarrollo de mejores robots y a la ampliación del abanico de aplicaciones disponibles. Así, este campo de la investigación está desarrollándose en todo momento, quedando aún mucho que recorrer.
- Teniendo en cuenta lo anterior, se puede ver que en el mundo y en especial en Japón y Estados Unidos (países líderes en el área de la robótica), existen muchas empresas dedicadas al desarrollo de nuevos robots y tecnologías de control, mientras que en Colombia no existen tales empresas que propendan al desarrollo de la industria nacional.

- Se estudiaron los diversos sistemas de locomoción que usan los robots móviles, y se hicieron consideraciones sobre estabilidad, maniobrabilidad, sincronización, y capacidad de control de los mismos.
- Fue necesario, para el desarrollo de los temas de cinemática y dinámica, hacer un resumen previo de ciertos conceptos, configuraciones y supuestos considerados enfocado a los robots móviles con ruedas para un mayor entendimiento de estos temas.
- En la cinemática se describieron formalmente las restricciones de las ruedas de forma individual, y luego se combinaron para expresar las restricciones cinemáticas del robot entero. Con estas herramientas, se evaluaron los caminos y las trayectorias que definen la maniobrabilidad un robot. También se realizaron consideraciones sobre la cinemática de robots bípedos.
- Tomando como base el estudio de la cinemática, se describieron los modelos dinámicos usados en los robots con ruedas, dando una mayor importancia a éstos, ya que ellos son los más usados en la actualidad pues tienen un amplio rango de aplicaciones por su estabilidad y otras características que hacen de ellos los robots adecuados para todo tipo de terreno. Se estudió también la dinámica de los robots móviles bípedos y se dejaron expuestos algunos modelos dinámicos simples de buques y helicópteros autónomos.
- Se clasificaron y explicaron las principales técnicas de percepción, fundamentos de procesamiento de imágenes, extracción de características, así como un completo resumen de los sensores usados actualmente en la robótica móvil autónoma, es decir todo aquello que el robot necesita para ser capaz de “entender” su entorno y desplazarse de forma satisfactoria en el terreno establecido para su funcionamiento.
- Dependiendo del entorno en que un robot móvil se desenvuelve, se escoje el sistema de percepción para el mismo ya que este es condicionado por las características ambientales y las tareas a desarrollar.
- A pesar que se han desarrollado actualmente sensores más avanzados, siempre hay un error asociado a sus mediciones y el medio ambiente afecta su funcionamiento; por tanto se hace necesario siempre para un robot tomar una serie de lecturas (y no una única lectura) que compensen ese error.

- La detección y seguimiento de color mediante una cámara integrada al robot tiene dos ventajas importantes: en primer lugar, la detección de color es una función sencilla de una sola imagen, por lo tanto, no hay problema de correspondencia que deba ser resuelto en este tipo de algoritmos (a diferencia de la visión estéreo). En segundo lugar, ya que la detección de color proporciona una señal nueva, señal independiente del medio ambiente, si se combina (es decir, fusión de sensores) con señales existentes, tales como los datos de la visión estéreo o de telemetría láser, se puede esperar un aumento significativo de la información que el robot adquiere de su entorno.
- Los algoritmos de detección de bordes y filtros de suavizado han cobrado un vital importancia en la percepción de un robot móvil ya que permiten extraer información del entorno de una forma simplificada.
- Para un robot móvil es esencial ubicar su posición aproximada en el entorno, por tanto se discutieron los principales métodos de localización usados en la actualidad (en especial el filtro de Kalman), representaciones de creencia y creación de mapas a priori o de forma simultánea.
- Se puede concluir que una creencia de múltiple hipótesis permite a un robot móvil corregir su posición en el movimiento a diferencia de una creencia de hipótesis única ya que ésta, al no considerar sino sólo una posición actual, genera ventajas ya que no hay ambigüedades en su creencia pero posteriormente si el robot cree estar en cierto punto pero no está realmente en ese punto, el robot se perderá fácilmente en su período de navegación.
- En la actualidad existen grandes limitaciones para que un robot sea capaz de localizarse en entornos dinámicos. Por tanto se está trabajando fuertemente en este aspecto para la creación de mejores sensores y nuevos métodos de representación de mapas, así como también en la fusión de estos sensores de manera que se perciba el entorno de forma óptima y los objetos transitorios (móviles) no representen un problema en la localización de un robot móvil.
- Todo robot que se considere completamente autónomo debe ser capaz de percibir su entorno, ubicarse en él, y moverse tomando decisiones que le permitan llegar a su objetivo.
- Un breve resumen de las competencias para la navegación fue hecho, dando a conocer los diferentes métodos de evasión de obstáculos y distintos tipos de arquitecturas de navegación usadas actualmente en la robótica móvil autónoma.

- De estos algoritmos de evasión de obstáculos que fueron tratados en esta investigación, se resaltó el funcionamiento del algoritmo de Bug dada su simplicidad ya que su método se centra solamente en circunnavegar cada obstáculo que el robot encuentre en su camino.
- Dado el creciente avance tecnológico, se preve de forma lógica que la robótica móvil autónoma crecerá aún más de lo que ya lo ha hecho y se abrirá camino incursionando en la creación de nuevos mecanismos de locomoción, y métodos de percepción, localización, navegación y control.
- Términos como “robótica evolutiva”, “robótica de desarrollo”, “robótica evolutiva”, “robótica probabilística” y “redes neuronales” son muy comunes hoy en día y hacen parte de un futuro prometedor en la robótica móvil autónoma ya que son ramas de la robótica que han surgido recientemente para mejorar el funcionamiento de los robots en todo sentido; es una certeza que pronto vendrán nuevos desarrollos que sorprenderán a la humanidad.
- A través del desarrollo de talleres en MATLAB y/o SIMULINK, se validaron los temas tratados en el seminario, llevando parte de la información recopilada a la práctica.
- La construcción de nuestro robot LEVI, permitió desarrollar ciertas rutinas, que sentaron las bases para que la Escuela de Ingeniería Mecánica de la Universidad Industrial de Santander incursione en competencias de robótica a nivel nacional.
- Este robot permitirá también establecer un punto de partida en la creación de una asignatura de Robótica Móvil Autónoma a nivel de pregrado en la escuela, ya que gracias a su fácil construcción y sencillez de programación, los futuros estudiantes harán mejoras al robot LEVI, o bien desarrollarán sus propios robots con los cuales afianzarán sus conocimientos.

## RECOMENDACIONES

- Ya que el objetivo de este seminario fue establecer las bases teóricas, los fundamentos de la robótica móvil autónoma, se sugiere incluir en el pensum de la carrera de Ingeniería Mecánica de la Universidad Industrial de Santander una asignatura teórico-práctica que profundice en esta rama de la robótica.
- Se necesita complementar el seminario de robótica móvil autónoma también a través de otros proyectos de investigación que profundicen en temas que no se trataron a fondo en este trabajo como por ejemplo:
  - Algoritmos de locomoción de robots móviles.
  - Modelos cinemáticos y dinámicos de robots trepadores, biorobots, AUV's y UAV's.
  - Robótica cognitiva.
  - Técnicas de evolución artificial.
  - Operadores genéticos.
- Teniendo en cuenta este proyecto de investigación y aquellos que se han desarrollado anteriormente, se recomienda a los futuros trabajos en esta modalidad desarrollar aplicaciones prácticas propias de la investigación.
- Es importante darle continuidad al estudio del robot "LEVI" desarrollado, ya que este se ha creado siguiendo algunas rutinas establecidas en las competencias de robótica realizadas a nivel nacional y vale la pena trabajar en ellas, y anexar más rutinas al robot si fuese posible o bien, modificar aquellas que fueron programadas.
- Con la creación de este robot se busca también que la escuela de Ingeniería Mecánica de la Universidad Industrial de Santander comience pronto a participar precisamente en esas competencias de robótica móvil a nivel local y nacional.

## BIBLIOGRAFÍA

- [1] AGUIAR António Pedro, ATASSI Ahmad N, PAACOAL António M, *Regulation of a Nonholonomic Dynamic Wheeled Mobile Robot with Parametric Modeling Uncertainty using Lyapunov Functions*.
- [2] ANGELES Jorge, *Fundamentals Of Robotic Mechanical Systems Theory, Methods, and Algorithms*, Third Edition, 2007.
- [3] ARRAS K.O, TOMATIS N, *Improving robustness and precision in mobile robot localization by using laser range finding and monocular vision in Proceedings of The Third European Workshop on Advanced Mobile Robots (Eurobot 99)*, Zurich, September, 1999.
- [4] BARRIENTOS Antonio, *Fundamentos de robótica. McGraw-Hill*, Segunda edición, 2007.
- [5] BARRIENTOS Antonio, *Nuevas aplicaciones de la Robótica - Robots de servicio*, Departamento de Automatica, Ingeniería Electrónica e Informática Industrial DISAM-UPM.
- [6] BERMÚDEZ BOHÓRQUEZ Giovanni Rodrigo, *Modelamiento Cinemático y Odométrico De Robots Móviles; Aspectos matemáticos*, 2003.
- [7] BLÖSCH M, WEISS S, SCARAMUZZA D, SIEGWART R, *Vision based MAV navigation in unknown and unstructured environments, IEEE International Conference on Robotics and Automation, ICRA 2010*.
- [8] BORENSTEIN J, HANSEN M, and NGUYEN H, *The OmniTread OT-4 Serpentine Robot for Emergencies and Hazardous Environments*, In Proceeding of the International Joint Topical Meeting: Sharing Solutions for Emergencies and Hazardous Environments, 2006.
- [9] BREAZEAL Cynthia L, *Designing Sociable Robots*, 2002.
- [10] BRÄUNL Thomas, *Embedded Robotics*, 2006.

- [11] CEBRECOS DEL CASTILLO Carolina, LOZANO PRIETO Abel, NIETO DOMÍNGUEZ Agustín, *Robots inteligentes autónomos de nueva generación*.
- [12] CHEN XiaoQi, CHEN Y.Q, CHASE J.G *Mobile Robots - State of the Art in Land, Sea, Air, and Collaborative Missions*, 2009.
- [13] CHEVALLEREAU Christine, BESSONNET Guy, ABBA Gabriel, AOUSTIN Yannick, *Bipedal Robots, Modeling, Design and Walking Synthesis*, 2009.
- [14] CHU Baeksuk, JUNG Kyungmo, HAN Chang-Soo and HONG Daehie, *A Survey of Climbing Robots: Locomotion and Adhesion*, 2010.
- [15] CORKE Peter, SUKKARIEH Salah, *Field and Service Robotics: Results of the 5th International Conference*, 2006.
- [16] CRESPI, A. BADERTSCHER, and GUIGNARD A, *Swimming and crawling with an amphibious snake robot*, In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005.
- [17] CSORBA M, *Simultaneous localization and map building*, 1997.
- [18] CUAYA SIMBRO German, *Procesos de Decisión de Markov Aplicados a la Locomoción de Robots Hexápodos*, 2007.
- [19] DAVISON A.J., *Active search for real-time vision*, In International Conference on Computer Vision, 2005.
- [20] DAVISON A.J., *Real time simultaneous localization and mapping with a single camera*, International Conference on Computer Vision, 2003.
- [21] DAVISON A.J, REID I. Molton, *MonoSLAM: Real time single camera SLAM*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007.
- [22] DEEPAK B. B. V. L, PARHI Dayal R, *Kinematic model of three wheeled mobile robot*, 2011.
- [23] DEVON David and BRETL Timothy, *Kinematic and Dynamic Control of a*

*Wheeled Mobile Robot*, 2007.

[24] FAHIMI Farbod, *Autonomous Robots Modeling, Path Planning, and Control*, 2009.

[25] FREEDMAN B, SHPUNT A, MACHLINE M, ARIELI Y, *Depth mapping using projected patterns*, 2010.

[26] GARCÍA J. A, VÁSQUEZ L. A, *Los Robots en el sector agrícola*, Universidad Politécnica de Madrid, Departamento de Automática, Ingeniería Electrónica e Informática Industrial.

[27] GARCÍA Juan Ignacio, *Robótica Evolutiva, Plataforma de Simulación Experimental (Tesis)*, 2009.

[28] GARCÍA ROBLEDO Pablo, TORRIJOS Jesús, *Robots de Seguridad y Defensa*, Universidad Politécnica de Madrid. Departamento de Automática, Ingeniería Electrónica e Informática Industrial.

[29] GIL ÁLVAREZ José Ignacio, *Evolución y futuro de la inteligencia artificial aplicada a la robótica*.

[30] GONZÁLEZ GÓMEZ J, *Diseño de Robots Ápodos*, 2003.

[31] GORROSTIETA Efrén, VARGAS SOTO Emilio, *Algoritmo difuso de locomoción libre para un robot caminante de seis patas*, 2007.

[32] GRANOSIK G, HANSEN M. G, and BORENSTEIN J, *The Omnitread Serpentine Robot for Industrial Inspection and Surveillance*, 2005.

[33] GÄTCHER S, HARATI A, SIEGWART R, *Incremental object part detection toward object classification in a sequence of noisy range images*, Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2008), Pasadena, USA, May 2008.

[34] HOLLAND John, *Designing Autonomous Mobile Robots*, 2004.

- [35] KAZUYA YOSHIDA, *Achievements in space robotics, Expanding the Horizons of Service and Exploration*, Tohoku University.
- [36] KIMURA Hiroshi, TSUCHIYA Kazuo, ISHIGURO Akio, WITTE Hartmut, *Adaptive Motion of Animals and Machines*, 2006.
- [37] LAVALLE S.M, *Planning Algorithms*, Cambridge, UK, Cambridge University Press, 2006.
- [38] LEONARD J.E, DURRANT-WHYTE H.F, *Directed sonar sensing for mobile robot navigation*, Norwood, MA, Kluwer Academic, 1992.
- [39] LOIZOU Savvas G, TANNER Herbert G, KUMARZ Vijay, KYRIAKOPOULOS Kostas J, *Closed Loop Motion Planning and Control for Mobile Robots in Uncertain Environments*.
- [40] MARQUES L, DE ALMEIDA A, *Advances in mobile robotics (Proceedings of the Eleventh International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines Coimbra, Portugal 8 -10 September 2008)*.
- [41] MCCOMB Gordon, PREDKO Mike, *Robot Builder's Bonanza*, 2006.
- [42] MEYER Jean-Arcady, GUILLOT Agnès, *Biologically Inspired Robots, Handbook of Robotics*, 2008.
- [43] MONDADA F, G. PETTINARO C, GUIGNARD A, KWEE I, FLOREANO D, DENEUBOURG J-L, NOLFI S, GAMBARDELLA L.M, and DORIGO M, *Swarm-bot: a new distributed robotic concept. Autonomous Robots*, 2004.
- [44] MONTIEL J.M.M, CIVERA J, DAVISON A.J, *Unified inverse depth parameterization for monocular SLAM*, 2006.
- [45] MORIN Pascal, SAMSON Claude, *Motion control of wheeled mobile robots*, 2007.
- [46] MURATA S, and KUROKAWA H, *Self-Reconfigurable Robot*, IEEE Robotics and Automation Magazine, 2007.

- [47] PAPOULIS A, *Probability, Random Variables, and Stochastic Processes*, 4th edition. New York, McGraw-Hill, 2001.
- [48] PEDERSEN L, KORTENKAMP D, WETTERGREEN D, NOURBAKHSI I, *A Survey of Space Robotics*, QSS group, Inc at NASA ARC, Metrica Inc. at NASA JSC, Carnegie Mellon University.
- [49] PHILLIPSEN R, SIEGWART R, *Smooth and efficient obstacles avoidance for a tourguide robot*, ICRA, 2003.
- [50] PIVTORAIKO, M, KNEPPER R. A, KELLY A, *Differentially constrained mobile robot motion planning in state lattices*, Journal of filed robotics 29, 2009.
- [51] RUFFLI M, FERGUSON D, SIEGWART R, *Smooth path planning in constrained environments*, ICRA, 2009.
- [52] RUFFLI M, SIEGWART R, *On the application of the  $D^*$  search algorithm to time based planning on lattice graphs*.
- [53] RUIZ DE GARIBAY PASCUAL Jonathan, *Robótica: Estado del arte*, Universidad de Deusto, 2007.
- [54] RUIZ DEL SOLAR J, SALAZAR R, *Robots móviles*, Universidad De Chile Facultad De CS. FS. y Matemáticas Departamento De Ingeniería Eléctrica.
- [55] SANTOS José, DURO Richard J, *Evolución artificial y robótica autónoma*, Primera edición, 2008.
- [56] SCARAMUZZA D, SIEGWART R, *Appearance guided monocular omnidirectional visual odometry for outdoor ground vehicles*, IEEE Transactions on Robotics 24 No 5, October, 2008.
- [57] SCARAMUZZA D, SIEGWART R, FRAUNDORFER F, *Real-time monocular visual odometry for on-road vehicles with 1-point RANSAC*, IEEE International Conference on Robotics and Automation, ICRA 2009.

- [58] SCARFOGLIERO U, STEFANINI C, DARIO P, *A bioinspired concept for high efficiency locomotion in micro robots: the jumping Robot Grillo*, Proc. IEEE Int. Conf. Robot. Autom, 2006.
- [59] SIEGWART Roland, NOURBAKHSH ILLAH R, SCARAMUZZA Davide, *Introduction to Autonomous Mobile Robots*, Second edition, 2011.
- [60] SOHANI Mohammad, KATSILIERIS Fotios, BAYRAKTAR Dogucan, *Mobile robot kinematics*.
- [61] WEINGARTEN J, GRUENER G and SIEGWART R, *A state-of-the-art 3D sensor for robot navigation*, Proceedings of IROS, Sendai, September, 2004.
- [62] WILLIAMS Karl, *Amphibionics, Build Your Own Biologically. Inspired Robot*, 2003.
- [63] XIE Ming, *Fundamentals of robotics linking perception to action*, 2003.
- [64] YAMADA H, and HIROSE S, *Development of Practical 3-Dimensional Active Cord Mechanism ACM-R4*, Journal of Robotics and Mechatronics, 2006.
- [65] ZUFFEREY J.C, *Bio-inspired flying robots: Experimental synthesis of autonomous indoor flyers*, 2008.

## **ANEXO A**

## OTRAS TÉCNICAS DE FILTRADO DE IMAGEN

Se trata de métodos para resaltar o suprimir, de forma selectiva, información contenida en una imagen a diferentes escalas espaciales, para destacar algunos elementos de la imagen, o también para ocultar valores anómalos.

El proceso de filtrado consiste en la aplicación a cada uno de los píxeles de la imagen de una matriz de filtrado de tamaño  $N \times N$  (generalmente de  $3 \times 3$  aunque puede ser mayor) compuesta por números enteros y que genera un nuevo valor mediante una función del valor original y los de los píxeles circundantes. El resultado final se divide entre un escalar, generalmente la suma de los coeficientes de ponderación. Los filtros se pueden expresar mediante la ecuación:

$$ND'_{i,j} = \frac{ND_{i-1,j-1} + ND_{i,j-1} + ND_{i+1,j-1} + ND_{i-1,j} + ND_{i,j} + ND_{i+1,j} + ND_{i-1,j+1} + ND_{i,j+1} + ND_{i+1,j+1}}{9}$$

Donde  $i$  y  $j$  representan la fila y la columna de cada píxel,  $ND_{i,j}$  su Nivel Digital y  $ND'_{i,j}$  el Nivel Digital obtenido tras hacer el filtrado. También pueden expresarse mediante tablas, por ejemplo las tablas que se representan a continuación. Mediante diferentes combinaciones de parámetros asignados a los diferentes píxeles circundantes se pueden conseguir diferentes efectos. En general los efectos deseados se relacionan con el aspecto borroso que tienen las imágenes de satélite, y también la fotografía aérea, debido a la dispersión atmosférica y la reflectividad lambertiana.

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} DIV = 9$$

Filtro de la media.

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} DIV = 10$$

Filtro de media ponderada que da mayor peso al valor central para evitar la pérdida de detalles.

Hay que tener en cuenta que los bordes de la imagen no podrán procesarse, ya que la ventana de filtrado saldría fuera de la imagen. De este modo se pierden  $N - 2$  filas y  $N - 2$  columnas por cada lado de la imagen.

El filtrado modifica necesariamente la imagen ya que pretenden extraer información cuantitativa que debe almacenarse como una nueva capa. Los filtros más utilizados son los de paso bajo (suavizan la imagen), de paso alto (aumentan el contraste), los filtros direccionales (detectan en la imagen estructuras que siguen una determinada dirección) y los de detección de bordes (permiten identificar y aislar objetos con propiedades homogéneas dentro de la imagen).

### ➤ **Filtros de paso bajo**

Su objetivo es suavizar la imagen, son útiles cuando se supone que la imagen tiene gran cantidad de ruido y se quiere eliminar. También pueden utilizarse para resaltar la información correspondiente a una determinada escala (tamaño de la matriz de filtrado); por ejemplo en el caso de que se quiera eliminar la variabilidad asociada a los tipos de cubierta presentes en la imagen uniformizando de esta manera su respuesta. Existen varias posibilidades:

- **Filtro de la media**, asigna al pixel central la media de todos los pixeles incluidos en la ventana. La matriz de filtrado estaría compuesta por unos y el divisor sería el número total de elementos en la matriz.
- **Filtro de media ponderada**, los elementos de la matriz de filtrado no son todos 1 sino que se da más peso a uno de ellos (generalmente el central) para obtener un resultado más parecido a la imagen original y evitar que aparezca borrosa.
- **Filtro de la mediana**, tiene la ventaja de que el valor final del pixel es un valor real presente en la imagen y no un promedio, de este modo se reduce el efecto borroso que tienen las imágenes que han sufrido un filtro de media. Además el filtro de la mediana es menos sensible a valores extremos. El inconveniente es que resulta más complejo de calcular ya que hay que ordenar los diferentes valores que aparecen en los pixeles incluidos en la ventana y determinar cuál es el valor central.

A continuación se pueden observar estas máscaras de filtro y su mecanismo de aplicación:

$$\begin{bmatrix} 20 & 23 & 30 & 31 \\ 22 & 21 & 29 & 30 \\ 23 & 24 & 32 & 33 \\ 29 & 31 & 34 & 37 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} N & N & N & N \\ N & 24.8 & 28.1 & N \\ N & 27.2 & 30.1 & N \\ N & N & N & N \end{bmatrix}$$

Filtro de media

$$\begin{bmatrix} 20 & 23 & 30 & 31 \\ 22 & 21 & 29 & 30 \\ 23 & 24 & 32 & 33 \\ 29 & 31 & 34 & 37 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} N & N & N & N \\ N & 24.5 & 28.2 & N \\ N & 26.9 & 30.3 & N \\ N & N & N & N \end{bmatrix}$$

Filtro de la mediana ponderada

$$\begin{bmatrix} 20 & 23 & 30 & 31 \\ 22 & 21 & 29 & 30 \\ 23 & 24 & 32 & 33 \\ 29 & 31 & 34 & 37 \end{bmatrix} \rightarrow \begin{bmatrix} N & N & N & N \\ N & 23 & 30 & N \\ N & 29 & 31 & N \\ N & N & N & N \end{bmatrix}$$

Filtro de mediana

- **Filtro adaptativos.** Son considerablemente más complejos ya que los coeficientes de ponderación se recalculan para cada uno de los píxeles en función del histograma de los ND que aparecen en la ventana. Se han utilizado con gran éxito filtros adaptativos para eliminar el *speckle* de las imágenes de radar y para detectar, con un solo filtro, diferentes elementos.
- **Filtros gaussianos.** Simulan una distribución gaussiana bivalente. El valor máximo aparece en el píxel central y disminuye hacia los extremos tanto más rápido cuanto menor sea el parámetro de desviación típica  $s$ . El resultado será un conjunto de valores entre 0 y 1. Para transformar la matriz a una matriz de números enteros se divide toda la matriz por el menor de los valores obtenidos. La ecuación para calcularla es:

$$g(x, y) = e^{-\frac{x^2+y^2}{2s^2}}$$

$$G(x, y) = \frac{g(x, y)}{\min_{x, y}(g(x, y))}$$

$$\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 20 & 33 & 20 & 4 \\ 7 & 33 & 55 & 33 & 7 \\ 4 & 20 & 33 & 20 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

Filtro gaussiano con  $s = 1$  y  $r = 2$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} DIV = 1$$

Matriz de filtrado identidad

Son una manera de obtener filtros de tipo genérico. Pueden ser útiles, por ejemplo, cuando se asume que la respuesta espectral de un pixel es función de la reflectividad de los pixeles vecinos atenuada en función de la distancia.

### ➤ Filtros de paso alto

Su objetivo es resaltar las zonas de mayor variabilidad eliminando lo que sería la componente media, precisamente la que detectan los filtros de paso bajo. Por otra parte la respuesta de cada pixel está *contaminada* por la de los pixeles vecinos ya que, considerando la superficie terrestre como lambertiana, la radiación reflejada por un pixel se reparte hacia los pixeles vecinos. Los filtros de paso alto consiguen también eliminar en parte esta contaminación. Existen diversos métodos:

- **Sustracción de la media.** Si se considera que un filtro de paso bajo sirve para resaltar componentes a gran escala eliminando la variabilidad local, si a la imagen original se le resta el resultado de pasarle un filtro de paso bajo se consigue resaltar esa variabilidad local. La matriz de filtrado de este filtro *menos media* puede calcularse directamente restando a la matriz de filtrado *identidad* la matriz de filtrado de la media.
- **Filtros basados en las derivadas.** La derivada de una función  $y = f(x)$  es el incremento de  $y$  para cada incremento infinitesimal de  $x$ . En el caso

de *Modelo Digital de Elevaciones* la derivada es la pendiente. La segunda derivada es la derivada de la derivada, en el caso de un *MDE* nos da información acerca de la forma (ladera recta, cóncava o convexa, valle, cresta o cima) del terreno. En el caso de una imagen de satélite nos va a informar de cómo son los cambios, más o menos bruscos, que se producen entre píxeles contiguos. A continuación aparecen las ecuaciones de las derivadas respecto a  $x$  e  $y$ , las segundas derivadas respecto a  $x$  e  $y$ , la derivada compuesta y la derivada segunda compuesta denominada también laplaciana.

$$\Delta xp(i, j) = p(i, j) - p(i - 1, j)$$

$$\Delta yp(i, j) = p(i, j) - p(i, j - 1)$$

$$\Delta px_2p(i, j) = \Delta xp(i + 1, j) - \Delta xp(i, j) = p(i + 1, j) + p(i - 1, j) - 2p(i, j)$$

$$\Delta py_2p(i, j) = \Delta yp(i, j + 1) - \Delta yp(i, j) = p(i, j + 1) + p(i, j - 1) - 2p(i, j)$$

$$\Delta xyp(i, j) = \text{sqr} \Delta xp(i, j)^2 + \Delta yp(i, j)^2$$

$$\Delta xy_2p(i, j) = [p(i + 1, j) + p(i - 1, j) + p(i, j + 1) + p(i, j - 1)] - 4p(i, j)$$

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} DIV 9$$

Filtro menos media

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} DIV = 1$$

Filtro laplaciano

El filtro laplaciano se recomienda para el realce de rasgos lineales en entornos urbanos. Otra opción para resaltar los elementos de mayor variabilidad es restar a la imagen original la obtenida mediante un filtrado laplaciano.

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} DIV = 1$$

Filtro menos-laplaciano

➤ **Filtros direccionales**

Se utilizan para detectar estructuras que siguen una determinada dirección en el espacio resaltando el contraste entre los píxeles situados a ambos lados de la estructura. A continuación se muestran como ejemplo las matrices para resaltar estructuras en dirección Este-Oeste y Norte-Sur respectivamente.

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix} DIV = 1 \quad \begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix} DIV = 1$$

Filtros Norte y Este

➤ **Filtros para la detección de bordes**

Uno de los más utilizados es el detector de bordes de Sobel que realiza la variación entre filas y columnas. Es algo más complejo que los anteriores:

$$S_{i,j} = \sqrt{C^2 + F^2}$$

Donde  $C$  y  $F$  resultan de pasar las máscaras de filtro a continuación:

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

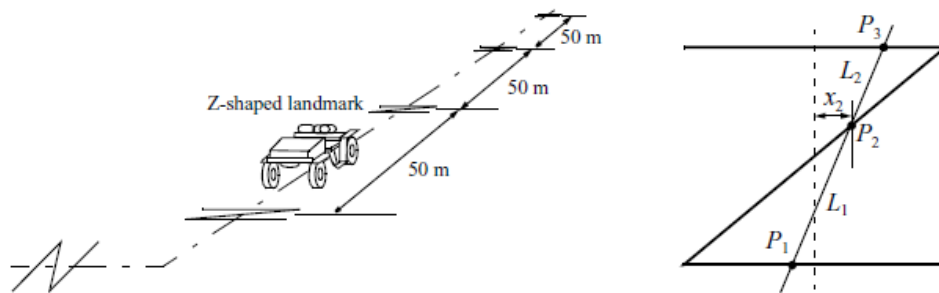
Filtro C y F de Sobel

## **ANEXO B**

## OTROS EJEMPLOS DE SISTEMAS DE LOCALIZACIÓN

La localización de Markov y la localización del filtro de Kalman han sido dos estrategias muy populares para los sistemas de investigación de robots móviles que navegan en ambientes interiores. Ellos tienen fuertes bases formales y por lo tanto un comportamiento bien definido. Sin embargo, hay un gran número de otras técnicas de localización que se han utilizado con grados variables de éxito en plataformas de robots móviles comerciales y de investigación. No se va a explorar el espacio de todos los sistemas de localización en detalle. No se entrará en detalle respecto a estas otras técnicas pero, sin embargo, hay varias categorías de técnicas de localización que merecen ser mencionadas.

No es sorprendente que muchas implementaciones de estas técnicas en robots comerciales empleen modificaciones del entorno del robot, algo que la localización de Markov y localización de filtro de Kalman evitan. En las secciones siguientes, se identifican brevemente la estrategia general incorporada por cada categoría y se referencian algunos ejemplos, incluyendo, en su caso, las que modifican el medio ambiente y las que operan sin modificación del medio ambiente.



*Puntos de referencia en forma de Z en el terreno. Komatsu Ltd., Japón.*

### ➤ Navegación basada en puntos de referencia (Landmark).

Los puntos de referencia o “señales” se definen generalmente como objetos pasivos en el medio ambiente que proporcionan un alto grado de precisión de localización cuando están dentro del campo de vista del robot. Los robots móviles que hacen uso de puntos de referencia para la localización suelen utilizar

marcadores artificiales que han sido colocados por los diseñadores del robot para hacer fácil la localización.

El sistema de control para un navegador basado en puntos de referencia consta de dos fases discretas. Cuando un punto está a la vista, el robot localiza con frecuencia y precisión, usando la actualización de acción y la actualización de percepción para rastrear su posición sin error acumulativo. Pero cuando el robot se encuentra en una "zona" donde no hay puntos de referencia, entonces, se produce la actualización de acción, y el robot acumula incertidumbre de posición hasta que el siguiente punto entre en el campo del robot de vista. El robot usa así efectivamente navegación por estima en la zona que hay entre un punto de referencia y otro.

Esto a su vez significa que el robot debe consultar su mapa con cuidado, asegurándose de que cada movimiento entre los puntos de referencia sea suficientemente corto, dado su modelo de movimiento, que será capaz de localizar con éxito al llegar a la siguiente marca. La figura anterior muestra una instancia de localización basada en puntos de referencia (landmark). La forma particular de los puntos de referencia permite una estimación de la posición por el robot fiable y precisa, el cual debe viajar utilizando navegación por estima entre los puntos de referencia. Una ventaja clave del enfoque de navegación basada en puntos de referencia en que es una teoría formal fuerte ha sido desarrollada para esta arquitectura del sistema general. La principal desventaja la navegación basada en puntos de referencia es que en general requiere una modificación ambiental significativa. Los puntos son locales, y por lo tanto se requiere un gran número para cubrir un área amplia de una fábrica o de un laboratorio de investigación. Por ejemplo, el laboratorio de robótica en Stanford hizo uso de aproximadamente treinta puntos de referencia discretos, todos colocados individualmente en el techo.

#### ➤ **Localización única global.**

El enfoque de navegación basado en puntos de referencia hace una suposición general fuerte: cuando el punto está en el campo de vista del robot, la localización es esencialmente perfecta. Una forma de alcanzar el "Santo Grial" de la localización en robótica móvil es efectivamente permitir que tal suposición sea válida sin importar dónde se encuentre el robot. Sería revolucionario si un vistazo a los sensores del robot identificara inmediatamente su situación particular, única y

repetidamente. Tal estrategia para la localización es seguramente agresiva, pero la cuestión de si se puede hacer es principalmente una cuestión de tecnología de sensores y de software de detección. Es evidente que tal sistema de localización tendría que utilizar un sensor que recoja una cantidad muy grande de información. Dado que la visión en efecto recoge mucha más información que los sensores anteriores, y se ha utilizado como el sensor de elección en la investigación hacia localización única global.

Si los humanos fueran capaces de mirar un cuadro catadióptrico e identificar la ubicación del robot en un entorno conocido, entonces se podría argumentar que la información para la localización única global existe dentro de la imagen. Uno de estos métodos se ha intentado por varios investigadores e implica la construcción de uno o más histogramas de las imágenes para representar el contenido de información de una imagen de forma. Un robot que utiliza un sistema de histograma de imágenes ha demostrado identificar exclusivamente habitaciones individuales en edificios de oficinas, así como aceras individuales en un entorno al aire libre. Sin embargo, tal sistema es muy sensible a la iluminación externa y proporciona sólo un nivel de resolución de localización igual a la huella visual de la óptica de la cámara.

El histograma angular (representado en capítulo de percepción) es otro ejemplo en el que los valores de los sensores del robot se transforman en un identificador de ubicación. Sin embargo, debido al contenido de la información limitada los picos de alcance de sonar, es probable que los dos lugares en el entorno del robot pueden tener histogramas angulares que son demasiado similares para ser diferenciados con éxito.

Una forma de intentar reunir información de sonar suficiente para localización global es permitir que el tiempo de robot recoja una gran cantidad de datos de sónar en una rejilla de evidencia local (es decir, red de ocupación) primero, luego, hacer coincidir la red de evidencia local con un mapa métrico global del entorno. En el trabajo desarrollado por los investigadores Schultz, A., Adams, W., "Continuous Localization Using Evidence Grids," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'98,*) demuestran que tal sistema es capaz de localizar sobre la marcha incluso cuando se realizan cambios significativos para el medio ambiente, degradando la fidelidad del mapa. Más interesante es que la cuadrícula de evidencia local representa la información de lo suficientemente bien que se puede utilizar para corregir y actualizar el mapa en el tiempo, lo que conduce a un sistema de localización que proporciona

retroalimentación correctiva para la representación ambiental directamente. Esto es similar en esencia a la idea de tomar características observadas rechazadas en el algoritmo de localización del filtro de Kalman y utilizarlos para crear nuevas características en el mapa.

Un nuevo método más prometedor, para la localización única global se llama localización basada en mosaico (*Kelly, A., "Pose Determination and Tracking in Image Mosaic Based Vehicle Position Estimation [2000]"*). Este enfoque fascinante aprovecha una característica ambiental que se utiliza muy poco por robots móviles: la textura fina del grano del piso. Este método se logra principalmente debido a la ubicuidad reciente de procesadores muy rápidos, cámaras muy rápidas, y medios de almacenamiento muy grandes. El robot se equipa con una cámara CCD de alta calidad y alta velocidad apuntando hacia el piso, situado entre las ruedas del robot, e iluminado por un modelo especializado de luz en el eje de la cámara para mejorar la textura del suelo. El robot comienza recogiendo imágenes de todo el suelo en su espacio de trabajo con esta cámara. Por supuesto, los requisitos de memoria son significativos, lo que requiere una unidad de 10 GB con el fin de almacenar la biblioteca de imagen completa de un área de 300 x 300.

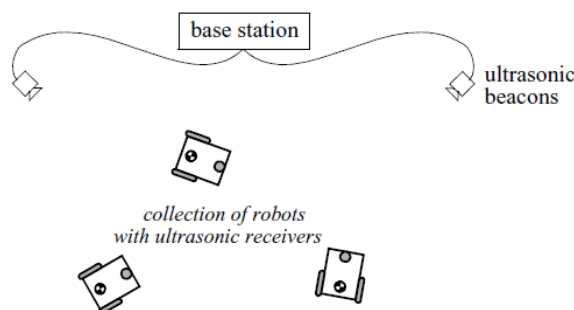
Una vez que el mosaico de imagen completo se almacena, el robot puede recorrer cualquier trayectoria en el piso mientras que sigue su propia posición sin dificultad. La localización se realiza simplemente grabando una imagen, realizando la acción de actualización, y luego, la actualización de percepción, haciendo coincidir la imagen a la base de datos de mosaico usando técnicas simples basadas en la coincidencia de imagen de la base de datos. El rendimiento resultante ha sido impresionante: un robot se ha demostrado poder localizar repetidamente con precisión de 1 mm mientras que se mueve a 25 km / h. La principal ventaja de la localización única global es que, cuando estos sistemas funcionan correctamente, simplifican en gran medida la navegación del robot. El robot puede moverse a cualquier punto y siempre estar seguro de localizar mediante la recopilación de un escaneo de sensor.

Sin embargo, la principal desventaja de localización única global es que es probable que este método nunca ofrezca una solución completa al problema de la localización. Siempre habrá casos en los que la información sensorial local es verdaderamente ambigua y, por lo tanto, es improbable que la localización única global usando sólo información actual del sensor tenga éxito. Los humanos a

menudo tienen excelentes sistemas de posicionamiento local, particularmente en ambientes no repetitivos y bien conocidos, tales como sus hogares. Sin embargo, hay una serie de entornos en los que la localización inmediata, es un reto incluso para los seres humanos: considérense laberintos de cobertura y grandes edificios de oficinas con pasillos repetidos que son idénticos. De hecho, el prototipo de localización basado en mosaico descrito anteriormente se encontró con un problema en su primera aplicación. El piso de la planta de la fábrica había sido recién pintado y por lo tanto carecía de suficientes micro fracturas para generar la textura para la correlación. Su solución fue modificar el entorno después de todo, pintando una textura al azar en el piso de la fábrica.

### ➤ **Sistemas de posicionamiento de baliza.**

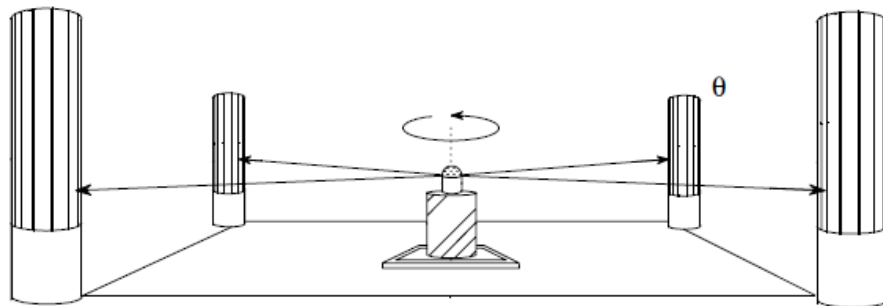
Una de las soluciones más fiables para el problema de localización es diseñar e implementar un sistema de baliza activa específicamente para el entorno de destino. Esta es la técnica preferida utilizada por la industria y aplicaciones militares como una manera de garantizar la máxima fiabilidad posible de localización. El sistema GPS se puede considerar simplemente como uno de estos sistemas. La figura siguiente representa tal disposición de baliza para una colección de robots. Así como con el GPS, mediante el diseño de un sistema en el que los robots localicen pasivamente mientras que las balizas están activas, cualquier número de robots pueden simultáneamente tomar ventaja de un sistema de baliza única. Como con la mayoría de los sistemas de baliza, el diseño representado depende ante todo de principios geométricos para efectuar la localización. En este caso, los robots deben conocer las posiciones de las dos balizas ultrasónicas activas en el marco de coordenadas global con el fin de localizarse a sí mismos para el sistema de coordenadas global.



*Balizas ultrasónicas Activas*

Un tipo popular de sistema de señal en aplicaciones robóticas industriales se representa en la siguiente figura. En este caso balizas son marcadores retrorreflectantes que pueden ser fácilmente detectados por un robot móvil basado en la reflexión de la energía de vuelta al robot. Teniendo en cuenta las posiciones conocidas de los retrorreflectores ópticos, un robot móvil puede identificar su posición cada vez cuando tiene tales tres balizas a la vista al mismo tiempo. Por supuesto, un robot con codificadores pueden localizarse en el tiempo también, y no necesita medir su ángulo con respecto a los tres balizas en el mismo instante.

La ventaja de tales sistemas mediante balizas es generalmente su extrema fiabilidad. Por la misma razón, la ingeniería generalmente rodea la instalación de tal sistema en un entorno comercial específico. Por lo tanto, mover el robot a otro piso de la fábrica será tanto, lento como también costoso. Por lo general, incluso cambiando las rutas utilizadas por el robot requeriría una seria re-ingeniería



*Balizas ópticas pasivas*

➤ **Ruta basada en la localización.**

Incluso más confiable que los sistemas basados en baliza son las estrategias de localización basadas en rutas. En este caso, el recorrido del robot está explícitamente marcado para que se pueda determinar su posición, no en relación con algún sistema de coordenadas global, pero relativo a la trayectoria concreta que se le permite viajar. Hay muchas técnicas para el marcado de una ruta y las intersecciones posteriores. En todos los casos, uno está creando efectivamente un sistema ferroviario, excepto que el sistema ferroviario es algo más flexible y ciertamente más amigable que un carril físico. Por ejemplo, una pintura ópticamente transparente de alta reflexión ultravioleta puede marcar la ruta de tal

manera que sólo el robot, utilizando un sensor especializado, fácilmente la detecte. Alternativamente, un alambre guía enterrado debajo del pasillo puede ser detectado usando bobinas inductivas situadas en el chasis del robot.

En todos estos casos, el problema de localización de robot está efectivamente trivializado forzando a que el robot siga siempre un camino prescrito. Para ser justos, hay nuevos vehículos industriales no tripulados guiados que se apartan brevemente de su ruta para evitar los obstáculos. Sin embargo, el costo de esta fiabilidad extrema es evidente: el robot es mucho más inflexible, dados tales medios de localización, y por lo tanto cualquier cambio en el comportamiento del robot requiere ingeniería y tiempo significativo.

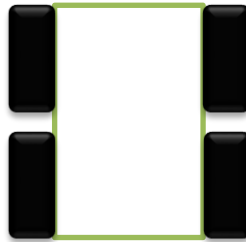
## **ANEXO C**

# PRUEBAS Y TALLERES

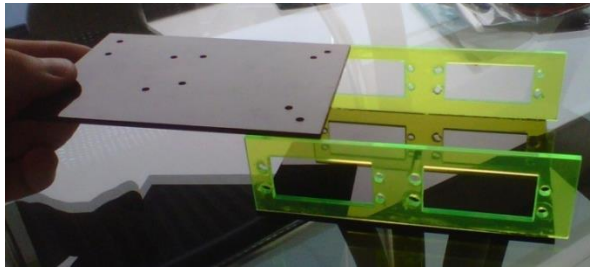
## ROBOT DE PRUEBAS LEVI

### Primera fase de construcción del robot de pruebas

Iniciamos con la construcción de un robot de cuatro ruedas, controladas individualmente, por sus ventajas, como estabilidad, potencia y seguridad al subir rampas. Ya que el objetivo era que el robot se desempeñara a nivel competitivo (seguir línea, cruces agudos, superar rampas, evitar choques y seguir pared).



- Se trabajó con un **Chasis en acrílico**, para evitar inconvenientes con la parte electrónica del robot. (Realizado en ACRILTEC).



- Se usaron **Ruedas de neopreno** para garantizar un mejor agarre y estabilidad debido a las propiedades del material.



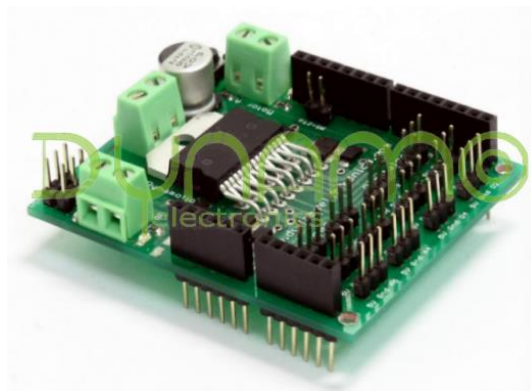
- **Servos Hitec HS-311** truncados, para el control de la velocidad mediante la modulación de la señal de entrada.



- **Arduino UNO**, una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar.



- **Dynamotion**, una shield para Arduino diseñada para controlar dos motores DC, con corriente max de 2A por cada motor, además cuenta con salidas y entradas para conectar sensores u otros dispositivos.

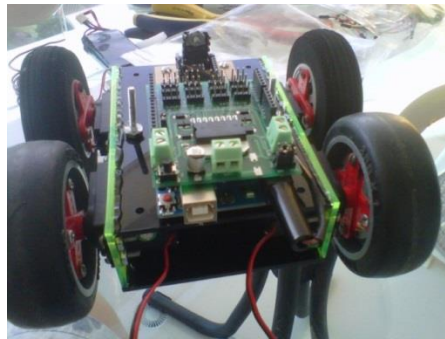


- Inicialmente se usó una **Batería polímero litio Ion 1000mA 7.4V**, pero luego de las primeras pruebas y al ver que la velocidad del robot era

limitada debido a la poca potencia de la batería, Se tomó la decisión de cambiar a una de **1500mA 11.1V**



### ***Resultado de la primera fase de construcción***

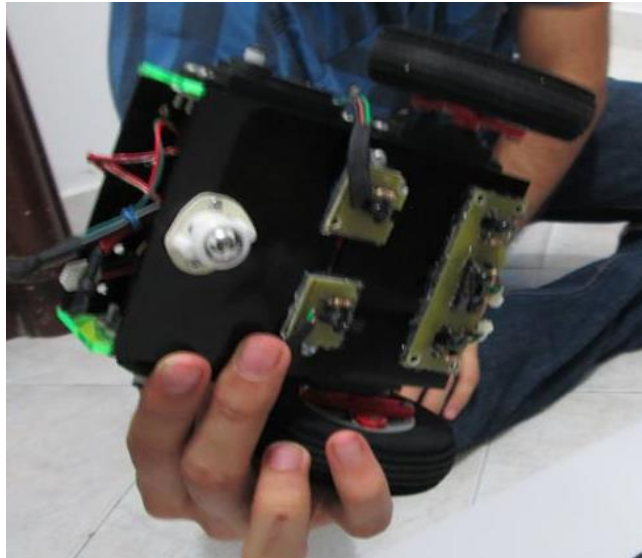


Al avanzar en la construcción del robot y la investigación del seminario, nos percatamos de los defectos y problemas de esta configuración a nivel cinemático y dinámico, su complejidad y grandes costos. Se decidió reestructurar el robot en base a la **configuración diferencial**, una configuración más sencilla y práctica para iniciar en la robótica móvil y cuyo modelo cinemático y dinámico ha sido estudiado ampliamente.



Configuración Diferencial

## ROBOT CON CONFIGURACION DIFERENCIAL



Los servos están conectados a pines de salida PWM de la tarjeta ARDUINO UNO que nos permite controlar su velocidad mediante la amplitud de la señal, ya que estos pines tienen una resolución de 8 bits, nos dan un rango de 256 posiciones (0-255)

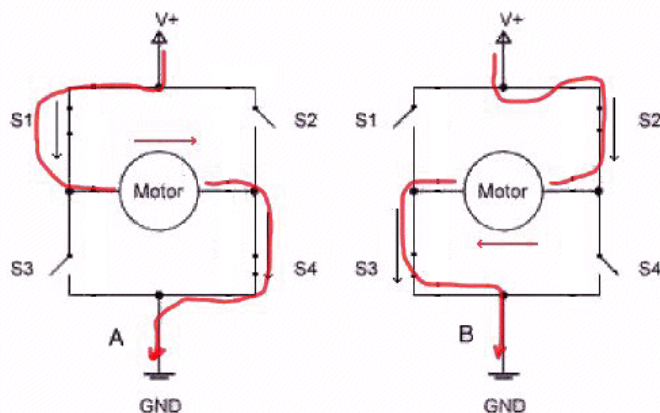
Servo Derecha:

- pin D3
- pin D5

Servo Izquierda:

- pin D6
- pin D9

La dirección de los servos es controlada por el dynamotion que básicamente cumple la función de un puente H que puede controlar hasta dos motores a la vez.



Motor Izquierdo Adelante

- analogWrite(3, 100);
- analogWrite(5, 0);

Motor Derecho Adelante

- analogWrite(6, 100);
- analogWrite(9, 0);

Motor Izquierdo Atrás

- analogWrite(3, 0);
- analogWrite(5, 100);

Motor Derecho Atrás

- analogWrite(6, 0);
- analogWrite(9, 100);

Motor Izquierdo Stop

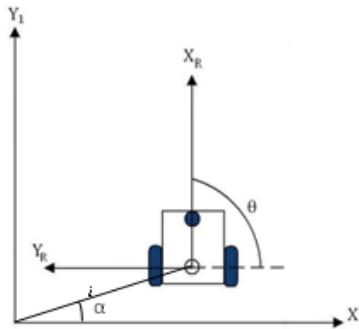
- analogWrite(3, 0);
- analogWrite(5, 0);

Motor Derecho Stop

- analogWrite(6, 0);
- analogWrite(9, 0);

## TALLER DE CINEMÁTICA

Usando el modelo cinemático para un robot diferencial, obtenido anteriormente en el capítulo 8, se desarrolló un programa que nos muestra la posición del robot en el tiempo. Según algunos parámetros iniciales y suponiendo que la velocidad angular de las ruedas se mantiene constante.

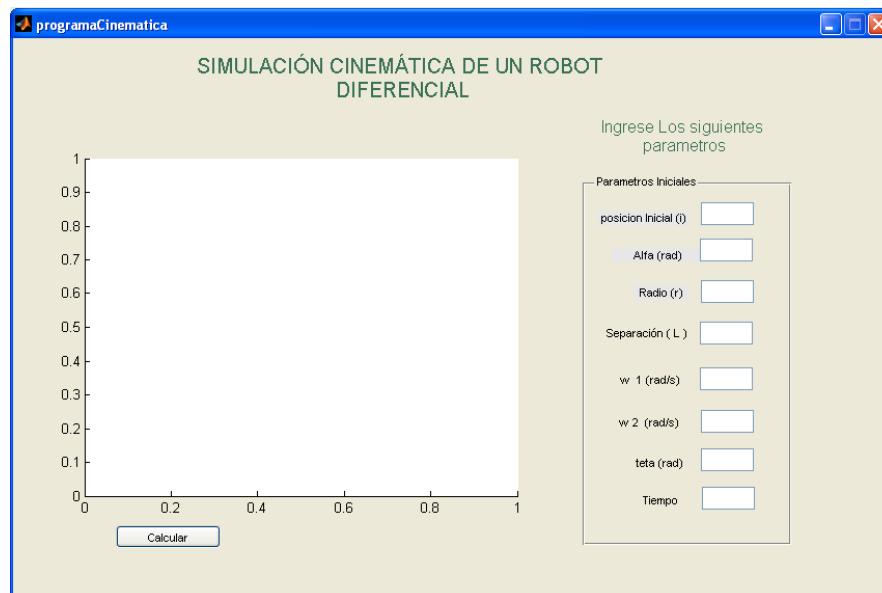


$$\dot{\xi}_I = R(\theta)^{-1} \begin{bmatrix} \frac{r\dot{\phi}_1}{2} + \frac{r\dot{\phi}_2}{2} \\ 0 \\ \frac{r\dot{\phi}_1}{2l} + \frac{-r\dot{\phi}_2}{2l} \end{bmatrix}$$

$$R(\theta)^{-1} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\dot{\xi}_I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}$$

El entorno gráfico se diseñó de la siguiente manera



## Programación

Dentro del código

```
%LECTURA DE PARÁMETROS INICIALES
i=str2double(get(handles.edit1,'String'));
alfa=str2double(get(handles.edit8,'String'));
r=str2double(get(handles.edit7,'String'));
l=str2double(get(handles.edit2,'String'));
w1=str2double(get(handles.edit3,'String'));
w2=str2double(get(handles.edit4,'String'));
teta=str2double(get(handles.edit5,'String'));
pasos=str2double(get(handles.edit6,'String'));
x=zeros(1,pasos);
y=zeros(1,pasos);

%POSICIÓN INICIAL
x(1,1)=i*cos(alfa);
y(1,1)=i*sin(alfa);

%APLICACIÓN DEL MODELO
for ind=2:pasos

    m1=[cos(teta), -sin(teta), 0; ...
        sin(teta),  cos(teta),  0; ...
        0,          0,          1];
    m2=[r*w1/(2)+r*w2/(2); ...
        0; ...
        r*w1/(2*1)-r*w2/(2*1)];
```

```

difer=m1*m2;% resultado ? ?_I

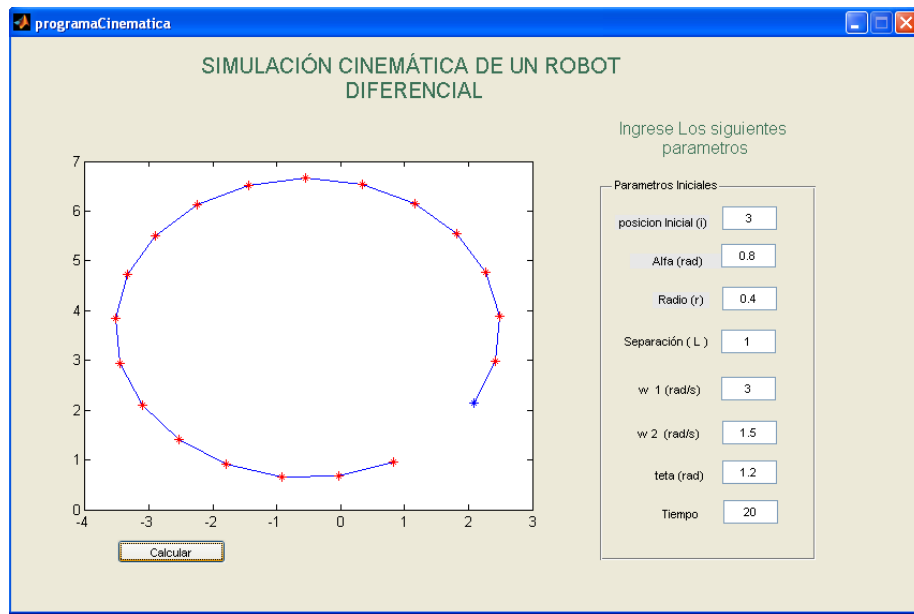
x(1,ind)=x(1,ind-1)+difer(1);
y(1,ind)=y(1,ind-1)+difer(2);
teta=teta+difer(3);
idif=sqrt(x(1,ind)^2+y(1,ind)^2);
i=i+idif;

end

%GRAFICAMOS
axis(handles.axes1);
plot(x,y);
hold on
for ind=2:pasos
plot(x(ind),y(ind),'r*');
plot(x(1),y(1),'b*')
end
hold off

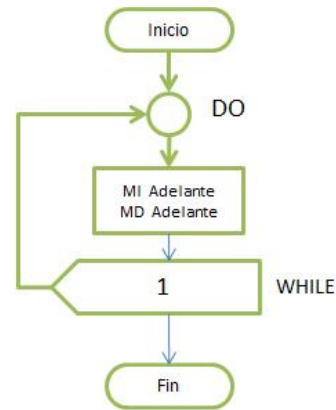
```

## Resultados



Utilizando el robot de pruebas LEVI, se pudo comprobar, que el análisis cinemático anteriormente mencionado es correcto.

Al definir la velocidad de la rueda derecha el doble de la velocidad de la rueda izquierda, la trayectoria que muestra el robot es una circunferencia, muy similar a la generada por el programa que se creó en Matlab.



- Motor Izquierdo Adelante  
analogWrite(3, 50);  
analogWrite(5, 0);

- Motor Derecho Adelante  
analogWrite(6, 0);  
analogWrite(9, 100);

## TALLER DE DINÁMICA

Usando la forma de primer orden de las ecuaciones dinámicas de un robot diferencial, realizar un programa que muestre gráficamente la variación en el tiempo de las ecuaciones de movimiento, según los valores de  $\tau_r$  y  $\tau_l$  y parámetros iniciales del robot.

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \theta \\ \dot{x}_{r1} \\ \dot{\theta} \end{bmatrix}$$

$$\dot{q}_1 = q_4 \cos q_3, \quad \dot{q}_2 = q_4 \sin q_3$$

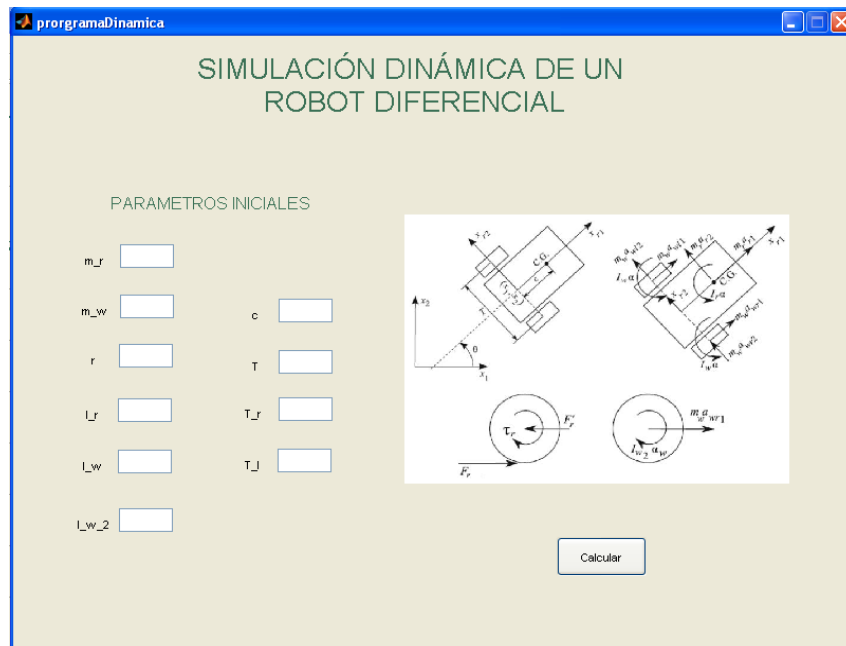
$$\dot{q}_3 = q_5, \quad \dot{q}_4 = \frac{1}{m} \left( \frac{1}{r} (\tau_r + \tau_l) + m_r c q_5^2 \right)$$

$$\dot{q}_5 = \frac{T}{2I} \frac{1}{r} (\tau_r - \tau_l) \quad (5.20)$$

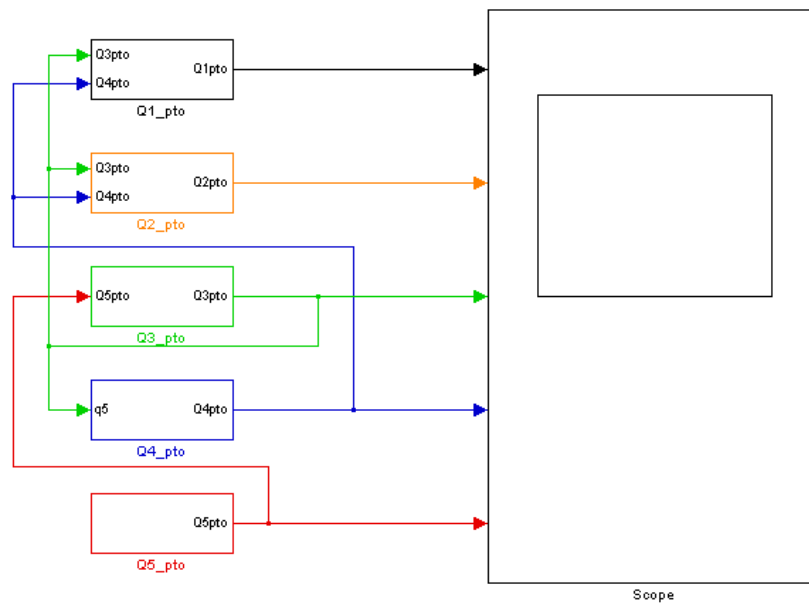
**Donde**

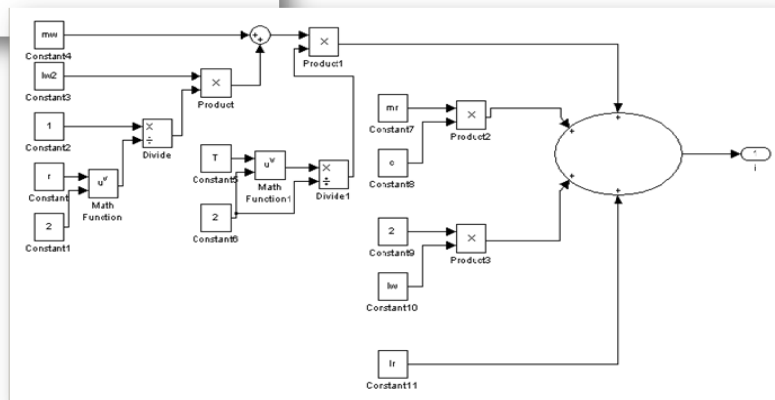
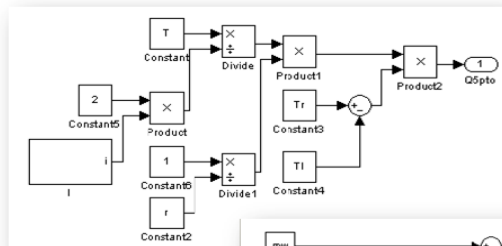
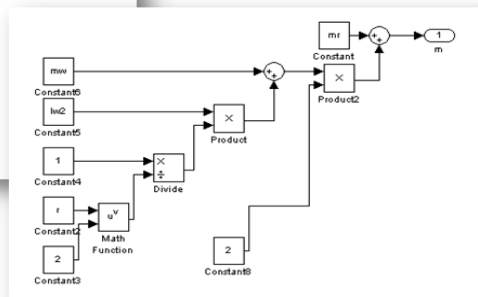
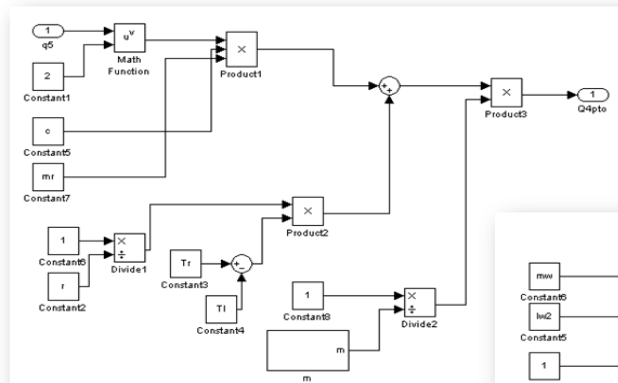
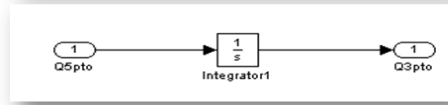
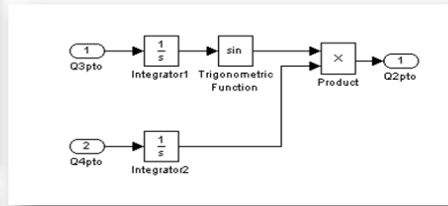
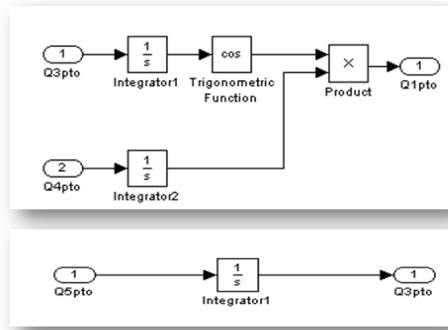
$$m = m_r + 2 \left( m_\omega + \frac{1}{r^2} I_\omega \right), \quad I = I_r + 2I_\omega + \left( m_\omega + \frac{1}{r^2} I_\omega \right) \frac{T^2}{2} + m_r c$$

## Entorno Grafico



## Programación Simulink

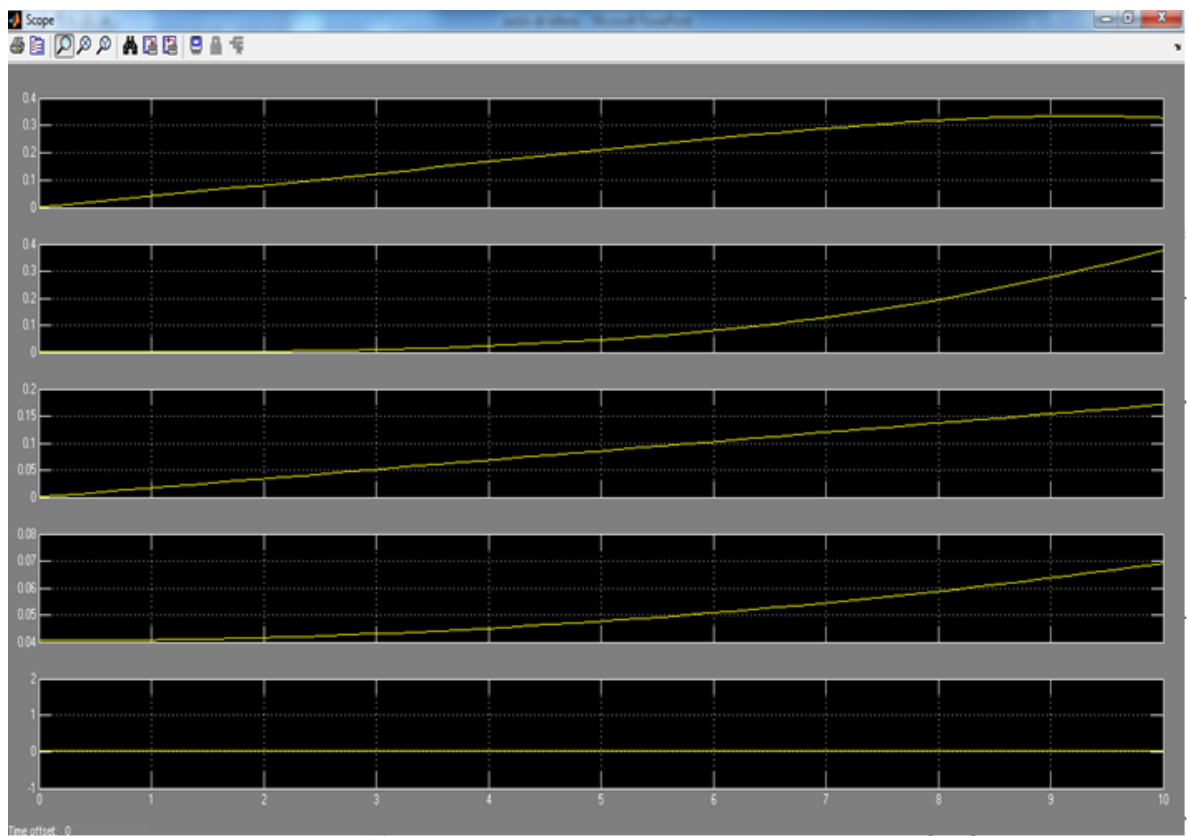




**Para los siguientes parámetros iniciales**

$m_r$	4		
$m_w$	2	$c$	2
$r$	3	$T$	4
$L_r$	4	$T_r$	3
$L_w$	5	$T_J$	2
$L_w_2$	1		

**El resultado es:**



## TALLERES DE PERCEPCIÓN

Se realizaron diversos programas en Matlab en base a los siguientes temas de este capítulo:

- Seguimiento de Color
- Tratamiento de Imágenes
- Procesamiento de Video

También sugerimos un programa de Correspondencia de puntos que se puede encontrar en la página de internet [www.mathworks.com](http://www.mathworks.com) y se desglosan las características de los sensores usados para el desarrollo del robot de pruebas LEVI y las rutinas que se pueden realizar con cada uno de ellos.

### SEGUIMIENTO DE COLOR

La aplicación más usada en el seguimiento de color es la CMUCAM, pero debido a inconvenientes con el tiempo de desarrollo y limitaciones de presupuesto se desarrolló un programa de detección y seguimiento de colores primarios a través de cámara Web, para ejemplificar los fundamentos de la visión por computador tratados en el capítulo de percepción. En la siguiente figura se puede apreciar el entorno gráfico del programa:

#### Entorno Gráfico



## Programación

El entorno gráfico posee tres buttons que permiten seleccionar el color que se quiere detectar. El código de detección de color azul inicia de la siguiente manera:

```
clear all;clc;
vid=videoinput('winvideo',1,'YUY2_320x240');
set(vid,'TriggerRepeat',Inf);
vid.ReturnedColorspace='rgb';
vid.FrameGrabInterval=2;
start(vid)
```

- Primero se captura un stream de video usando *videoinput*, con argumento de *winvideo*, número de dispositivo y formato de la cámara.
- Luego se configuran las opciones de adquisición de video *set* y *framegrabinterval* significa que tomara cada 2 frame del stream de video adquirido; con *start(vid)* se activa la adquisición de video.

Posteriormente:

```
while(vid.FramesAcquired<=200)
    data1=getdata(vid,1);
    data=imcomplement(data1);
    diff_im = imsubtract(data(:,:,1), rgb2gray(data));
    diff_im=medfilt2(diff_im,[3 3]);
    diff_im=im2bw(diff_im,0.1);
    diff_im=bwareaopen(diff_im,50);
    bw=bwlabel(diff_im,8);
    stats=regionprops(bw,'BoundingBox','Centroid');
```

- Se crea un bucle while, y despues de 200 frames adquiridos se sale del bucle para evitar inconvenientes.
- Con *data = getdata(vid)*; se toma un snapshot del stream y se le almacena en *data* para trabajar más fácil
- Se reconoce el color a continuación de la imagen en escala de grises de la imagen adquirida en *data*: *diff\_im = imsubtract(data(:,:,1), rgb2gray(data))*;
- *medfilt2* filtra la señal del ruido
- *im2bw* convierte la imagen en escala de grises a una imagen binaria.
- *Bwareaopen* determina el tamaño a reconocer para descartar imagen de menos de 50 pixels y con *regionprops* se configura la región etiquetada.

Con el siguiente bucle lo que se hace es encerrar al objeto de color azul en un rectángulo con una cruz en el centroide; los colores del rectángulo y la cruz se pueden variar fácilmente:

```
for object=1:length(stats)
```

```

bb=stats(object).BoundingBox;
bc=stats(object).Centroid;
rectangle('Position',bb,'EdgeColor','b','LineWidth',2);
plot(bc(1),bc(2),'-m+');
a=text(bc(1)+15,bc(2),strcat('X:',num2str(round(bc(1))),'Y:',
num2str(round(bc(2)))));
set(a,'FontName','Arial','FontWeight','bold','FontSize',
12,'Color','green');

```

Con *stop(vid)* se detiene la captura. El código para la detección de colores rojo y amarillo es exactamente igual, la única diferencia radica en la función:

```
diff_im = imsubtract(data(:,:,1)
```

Donde sólo cambia el número a 2 y 3 para la detección de colores primarios.

Los resultados fueron los siguientes:

### ***Detección de color rojo***



### ***TRATAMIENTO DE IMÁGENES***

Este tema es bastante importante, pues nos permite extraer de una manera fácil y confiable las características del entorno, para usar en el robot

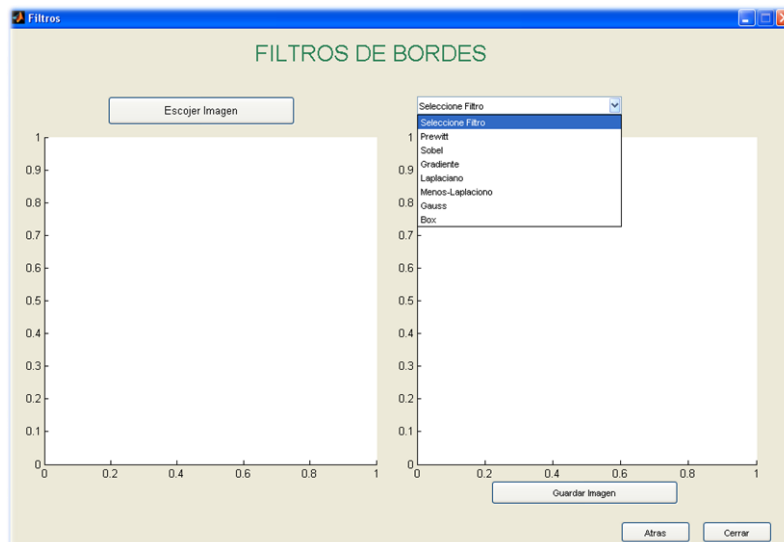
## Entorno Grafico



Cada uno de estos cuatro temas se manejó de la siguiente manera:

### ***Detección de Bordes***

Aplicando los diferentes filtros de bordes desarrollados en el capítulo de percepción y algunos extras que se muestran en los anexos, a imágenes prediseñadas, podemos demostrar los efectos de cada uno de ellos y sus diferencias.



## **Programación**

### **case 2 % Filtro Prewitt**

```
Img = getimage(handles.axes1);
if Img == 0
    return
end

Im=rgb2gray(Img);
id=double(Im);
Hx=[-1 0 1;-1 0 1;-1 0 1]
Hy=[-1 -1 -1;0 0 0;1 1 1]
Gx=imfilter(id,Hx);
Gy=imfilter(id,Hy);
Ix=abs(Gx);
Iy=abs(Gy);
Gt=Ix+Iy;
subplot (handles.axes2),imshow(uint8(Gt)),title('Filtro Prewitt');
```

### **case 3 % Filtro Sobel**

```
Img = getimage(handles.axes1);
if Img == 0
    return
end

Im=rgb2gray(Img);
id=double(Im);
Hx=[-1 0 1;-2 0 2;-1 0 1]
Hy=[-1 -2 -1;0 0 0;1 2 1]
Gx=imfilter(id,Hx);
Gy=imfilter(id,Hy);
Ix=abs(Gx);
Iy=abs(Gy);
Gt=Ix+Iy;
subplot (handles.axes2),imshow(uint8(Gt)),title( 'Filtro Sobel');
```

### **case 4 % Filtro Gradiente**

```
Img = getimage(handles.axes1);
if Img == 0
    return
end

Im=rgb2gray(Img);
```

```

id=double(Im);
Hx=[-0.5 0 0.5];
Hy=[-0.5; 0; 0.5]
Gx=imfilter(id,Hx);
Gy=imfilter(id,Hy);
Ix=abs(Gx);
Iy=abs(Gy);
Gt=Ix+Iy;
subplot(handles.axes2),imshow(uint8(Gt)),title('Filtro Gradiente');

```

case 5 **%Filtro Laplaciano**

```

Img = getimage(handles.axes1);
if Img == 0
    return
end
Im=rgb2gray(Img);
id=double(Im);
DD=[0 1 0;1 -4 1;0 1 0]
n=imfilter(Im,DD);
subplot(handles.axes2),imshow(n*6),title('Filtro Laplaciano');

```

case 6 **% Filtro Menos-Laplaciano**

```

Img = getimage(handles.axes1);
if Img == 0
    return
end
Im=rgb2gray(Img);
id=double(Im);
X=[0 -1 0;-1 5 -1;0 -1 0]
n=imfilter(Im,X);
subplot(handles.axes2),imshow(n*6),title('Filtro Menos-Laplaciano');

```

case 7 **% Filtro Gauss**

```

Img = getimage(handles.axes1);
if Img == 0
    return
end
I1=rgb2gray(Img);

H=[0 1 2 1 0; 1 3 5 3 1; 2 5 9 5 2; 1 3 5 3 1; 0 1 2 1 0];
FG=imfilter(I1,H);
sum(H);
sum(sum(H));

```

```
H=H/57;  
FG=imfilter(I1,H);  
CG=edge(FG,'canny');  
  
subplot(handles.axes2),imshow(CG),title('Filtro Gauss');
```

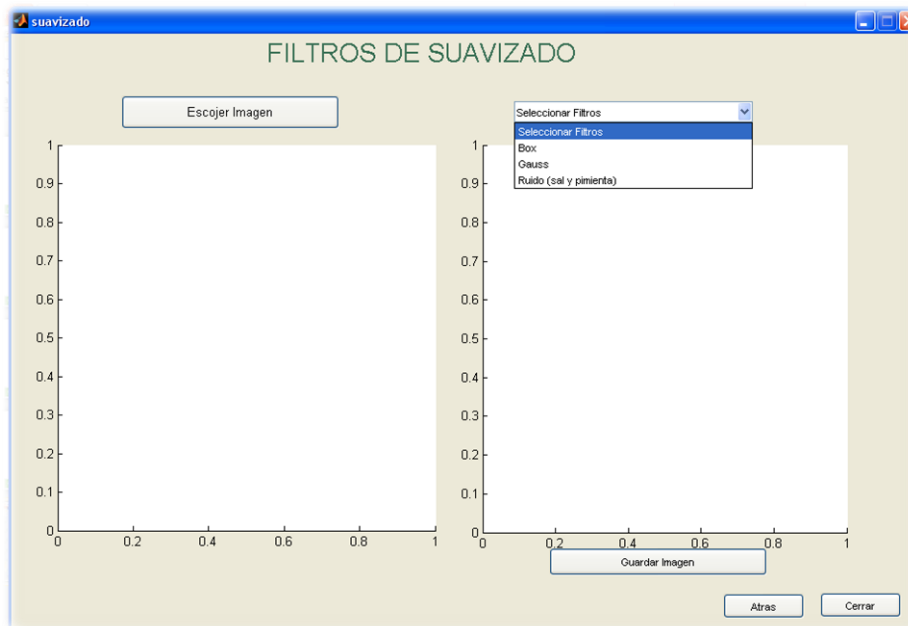
## Resultados

Detección de Bordes mediante el prewitt



## Suavizado

Aplicando los diferentes filtros de suavizado desarrollados en el capítulo y algunos extras que se muestran en los anexos, a imágenes prediseñadas, podemos demostrar los efectos de cada uno de ellos y sus diferencias.



### Programación

#### case 2 % **Filtro BOX**

```

Img = getimage(handles.axes1);
if Img == 0
    return
end
HB=ones(5,5)/25;
FB=imfilter(Img,HB);
subplot (handles.axes2),imshow(FB),title('Filtro BOX');

```

#### case 3 % **Filtro GAUSS**

```

Img = getimage(handles.axes1);
if Img == 0
    return
end
H=[0 1 2 1 0; 1 3 5 3 1; 2 5 9 5 2; 1 3 5 3 1; 0 1 2 1 0];
sum(H);
sum(sum(H));
H=H/57;
FG=imfilter(Img,H);
subplot (handles.axes2),imshow(FG),title('Filtro GAUSS');

```

#### otherwise % **ruido(Sal Pimienta)**

```

Img = getimage(handles.axes1);
if Img == 0
    return

```

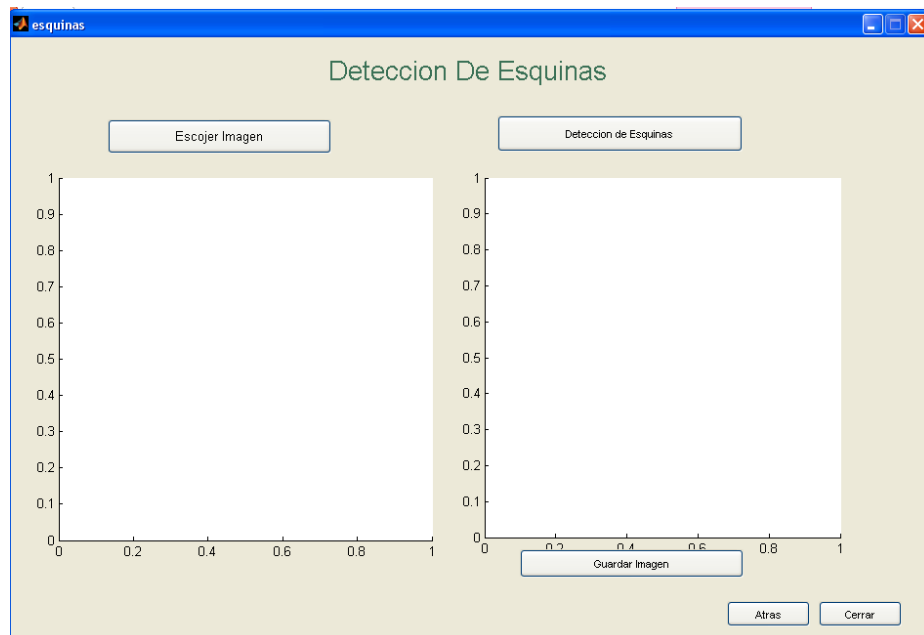
```
end
IR=imnoise(Img,'salt & pepper',0.01);
subplot(handles.axes2),imshow(IR),title('ruido(Sal Pimienta)');
```

### Resultado



### Detectores De Esquinas

Se desarrolló un programa para detección de esquinas según Harris.



## Programación

```
% Leemos nuestra imagen y la convertimos a escala de grises
Img = getimage(handles.axes1);
    if Img == 0
        return
    end
lr=rgb2gray(Img);

%se guardan las dimensiones de la imagen en una matriz MXN
[m n]=size(lr);
%declaramos 2 imágenes llenas de ceros para poner los resultados
U=zeros(size(lr));
S=zeros(size(lr));

h=ones(3*3)/9;
%Convertir imagen a tipo double
ld=double(lr);
%aplicación del filtro BOX
lf=imfilter(ld,h);
%aplicación del filtro Sobel
Hx=[-0.5 0 0.5];
Hy=[-0.5; 0; 0.5];
lx=imfilter(lf,Hx);
ly=imfilter(lf,Hy);
%Matriz de estructuras
HE11=lx.*lx;
HE22=ly.*ly;
HE12=lx.*ly;
%aplicación filtro gaussiano
Hg=[0 1 2 1 0; 1 3 5 3 1; 2 5 9 5 2; 1 3 5 3 1; 0 1 2 1 0]*(1/57);
A=imfilter(HE11,Hg);
B=imfilter(HE22,Hg);
C=imfilter(HE12,Hg);
%declaramos alfa para la sensibilidad de la detección de esquinas
alfa=0.04;
%se determina y obtiene el valor de la esquina
Rp=A+B;
Rp1=Rp.*Rp;
%Aplicación de la fórmula de Harris
Q=((A.*B)-(C.*C))-(alfa*Rp1);
U=Q>1000;
pixel=10;
%leemos la imagen por Filas y columnas
for r=1:m
    for c=1:n
        if (U(r,c))
```

```

I1=[r-pixel 1];
I2=[r+pixel m];
I3=[c-pixel 1];
I4=[c+pixel n];
datxi=max(I1);
datxs=min(I2);
datyi=max(I3);
datys=min(I4);
Bloc=Q(datxi:1:datxs,datyi:1:datys);
MaxB=max(max(Bloc));
%donde encuentre el max nos pone el valor de 1
if (Q(r,c)==MaxB)
    S(r,c)=1;
end
end
end
end
subplot(handles.axes2),imshow(Ir),title('Deteccion de Esquinas');

hold on

for r=1:m
    for c=1:n
        if(S(r,c))

            plot(c,r,'+');

        end
    end
end
end

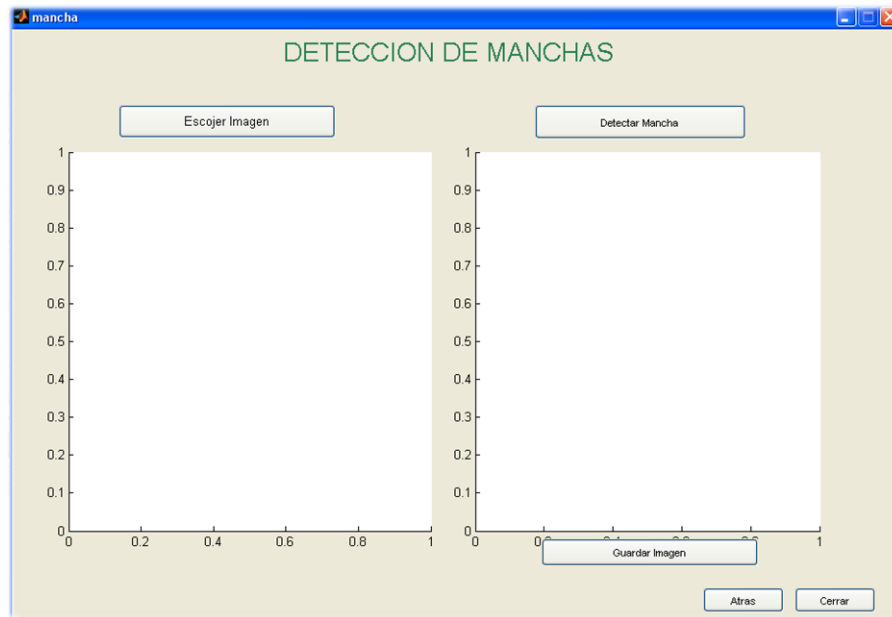
```

**Resultado:**



## ***Detectores De Manchas***

Se realizó un programa en Matlab que muestra gráficamente un detector de manchas. El entorno gráfico de dicho programa es el siguiente.



### ***Pasos del algoritmo***

- Aproximar un filtro de ruidos laplaciano por una diferencia de gaussianas. Por ejemplo, el laplaciano de ancho de dos se puede aproximar por una gaussiana de ancho de 2 menos un gaussiano de anchura  $\sqrt{2}$ . Calcular estas diferencias de gaussianas más de los anchos de 1,  $\sqrt{2}$ ,  $2 * \sqrt{2}$ , ... 32, y convolucionar cada uno con la imagen. Un ancho de 32 sería tan grande como lo que se necesitaba; que conduce a una gaussiana que se extiende por  $2 * (32) * 3$ , o 192 píxeles de ancho 3 sigma en ambos lados. Esta es una gran mancha.
- En cada píxel en la imagen, encontrar el ancho de la diferencia de gaussianas (DoG) que conducen a la mayor respuesta. Este es el máximo global en escala, y es del tamaño aproximado de una función si está presente. Para determinar si una característica de ese tamaño está centrada en el píxel en cuestión, asegúrese de que la respuesta a esa escala es un máximo local en los 26 píxeles alrededor del espacio.

## **Programación**

```
imcolor = filename;
if length(size(imcolor)) > 2
    gray = rgb2gray(imcolor);
else
    gray = imcolor;
end
im = double(gray) / 255;
max_sigma = 32;
numG = uint8(log(max_sigma) / log(sqrt(2))) + 1;
PixelDim = size(im);
Features = 0 * im;
for i=1:numG
    sigma = exp(log(sqrt(2))*double((i - 1)));
    width = round(3*sigma);
    t = -width: 1: width;
    gaussf = exp(-t.*t ./ (2*sigma^2));
    gaussf = gaussf / sum(sum(gaussf));
    Gaussians{i} = conv2(gaussf, gaussf, im, 'same');
end

for i = 2:numG
    DOG{i - 1} = Gaussians{i} - Gaussians{i - 1};
End
for i = 3: PixelDim(1) - 2
    for j = 3: PixelDim(2) - 2
        index = numG/2;
        greatest = 0;
        for k = numG/2:numG - 1
            if DOG{k}(i, j) > greatest
                index = k;
                greatest = DOG{k}(i, j);
            end
        end
        if (greatest > T)
            if (greatestSpatially(i, j, index))
                Features(i, j) = exp(log(sqrt(2))*double(index));
            end
        end
    end
end
imout = imcolor;
for i = 1: PixelDim(1)
    for j = 1: PixelDim(2)
        if (Features(i, j) > 0)
            drawSquare(i, j, round(Features(i, j)));
        end
    end
end
```

```

    end
end

fileout=imout;

function ret = greatestSpatially(ci, cj, index)
    val = DOG{index}(ci, cj);
    ret = 1;
    if val ~= max(max(DOG{index}(ci-2:ci+2, cj-2:cj+2)))
        ret = 0;
    end
end
function drawSquare(ci, cj, width)
    l = max(1, ci - width);
    r = min(PixelDim(1), ci+width);
    t = max(1, cj - width);
    b = min(PixelDim(2), cj+width);
    for leftright = l:r
        imout(leftright, t, 1) = 255;
        imout(leftright, t, 2) = 0;
        imout(leftright, t, 3) = 0;
        imout(leftright, b, 1) = 255;
        imout(leftright, b, 2) = 0;
        imout(leftright, b, 3) = 0;
    end
for updown = t:b
    imout(l, updown, 1) = 255;
    imout(l, updown, 2) = 0;
    imout(l, updown, 3) = 0;
    imout(r, updown, 1) = 255;
    imout(r, updown, 2) = 0;
    imout(r, updown, 3) = 0;
end
end
end

```

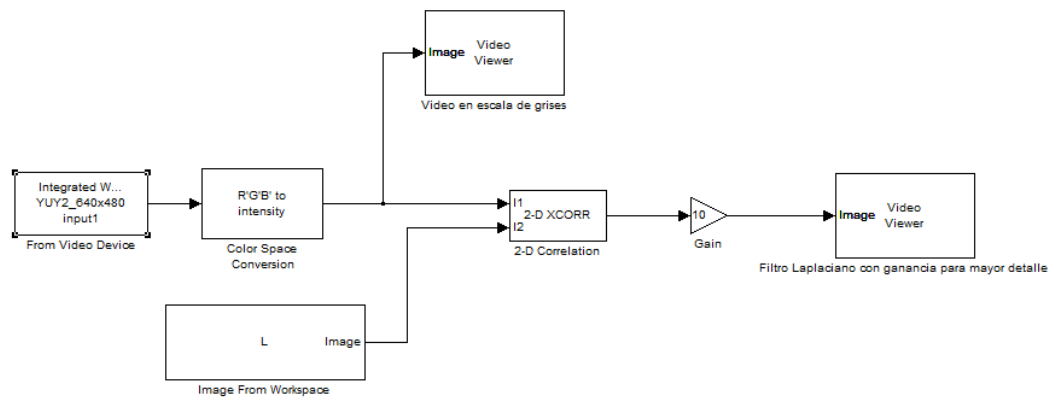
**Resultado:**



**PROCESAMIENTO DE VIDEO**

**Detección de Bordes en video (SIMULINK)**

Anteriormente se trabajó con filtrado de imágenes, pero en este caso mostraremos el proceso detección de bordes en tiempo real usando la WebCam con dos tipos de filtro: Gauss y Laplace. A continuación se muestra el esquema general del programa:

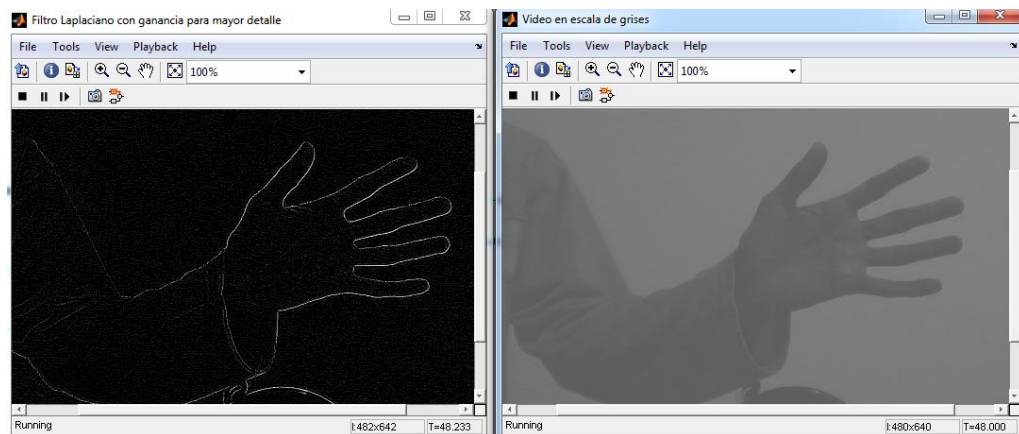


- *From video device*: Es el primer paso del programa, pues permite capturar el video de la webcam, donde se puede definir el tamaño del video, en nuestro caso fue 640 × 480 pixeles.
- *Color space conversion*: Convierte el video a escala de grises para mayor facilidad a la hora de realizar la detección de bordes posteriormente.
- *Image from workspace*: Allí se almacena la matriz laplaciana de filtrado de imagen definida en el *comand window* como:

$$L = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

- *2-D correlation*: Se correlaciona el video con la matriz de filtro laplaciano.
- *Gain*: Para mejorar la visualización del filtro se aplica un Gain, en nuestro caso con un valor de 10 aunque puede definirse con cualquier numero dependiendo de la forma como se quiera visualizar el filtro, ya que sin un gain no es fácilmente apreciable.
- *Video Viewer*: Se usaron dos de ellos, uno a la salida del *color space conversión* y otro a la salida del *gain*. el objetivo de ello es comparar el video real en escala de grises, con la aplicación del filtro laplaciano.

### Resultado:



### CORRESPONDENCIA DE PUNTOS

Esta función OPENSURF es una aplicación sugerida la cual se puede descargar de la siguiente página: <http://www.mathworks.com/matlabcentral/fileexchange/28300-opensurf-including-image-warp> . Para más programas relacionados, consultar la

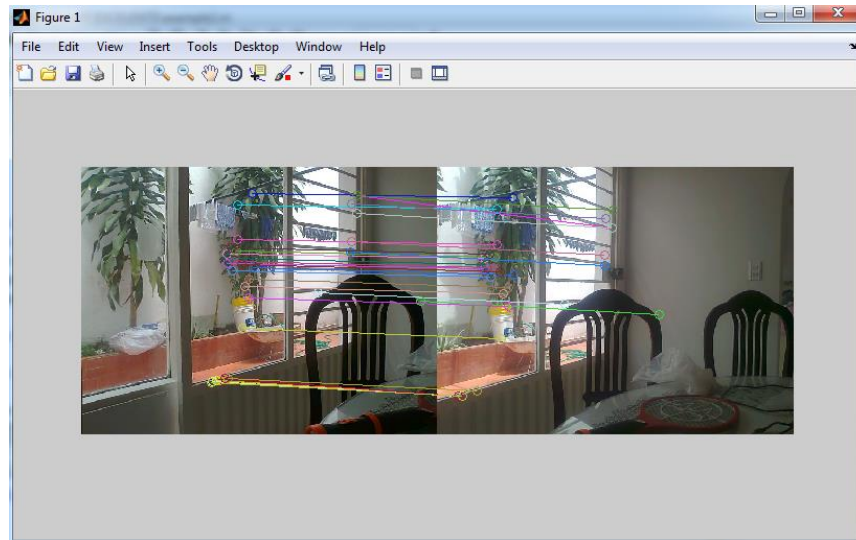
página del programador: <http://www.chrisevansdev.com/computer-vision-opensurf.html> .  
 La aplicación consiste en obtener puntos coincidentes de dos imágenes con cierta disparidad horizontal (Ejemplo 1) y también unir esas dos imágenes en una sola (Ejemplo 2). El objetivo de su uso es ejemplificar el funcionamiento de una cámara estéreo.

### Ejemplo 1.

El código y su explicación es el siguiente:

```
% Carga de imágenes
I1=imread('Foto0041.jpg');
I2=imread('Foto0042.jpg');
% Obtiene los puntos claves
Options.uptight=true;
Options.tresh=0.0001;
Ipts1=OpenSurf(I1,Options);
Ipts2=OpenSurf(I2,Options);
% Coloca los descriptores de puntos de interés en una matriz
D1 = reshape([Ipts1.descriptor],64, []);
D2 = reshape([Ipts2.descriptor],64, []);
% Encuentra las mejores coincidencias
err=zeros(1,length(Ipts1));
cor1=1:length(Ipts1);
cor2=zeros(1,length(Ipts1));
for i=1:length(Ipts1),
    distance=sum((D2-repmat(D1(:,i),[1 length(Ipts2)]))^2,1);
    [err(i),cor2(i)]=min(distance);
end
% Ordena coincidencias en un vector distancia
[err, ind]=sort(err);
cor1=cor1(ind);
cor2=cor2(ind);
% Muestra ambas imágenes
I = zeros([size(I1,1) size(I1,2)*2 size(I1,3)]);
I(:,1:size(I1,2),:)=I1; I(:,size(I1,2)+1:size(I1,2)+size(I2,2),:)=I2;
figure, imshow(I/255); hold on;
% Muestra las mejores coincidencias
for i=1:30,
    c=rand(1,3);
    plot([Ipts1(cor1(i)).x
Ipts2(cor2(i)).x+size(I1,2)], [Ipts1(cor1(i)).y Ipts2(cor2(i)).y], '-
','Color',c)
    plot([Ipts1(cor1(i)).x
Ipts2(cor2(i)).x+size(I1,2)], [Ipts1(cor1(i)).y
Ipts2(cor2(i)).y], 'o', 'Color',c)
end
```

El resultado es:



## EJEMPLO 2.

El código y su explicación es el siguiente:

```
% EJEMPLO 2: Unión de imágenes
% Carga Imágenes
I1=im2double(imread('Foto0041.jpg'));
I2=im2double(imread('Foto0042.jpg'));

% Obtiene los puntos claves
Options.uptight=true;
Options.tresh=0.0001;
Ipts1=OpenSurf(I1,Options);
Ipts2=OpenSurf(I2,Options);

% Coloca los descriptores de puntos de interés en una matriz
D1 = reshape([Ipts1.descriptor],64, []);
D2 = reshape([Ipts2.descriptor],64, []);

% Encuentra las mejores coincidencias
err=zeros(1,length(Ipts1));
cor1=1:length(Ipts1);
cor2=zeros(1,length(Ipts1));
for i=1:length(Ipts1),
    distance=sum((D2-repmat(D1(:,i),[1 length(Ipts2)]))^2,1);
    [err(i),cor2(i)]=min(distance);
end

% Ordena las coincidencias en un vector de distancia
[err, ind]=sort(err);
cor1=cor1(ind);
cor2=cor2(ind);
```

```

% Crea vectores con las coordenadas de las mejores correspondencias
Pos1=[[Ipts1(cor1).y],[Ipts1(cor1).x]];
Pos2=[[Ipts2(cor2).y],[Ipts2(cor2).x]];
Pos1=Pos1(1:30,:);
Pos2=Pos2(1:30,:);

% Muestra ambas imágenes
I = zeros([size(I1,1) size(I1,2)*2 size(I1,3)]);
I(:,1:size(I1,2),:)=I1; I(:,size(I1,2)+1:size(I1,2)+size(I2,2),:)=I2;
figure, imshow(I); hold on;

% Muestra las mejores correspondencias
plot([Pos1(:,2) Pos2(:,2)+size(I1,2)], [Pos1(:,1) Pos2(:,1)]', '-');
plot([Pos1(:,2) Pos2(:,2)+size(I1,2)], [Pos1(:,1) Pos2(:,1)]', 'o');

% Calcula matriz afín
Pos1(:,3)=1; Pos2(:,3)=1;
M=Pos1'/Pos2';

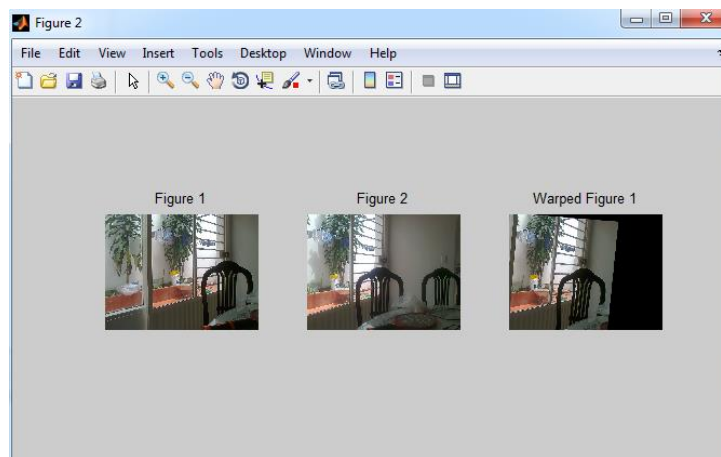
% Adiciona subfunciones a la búsqueda de ruta de Matlab
functionname='OpenSurf.m';
functiondir=which(functionname);
functiondir=functiondir(1:end-length(functionname));
addpath([functiondir '/WarpFunctions'])

% Define la imagen
I1_warped=affine_warp(I1,M,'bicubic');

% Muestra el resultado
figure,
subplot(1,3,1), imshow(I1);title('Figure 1');
subplot(1,3,2), imshow(I2);title('Figure 2');
subplot(1,3,3), imshow(I1_warped);title('Warped Figure 1');

```

El resultado es:



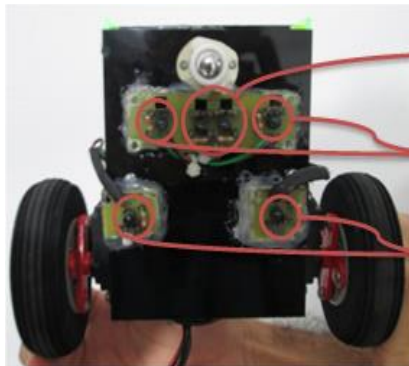
## **Segunda fase de construcción del robot de pruebas**

### **SENSORES USADOS EN EL ROBOT**

Se usaron una total de 6 sensores **Sharp is471f digitales** para la detección de línea, montados en una baquelita provista de un emisor y una pequeña resistencia que nos define el alcance del sensor.

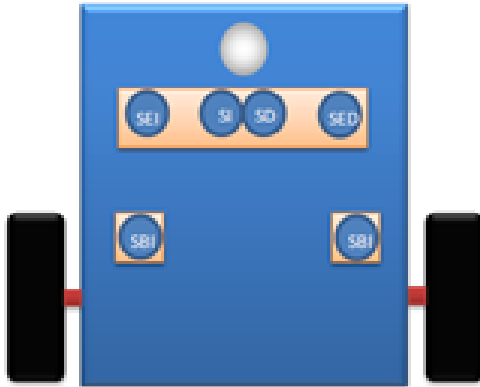


Los sensores Sharp digitales solo detectan blanco y negro o HIGH y LOW y se dispusieron en el robot de la siguiente manera:



- Los dos sensores frontales del centro, son usados para seguimiento de línea.
- Los dos sensores frontales de los extremos, sirven para detectar los cruces de 90°.
- Mientras los dos sensores inferiores se usan para llevar el centroide del robot hasta el cruce y permitir que el robot gire sin perder la línea.

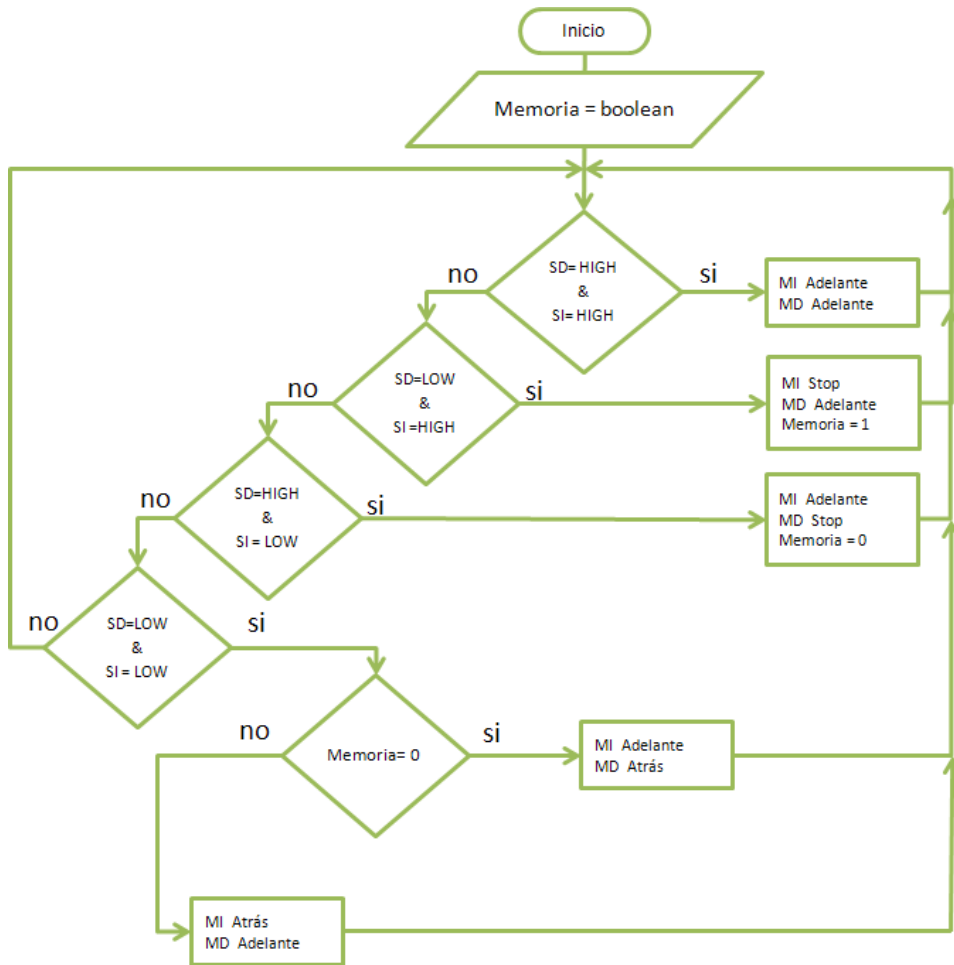
Estos sensores están conectados en los pines 14 a 18 (pines analógicos) y 4 (pin digital).



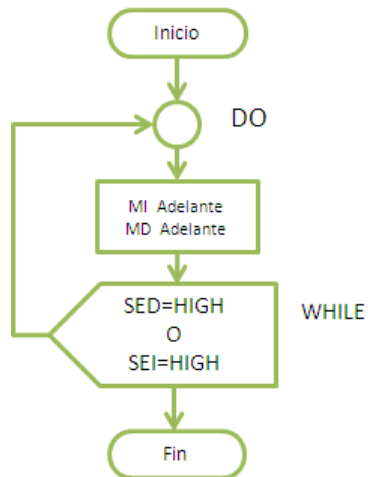
- Sensor Izq(SI) = Pin 14
- Sensor ExtDer(SE D) = Pin 15
- Sensor Der(SD) = Pin 16
- Sensor Extlzq(SE I) = Pin 17
- Sensor Backlzq(SBI) = Pin 18
- Sensor BackDer(SBD) = pin 4

Con estos sensores podemos hacer las siguientes rutinas (seguidor de línea, cruces a 90° e ignorar cruces si es necesario) y muchas otras.

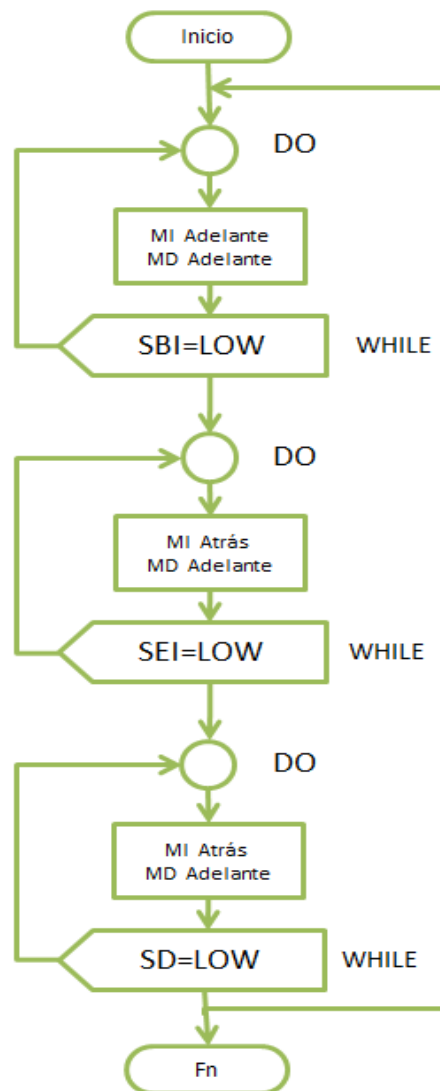
## Seguidor de línea



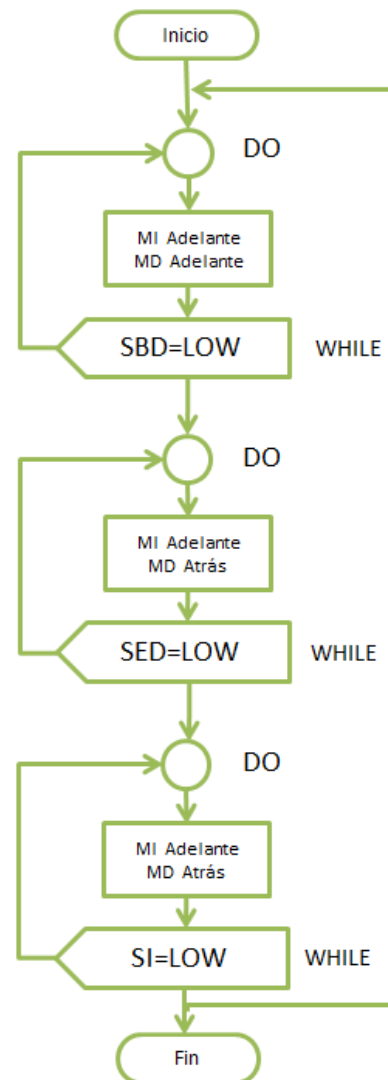
## Ignorar Cruce



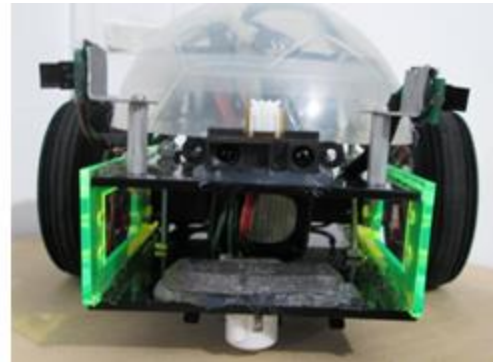
### Cruce 90° a la Izquierda



### Cruce 90° a la Derecha



Se usó un **Sensor infrarrojo GP2D120XJ00F análogo** para medir distancias de 4 a 30cm, usando el principio de triangulación el cual mide la distancia por medio de la medida del ángulo de luz que incide en el fotodetector por lo tanto el tipo de objeto, temperatura ambiente, y otros factores no alteran la medición de la distancia.



El sensor se conectó en el pin de entrada analógico 19, resolución de 10 bits, (1024 posiciones 0-1023).

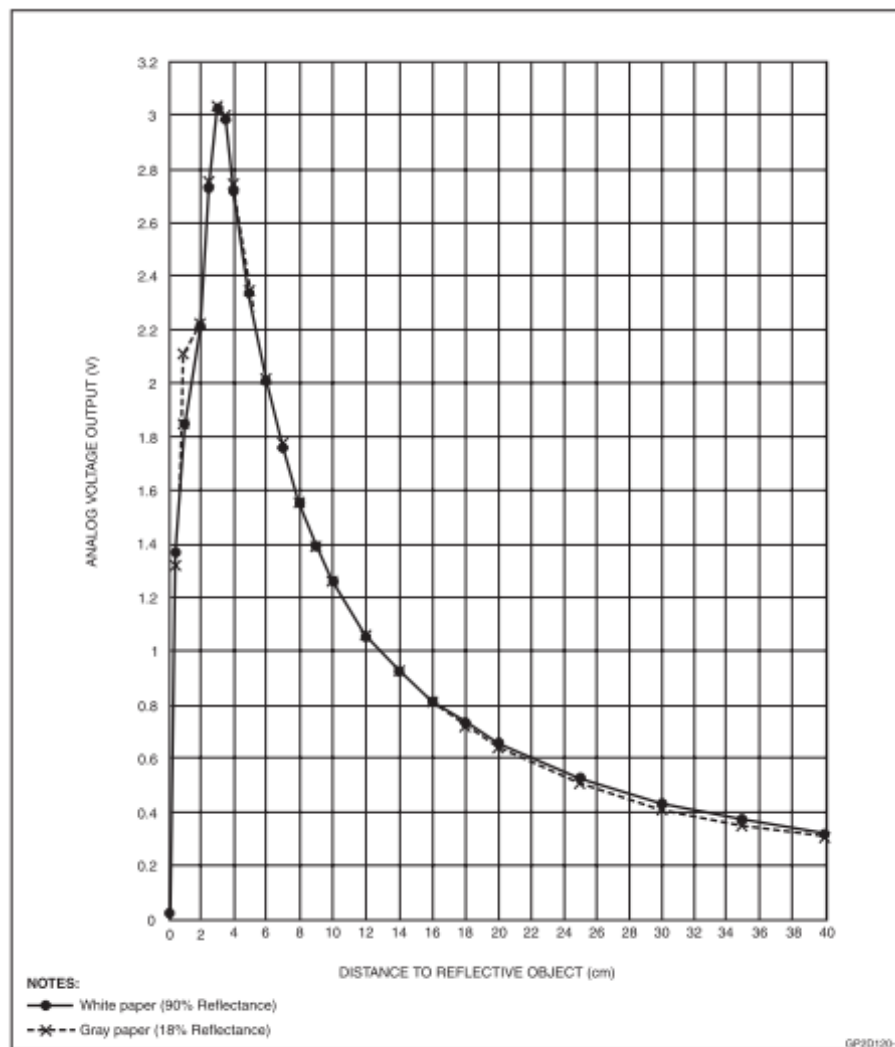


Figure 4. GP2D120 Example of Output Distance Characteristics

Al intentar caracterizar el sensor se pudo notar que el sensor no maneja 5v que es el rango completo del pin, sino solo 3,2 v. por lo tanto su número de posiciones o rango es menor, con una simple regla de tres podemos determinarlo.

$$x = \frac{3,2 * 1024}{5} = 655,36$$

Si aproximamos tenemos 655 posiciones para el sensor.

### ***Control de velocidad por aproximación a obstáculos.***

El robot comienza a reducir la velocidad gradualmente a medida que se acerca a un objeto y se detiene a unos 4cm de él.

Para lograr esto tenemos que definir una variable de distancia que nos permita controlar directamente la señal que se envía a los servos con la señal que lee el sensor Analógico.

Lo primero que se debe hacer es tratar de igualar los rangos de la señal leída por el sensor (655 posiciones) y las enviadas a los servos (256 posiciones).

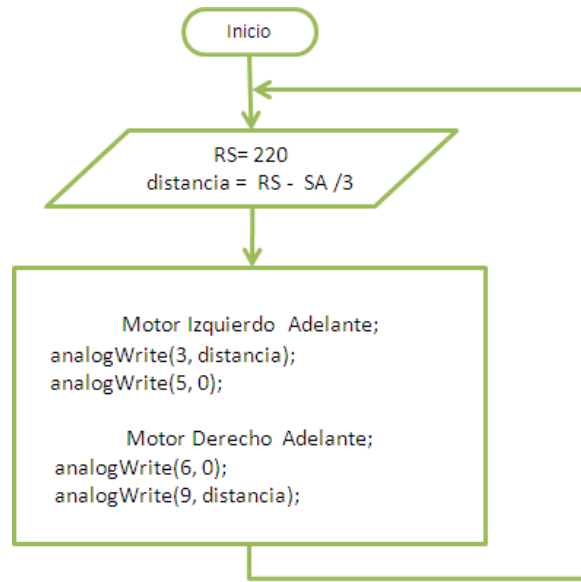
No se usa el rango completo del PWM para evitar que el servo se mantenga trabajando a máximo voltaje cuando no se detecte un objeto.

El Rango que manejaremos en los servos = 220

$$\frac{\text{Señal Entrada SA}}{C} = \text{Señal Salida M} ; \quad C = \frac{655}{220} = 2,97 ; \quad C \cong 3$$

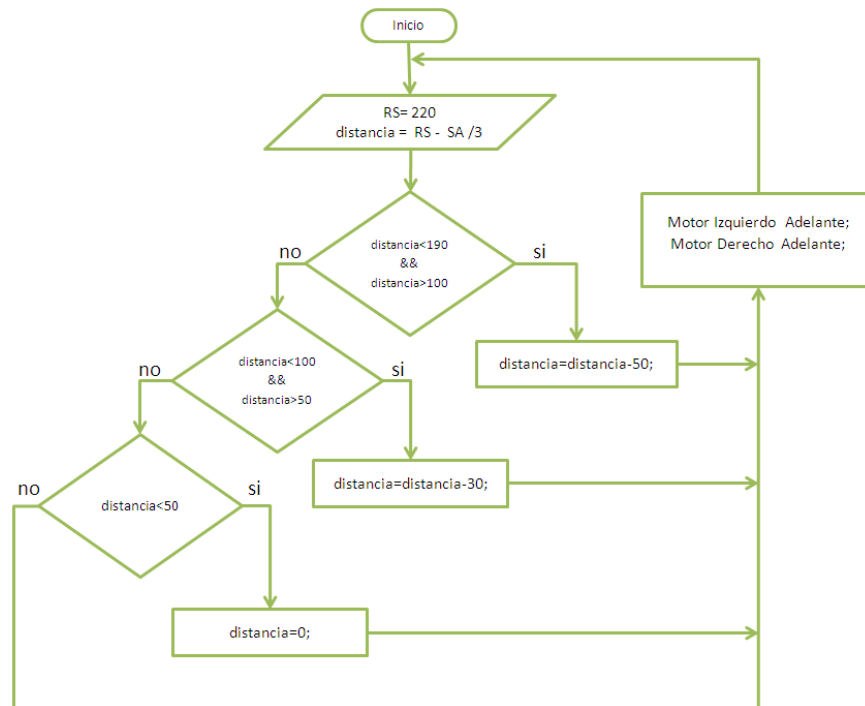
Luego debemos invertir la señal del sensor ya que esta incrementa a medida que el robot se acerca al objeto y la señal enviada a los servos debe disminuir.

$$\text{Distancia} = \text{Rango servo} - \frac{\text{Señal Entrada SA}}{C}$$

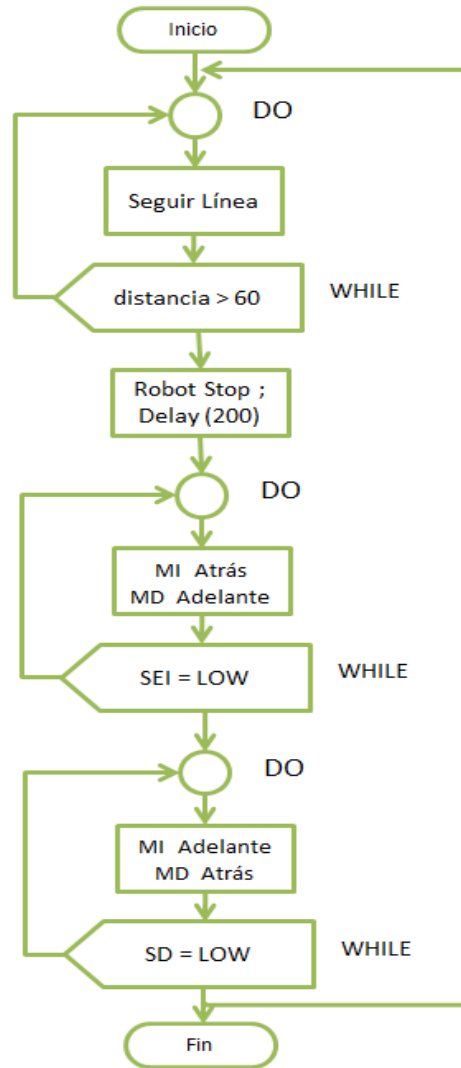


Si se deja el control de forma directa , no se puede percibir claramente el control de velocidad debido a que el sensor no tiene un comportamiento lineal, por esto se definieron los siguientes rangos para mejorar el control.

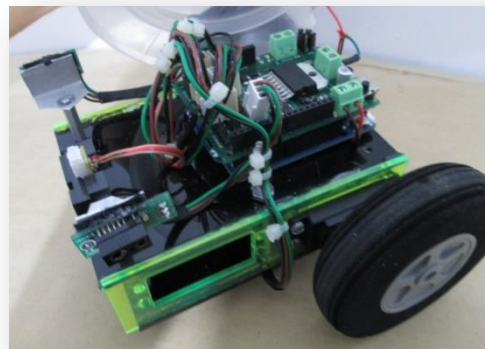
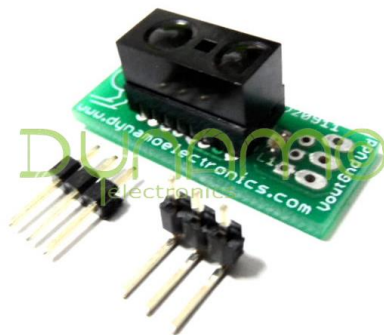
### **Adelante**



### Giro 180° o Turn Back.



También se usaron 2 **Sensores infrarrojos digitales GP2Y0D810Z0F** para medir distancia de 10 y 5 cm montados en boards.

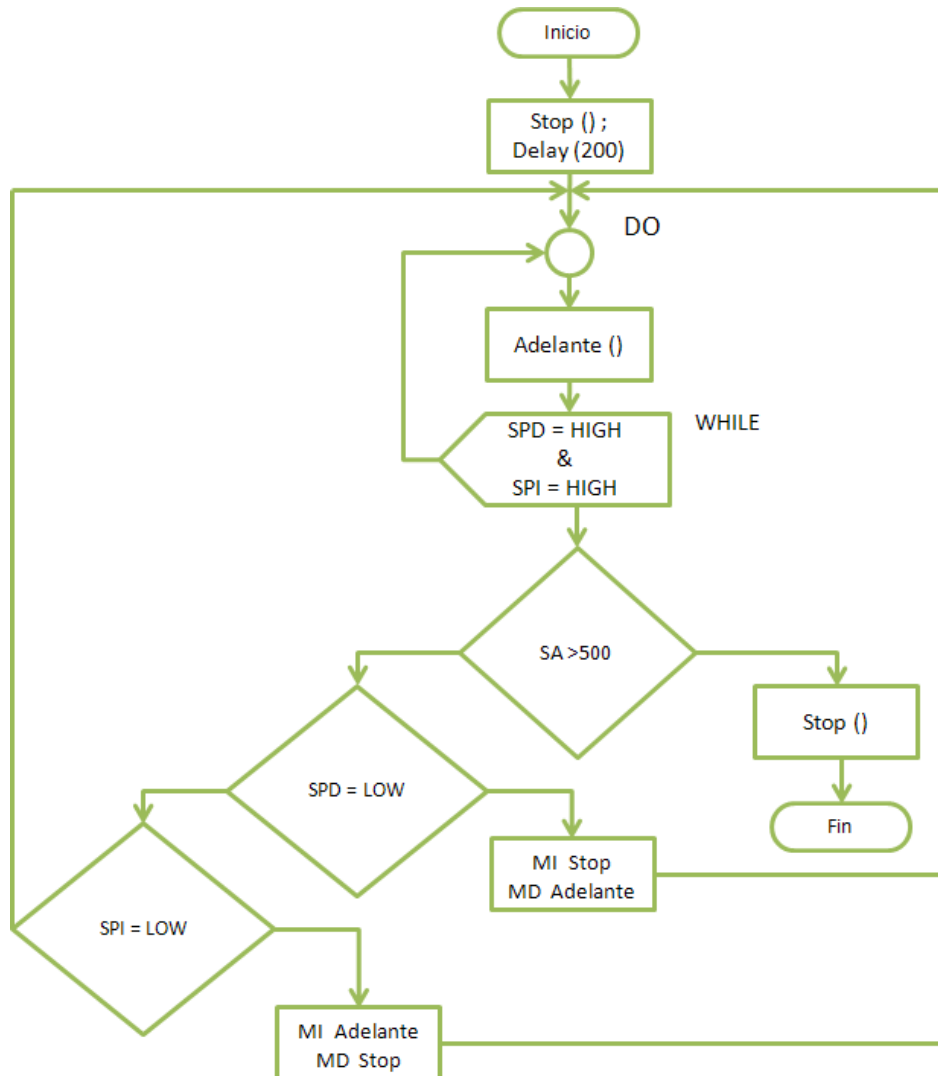


Montados en la parte superior a derecha e izquierda del robot para la detección de pared, conectados en pines digitales.

- sensorParedIzq = Pin 10;
- sensorParedDer = Pin 8;

Estos sensores se usan para seguir pared y la rutina que se realiza es la siguiente:

### Seguir Pared



## **TALLER DE LOCALIZACIÓN**

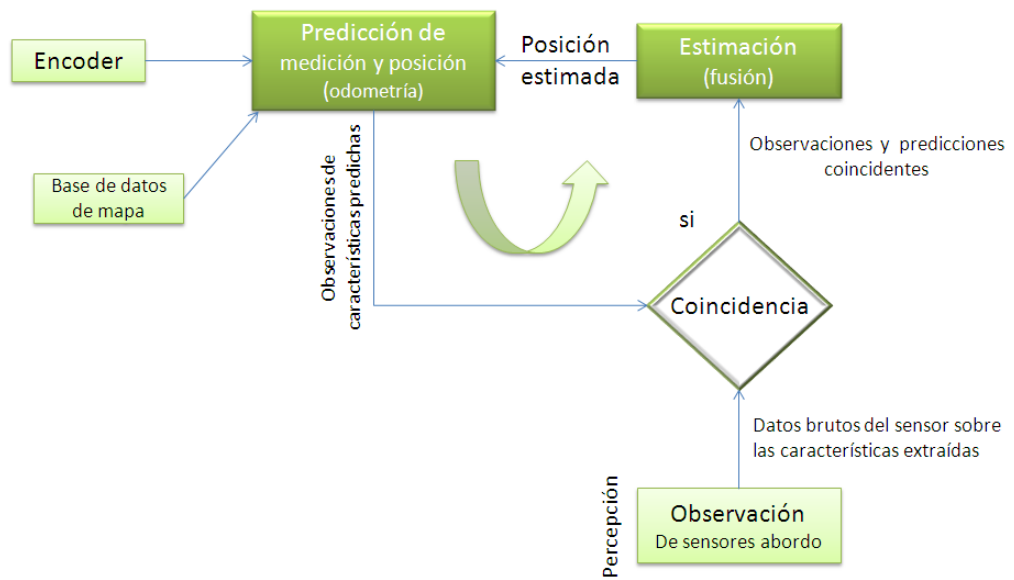
Para este capítulo en particular se implementó un programa que nos muestra gráficamente el funcionamiento del EKF SLAM, Dicho programa se puede encontrar en [http://www.mathworks.com/matlabcentral/fileexchange/39992-ekf-slam-example/content/Slam\\_Tested\\_4Beacons/EKFSLAM.m](http://www.mathworks.com/matlabcentral/fileexchange/39992-ekf-slam-example/content/Slam_Tested_4Beacons/EKFSLAM.m).

La localización y mapeo simultáneos (SLAM) Es una técnica usada por robots y vehículos autónomos para construir un mapa en entornos desconocidos y a la vez estimar su trayectoria mientras se desplaza en dicho entorno.

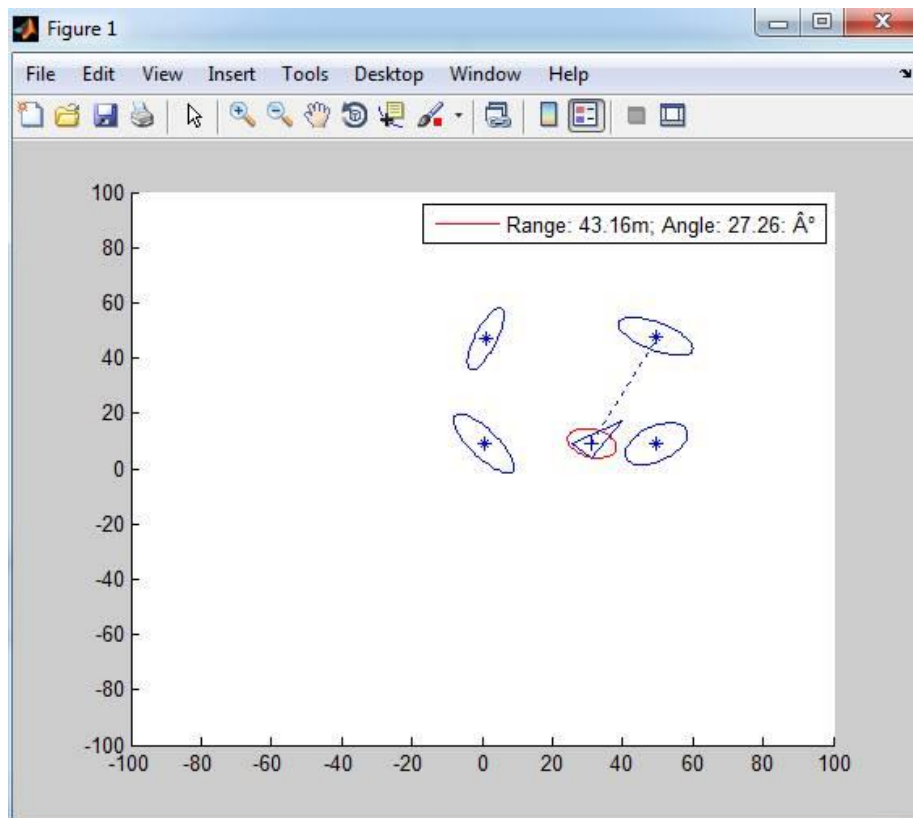
Las soluciones que mejores resultados han obtenido a la hora de abordar el problema del SLAM son aquellas basadas en técnicas probabilísticas, tales como la localización de Markov, filtro de Kalman, filtro de Kalman extendido, entre otros

En el EKF SLAM se utiliza el filtro de Kalman Extendido, el cual consta de múltiples etapas que se resumen a continuación.

1. En la etapa de la observación, el robot recoge datos reales de sensores y características apropiadas extraídas (por ejemplo, líneas, puertas, o incluso el valor de un sensor específico).
2. Al mismo tiempo, sobre la base de su posición predicha en el mapa, el robot genera una predicción de medición que consiste en las características que el robot espera observar a partir de la posición en la que se piensa que está (por ejemplo, la posición estimada en el paso de predicción).
3. En la etapa de coincidencia el robot calcula el mejor ajuste entre las características extraídas durante la observación y las características esperadas seleccionadas durante la predicción de medición.
4. Finalmente, el filtro de Kalman fusiona la información proporcionada por todas estas coincidencias para actualizar el estado de creencia robot en la etapa de estimación.



## EKF SLAM



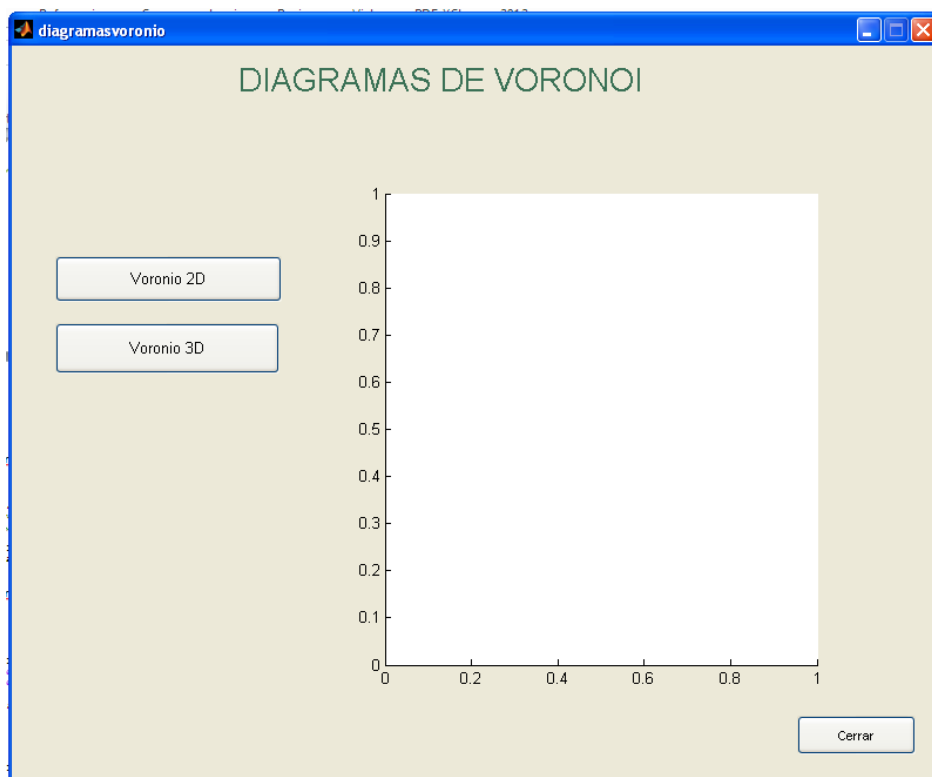
# TALLER DE NAVEGACIÓN

## Diagrama de Voronoi

El diagrama de Voronoi nos muestra las posibles rutas a seguir al trazar líneas construidas a partir de todos los puntos que son equidistantes de dos o más obstáculos.

Matlab posee un comando que realiza esta tarea y lo mostraremos a continuación.

### Entorno grafico



### Programación

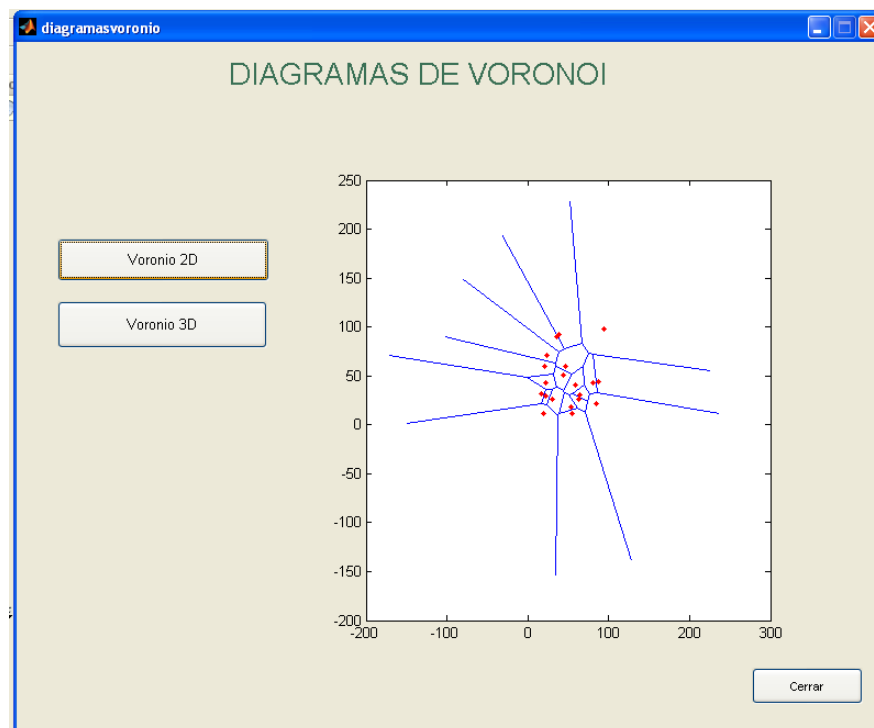
#### Diagrama de Voronoi 2D

```
%generar un campo aleatorio con 20 puntos u obstáculos con el comando  
rand  
Map = 100*rand(20,2)  
%trazar líneas equidistantes a cada punto mediante el comando voronoi  
[Vx,Vy] = voronoi (Map (:,1),Map (:,2));  
%graficar  
axis (handles.axes1);
```

```
plot (Vx,Vy, 'b-',Map (:,1),Map (:,2), 'r.');
```

## Diagrama de Voronoi 3D

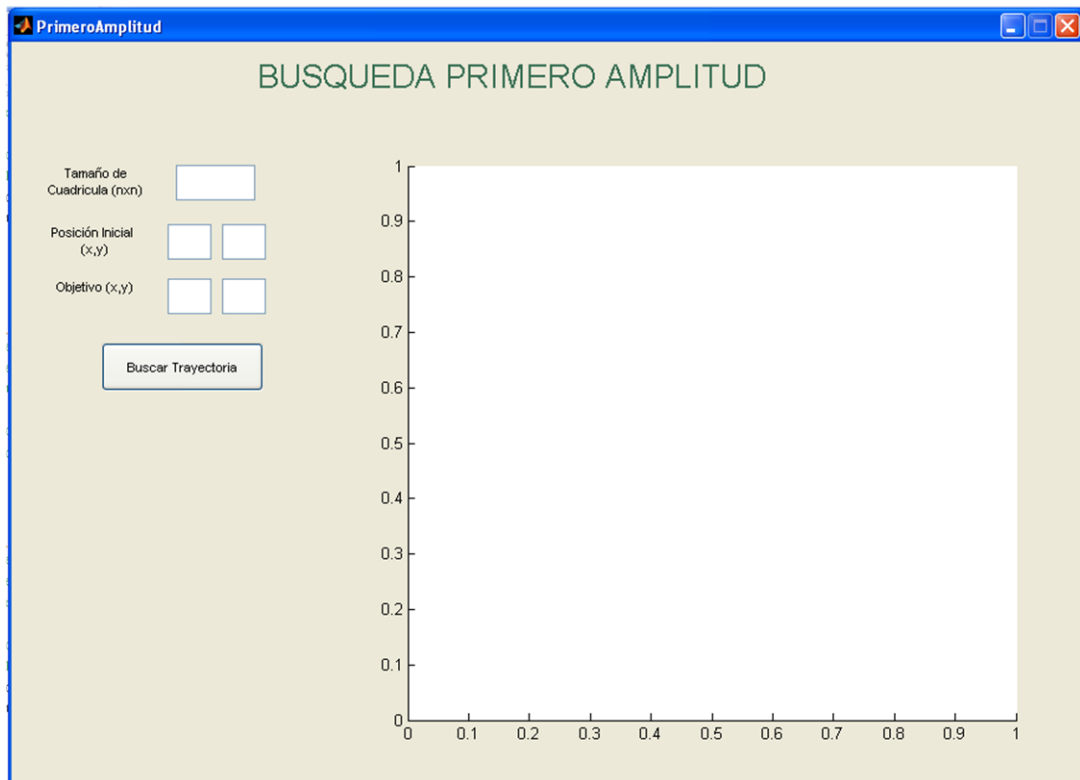
```
k=40;
x=rand(k,1);
y=rand(k,1);
z=rand(k,1);
axes(handles.axes1);
plot3(x,y,z, 'Marker', '.', 'MarkerEdgeColor', 'r', 'MarkerSize', 10,
'LineStyle', 'none')
X=[x y z];
[V,C]=voronoin(X);
V;
for k=1:length(C)
    disp(C{k})
end
for k=1:length(C)
    if all(C{k}~=1)
        VertCell = V(C{k},:);
        KVert = convhulln(VertCell);
    end
end
patch('Vertices',VertCell,'Faces',KVert,'FaceColor','b','FaceAlpha',0.5)
end
end
```



## Búsqueda Primero en Amplitud

Este algoritmo de búsqueda gráfica comienza con el nodo de inicio y explora la totalidad de sus nodos vecinos. Entonces, para cada uno de estos nodos, este explora todos sus vecinos inexplorados y así sucesivamente. Los nodos se expanden en orden de proximidad al nodo de inicio con la proximidad definida como el menor número de transiciones de borde. El algoritmo procede hasta que llega al nodo objetivo.

### Entorno grafico



### Programación

```
%LEER VALORES DE ENTRADA
n=str2double(get(handles.edit1,'String'));
x0=str2double(get(handles.edit2,'String'));
y0=str2double(get(handles.edit3,'String'));
xf=str2double(get(handles.edit4,'String'));
```

```

yf=str2double (get (handles.edit5, 'String'));

%%%   SE DEFINE LA POSICIÓN DE LOS OBSTÁCULOS   %%%

obstacles (1, :)= [1, 4];
obstacles (2, :)= [2, 2];
obstacles (3, :)= [3, 1];
obstacles (4, :)= [4, 3];
obstacles (5, :)= [4, 4];

%%%   EL USUARIO DEFINE LA POSICIÓN DE PARTIDA Y EL OBJETIVO   %%%

startingPosition=[x0,y0];
goal=[xf,yf];

% Se definen los colores que se mostrarán en los resultados:
obstacleColor=[1,0,0]; % Rojo
nodeColor=[0,1,0]; % Verde
expandColor=[0,0,0]; % Negro
goalColor=[0,0,1]; % Azul
pathColor=[0,1,1]; % Celeste

% Grafica la cuadrícula y los obstáculos:
scatter (obstacles (:,1), obstacles (:,2), 100, obstacleColor, 'filled');
grid ;
AXIS ([0 n 0 n]);
hold on;

% Grafica el objetivo:
scatter (goal (1,1), goal (1,2), 100, goalColor, 'filled');

% Inicializa variables:
fringeCount=1; % Se utiliza para recorrer el conjunto marginal (no
pierde de vista el nodo actual en el conjunto de franja)
tempCount=1; % Usado para expandir el conjunto marginal (No pierde de
vista el extremo de dicho conjunto)

% Inicializa el set de franjas cuya estructura es:

fringe (fringeCount, :)= [startingPosition, fringeCount];

% Este bucle se ejecuta hasta que se encuentra el objetivo:
while (~ ((fringe (fringeCount, 1)==goal (1,1)) &
(fringe (fringeCount, 2)==goal (1,2))))

    % Grafica el nodo actual:

scatter (fringe (fringeCount, 1), fringe (fringeCount, 2), 100, nodeColor, 'filled
');
```

```

    % Expande el set de franjas a la izquierda, derecha, encima y fondo
del nodo actual:
    for x=-1:1
        for y=-1:1
            % Esta es una prueba simple para asegurar que el conjunto de
            % bordes no se expanda en diagonal pues el robot no puede
            % moverse en esa dirección:

            if (x*y==0)

                % 'failsTest' es usado para determinar cuando un nodo no
                % puede expandirse porque está fuera de la cuadrícula, en
                % un obstáculo, o ya ha sido expandido:
                failsTest=0;

                %'tempNode' es el nodo actual que está tratando de
                %expandirse:

tempNode=[fringe (fringeCount,1)+x, fringe (fringeCount,2)+y, fringeCount];

                % Prueba para ver si el nodo está fuera de la cuadrícula:
                if ( (tempNode(1,1)<0) | (tempNode(1,2)<0) ) | (
(tempNode(1,1)>n) | (tempNode(1,2)>n) )
                    failsTest=failsTest+1;
                end

                % Si no falló el primer test, test para ver si el nodo
está
                % ya en un set de franjas o bordes:
                if (failsTest<1)
                    for i=1:size (fringe,1)
                        if (tempNode(1,1)==fringe(i,1)) &
(tempNode(1,2)==fringe(i,2))
                            failsTest=failsTest+1;
                        end
                    end
                end

                % Si no fallaron los anteriores test, prueba para ver si
el
                % nodo es un obstáculo:
                if (failsTest<1)
                    for i=1:size (obstacles,1)
                        if
(tempNode(1,1)==obstacles(i,1)) & (tempNode(1,2)==obstacles(i,2))
                            failsTest=failsTest+1;
                        end
                    end
                end

                % Si ninguna prueba falló, agregar al final del set de

```

```
        % bordes. En la búsqueda de primero amplitud, los nodos
son removidos desde el fin del set de franjas, entonces se agregan nuevos
nodos al final:
```

```
        if (failsTest<1)
            fringe(fringeCount+tempCount,:)=tempNode;
scatter(tempNode(1,1),tempNode(1,2),100,expandColor,'filled');
        tempCount=tempCount+1;
        end
    end
end
end
```

```
    % Incrementa el siguiente nodo:
    fringeCount=fringeCount+1;
    tempCount=tempCount-1;
    pause(.1);
end
```

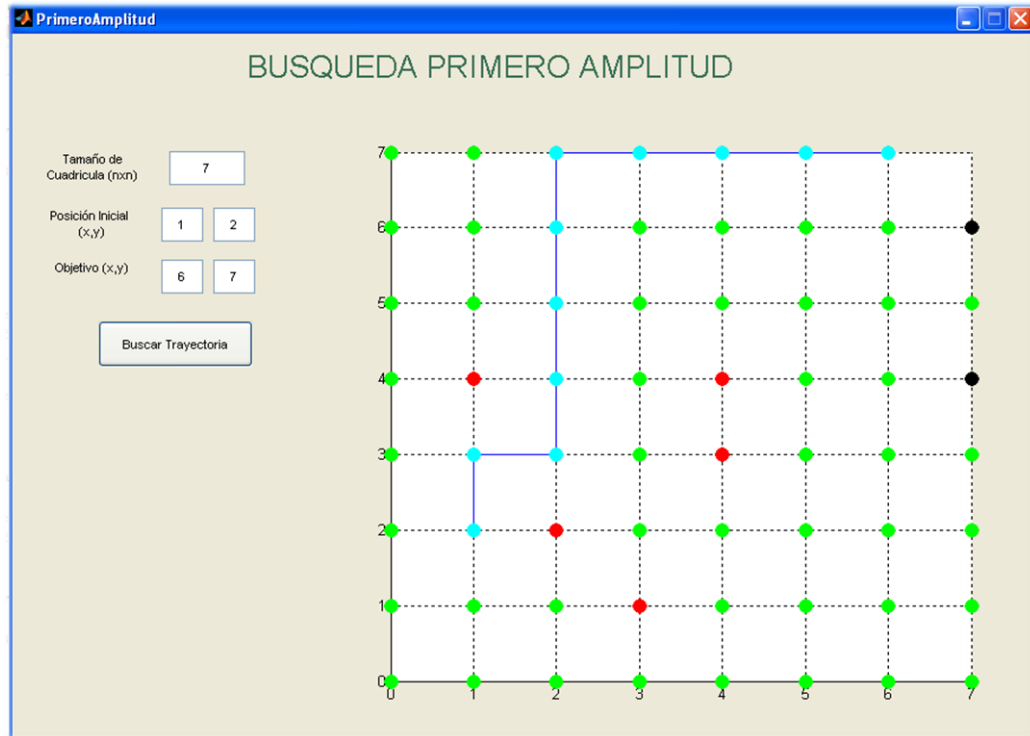
```
% Inicia un contador:
i=1;
```

```
% Rastrea a través de los nodos principales para recuperar el camino:
while ~(fringeCount==1)
    path(i,:)=fringe(fringeCount,1),fringe(fringeCount,2)];
    fringeCount=fringe(fringeCount,3);
    i=i+1;
end
```

```
% Agrega la posición inicial a la ruta:
path(i,:)=startingPosition;
```

```
% Grafica la ruta:
axis(handles.axes1);
plot(path(:,1),path(:,2));
scatter(path(:,1),path(:,2),100,pathColor,'filled');
```

## Resultado:



## Algoritmo De Bug

La idea básica del algoritmo de bug es seguir el contorno de cada obstáculo en el camino del robot y por lo tanto circunnavegarlo.

Para poder mostrar su funcionamiento usaremos el siguiente código que se puede encontrar en [http://vincentjoy.com/a4\\_html/a4.html](http://vincentjoy.com/a4_html/a4.html), dicho código fue desarrollado en base a:

### Paso 1

Se crean algunos obstáculos de formas simples y se determina si un punto dado choca con los obstáculos.

## Paso 2

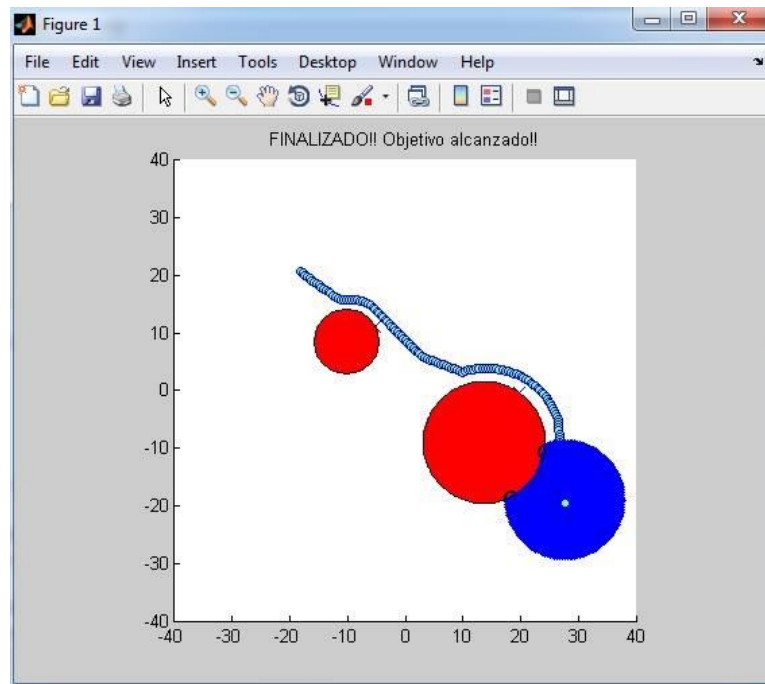
No hay pre-conocimiento de los obstáculos por el robot. El programa tiene la información de los obstáculos y determina si los rayos ultrasónicos han alcanzado los obstáculos y devuelve la distancia.

## Paso 3

El algoritmo implementado en el código de Matlab se muestra a continuación

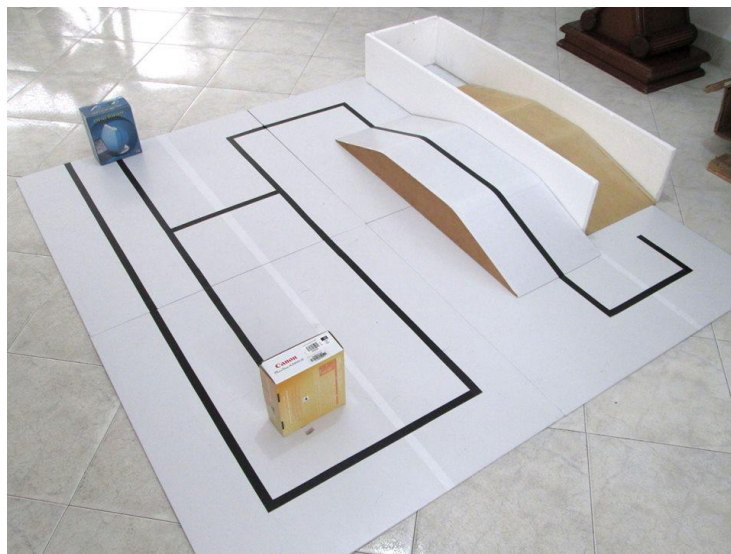
1. Mientras sea Verdadero hacer
2. Moverse al Objetivo
3. Repetición
4. Moverse continuamente hacia el punto  $n_T$ ,  $O_i$  (T es la línea que conecta el robot y el objetivo, si no se han detectado obstáculos)
5. Hasta
6. El objetivo es encontrado o
7. La dirección que reduzca al mínimo  $d(x, n) + d(n, q_{goal})$  comienza a aumentar  $h(\text{robot}, q_{goal})$  es decir, "mínimo local". Cambia a modo de seguimiento de frontera.
8. Seguimiento de frontera
9. Elegimos una dirección de seguimiento de frontera que continúe en la misma dirección que la dirección del movimiento al objetivo más reciente.
10. repetir
11. Actualización  $d_{reach}$  (distancia más corta entre la meta y la frontera detectada en este paso),  $d_{followed}$  (la distancia más corta entre el obstáculo que bloquea y el objetivo o la distancia desde el propio robot a la meta, si no hay obstáculo que bloquea alrededor), y la frontera  $O_i$ .
12. detectada.
13. se mueve en la "frontera tangente" del obstáculo en la dirección de frontera elegida.
14. hasta
15. El objetivo se alcanza o
16. El robot completa un ciclo alrededor del obstáculo en cuyo caso el objetivo no puede ser alcanzado o
17. Ningún obstáculo bloquea y  $d_{reach}$  y  $d_{followed}$  en este caso, cambia al movimiento a modo objetivo
18. Fin Mientras.

## Resultado:



### ***Tercera fase de construcción del Robot de pruebas***

Se construyó un laberinto que nos permite mostrar el funcionamiento de cada una de las rutinas que se implementaron en el robot, generando una ruta predefinida.



En el siguiente diagrama se muestran las rutinas que sigue el robot para recorrer el laberinto en su totalidad.

