

**DESARROLLO DE ESTRATEGIAS PARA EL CONTROL DE MÓVILES
AUTÓNOMOS BASADO EN REDES NEURONALES**

JULIO CÉSAR ROJAS GUALDRÓN

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA ELECTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
BUCARAMANGA
2007**

**DESARROLLO DE ESTRATEGIAS PARA EL CONTROL DE MÓVILES
AUTÓNOMOS BASADO EN REDES NEURONALES**

JULIO CÉSAR ROJAS GUALDRÓN

**Tesis de Grado presentada como requisito para optar al título de
Ingeniero Electrónico**

Director

M.S.c. JORGE HERNANDO RAMÓN

Codirector

Ph.D. OSCAR EDUARDO GUALDRÓN GUERRERO

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
BUCARAMANGA
2007**

DEDICATORIA

Quiero dedicar todo mi esfuerzo durante estos últimos años a mis padres y a mis hermanos porque todos los días estoy mas convencido de que tengo la mejor familia que cualquier ser humano pudiera desear y porque me enseñaron que no hay cosas imposibles solo hombres incapaces

A mi novia por su apoyo, compañía y confianza, por que sin importar los momentos difíciles siempre me recordó el nunca poder desfallecer

A los verdaderos amigos que me han ayudado día a día a entender lo importante que es siempre exigirse por ser mejor en todos los aspectos de nuestra existencia

Al grupo ERA y a la empresa DINAMO por su apoyo, ejemplo y guía sobre todo momento no solo para el desarrollo de este proyecto sino también para mi crecimiento como profesional.

A Dios por hacerme quien soy y por darme la fuerza para nunca, nunca darme por vencido

JULIO CESAR

CONTENIDO

	Pág.
INTRODUCCIÓN	1
1. DESCRIPCIÓN DEL PROYECTO	2
1.1. TÍTULO DEL PROYECTO	2
1.2. OBJETIVO GENERAL	2
1.3. OBJETIVOS ESPECÍFICOS	2
1.3.1. Objetivos en el campo de acción de Ingeniería Electrónica.	2
1.4. JUSTIFICACIÓN	3
2. COMPRENSIÓN DEL PROBLEMA	5
2.1. METODOLOGÍA	5
2.1.2. Metodología para el desarrollo del proyecto	5
2.2. MARCO TEÓRICO	7
2.2.1. Hacia La Construcción De Maquinas Inteligentes	7
2.2.2. Robótica Móvil	8
2.2.3. Redes Neuronales Artificiales	8
2.3. PRECISIÓN DEL PROBLEMA	10
2.3.1. Requerimientos generales	10
2.4. ANÁLISIS DE SOLUCIONES EXISTENTES	12
3. PROPUESTA DE NAVEGACION	15
3.1. INTRODUCCION A LA METODOLOGIA DE NAVEGACION	15
3.2. SELECCIÓN DE PARAMETROS	17
3.3. PLANTEAMIENTO DE METODOLOGIA DE NAVEGACION	20
3.4. COMPARACION CON OTRAS ALTERNATIVAS	23
3.5. SELECCIÓN DE METODOLOGIA DE NAVEGACION	25
4. DESARROLLO DE ALGORITMO	26
4.1. SELECCIÓN DEL TIPO DE RED	26
4.2. PROGRAMACION DEL ALGORITMO DE CONTROL	27
5. EXPERIMENTACION	29
5.1. SIMULACIÓN	29
5.2. COMPROBACION TECNICA	30
CONCLUSIONES Y RECOMENDACIONES	32
BIBLIOGRAFIA	34
ANEXOS	37

LISTA DE TABLAS

	Pág.
Tabla 1. Características generales de los proyectos seleccionados	13
Tabla 2. Acciones de control de desplazamiento del móvil.	23

LISTA DE FIGURAS

	Pág.
Figura 1. Sistema de coordenadas global, y trayectoria de desplazamiento	16
Figura 2. Ubicación de los sensores en el móvil.	21
Figura 3. Ángulos permitidos en la orientación del carro	21
Figura 4. Estados posibles de detección percibidos por los sensores	22
Figura 5. Posibles movimientos del carro programados en la red PNN.	23
Figura 6. Topología de la red neuronal para el trazado de trayectorias	27

LISTA DE ANEXOS

	Pág.
ANEXO A. ANÁLISIS DE SOLUCIONES EXISTENTES	38
ANEXO B. MÉTODOS CLÁSICOS DE PLANIFICACIÓN	52
ANEXO C. ALGORITMOS SELECCIONADOS	64
ANEXO D. VALORACIÓN DE CRITERIOS	79
ANEXO E. RESUMEN RED PNN	84
ANEXO F. PROGRAMA	87
ANEXO G. COMPROBACIÓN TÉCNICA	90

GLOSARIO

ACTUADORES: dispositivos que transforman una señal de entrada (eléctrica) en movimiento.

ALGORITMO: conjunto finito de instrucciones o pasos que sirven para ejecutar una tarea o resolver un problema.

CONFIGURACION: Disposición de las partes y programas que componen un móvil que constituyen sus propiedades.

ENTORNO: espacio determinado con la posibilidad de tener o no obstáculos, en donde el móvil cumple su acción de desplazamiento.

HARDWARE: termino que se utiliza generalmente para describir el soporte físico de una tecnología.

NAVEGACION: desplazamiento sobre un entorno determinado.

LOCOMOCION: sistema del movil encargado del movimiento y al configuracion de los actuadores .

PARADIGMA: modelo o patrón particular que define un tema específico en cualquier disciplina.

PNN: configuracion de red emparentada con la red de base radial (probabilistic neural network).

PROCESADOR: componente electrónico que interpreta las [instrucciones](#) y procesa los [datos](#) contenidos en los programas.

RAM: memoria de acceso aleatorio (random acces memory)

REDES NEURONALES: tecnica de procesamiento de informacion inspirado en la forma en que funciona el sistema nervioso, que se desarrolla para producir una salida.

ROBOTICA MOVIL: área de la [robótica](#) que desarrolla [robots](#) capac desplazarse; para ello el robot debe *percibir* su entorno y *actuar* de la adecuada para llevar a cabo su tarea.

ROM: memoria de solo lectura (read only memory).

SENSOR: dispositivo que detecta algunas cantidades externas. Los sensores convierten una cantidad física en otra, estos producen señales eléctricas relacionadas a las cantidades detectadas.

SOFTWARE: conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica.

RESUMEN

TÍTULO: DESARROLLO DE ESTRATEGIAS PARA EL CONTROL DE MÓVILES AUTÓNOMOS BASADO EN REDES NEURONALES.*

AUTOR: ROJAS GUALDRÓN, Julio César**

PALABRAS CLAVE:

Robótica móvil, Redes neuronales, Sensores, Planificación de trayectorias, Actuadores.

DESCRIPCIÓN:

Este proyecto tiene como objetivo desarrollar un algoritmo de control basado en redes neuronales para la navegación de un robot móvil que se desplace de un punto a otro, con obstáculos fijos.

Para la consecución de este objetivo se propuso partir de un estudio previo de las metodologías clásicas de planificación de trayectorias, simultaneo con un estudio de algunas propuestas de aplicación de redes neuronales en el campo de la robótica móvil

Según la información previa se propone una estrategia de navegación asociada a la mejor arquitectura de red neuronal escogida, (en este caso la red PNN), para luego realizar un proceso de comparación según los parámetros seleccionados tomando en cuenta los requerimientos para su montaje en un sistema embebido

Por ultimo se realiza una comprobación técnica de la metodología propuesta utilizando un simulador desarrollado para este objetivo.

* Trabajo de Grado

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería Electrónica. Director del Proyecto: M.S.c. Jorge Hernando Ramón

ABSTRACT

TITLE: DEVELOPMENT OF STRATEGIES FOR THE CONTROL OF INDEPENDENT MOBILE ROBOT BASED ON NEURONAL NETWORKS*

AUTHOR: ROJAS GUALDRÓN, Julio César**

KEYWORDS:

Mobile robots, Neural Networks, Sensors, Planning of Trajectories, Actuator

DESCRIPTION:

This project has the objective to develop an algorithm of control based on neuronal networks for the navigation of a mobile robot that move from one place to another one, with fixed obstacles.

For the attainment of this objective one seted out to start off of a previous study of the classic methodologies of planning of trajectories, it is synchronized with a study of some proposals of application of neuronal networks in the field of the movable robotics.

According to the previous information a strategy of navigation associated to the best selected neuronal network architecture sets out, (in this case PNN network), then make a comparison process according to the selected parameters taking in account the requirements for its assembly in a contracted system.

Finally a technical verification of the propose methodology is made using a simulator developed for this objective.

* Thesis

** Faculty of Physical - Mechanical Engineering. School of Electronic Engineering
Thesis Director: M.S.c. Jorge Hernando Ramón

INTRODUCCIÓN

Es común que el concepto de robótica en estos momentos haga referencia a androides presentados en las películas de ciencia ficción que poseen características humanas y que cumplen funciones de ayuda a los seres humanos, pero se olvida que este mismo objetivo es el que asumieron en sus orígenes hace muchos años los autómatas como herramienta de trabajo y entretenimiento de su dueño.

En la actualidad para la industria, la robótica es un concepto que enmarca los procesos repetitivos que se tienen designados sobre máquinas específicas (Barrientos 1.997), pero una de sus principales limitantes es la característica de estar fija; es aquí donde aparecen como una alternativa los robots móviles, como solución a los problemas en donde el desplazamiento es una necesidad.

Aunque en sus inicios el trabajo sobre la robótica móvil fue planteado únicamente de carácter didáctico, la investigación y las nuevas aplicaciones le han dado la posición y el auge que posee en este momento, enfocando todos los esfuerzos en el perfeccionamiento de estrategias de navegación y en la búsqueda de la completa autonomía a la hora de enfrentar situaciones no planeadas.

El proyecto que se plantea propone desarrollar un algoritmo de control basado en redes neuronales usando un lenguaje de alto nivel, el cual busca simular el control de la navegación de un robot en un entorno semicontrolado, teniendo como proyección la implementación de este algoritmo en un robot móvil, donde se debe tener en cuenta las limitaciones computacionales.

Este proyecto forma parte de la continuación de trabajo de grado SRM¹ (Carvajal 2006), y además forma parte del plan de propuesto por ERA², grupo de investigación estudiantil, el cual ha enmarcado sus proyectos de forma asociativa para permitir que el conocimiento alcanzado aquí, dé paso al planteamiento de nuevos proyectos que permitan promulgar un enfoque para la solución de problemáticas determinadas.

¹ SRM Sistema Robótico Modular

² ERA Electrónica Recreativa Avanzada

1. DESCRIPCIÓN DEL PROYECTO

En este proyecto se hace un estudio previo del concepto de redes neuronales, sus clases y ejemplos de implementación, que permiten entender su estructura y proporciona las características para escoger el mejor tipo de red, que se adapte a la implementación de un algoritmo de control para la navegación de un robot móvil.

El entorno para el cual se desarrollara este problema simulado, es una celda dividida por una cuadrícula en cuyos espacios se ubicaran obstáculos fijos, se tomara un robot móvil como modelo, el cual recibe las órdenes básicas de locomoción, adelante, atrás, izquierda, derecha, la rotación se realiza en el eje (Giros diferenciales), y tendrá ubicados sensores que detecten los obstáculos, conservando la medida de su desplazamiento sobre la celda.

Por ultimo se presenta el algoritmo propuesto simulado en una herramienta desarrollada para verificar la solución del problema.

Este algoritmo es el primer paso para su posterior implementación sobre un robot móvil, recibiendo las señales desde los sensores del móvil, procesándolas en el procesador, para luego transmitir la mejor acción de control a los actuadores desplazándose de un punto a otro.

1.1. TÍTULO DEL PROYECTO

“DESARROLLO DE ESTRATEGIAS PARA EL CONTROL DE MÓVILES AUTÓNOMOS BASADO EN REDES NEURONALES”.

1.2. OBJETIVO GENERAL

Desarrollar un algoritmo de control basado en redes neuronales para la navegación de un robot móvil que se desplace de un punto a otro, con obstáculos fijos.

1.3. OBJETIVOS ESPECÍFICOS

1.3.1. Objetivos en el campo de acción de Ingeniería Electrónica.

- Definir la configuración de la plataforma de navegación del robot así como la estructura funcional del mismo, con el fin de simular su navegación de un punto a otro con obstáculos fijos.

- Seleccionar el tipo de red neuronal que mejor se adapte a la configuración de la plataforma de navegación planteada.
- Realizar una comparación, entre el algoritmo propuesto y otras alternativas existentes para la solución del problema.
- Presentar una amplia bibliografía y los conceptos básicos de redes neuronales como herramienta inicial para el desarrollo de proyectos acerca de esta misma metodología.
- Continuar el desarrollo propuesto del proyecto, sistema robótico modular SRM, y dejar las bases sentadas para su implementación sobre una plataforma escogida.

1.4. JUSTIFICACIÓN

El nivel de aplicación de la robótica móvil está directamente relacionado con el ambiente industrial (Barrientos 1.997); el poder utilizar estos dispositivos sobre aplicaciones peligrosas, insalubres o simplemente mundanas surgen como la solución mas viable en la cual no es necesario arriesgar o subutilizar la vida de un ser humano.

Las necesidades de autonomía para las máquinas que cumplen estos diversos procesos en la sociedad actual, exigen el planteamiento de estrategias de control que posean la capacidad de enfrentar situaciones inesperadas y plantear de manera coherente una solución acertada.

La necesidad de solucionar el problema de la navegación en entornos no controlados es entonces uno de los primeros pasos a la hora de concebir cualquier tipo de móvil que desarrolle actividades específicas sin supervisión.

La cantidad de herramientas y metodologías de navegación que son planteadas a nivel mundial por las Universidades son incalculables, pero solo muy pocas de estas alternativas han tenido un seguimiento continuo desde su análisis y pruebas sobre prototipos hasta su implementación en modelos reales utilizados en la industria de manera exitosa (Martínez 2001).

A nivel regional es limitado el avance sobre metodologías de control para móviles autónomos, el impulso mundial propicia desde hace escasos años su estudio y gracias al desarrollo de la robótica móvil en los grupos de investigación de universidades como, la Universidad Industrial de Santander, la Corporación Universitaria de Investigación y Desarrollo - UDI, la Universidad Pontificia Bolivariana y La Universidad Santo Tomás, se ha podido propiciar un camino de

investigación sobre esta necesidad.

La escuela de ingeniería eléctrica electrónica y telecomunicaciones ha venido desarrollando diferentes proyectos que han potencializado el estudio del área de la robótica en la universidad, pero no ha sido sino hasta hace muy poco que se ha dado un interés real en el desarrollo de proyectos que propician el adelanto en la robótica móvil, una rama que aunque no fué considerada muy relevante en sus inicios hoy toma importancia gracias al trabajo de grupos de investigación como CEMOS³, grupo de investigación en control, electrónica, modelado y simulación, y ERA, grupo de investigación estudiantil desarrollando prototipos, primer paso para el avance de modelos finales.

El avance que se busca con este adelanto es el de presentar un camino de desarrollo en la robótica móvil basado en la utilización de la técnica de redes neuronales para solucionar problemas específicos de navegación, y además dejar las bases sentadas acerca de las limitaciones del hardware que se requiere para implementar este tipo de algoritmo sobre un prototipo autónomo.

Es así que al ubicarse en el contexto actual, este proyecto cobra un carácter significativo para la Universidad Industrial de Santander y en especial para la Escuela de Ingeniería Eléctrica Electrónica y Telecomunicaciones, ya que se convierte en un nuevo aporte al presentar otra alternativa de solución al problema de la navegación autónoma de este tipo de dispositivos móviles, dejando las bases para poder implementar esta metodología en otro tipo de aplicaciones y además para tener una referencia para comparar esta propuesta con otros planteamientos futuros que se hagan a nivel universitario o mayor.

Cabe resaltar el aporte al grupo de investigación electrónica recreativa avanzada ERA, para el cual la solución de este tipo de problemáticas son un adelanto hacia la consecución de una herramienta para futuros desarrollos y que se usen de forma activa posiblemente en otros trabajos del grupo de investigación.

³ CEMOS Grupo de Investigación en Control, Electrónica, Modelado y Simulación

2. COMPRENSIÓN DEL PROBLEMA

En este capítulo se hace una descripción de la metodología utilizada para el desarrollo del proyecto y la investigación llevada a cabo previamente para llegar a una definición más precisa del problema.

2.1. METODOLOGÍA

Para lograr los objetivos planteados en el desarrollo de este proyecto se planea una estrategia de desarrollo que enmarca seis módulos principales que a su vez se subdividen en actividades específicas, cada uno de estos módulos es desarrollado de manera consecutiva para así, asegurar una continuidad y claridad de conceptos a la hora de iniciar cada nueva etapa .

2.1.2. *Metodología para el desarrollo del proyecto*

Módulo de Recopilación Bibliografía. Este módulo permite un conocimiento del estado del arte de las diferentes disciplinas que deben ser abordadas a la hora de resolver la problemática planteada especificando los requerimientos necesarios para proponer correctamente una solución adecuada.

Esta dividida en dos actividades específicas que entregarán al final una selección de los mejores documentos

Documentación: En esta primera actividad se realizará un trabajo de recopilación de información con relación al proyecto planteado buscando así obtener una fuerte base teórica sobre percepción, ubicación, navegación de robots móviles entre otros conceptos, además sobre la utilización y posterior aplicabilidad de redes neuronales.

Selección de información: Luego se procederá a realizar una selección específica de la bibliografía para el desarrollo de este proyecto dentro de la cual se encontrarán los planteamientos existentes que se compararán con el desarrollo final.

Módulo de Análisis de Parámetros del Entorno y Navegación. En este módulo se propone hacer un ejercicio inicial sobre un individuo donde se le proponga dejar

de lado los paradigmas previos sobre navegación e interpretar una analogía con un robot móvil que permita entender las limitaciones y requerimientos de este para desplazarse de un lugar a otro.

Cabe aclarar que no se trata de asumir la solución entregada por el individuo como única y absoluta si no que esta se usa como punto de partida particular hacia una solución general.

Módulo de Planteamiento y Comparación de Soluciones de Navegación. Este módulo también está dividido en dos actividades específicas consecuentes cuyo único objetivo es encontrar la mejor metodología de navegación para este proyecto

Planeamiento de alternativas: De acuerdo a las conclusiones obtenidas en el módulo anterior y teniendo en cuenta las limitaciones del entorno y las características del robot móvil propuesto se plantearán diferentes soluciones para la navegación simulada.

Comparación de soluciones: Se analizarán diferentes parámetros en los planteamientos propuestos y algunos en los tradicionales buscando la selección de una solución.

Módulo de Selección e Implementación del tipo de Red sobre el Algoritmo de Control. El objetivo de este módulo es integrar la alternativa de navegación escogida previamente con la propuesta de red neuronal mediante la programación de un algoritmo de control utilizando lenguaje de alto nivel, para esto se proponen dos actividades complementarias que son:

Propuesta de tipo de red: De acuerdo con la selección previa de la alternativa de navegación, se escoge el tipo de red que tenga un buen desempeño para las necesidades del proyecto

Programación del algoritmo de control: a continuación se procede al desarrollo de la programación en lenguaje de alto nivel, con lenguaje C en nuestro caso, que tendrá como objetivo principal integrar el tipo de red escogido con la alternativa de navegación.

Módulo de Comparación del Algoritmo de Control con otras Soluciones

existentes. Se tomaran las alternativas encontradas en la revisión bibliografía y se compararán con la solución propuesta, teniendo en cuenta los parámetros de entorno y navegación propuestos anteriormente.

Módulo de Documentación. Esta actividad enmarca la culminación del proyecto y recopila los avances conclusiones y recomendaciones que darán el punto de partida de otras investigaciones sobre este tema.

2.2. MARCO TEÓRICO

2.2.1. Hacia La Construcción De Maquinas Inteligentes: Esta búsqueda incansable nace en la segunda guerra mundial con el diseño de ordenadores analógicos para asumir la responsabilidad de controles antiaéreos o de navegación; a partir de este momento algunos investigadores enfocaron sus esfuerzos en la búsqueda de maquinas capaces de responder y aprender como los seres humanos, dando le paso a varias metodologías como la cibernética, que estudia el control y la comunicación entre los animales y las maquinas, y las redes neuronales las cuales surgieron como un intento de emular estructuras biológicas.

De manera simultanea, se inicia el desarrollo de los ordenadores digitales, catalogados por algunos como computación algorítmica, son los computadores tal y como los conocemos hoy en día, que debido a su eficacia, flexibilidad y versatilidad y apoyados en el impresionante desarrollo de las tecnologías electrónicas han convertido al binomio, lógica booleana-Von Neuman, en el enfoque dominante de las ultimas décadas (Martín del Brío 2002).

Pioneros como Turing y Von Neuman abrigaban la posibilidad de incorporar en las maquinas la capacidad de pensar, y tras esta inquietud nace la inteligencia artificial, termino que definía los métodos algorítmicos capaces de hacer pensar en los ordenadores, el extraordinario desarrollo de la IA en la época eclipso la cibernética y las redes neuronales, a finales de los 60 se llego a pensar que mediante esta metodología, en una década se podría alcanzar una maquina realmente inteligente pero aun hoy con computadores mas poderosos, estos no resultan mucho mas inteligentes que en sus inicios, debido a que en el momento de abordar ciertas tareas como aquellas denominadas del mundo real, donde la información que se presenta es masiva, imprecisa y distorsionada, la arquitectura Von Neuman sobre la que se asienta la IA, no es muy apropiada debido a los altos requerimientos computacionales de este tipo de tareas (Martín del Brío 2002).

Así se da la oportunidad del renacer de los paradigmas de cómputo alternativo como las redes neuronales y el nacimiento de nuevas alternativas como los sistemas borrosos, algoritmos genéticos o la computación evolutiva, donde las redes neuronales emulan el hardware del cerebro y los sistemas borrosos se ocupan del software (Delgado 1998).

Hoy aparecen dos corrientes importantes dentro de la búsqueda de inteligencia que en realidad son complementarias, la IA convencional basada en algoritmos manipuladores de información simbólica que se ejecutan sobre ordenadores von Neuman y la inteligencia computacional o soft computing donde encontramos las redes neuronales, los sistemas borrosos entre otros.

2.2.2. Robotica Movil: Aunque la palabra robot se concibió en sus inicios como autómatas con características humanas que enmarcaban historias de ciencia ficción de mediados del siglo XX (Delgado 1998), el parecido con los primeros modelos de maquinas antecesoras, los telemanipuladores, distaban mucho de ser cercano; pero gracias a la mitificación producida por su frecuente presencia en medios como la literatura o el cine, se ha permitido que el término sea familiar, originando que le abramos la puerta de nuestra cotidianeidad. (Barrientos 1997)

Esta ciencia encontró su primera oportunidad de saltar de la ficción a la realidad en los procesos de automatización de niveles de producción; poder resolver de manera eficiente algunas tareas repetitivas, como por ejemplo la alimentación de las distintas máquinas componentes de una célula de fabricación robotizada (Muñoz 1995), pero las necesidades de la industria aumentan y exigen el hecho de tener la capacidad de movilizarse en la célula de trabajo, así se plantea la opción de diseñar robots que ante cualquier cambio que afecte el desarrollo normal de la navegación, ejecute acciones alternativas que le permitan reanudar su labor (Muñoz 1995).

Sin embargo la robótica experimenta una ampliación de sus campos de acción en aplicaciones fuera del ámbito industrial, donde resulta costoso o imposible estructurar el entorno (Barrientos 1997); aplicaciones donde no existe la posibilidad de sistematizar o clasificar, y que exige soluciones aisladas a problemas concretos lo que significa un mayor grado de inteligencia y percepción, estas se convierten entonces en las características de un robot móvil, con una conexión inteligente entre las operaciones de percepción y acción para la consecución de los objetivos trazados así se enfrente cierto grado de incertidumbre (Muñoz 1995).

2.2.3. Redes Neuronales Artificiales: El continuo trabajo sobre redes neuronales

artificiales comúnmente conocidas solo como redes neuronales, son la viva esencia del seguimiento y desarrollo de técnicas que simulan la capacidad de procesamiento del cerebro a la hora de enfrentar problemáticas como reconocimiento de patrones, percepción y control etc.

Después de aparecer como una perspectiva de desarrollo importante, un análisis de sus principales limitaciones algunos años después, interrumpió su avance, pero luego gracias a investigadores que no desistieron de su trabajo se presentó un renacer donde innumerables modelos y la convergencia de diferentes áreas, las muestran como alternativas en tareas de procesamiento de bajo nivel (Martín del Brío 2002).

La composición de una red neuronal está dada por capas de entrada y salida, entre las cuales algunos modelos poseen además una capa oculta; la asociación entre las neuronas se hace mediante pesos y según las necesidades del usuario son agrupadas de manera que se desenvuelva de manera exitosa en la aplicación deseada, se pueden encontrar de forma monocapa o multicapa.

La red da la posibilidad de propagar la información de adelante hacia atrás y viceversa según lo requiera el proceso tratado.

Las cuatro características representativas de la red son:

- Su mecanismo de aprendizaje
- Su topología
- La información de entrada-salida
- La forma de representación de estas informaciones

Los dos tipos de mecanismos de aprendizaje que dan una clasificación en el tipo de red son el supervisado y el no supervisado, la diferencia radica en la necesidad o no de un supervisor que controle el aprendizaje de la red.

La información que recibe la red requiere realizar asociaciones entre la entrada y la salida para así almacenar dicha información, con este requerimiento aparecen dos tipos la heteroasociación en el cual se generan las salidas a partir de las entradas y la autoasociación donde se generan salidas solo para los datos más parecidos a los de entrada; dando así dos tipos de redes neuronales.

Por ultimo la clasificación de las redes según el estilo de representación de la información de entrada o de los datos de salida, de naturaleza analógica o de valores binarios o discretos, son condicionados con las funciones de activación que determinan según los requerimientos del sistema la forma más conveniente para presentar la información.

2.3. PRECISIÓN DEL PROBLEMA

2.3.1. Requerimientos generales: Para poder proponer una estrategia de navegación y para comparar esta con soluciones existentes a esta misma problemática se debe antes entender el concepto de navegación, sus implicaciones y requerimientos por esta razón se propuso una estrategia de análisis, que busca presentar como resultado el conocimiento de cuales son las limitaciones y requerimientos que se deben tener en cuenta para proponer un planteamiento de solución a la problemática de navegación.

Como se ubican los seres humanos?, como se desplazan ?, como llegan de un lugar a otro?, estos cuestionamientos, no son comunes en la vida diaria de las personas debido a que son acciones de ejecución automática, preconcebidas y mecanizadas en los primeros años de vida, pero que pasaría si se le pidiera a un ser humano que hiciera una reflexión de este tipo y se pusiera en la situación hipotética de tener la mismas limitaciones de un robot a la hora de emprender un desplazamiento o encontrar su ubicación?, el cual posee la capacidad de adquirir la información necesaria para la consecución de sus objetivos a través del planteamiento y las indicaciones del diseñador simulando inteligencia, permitiendo hacer una analogía con un recién nacido que inicia su proceso de aprendizaje, el cual, “posee una inteligencia muy limitada y debe iniciar un crecimiento cognitivo posterior”⁴

- **El hombre y la ubicación en el medio**

En primer lugar se propone olvidar por un momento todos los sentidos y ubicarse en una celda cuadrada con un área de 100 m² , con obstáculos que se desconociera por completo; se describe un robot que recibe las ordenes básicas de locomoción adelante atrás izquierda derecha, la rotación la realiza en el eje (diferencial); teniendo claro que se tiene como misión encontrar en ese ambiente una posición objetivo, se inicia entonces una serie de cuestionamientos que

⁴ <http://www.waece.org/educacionprepostnatal/seccion4/desarrollopsicoev.htm>

buscan plantear cuales serian las variables fundamentales a analizar para culminar exitosamente la misión.

Así inicia el trabajo de analogía; en este caso después de iniciar este ejercicio previo, se percibe que para poder realizar la misión de desplazarse de un punto a otro, el individuo necesita en primer lugar definir su posición, saber en que punto de la celda se encuentra y hacia donde se dirige; estimar la posición de un vehiculo es un objetivo sobre el cual se proponen gran cantidad de técnicas que varían según las necesidades del problema el entorno de navegación, pero antes que nada esta acción depende de la aplicación para la que el diseñador haya planteado el móvil.

- **Como encontrar el mejor camino**

Teniendo claro el punto de partida y el punto de llegada se propone encontrar el trayecto más adecuado para alcanzar el objetivo trazado; en el caso del ejercicio propuesto, el individuo plantea esta acción mediante un proceso de visualización donde mentalmente se proponen los caminos posibles buscando el recorrido mas corto que asegure la llegada al destino.

Para el caso de un robot móvil el planteamiento de una trayectoria requiere la capacidad de proponer caminos mediante algoritmos matemáticos sobre entornos conocidos o no previamente, por lo tanto esto indica que se requiere la elección de un método de planificación

- **Ponerse de pie y caminar**

Antes de iniciar cualquier tipo de desplazamiento errático es primordial tener claro el objetivo de este desplazamiento por lo tanto en el caso de un móvil y basándose en la configuración planteada es necesario definir la posición objetivo y comparar este dato en cada momento con su nueva posición.

Preparar el desplazamiento, es la siguiente acción que propone el individuo, que haciendo la analogía con un móvil, seria utilizar su sistema de tracción para iniciar su movimiento; para cualquier posición del ambiente hacia donde se dirija, el móvil debe ubicar los obstáculos y plantear una acción que permita su evasión, es por eso que se requiere una planteamiento que integre la percepción de los sensores con la debida acción de los actuadores y que además retome la trayectoria propuesta inicialmente.

Por ultimo el individuo necesito darse cuenta de que llego a su a la posición deseada, lo que en el caso de un móvil se convierte en el hecho de adicionar un cuestionamiento que indique el alcance o no de la misión planteada

2.4. ANÁLISIS DE SOLUCIONES EXISTENTES

A continuación se procede a presentar el análisis de los planteamientos existentes donde se aplican las redes neuronales a la robótica móvil, escogidos según la selección de la bibliografía, se analiza cada uno teniendo en cuenta sus características y buscando definir sus ventajas y desventajas para tenerlas como referencia al proponer una solución a la problemática planteada

Los planteamientos analizados fueron: (ver tabla 1):

1. *Una Arquitectura Distribuida para el Control de Robots Autónomos Móviles: un Enfoque Aplicado a la Construcción de la Plataforma Quaky-Ant.* – Martínez Barbera, Humberto.

2. *Navegación Planificada de un Robot Móvil en Entornos Interiores Desconocidos.* - Zamora, L.M., et al.

3. *A Neural Network Based Torque Controller for Collision-free Navigation of Mobile Robots.* – Yang, Simon X., et al.

4. *A Neural Network Approach to Real-Time Trajectory Generation.* – Meng, Max, y Xianyi Yang.

5. *A Neural Network Based Testbed for Modelling Sensorimotor Integration in Robotic Applications.* - Fagg, Andrew H., et al.

6. *An Introduction to Neural Networks.* – Krose, Ben, y Patrick Van Der Smagt.

7. *Método de Aprendizaje Simple para Navegación de Minirobots Móviles Rodantes.* - Ramírez, Gustavo.

8. *Concepción e implementación de un simulador de trayectorias para robots autónomos móviles.* – Vega Yance, Mac Gabriel.

En la tabla 1 se muestran algunas de las características de los planteamientos estudiados, y el análisis completo de estas soluciones existentes está documentado en el Anexo A.

Tabla 1. Características generales de los proyectos seleccionados

Proyecto	Nivel de Desarrollo	Aplicación de la Red Neuronal	Procesamiento	Autor
<i>Una Arquitectura Distribuida para el Control de Robots Autónomos Móviles</i>	Aplicación sobre un robot autónomo móvil (plataforma Quaky-Ant)	Fusión sensorial	tarjeta PC industrial (486 a 133MHz)	Martínez Barbera, Humberto. España (Tesis Doctoral)
<i>Navegación Planificada de un Robot Móvil en Entornos Interiores Desconocidos</i>	Aplicación sobre un robot autónomo móvil (plataforma Quaky-Ant)	Fusión sensorial	tarjeta PC industrial (486 a 133MHz)	Zamora, L.M., et al. España (Tesis de grado)
<i>A Neural Network Based Torque Controller for Collision-free Navigation of Mobile Robots</i>	Evaluación a través de la simulación	Control de torque	Ordenador común	Yang, Simon X., et al. Taiwán (Artículo)
<i>A Neural Network Approach to Real-Time Trajectory Generation</i>	Evaluación a través de la simulación	Coordinación de generación de trayectorias	Ordenador común	Meng, Max, y Xianyi Yang Belgica (Artículo)
<i>A Neural Network Based Testbed for Modelling Sensorimotor Integration in</i>	Aplicación sobre un robot de extremidades (plataforma Meno)	procesamiento de información visual y la coordinación del motor	un microcontrolador 68332	Fagg, Andrew H., et al. Estados Unidos (Artículo)

<i>Robotic Applicatioons</i>				
<i>An Introduction to Neural Networks</i>	Evaluación a través de la simulación	procesamiento de información sensorial y visual.	IBM AT	Krose, Ben, y Patrick Van Der Smagt Holanda (Libro)
<i>Método de Aprendizaje Simple para Navegación de Minirobots Móviles Rodantes</i>	Aplicación sobre un robot autónomo móvil	Navegación reactiva, percepción sensorial y respuesta de motores	PIC16F84A	Ramírez, Gustavo. Colombia (Tesis de Grado)
<i>Concepción e implementación de un simulador de trayectorias para robots autónomos móviles</i>	Evaluación a través de la simulación	Navegación reactiva, percepción sensorial y respuesta de motores	Ordenador común	Vega Yance, Mac Gabriel Colombia (Tesis de Grado)

Fuente: Autores del Proyecto

3. PROPUESTA DE NAVEGACION

En este capítulo se realiza primero una introducción a la navegación, haciendo el ejercicio de proponer una metodología didáctica para trazar un recorrido de un punto a otro, con las restricciones respectivas, posteriormente se escogen algunos de los métodos clásicos de planificación y basado en su revisión y el ejercicio inicial se seleccionan los parámetros que según el desarrollo son las limitantes fundamentales del proyecto, posteriormente se propone una metodología de navegación y a continuación se desarrolla una comparación de los algoritmos de planificación seleccionados según los parámetros escogidos para evaluar su desempeño, entregando al final el método de planificación de trayectorias mas adecuado para este proyecto.

3.1. INTRODUCCION A LA METODOLOGIA DE NAVEGACION

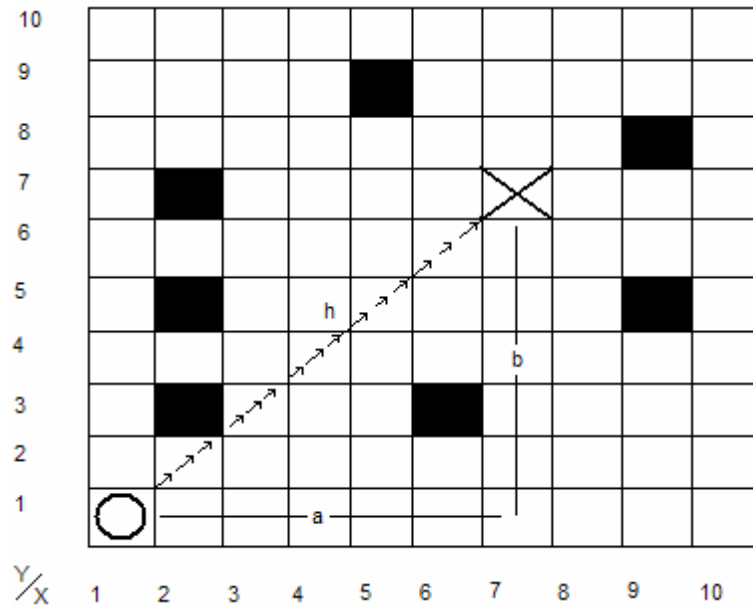
Si se da continuidad al ejercicio presentado en el capítulo dos, para la precisión del problema, se indica que es una necesidad la elección de un método de planificación de trayectorias, pero para entender los requerimientos de una metodología de navegación y para seleccionar posteriormente los parámetros de evaluación de las tácticas seleccionadas, se plantea hacer el ejercicio de proponer una metodología didáctica de planificación.

En primer lugar debemos establecer claramente las limitaciones del ambiente y las características del móvil las cuales serán el punto de partida del planteamiento. Se propone tener codificada una celda de 10 x 10, los requerimientos de percepción cantidad, ubicación y tipo de sensores, están relacionados directamente con las metodologías de navegación; se describe un robot que recibe las ordenes básicas de locomoción adelante atrás izquierda derecha, la rotación la realiza en el eje (diferencial); teniendo claro que se tiene como misión encontrar en el ambiente una posición objetivo.

Este procedimiento propuesto de manera didáctica, se aplica a entornos conocidos, entregando una trayectoria de cuadrados consecutivos hacia el punto objetivo.

A continuación se describe el proceso:

Figura 1. Sistema de coordenadas global, y trayectoria de desplazamiento asociado al robot



Fuente-. Autores del proyecto

Se asume un conocimiento previo del entorno, se toma la decisión de dividir el ambiente en una cuadrícula que fraccione el mapa en una matriz de 10*10, esto para asumir inmediatamente una distribución con coordenadas, los obstáculos se asumen de el mismo tamaño de los cuadrados que componen la distribución

Se calcula la hipotenusa del triangulo compuesto por los vectores que aparecen al hallar el desplazamiento por cada uno de los ejes, utilizando para este fin la diferencia de las coordenadas tanto del punto de partida como del punto de llegada entregando un numero de cuadrados de referencia.

Se toma el valor de la operación

$$h = (a^2 + b^2)^{\frac{1}{2}} \quad (1)$$

Donde

$$a = (x_f - x_o) \quad \text{y} \quad b = (y_f - y_o) \quad (2)$$

El resultado de esta operación nos entrega la cantidad mínima de cuadrados que

aseguran una trayectoria hacia el punto objetivo si no existieran obstáculos

- Si el resultado es entero se toma este dato como referencia.
- Si el resultado es un entero multiplicado por raíz de 2, solo se toma el valor entero como referencia.
- Si el resultado es un número decimal cuyo número después de la coma es inferior a 5, se toma como referencia el número antes de la coma.
- Si el resultado es un número decimal cuyo número después de la coma es superior a 5, se le suma 1 y este será el valor referencia.

Luego se trazan todos los posibles caminos desde la posición de inicio hasta la posición objetivo y se compara luego el número de cuadrados necesarios por cada uno de estos caminos con el valor referencia escogiendo el camino que entregue el valor mas cercano o igual al valor de referencia.

3.2. SELECCIÓN DE PARAMETROS

Para poder elegir un método de planificación se debe realizar un proceso de selección en el cual, según ciertos parámetros escogidos previamente, se comparen características de cada una de las propuestas, en esto radica la importancia de una correcta selección de los parámetros adecuados ya que si esto no se hace de manera adecuada la selección de la metodología de navegación idónea para el proyecto no será acertada

Como propuestas de planificación de trayectorias se escogieron algunos de los métodos clásicos de planificación descritos a continuación y presentados en el Anexo B

- *Planificación basada en grafos de visibilidad.*
 - Es necesario previamente el conocimiento del entorno o algún mecanismo de identificación de los vértices de los obstáculos.
 - Se aconseja el uso de sensores para la navegación local evitando así inconvenientes en el caso de escoger trayectorias no adecuadas.

- La estrategia es la siguiente.
 - o Identificar todos los vértices de los obstáculos del entorno.
 - o Unir mediante rectas todos aquellos nodos (vértices identificados) visibles.
 - o Escoger la mejor trayectoria generada.

- *Panificación basada en diagramas de Voronoi.*
 - Además de los vértices debe distinguir las aristas o en su defecto es necesario previamente el conocimiento del entorno.
 - Se aconseja el uso de sensores para la navegación local evitando así inconvenientes en el caso de escoger trayectorias no adecuadas.
 - La estrategia es la siguiente.
 - o Identificar todos los vértices del entorno.
 - o Identificar todas las aristas del entorno.
 - o Si se encuentran dos aristas enfrentadas, trazar una recta en el medio.
 - o Si se encuentra una arista y un vértice enfrentados, trazar una parábola en el medio.
 - o Unir mediante funciones los segmentos generados.
 - o Escoger la mejor trayectoria generada.

- *Planificación basada en modelado del espacio libre*
 - Es necesario previamente el conocimiento del entorno o un mecanismo de percepción que proporcione las características necesarias del ambiente.
 - Se aconseja el uso de sensores para la navegación local evitando así

inconvenientes en el caso de escoger trayectorias no adecuadas.

- La estrategia es la siguiente.
 - Identificar todas las aristas de los obstáculos del entorno.
 - Se identifica si las aristas encontradas cumplen los requerimientos para generar un cilindro rectilíneo generalizado (CRG).
 - Se calcula el ángulo formado por el corte de las rectas que contienen las aristas
 - Se calcula la bisectriz del ángulo.
 - Por ambos lados de las aristas se construyen segmentos rectilíneos paralelos al *eje* (bisectriz), con origen en los vértices de las aristas implicadas
 - Se detiene la proyección de eje hasta encontrar un obstáculo que lo corte
 - Se repite este proceso uniendo mediante funciones los segmentos generados de tal modo que la configuración de partida se encuentre en el primer cilindro de la sucesión y la final en el último.
 - Escoger la mejor trayectoria generada.
- *Planificación basada en la descomposición en celdas.*
- Es necesario previamente el conocimiento del entorno.
- Se aconseja el uso de sensores para la navegación local evitando así inconvenientes en el caso de escoger trayectorias no adecuadas.
- La estrategia es la siguiente.
 - Descomposición del espacio libre en celdas
 - Identificación de espacios libres.
 - Construcción de un grafo de conectividad.
 - Escoger la mejor trayectoria generada.

- *Planificación Basada En Campos Potenciales.*
- Es necesario previamente el conocimiento del entorno.
- Se aconseja el uso de sensores para la navegación local evitando así inconvenientes en el caso de escoger trayectorias no adecuadas.
- La estrategia es la siguiente.
 - Calcular el potencial que actúa sobre el vehículo en la posición actual según la información recabada de los sensores.
 - Determinar el vector fuerza artificial.
 - En virtud del vector calculado construir las consignas adecuadas para los actuadores del vehículo que hagan que éste se mueva según el sentido, dirección y aceleración especificadas por $F(p)$.

De acuerdo con todo lo investigado hasta esta fase del proyecto, al análisis de ejercicio didáctico propuesto, y a la revisión de algunos de los métodos de navegación escogidos, se eligieron los siguientes criterios de evaluación.

- Criterio A: Percepción.
- Criterio B: Memoria.
- Criterio C: Procesamiento.

3.3. PLANTEAMIENTO DE METODOLOGIA DE NAVEGACION

Para el planteamiento de la metodología de navegación y la programación del algoritmo de control se ha tomado como base fundamental la experiencia adquirida mediante el estudio de la problemática de planificación de trayectorias, la participación en diferentes concursos de robótica, el trabajo presentado en (Vega 2005) y prácticas sobre redes neuronales desarrolladas previamente.

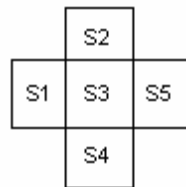
El punto inicial del planteamiento debe ser su objetivo fundamental, teniendo el móvil, la referencia de su punto de partida y un punto de llegada, encontrar una posición de meta en un espacio de trabajo con obstáculos fijos, usando para su consecución una red neuronal que permita la evasión de obstáculos, el control de

los actuadores y el alcance de la meta.

- *Sensores.*

El microbot dispone de cuatro sensores que se distribuyen como se aprecia en la figura 2. Estos sensores detectan la presencia de un obstáculo y su respuesta es de dos estados, si hay detección se activara la señal lógica de 1 de lo contrario la señal será un cero, el punto indicado como S3 es la ubicación del móvil, no posee sensor.

Figura 2. Ubicación de los sensores en el móvil.

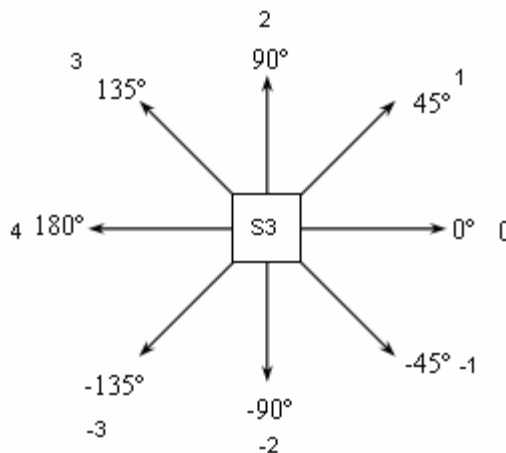


Fuente: Autores del proyecto

- *Orientación.*

Se considera que el punto de inicio para el carro es aleatorio y lo selecciona el usuario. La orientación del microbot en cada píxel se aproxima a una de las ocho posiciones siguientes.

Figura 3. Ángulos permitidos en la orientación del carro

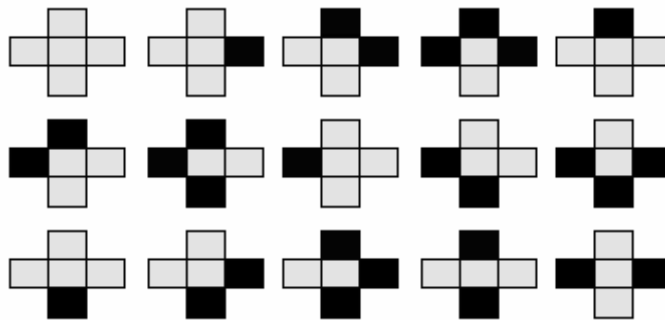


Fuente: Autores del proyecto

- *Casos de Entrenamiento de la Red.*

Para el logro del objetivo propuesto se propone el manejo de 4 sensores que permitan la detección de obstáculos en la periferia del móvil, y la utilización de un sistema de brújula para poder contar con el dato de orientación hacia la meta; a continuación se presentan algunos de los estados posibles de los sensores, que sumados con el ángulo medido entre el robot y la meta son utilizados para entrenar la red

Figura 4. Estados posibles de detección percibidos por los sensores



Fuente: Autores del proyecto

- *Acciones de Control.*

La red se entrena con un aprendizaje híbrido, que se compone de un nivel de aprendizaje supervisado y otro no supervisado, lo que indica que los vectores que muestran los valores de las entradas percibidas por los sensores deben ir acompañados por las salidas deseadas que proporcionan la más adecuada acción de control para las diferentes situaciones que enfrenta un móvil en su acción de evasión de obstáculos.

Se considera que el robot tiene en cada punto de la trayectoria la posibilidad de moverse a la siguiente posición en cualquier dirección. El valor del incremento puede ser +1 o -1, de no haber incremento el valor será 0. Según estos criterios las acciones básicas de control que debe proporcionar la red son ocho:

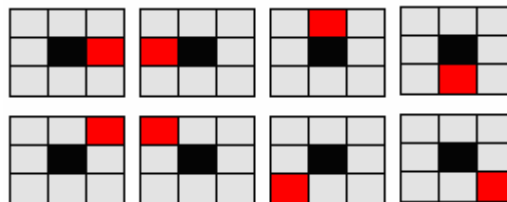
Tabla 2. Acciones de control de desplazamiento del móvil.

<i>Desplazamiento</i>	<i>x</i>	<i>y</i>
<i>derecha</i>	1	0
<i>Izquierda</i>	-1	0
<i>Arriba</i>	0	1
<i>abajo</i>	0	-1
<i>diagonal superior derecha</i>	1	1
<i>diagonal superior izquierda</i>	-1	1
<i>diagonal inferior izquierda</i>	-1	-1
<i>diagonal inferior derecha</i>	1	-1

Fuente: Autores del proyecto

En la figura 5 se muestra los posibles posiciones a las que puede ir el móvil (el color negro representa el origen y los cuadros negros representan el destino).

Figura 5. Posibles movimientos del carro programados en la red PNN.



Fuente: Autores del proyecto

3.4. COMPARACION CON OTRAS ALTERNATIVAS

Para la selección de la más adecuada metodología se hizo una evaluación ponderada de acuerdo a tres criterios A, B y C, (que según el desarrollo son las limitantes fundamentales del proyecto), de cada una de las alternativas de planificación seleccionadas.

Teniendo en cuenta que para poder realizar una comparación basada en los criterios escogidos, las alternativas deben presentarse en el mismo lenguaje de programación y sobre la misma plataforma de desarrollo, por tal razón los

algoritmos seleccionados se presentan en MATLAB 7.0 tomando como referencia talleres de robótica para el control de robots móviles y el desarrollo presentado en (Vega 2005). Los algoritmos escogidos son presentados en el Anexo C.

La valoración se desarrolla con base a las preguntas que se enuncian a continuación, las cuales fueron respondidas teniendo como respaldo toda la investigación desarrollada hasta el momento. La valoración detallada de la forma en que se selecciono la más adecuada metodología de planeación de trayectorias está documentada en el Anexo D.

Criterio A: Percepción. (Peso: 4/10)

La necesidad de medición de las variaciones físicas del entorno para cada método, identifica los requerimientos en puertos del sistema?

Aunque no parece una característica relevante a la hora de evaluar, la información que se recibe a través de los sensores es la base fundamental para el planeamiento de cualquier tipo de trayectoria, si llegara a faltar la información de algún sensor la planeación sería errada, por esta razón debe tener un número adecuado de entradas y salidas, el cual permita manejar todos los posibles módulos del sistema.

Criterio B: Memoria. (Peso: 4/10)

Los requerimientos de memoria podrían convertirse en un aspecto diferenciador entre las propuestas presentadas?

Para poder implementar el algoritmo o para realizar operaciones entre cualquier tipo de dato escogido es claro suponer la necesidad de un espacio que me permita el desarrollo de estas acciones, es por eso que los sistemas presentan dos tipos de memoria, una memoria de programa (ROM) y una de datos (RAM) o variables.

Criterio C: Procesamiento. (Peso: 2/10)

La cantidad y calidad de operaciones depende directamente de la metodología escogida para la solución del problema?

Aunque este criterio recibió un peso menor que los dos anteriores es de suma importancia a la hora de plantear estrategias con necesidades altas de velocidad de procesamiento ya que poseer dificultades con este criterio limita alternativas como la posibilidad de alcanzar mayor velocidad de respuesta o aumentar el nivel de complejidad del algoritmo que derivan para procesadores no adecuados en retardos en el tiempo o la imposibilidad de la ejecución del programa.

3.5. SELECCIÓN DE METODOLOGIA DE NAVEGACION

La metodología planificación de trayectorias escogida fue la basada en redes neuronales teniendo en cuenta que el programa presentado para la evaluación es el expuesto en (Vega 2005), el cual servirá como guía para el algoritmo de control propuesto en este desarrollo de acuerdo con el planteamiento de metodología de navegación presentada en el apartado 3.3.

4. DESARROLLO DE ALGORITMO

En el siguiente capítulo se presenta la selección del mejor tipo de red y su programación buscando la creación del algoritmo de control.

4.1. SELECCIÓN DEL TIPO DE RED

La elección del tipo de red neuronal es la base fundamental para poner en práctica la metodología de navegación propuesta, como se observó en los proyectos seleccionados para analizar las soluciones existentes; la mayoría de metodologías utilizaban redes de tipo perceptrón multicapa con aprendizaje de retropropagación del error (backpropagation), para la acción de fusión sensorial; teniendo en cuenta las limitantes que presenta su implementación en dispositivos embebidos (memoria, procesamiento, percepción, etc.) por lo tanto se propone la utilización de un otro tipo de redes con características semejantes pero con mejor desempeño en muchos aspectos (Demuth 1998), las escogidas fueron las redes PNN, emparentadas con las redes de base radial (RBF).

De la misma manera que las MLP las RBF permiten modelar con relativa facilidad sistemas no lineales arbitrarios, ambas son arquitecturas en capas y unidireccionales, las dos son también aproximadores universales y se aplican al mismo tipo de tareas (Haykin 94).

La principal diferencia entre estas arquitecturas está en la operación de las neuronas ocultas es decir, así como para las MLP las neuronas ocultas poseen una respuesta de rango infinito (cualquier vector de entrada, con independencia del lugar del espacio de entrada de donde proceda puede causar que la neurona se active), en las RBF las neuronas son de respuesta localizada, pues sólo responden con una intensidad apreciable cuando el vector de entradas presentado y el centroide (distancia que separa al vector de entradas respecto del vector sináptico que cada una almacena), de la neurona pertenecen a una zona próxima en el espacio de las entradas .

El tiempo requerido para el entrenamiento de las RBF suele ser mucho más reducido que el del BP, el problema es que cuando se simulan sobre computadores convencionales, las primeras son más lentas debido a que normalmente necesitan un alto número de nodos ocultos. La razón las neuronas intermedias MLP cubren una mayor parte de espacio de entradas a diferencia de

los nodos ocultos RBF que actúan localmente (Vega 2005).

Como siempre, el empleo de uno u otro modelo con mayor o menor éxito dependerá de la aplicación concreta (Martín del Brío 2002), y aunque de reciente introducción, la red RBF cada vez cuenta con más aplicaciones prácticas gracias a su simplicidad, generalidad y facilidad de aprendizaje considerándose de tipo híbrido por incorporar aprendizaje supervisado y no supervisado.

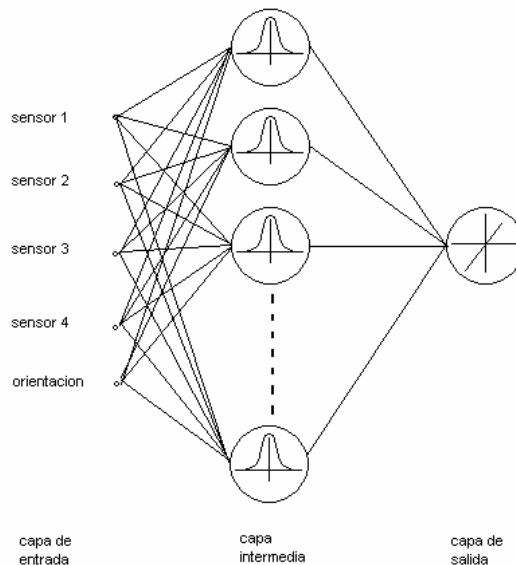
Para conocer claramente las características de la red PNN se presenta un resumen de estas en el Anexo E.

4.2. PROGRAMACION DEL ALGORITMO DE CONTROL

Cuando se inicia la programación del algoritmo de control es muy importante escoger la cantidad adecuada de casos de entrenamiento que indicaran la cantidad de neuronas ocultas, debido a que al aumentar el número de este tipo de neuronas se reduce la velocidad de procesamiento

La red neuronal usada fue una red PNN emparentada con la red de base radial, cuya configuración aparece en al figura 6. Consta de tres capas.

Figura 6. Topología de la red neuronal para el trazado de trayectorias



Fuente: Autores del proyecto

La capa de entrada posee cinco neuronas ordenadas en un vector columna, las cuatro primeras envían información del estado de activación o desactivación de cada sensor y la quinta posición representa el ángulo que hay entre la horizontal y la línea imaginaria que une el carro y la meta como se puede observar en la figura 3, proporcionando un grado de libertad adicional para hacer que la red converja con mayor facilidad en los casos que presentan redundancia en la fase entrenamiento.

La capa intermedia esta compuesta por 136 neuronas del tipo competitivo, correspondientes a los casos de entrenamiento. Con función de evaluación del tipo distancia euclidiana y función de transferencia gaussiana. En esta capa se procesa la información combinatoria de las diferentes entradas a la red.

La capa de salida esta compuesta por una red competitiva que proporciona el dato que representa alguna de las posibles acciones de control ejecutadas por los actuadores

El entrenamiento se realizó a partir de 136 casos donde se tuvieron en cuenta las diferentes posibilidades de activación de los sensores y el ángulo que hay entre la horizontal y la línea imaginaria que une los el carro y la meta.

Los casos utilizados pueden encontrarse en el archivo "casos_net_pnn.m" y el programa se ejecuta con el archivo "plantra.m" en el programa de simulación adjunto.

En el Anexo F se presenta el algoritmo de control propuesto limitando las funciones enfocadas al desarrollo de proyectos con redes neuronales que presenta Matlab, las cuales se convierten en una gran ayuda para su implementación, con el objetivo de aproximar la programación al lenguaje de alto nivel (C), teniendo como proyección la implementación de este algoritmo en un robot móvil, donde se debe tener en cuenta las limitaciones computacionales.

5. EXPERIMENTACION

A continuación se presenta una herramienta de simulación, la cual permite verificar la metodología de navegación planteada proporcionando la posibilidad de la generación de diferentes espacios con obstáculos.

5.1. SIMULACIÓN

Para poder simular adecuadamente el comportamiento del algoritmo de control propuesto se decidió tomar como referencia las alternativas de simulación presentadas en practicas de redes neuronales desarrolladas previamente y el trabajo presentado en (Vega 2005), para a partir de estos modelos presentar una herramienta de simulación adecuada

La Herramienta de simulación fue desarrollada en Matlab 7.0 aprovechando su gran cantidad de herramientas graficas y flexibilidad en el manejo de funciones que facilitan el trabajo de programación, a continuación se describe paso a paso su funcionamiento.

Reseteado de variables: en esta parte del programa se borran todas las variables del programa que se deben volver a reutilizar cada vez que el programa se corra. Entre ellas se pueden encontrar, plano, tiempo de hallazgo del obstáculo, distancia recorrida en por el móvil, etc.

Introducción de los puntos de partida y llegada: este sección se implemento con el comando ginput; que permite obtener un punto cualquiera en una gráfica y no hay necesidad de introducir por teclado ningún número. Además hubo la necesidad de redondear el número al entero más cercano. Una vez introducido los puntos en la gráfica aparecen unos cuadrados de color verde marcando esos puntos.

Ubicar el carro en la posición de inicio: esto se logra haciendo las coordenadas del carro igual a las coordenadas de la entrada.

Evitar el desborde de las coordenadas del plano: el carro solo se debe mover en un plano de 10x10 cuadros y si está fuera de este rango no se podrá obtener el estado de los sensores alrededor del móvil.

Tomar el estado de los sensores: el estado de los sensores se obtiene de la lectura de una cuadrícula de 3 por 3 pixeles alrededor de la posición del carro, aclarando que los sensores están en las posiciones (N, norte; S, sur; E, este; W, oeste) del móvil y que los espacios vacíos de la matriz analizada se rellenan con

ceros para evitar el errores, un estado de 1 indica que el puesto está ocupado por un obstáculo y un 0 indica que el espacio puede ser ocupado por el móvil.

Obtención del ángulo entre el carro y la meta: Esto se logró hallando el ángulo entre el carro y la meta por medio de la ecuación que aparece en seguida, esto se redondeo a un ángulo de 45°.

$$\theta = \text{round}\left(a \tan\left(\frac{y_f - y_c}{x_f - x_c}\right)\right) / 45 \quad (3)$$

Donde x_f , y_f es la ubicación de la meta y x_c , y_c es la ubicación del carro.

Arreglo del vector de consulta de la red neuronal. Este se logra, cambiando la matriz del estado de sensores a un vector del tipo columna y luego se le agrega el ángulo entre la meta y el carro. La matriz se puede ver a continuación:

$$\text{Sensores} = [0 \ s_1 \ 0 \ s_2 \ s_3 \ s_4 \ 0 \ s_5 \ 0 \ \theta] \quad (4)$$

Consulta de la matriz a la red: una vez obtenida la matriz del estado de los sensores se consulta la red neuronal con el comando

$$A = \text{sim}(\text{netpnn}, \text{sensc}'). \quad (5)$$

Y la red neuronal responde con el incremento en el eje x y eje y que se le debe hacer al carro. Una vez se tengan los incrementos se debe actualizar las coordenadas del carro.

Mover los motores respectivos: los motores se semejan a un incremento o decremento de un píxel en el plano de trabajo, estos valores pueden ser de 1 para avanzar, 0 para mantener quieto el motor.

Cuando ya se ha encontrado la meta el programa imprime la ruta trazada y si no la encuentra imprime un mensaje que no se ha encontrado la trayectoria, a continuación el proceso es repetitivo hasta que el carro llegue encuentre la meta.

5.2. COMPROBACION TECNICA

La prueba del desempeño del algoritmo de control se obtiene por medio de 8 casos sobre los cuales para cada uno se ubicara un punto objetivo fijo y de manera aleatoria se escoge el punto de origen tabulando los resultados y repitiendo el proceso en 10 ocasiones; los resultados se presentan en el Anexo G.

Los tres resultados posibles en una simulación son:

Correcto: se traza de manera clara un camino que une el punto objetivo con el punto de origen sin pasar por recorridos físicamente imposible para un móvil.

Incorrecto: el trayecto propuesto plantea un movimiento incorrecto al ignorar los obstáculos ubicados en el plano atravesándolos.

No realizado: el camino propuesto en el simulador queda inconcluso.

Los ocho casos seleccionados para la prueba son:

Sin obstáculos: El plano no posee ningún obstáculo.

Baja densidad de obstáculos aleatorios: se toma la opción aleatorio en el simulador y se inicia el proceso.

Media densidad de obstáculos aleatorios: Además de seleccionar la opción aleatorio en el simulador se añaden de manera manual algunos obstáculos al azar.

Alta densidad de obstáculos aleatorios: El número de obstáculos debe ser mucho mayor que las anteriores opciones.

Alcobas: Se determina una distribución uniforme de los obstáculos simulando un pequeño apartamento.

Parqueadero: Se organizan obstáculos en líneas rectas aparentando lugares de estacionamiento.

Centro comercial: La distribución rectangular de los obstáculos reubica hacia el interior del plano.

Parque: El obstáculo en el centro del plano representa la fuente de un parque.

Cada una de las anteriores casos es presentado gráficamente en el Anexo G nuevamente.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones, recopila las deducciones adquiridas durante el desarrollo del trabajo de grado.

Se desarrollo un algoritmo de control de robots móviles basado en redes neuronales PNN, teniendo en cuenta los parámetros necesarios para su implementación en un sistema embebido

La ventaja de usar redes neuronales PNN radica en características como versatilidad de diseño debido a que son pocos los parámetros necesarios para su desarrollo y su carácter probabilístico, ya que este permite dar respuesta a casos que no fueron especificados en la etapa de entrenamiento.

La comprobación técnica sirvió para evaluar el algoritmo de evasión de obstáculos en diferentes entornos, en cada una de las pruebas se dio un número de aciertos mayor al 60% en todos los casos exceptuando la prueba de alta densidad y la prueba de alcoba, en general el vehículo es capaz de encontrar trayectorias en zonas poco pobladas o con bajo número de obstáculos; si bien algunos de los trayectos no son óptimos proporcionan una respuesta aplicable en la realidad.

El simulador desarrollado en MATLAB es una herramienta que permite visualizar el comportamiento de diferentes metodologías de navegación que se propongan en trabajos futuros en un terreno de dos dimensiones para un robot móvil autónomo.

Se presentan las bases sobre las necesidades de procesamiento, percepción y memoria necesarias para decidir la plataforma más adecuada donde se deba implementar el algoritmo de control propuesto en un robot móvil autónomo.

Los trayectos errados que se presentan en algunos casos se deben a la metodología de adquisición de los valores de la entrada de cada uno de los sensores los cuales no se activan de manera simultánea, sino que siguen un orden de percepción establecido en el simulador.

Se debe tener en cuenta que la relación entre capacidad de percepción y

velocidad de respuesta es proporcional debido a que entre mayor cantidad de sensores se posean los casos de entrenamiento aumentaran, lo que se traduce en mas nodos en la capa oculta que al final requieren mayor nivel de procesamiento

Se entrega una recopilación bibliográfica relacionada con temas como redes neuronales, alternativas para su aplicación, metodologías de navegación, entre otros, que puede ser necesaria para trabajos posteriores que se planteen sobre esta línea de investigación.

Para trabajos futuros se propone continuar la investigación sobre la aplicabilidad de las redes neuronales en otro tipo de metodologías enfocadas a resolver problemáticas de la robótica móvil y además analizar alternativas de control que también aporten a la solución de planteamientos de navegación.

BIBLIOGRAFIA

Amaya Quintero, Yamit Danilo, y John Alexander Ruiz Hernandez. "Localización Dinámica de Móviles y Obstáculos en una Escena Controlada para Aplicaciones en Robotica". Tesis de Grado, Universidad Industrial de Santander, Colombia, 2005.

Barrientos, Antonio, et al. **Fundamentos de Robótica**. 1a. ed. España: McGraw-Hill, 1997.

Blynel, Jesper, y Dario Floreano. Levels of Dynamics and Adaptive Behavior in Evolutionary Neural Controllers. **Swiss Federal Institute of Technology (EPFL)**, Institute of Systems Engineering, Edmonton. pp. 1-10.

Carvajal, Omar Orlando, et al. "Diseño y construcción de un sistema robótico modular que permita la configuración de distintos robots móviles programables. Proyecto interdisciplinario desarrollado entre la escuela de diseño industrial y la escuela de ingeniería electrónica". Tesis de Grado, Universidad Industrial de Santander, Colombia, 2006.

Delgado, Alberto. **Inteligencia Artificial y Minirobots**. Colombia: Ecoe ediciones, 1998.

Demuth, Howard, y Mark Beale. **Neural Network Toolbox**. Version 3.0. U.S.: The Math Works, 1998.

Fagg, Andrew H., et al. A Neural Network Based Testbed for Modelling Sensorimotor Integration in Robotic Applications. **Center for Neural Engineering**, University of Southern California, 1992, pp. 186-191.

Fernández León, José A. **Robótica Evolutiva: La Próxima Generación de Robots Autónomos**. Argentina: Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), 2005.

González Jiménez, Javier, y Anibal Ollero Baturone. "Estimación de la Posición de un Robot Móvil". Tesis de Maestría, Universidad de Málaga, España, 1997.

Haykin, Simon. **Neural Networks**. United States of America: Macmillan College Publishing Company, 1994.

Jain, Anil K, Jianchang Mao y K.M. Mohiuddin. Artificial Neural Networks: A

Tutorial. *IEEE Transaction on Neural Networks*, March 1996, pp 31-44.

Krose, Ben, y Patrick Van Der Smagt. *An Introduction to Neural Networks*. 8a. ed. Holanda: University of Amsterdam, 1996.

Martín del Brío, Bonifacio, y Alfredo Sanz Molina. *Redes Neuronales y Sistemas Difusos*. 2a. ed. España: RA-MA, 2002.

Martínez Barbera, Humberto. “Una Arquitectura Distribuida para el Control de Robots Autónomos Móviles: un Enfoque Aplicado a la Construcción de la Plataforma Quaky-Ant”. Tesis Doctoral, Universidad de Murcia, España, 2001.

Meng, Max, y Xianyi Yang. A Neural Network Approach to Real-Time Trajectory Generation. *International Conference on Robotics y Automation*, Proceedings of the 1998 IEEE, Mayo 1998. pp. 1725-30.

Meng, Max, y Xianyi Yang. A Neural Network Approach to Real-Time Collision-Free Navigation of 3-D.O.F. Robots in 2D. *International Conference on Robotics y Automation*, Proceedings of the 1999 IEEE, Mayo 1999, pp. 23-28.

Muñoz Martínez, Víctor Fernando. “Planificación de Trayectorias para Robots Móviles”. Tesis Doctoral, Universidad de Málaga, España, 1995.

Pinzón, Claudia Patricia. “Herramienta de apoyo para el Conocimiento Tecnologías Adaptativas en el Campo de las Redes Neuronales”. Tesis de Grado, Univesidad Industrial de Santander, Colombia, 1997.

Ramírez, Gustavo. “Método de Aprendizaje Simple para Navegación de Minirobots Móviles Rodantes”. Tesis de Grado, Universidad Nacional de Colombia, 2003.

Tibaduiza Burgos, Diego Alexander, “Planeamiento de Trayectorias de un Robot Móvil”. Tesis de Maestría, Universidad Industrial de Santander, Colombia, 2006.

Torben, B., et al. Timing is important: delaying action execution in Plastic Neural Networks. *Institute for Knowledge and Agent Technologies (IKAT)*, Universiteit Maastricht, pp. 1-8.

Torres Albarracin, Carlos Humberto, y Edward Yesith Mendoza Gonzales. “Control de Dos Móviles en un Entorno Dinámico”. Tesis de Grado, Universidad Industrial de Santander, Colombia, 2006.

Vaughan, Eric D. Bilaterally Symmetric Segmented Neural Networks for Multi-Jointed Arm Articulation. *Adaptive Systems*, University of Sussex, 2003, pp. 1-25.

Vega Yance, Mac Gabriel. "Concepción e implementación de un simulador de trayectorias para robots autónomos móviles". Tesis de Grado, Universitaria de Investigación y Desarrollo, Colombia, 2005.

Zamora, L.M., et al. "Navegación Planificada de un Robot Móvil en Entornos Interiores Desconocidos". Tesis de Grado, Universidad de Murcia, España, 2000.

ANEXOS

ANEXO A. ANÁLISIS DE SOLUCIONES EXISTENTES

ANÁLISIS DE SOLUCIONES EXISTENTES.

A continuación se procede a realizar el análisis de los planteamientos existentes donde se aplican las redes neuronales a la robótica móvil, escogidos según la selección de la bibliografía; se analiza cada uno teniendo en cuenta sus características y buscando definir sus ventajas y desventajas para tenerlas como referencia al proponer una solución a la problemática planteada.

Los planteamientos analizados son:

1. *Una Arquitectura Distribuida para el Control de Robots Autónomos Móviles: un Enfoque Aplicado a la Construcción de la Plataforma Quaky-Ant.* – Martínez Barbera, Humberto.

2. *Navegación Planificada de un Robot Móvil en Entornos Interiores Desconocidos.* - Zamora, L.M., et al.

3. *A Neural Network Based Torque Controller for Collision-free Navigation of Mobile Robots.* – Yang, Simon X., et al.

4. *A Neural Network Approach to Real-Time Trajectory Generation.* – Meng, Max, y Xianyi Yang.

5. *A Neural Network Based Testbed for Modelling Sensorimotor Integration in Robotic Applications.* - Fagg, Andrew H., et al.

6. *An Introduction to Neural Networks.* – Krose, Ben, y Patrick Van Der Smagt.

7. *Método de Aprendizaje Simple para Navegación de Minirobots Móviles Rodantes.* - Ramírez, Gustavo.

8. *Concepción e implementación de un simulador de trayectorias para robots*

autónomos móviles. – Vega Yance, Mac Gabriel.

El desarrollo de metodologías de navegación para robots móviles a nivel mundial, va ligado al desarrollo de este último tópico, que dejando de ser una alternativa didáctica de la robótica trata de involucrarse de manera mas directa en otro tipo de aplicaciones de mayor impacto, que requieran el planteamiento de algoritmos que aprovechen la integración de la gran cantidad de ciencias que convergen para permitir a un robot alcanzar un relativo nivel de inteligencia.

A nivel de la Universidad Industrial de Santander es necesario referir algunos trabajos relacionados con la robótica móvil y metodologías de navegación, que aunque sin ser muchos, con su adelanto, presentan la posibilidad de desarrollo sobre esta línea de investigación, y motivan a los estudiantes a trabajar en temas que requieren gran calidad de desarrollo.

Una plataforma para aplicaciones en robótica móvil (Torres 2006), que es utilizada para el desarrollo de dos técnicas de planeamiento de trayectorias, una campos de potencial y la otra corresponde a los algoritmos genéticos (Tibaduiza 2006), la descripción de como por medio del procesamiento digital de imágenes y algoritmos de detección de movimiento, se determina la localización de objetos móviles en un escenario controlado donde están presentes algunos obstáculos (Amaya 2005), o el SRM, un sistema robótico que tiene como objetivo ser una herramienta para el aprendizaje y el desarrollo de la robótica móvil (Carvajal 2006), son solo algunos de estos primeros pasos hacia el avance de la robótica móvil dentro de nuestra Universidad

El análisis que se presentara a continuación muestra un resumen de las características que poseen los planteamientos, para poder inferir sus ventajas y desventajas; previamente se presenta una definición de cada una de estas características escogidas.

- Metodología de Navegación: es un resumen de la estrategia o el enfoque con el cual se plantea una solución a la problemática de navegación
- Aplicación de la Red Neuronal: Se presenta la acción que cumplen las redes neuronales en cada uno de los trabajos escogidos
- Arquitectura de control: Se describe el lenguaje de programación y el compilador utilizado en los casos que se puede interpretar esta información

de los documentos

- Nivel de desarrollo: Se indica el alcance de cada uno de los proyectos descritos, teniendo como base dos niveles, simulación e implementación.
- Procesamiento: en este ítem se hace una presentación del mecanismo computacional utilizado en cada una de las metodologías según su nivel de desarrollo.

***Una Arquitectura Distribuida para el Control de Robots Autónomos Móviles:
un Enfoque Aplicado a la Construcción de la
Plataforma Quaky-Ant.***

Descripción

1 **Autor:** Martínez Barbera, Humberto.

2 **Tipo de desarrollo:** Tesis Doctoral

3. **Lugar:** Universidad de Murcia, España

4. **Año:** 2001

Esta tesis presenta una serie de trabajos relacionados a la robótica móvil, que abarcan los distintos niveles necesarios para conseguir que un robot móvil realice tareas de forma independiente en un entorno previamente desconocido. Se emplean técnicas de Soft-Computing (redes neuronales, algoritmos genéticos, y especialmente lógica difusa) para el desarrollo de los distintos niveles.

Metodología de Navegación: Se resuelve, en dos etapas, por la aplicación de dos técnicas de navegación. En la primera etapa se aplica un mapa de celdillas difusas para el problema de la elaboración de mapas, así como una búsqueda para el problema de la generación de trayectorias. En la segunda etapa se aplica un mapa de segmentos difusos para el problema de la elaboración de mapas, así como un sistema de localización basado en el mapa de segmentos difusos.

Aplicación de la Red Neuronal: Se presenta en una solución al problema de la

fusión sensorial, dicha fusión se resuelve inicialmente con una red neuronal simple; se entrenan varias redes con distintos parámetros, y la mejor red, aquella con el menor error sobre ejemplos de prueba, se utiliza como salida del proceso de aprendizaje. Para acelerar la ejecución del proceso de aprendizaje se utilizan simultáneamente cuatro equipos que ejecutan un algoritmo de backpropagation.

Arquitectura de control: Para facilitar el desarrollo de programas de control por medio de la arquitectura BGA, se ha definido un lenguaje (BG) que hace uso de la lógica difusa para especificar y definir los distintos módulos de control. La arquitectura BGA y el lenguaje BG se han integrado en un entorno de programación y simulación (BGen) desarrollado íntegramente en Java, con el objetivo de poder ser integrado en distintos robots móviles.

Nivel de desarrollo: Tomando como base la arquitectura hardware que da lugar a la plataforma Quaky-Ant, se realiza la aplicación sobre este robot autónomo móvil, que es la base de la mayor parte de las pruebas de los distintos algoritmos en el entorno real

Procesamiento: La computadora central del robot está basada en una tarjeta PC industrial (486 a 133MHz), donde se ubica la arquitectura de control del robot. Dicha tarjeta se encuentra en comunicación con el exterior a través de un modem radio. La arquitectura del sistema sensorial está compuesta de diversas tarjetas comunicadas a través de un bus I2C.

Navegación Planificada de un Robot Móvil en Entornos Interiores Desconocidos.

Descripción

- 1 **Autor:** Zamora, L.M., et al.
- 2 **Tipo de desarrollo:** Tesis de grado
3. **Lugar:** Universidad de Murcia, España
4. **Año:** 2000

La construcción del mapa del entorno y la localización del vehículo en el mismo son tareas fundamentales en las arquitecturas de control con navegación planificada, ante todo cuando no se dispone del mapa a priori o éste es incompleto. Por otra parte la utilización de sensores de distinto tipo para adquirir la información del entorno permite mejorar las deficiencias de cada uno de ellos por separado.

Metodología de Navegación: Se presenta un sistema de construcción de mapas basado en lógica fuzzy, estimando la posición de robot en el mismo con un filtro de Kalman extendido.

Aplicación de la Red Neuronal: Los datos obtenidos en los sensores utilizados de ultrasonido e infrarrojo son fusionados mediante el uso de una red neuronal que proporciona una estimación de la medida de las distancias, usada posteriormente en las tareas de navegación. Para realizar el entrenamiento de la red neuronal se ha utilizado el algoritmo de backpropagation, sobre un sistema distribuido formado por cuatro equipos que realizan experimentos en una parte del espacio de parámetros del algoritmo.

Arquitectura de control: se decidió definir e implementar un nuevo lenguaje de programación de alto nivel basado en la sintaxis del lenguaje de programación C, así como en la semántica y en muchas características de COLBERT. Este lenguaje, que se denomina BG, está basado en el paradigma multiagente donde cada agente posee una serie de comportamientos, siendo su destino principal la robótica y las aplicaciones de control, principalmente los robots móviles.

Nivel de desarrollo: se realizan numerosas pruebas en el simulador del robot con distintos entornos (y plataformas móviles), para posteriormente realizar las pruebas con el robot real (plataforma Quaky-Ant).

Procesamiento: La computadora central del robot está basada en una tarjeta PC industrial (486 a 133MHz), donde se ubica la arquitectura de control del robot. Dicha tarjeta se encuentra en comunicación con el exterior a través de un modem radio. La arquitectura del sistema sensorial está compuesta de diversas tarjetas comunicadas a través de un bus I2C.

A Neural Network Based Torque Controller for Collision-free Navigation of Mobile Robots.

Descripción

1 **Autor:** Yang, Simon X., et al.

2 **Tipo de desarrollo:** Artículo conferencia Internacional en robótica y automatización.

3. **Lugar:** Taipei, Taiwan

4. **Año:** 2003

Metodología de Navegación: Se propone el control de torque basado en una red neuronal para una navegación libre de colisiones en tiempo real en robots móviles. La propuesta de control de torque resulta de los obstáculos incorporados en un diseño de control basado en una técnica de potencial artificial la cual localmente coloca el robot lejos de los obstáculos para evitar las colisiones. Toda la información requerida del ambiente puede ser obtenida por los sensores a bordo del robot que tienen una visibilidad limitada solo a su rango de acción

Aplicación de la Red Neuronal: Para el control de torque se usa una red neuronal de capa sencilla, empleada para aprender por completo la desconocida dinámica del robot. La estabilidad es garantizada por la teoría de estabilidad de Lyapunov. El control fino de tiempo real de un robot móvil es llevada a cabo es a través de el aprendizaje on-line de la red neuronal.

Arquitectura de control: No es especificada de manera clara en el artículo debido a que este proceso solo es desarrollado hasta el nivel de simulación.

Nivel de desarrollo: La eficiencia del propósito de control es demostrada solo por estudios de simulación en los dos ambientes dinámico y estático.

Procesamiento: Este tópico depende de la selección de las especificaciones del ordenador que halla hecho el encargado de desarrollar el proyecto lo cual no es especificado en este articulo

A Neural Network Approach to Real-Time Trajectory Generation.

Descripción

1 **Autor:** Meng, Max, y Xianyi Yang

2 **Tipo de desarrollo:** Artículo conferencia Internacional en robótica y automatización.

3. **Lugar:** Leuven, Belgica

4. **Año:** 1998

Metodología de Navegación: Una estrategia basada en redes neuronales es propuesta para la generación de trayectorias libres de colisiones en un ambiente con varios obstáculos y un objetivo móvil. La red neuronal propuesta es aplicada a la formación de trayectorias para un robot móvil resolviendo problemas de tipo laberintos, localización dinámica de objetivos móviles, y la evasión de varios obstáculos.

Aplicación de la Red Neuronal: La dinámica de cada neurona es caracterizada por una ecuación de activación y una ecuación de adición. Cada neurona tiene solo conexiones locales, y la trayectoria optima es generada sin ninguna función de costo optima explicita y sin aprendizaje por lo tanto el modelo computacional es eficiente, La estabilidad de la red es analíticamente probada usando la funcion candidato de Lyapunov

Arquitectura de control: No es especificada de manera clara en el artículo.

Nivel de desarrollo: La eficiencia de este propósito abordado es demostrada a través de la simulación y estudios de comparación.

Procesamiento: Este tópico depende de la selección de las especificaciones del ordenador que halla hecho el encargado de desarrollar el proyecto lo cual no es especificado en este articulo.

A Neural Network Based Testbed for Modelling Sensorimotor Integration in Robotic Applications.

Descripción

1 **Autor:** Fagg, Andrew H., et al.

2 **Tipo de desarrollo:** Artículo Centro para la Ingeniería Neural

3. **Lugar:** Universidad del sur de California, Estados Unidos

4. **Año:** 1992

El reto de este trabajo es mostrar un ambiente factible en donde ambos, el procesamiento sensorial como la acción de los motores son integrados simultáneamente y adheridos para enfrentar el mundo exterior

Metodología de Navegación: Este reporte se enfoca hacia la integración de las acciones sensoras-motoras usando el lenguaje de simulación neural (NSL), y el ambiente desarrollo de aplicación de robótica rápida (R²AD) para la computación de la información sensorial y la coordinación del comportamiento de un motor respectivamente, con el propósito de la coordinación visual-motora en el mundo real

Aplicación de la Red Neuronal: El objetivo es entonces proveer un espacio de prueba usando redes neuronales para el procesamiento de información visual y la coordinación del comportamiento del motor, un modelo simple de integración del procesamiento de los sensores con el acoplamiento del comportamiento del robot.

Arquitectura de control: Este desarrollo utiliza el (NLS), un lenguaje de programación de propósito general, este sistema ofrece la interpretación de comandos interactivos que traducen el lenguaje de alto nivel a modelos neuronales específicos, es compuesto por módulos que organizan en dos subsistemas independientes un simulador y una ventana de internas; pero para soportar el desarrollo de los programas de control de el robot se utiliza un programa experimental llamado (R² AD), usado para la integración de todos los subsistemas físicos utilizados en el proyecto.

Nivel de desarrollo: Para este experimento en particular se logra alcanzar el nivel de la aplicación sobre una plataforma llamada Meno, que es una maquina que se desplaza sobre cuatro extremidades, a la cual se integran un brazo puma 560 y una cámara que entrega imágenes de resolución reducida.

Procesamiento: Esta parámetro depende de las características que posea para este experimento en particular la plataforma Meno, maquina que tiene un microcontrolador 68332

An Introduction to Neural Networks.

Descripción

1 **Autor:** Krose, Ben, y Patrick Van Der Smagt

2 **Tipo de desarrollo:** Libro

3. **Lugar:** Universidad de Ámsterdam, Holanda

4. **Año:** 1996

A diferencia de los anteriores planteamientos en esta referencia se presenta el punto de vista de Jorgensen (Jorgensen, 1987), que da algunas alternativas de solución para la navegación utilizando redes neuronales, incluidas en el *libro An Introduction to Neural Networks*.

Metodología de Navegación: El control de un brazo robótico y el control de un robot móvil es muy similar. El control jerárquico primero plantea un camino; este camino es transformado de un mundo cartesiano dominado por la unión del dominio de las ruedas usando la cinemática del sistema y finalmente un controlador dinámico que tome en cuenta la cartografía de los puntos de referencia en el dominio de la señal de los actuadores

Aplicación de la Red Neuronal: En la Primera alternativa se propone que las redes neuronales sean usadas conjuntamente con las lecturas directas de un escáner sonar sencillo para de manera asociativa pueda aproximar los rasgos globales del terreno no observables desde la perspectiva sencilla de un robot, para la segunda se prueba si la red neuronal es lo suficientemente rápida para ser usada en la planificación de caminos donde para el robot es requerida optimizar el movimiento y se posea la situación de una cámara ubicada en el móvil y las imágenes de un detector de rango láser.

Arquitectura de control: No es especificada de manera clara en el libro.

Nivel de desarrollo: Solo son presentadas las alternativas basadas en el trabajo de simulación desarrollado en el laboratorio por los autores, sin desarrollar

ninguna labor de implementación sobre alguna plataforma específica

Procesamiento: Este tópico depende de la selección de las especificaciones del ordenador que haya hecho el encargado de desarrollar los trabajos de simulación, que para este caso fue un IBM AT.

Método de Aprendizaje Simple para Navegación de Minirobots Móviles Rodantes

Descripción

1 **Autor:** Ramírez, Gustavo.

2 **Tipo de desarrollo:** Tesis de grado

3. **Lugar:** Universidad Nacional de Colombia.

4. **Año:** 2003

Metodología de Navegación: El robot no tiene una ruta definida, simplemente se desplaza por el recinto, tratando de no colisionar y mientras esto no suceda, su misión se limita a dirigirse hacia adelante siempre y cuando no se encuentre con un obstáculo.

Aplicación de la Red Neuronal: El procedimiento seleccionado apunta a una red neuronal asociativa tipo B.A.M. En este tipo de redes los elementos de ambas capas están interconectados entre si. Las unidades pueden o no tener conexiones de realimentación consigo mismas. Sin perder la generalidad de las redes neuronales, en estas arquitecturas tipo B.A.M. hay pesos asociados a las conexiones entre elementos del proceso.

Arquitectura de control: No es especificada de manera clara.

Nivel de desarrollo: Se propone la implementación del algoritmo de aprendizaje en un mini-robot móvil para que pueda navegar en un ambiente desconocido.

Procesamiento: Uno de los objetivos planteados del sistema es cumplir la condición de ser embebido en la memoria e implementar el sistema operativo sobre un microcontrolador de ocho bits tipo PIC16F84A.

Concepción e implementación de un simulador de trayectorias para robots autónomos móviles

Descripción

- 1 **Autor:** Vega Yance, Mac Gabriel.
- 2 **Tipo de desarrollo:** Tesis de grado
3. **Lugar:** Universitaria de Investigación y Desarrollo, Colombia
4. **Año:** 2005

La realización de esta tesis se dedicó al desarrollo de una herramienta software para llevar a cabo la planeación y seguimiento de trayectorias para Robots Autónomos Móviles.

Metodología de Navegación: Se busca la solución progresiva mediante diferentes técnicas o métodos clásicos e inteligentes, al problema de la planificación y seguimiento de trayectorias.

Aplicación de la Red Neuronal: La topología de la red neuronal usada es la de base radial, esta consta de tres capas. La capa de entrada la conforman los datos que van a ser procesados por la red, la segunda capa es una red competitiva o red inestable, esta red tiene la posibilidad de clasificar las entradas de acuerdo a la similitud que haya con los pesos previamente entrenados y la capa de salida, se usa como un acondicionador, es una red pureline donde la entrada proveniente de la capa intermedia se acomoda a un tipo de respuesta dado en el entrenamiento

Arquitectura de control: Se presenta un software de Planeación de Trayectorias desarrollado en *Matlab*.

Nivel de desarrollo: Solo son presentadas las técnicas sobre el trabajo de simulación, sin desarrollar ninguna labor de implementación sobre alguna plataforma específica.

Procesamiento: Este tópico depende de la selección de las especificaciones del ordenador que haya hecho el encargado de desarrollar los trabajos de simulación

ANEXO B. MÉTODOS CLÁSICOS DE PLANIFICACIÓN

MÉTODOS CLÁSICOS DE PLANIFICACIÓN

Todos ellos se fundamentan en una primera fase de construcción de algún tipo de grafo sobre el espacio libre, según la información poseída del entorno, para posteriormente emplear un algoritmo de búsqueda en grafos que encuentra el camino óptimo según cierta función de coste.

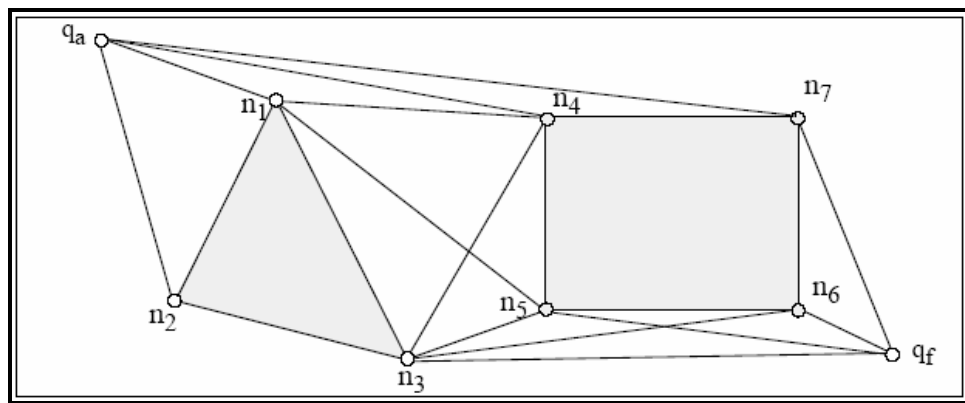
- **Planificación basada en grafos de visibilidad.**

Los grafos de visibilidad proporcionan un enfoque geométrico útil para resolver el problema de la planificación. Supone un entorno bidimensional en el cual los obstáculos están modelados mediante polígonos. Para la generación del grafo este método introduce el concepto de visibilidad, según el cual define dos puntos del entorno como visibles si y solo si se pueden unir mediante un segmento rectilíneo que no intercepte ningún obstáculo (si dicho segmento resulta tangencial a algún obstáculo se consideran los puntos afectados como visibles). En otras palabras, el segmento definido debe yacer en el espacio libre del entorno C_i .

Así, si se considera como nodos del grafo de visibilidad la posición inicial, la final y todos los vértices de los obstáculos del entorno, el grafo resulta de la unión mediante arcos de todos aquellos nodos que sean visibles. (Nilsson 1969)

En la figura que se presenta a continuación se muestra el grafo de visibilidad construido en función a los obstáculos poligonales existentes en el entorno y las configuraciones inicial q_a y final q_f .

Grafo de visibilidad en un entorno de dos obstáculos.



Fuente: OLLERO. Aníbal. Manipuladores y Robots móviles. Barcelona: Marcombo, 2001.p310

En el grafo mostrado previamente, se puede observar cómo sólo están unidos los nodos directamente visibles, de tal forma que el conjunto de arcos estará formado por las aristas de los obstáculos, más el resto de líneas que relacionan los vértices de los diferentes polígonos.

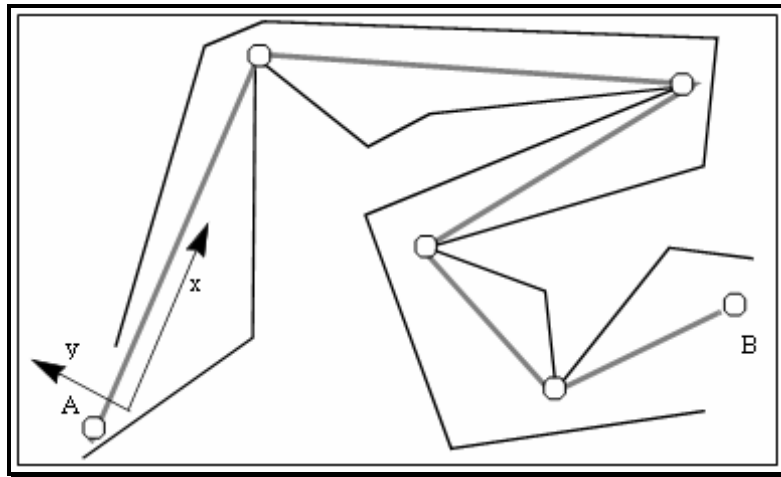
Mediante un algoritmo de búsqueda en grafos se elige la ruta que una la configuración inicial con la final minimizando alguna función de coste. La ruta que cumple el objetivo de la navegación queda definida como una sucesión de segmentos que siguen los requisitos especificados.

Aunque en principio el método está desarrollado para entornos totalmente conocidos, existe una versión denominada LNAV capaz de efectuar una planificación local a medida que se realiza la labor de navegación. (Rao 1988)

Este algoritmo, que parte de una determinada posición, determina los nodos visibles desde el punto actual. Elige el más cercano de los nodos visibles, según distancia euclídea a la posición final, para desplazarse posteriormente al nodo seleccionado y marcarlo como visitado. Desde esta nueva posición se vuelve a iterar el proceso hasta llegar a la posición final (éxito), o bien no existen más nodos sin visitar (fracaso).

Dentro de los métodos basados en grafos de visibilidad, se encuentran algoritmos especializados en la búsqueda de la ruta óptima que lleve al vehículo desde la posición inicial *A* hasta la final *B* a través de un entorno en el cual el espacio libre entre obstáculos está modelado mediante el uso de dos cadenas de segmentos. (Lodares y Abellanas 89)

Planificación con el espacio libre de obstáculos modelado mediante cadenas.



Fuente: OLLERO. Aníbal. Manipuladores y Robots móviles. Barcelona: Marcombo, 2001.p325

Aunque están restringidos a esquemas de entornos muy concretos, el uso queda justificado debido a su bajo coste computacional. Como se puede observar en la figura anterior, los algoritmos desarrollados para encontrar la ruta óptima bajo las condiciones descritas, se basan en enlazar los nodos situados en las zonas convexas del entorno tal que dos nodos consecutivos son visibles.

El uso de métodos de planificación y seguimiento basados en grafos de visibilidad está muy extendido, debido a que se pueden construir algoritmos a bajo coste computacional que resuelvan el referido problema. Sin embargo, utilizar como nodos los vértices de los obstáculos implica que no son inmediatamente aplicables en la práctica, ya que un robot móvil real no consiste en un punto. Por ello, algunos autores denominan a la ruta planificada semi-libre de obstáculos. (Latombe 91)

- **Planificación basada en diagramas de Voronoi.**

Al contrario que los métodos basados en grafos de visibilidad, la planificación basada en diagramas de Voronoi sitúa la ruta lo más alejada posible de los obstáculos. Con ello elimina el problema presentado por los grafos de visibilidad de construir rutas semi- libres de obstáculos.

Los diagramas de Voronoi se definen como una proyección del espacio libre del

entorno en una red de curvas unidimensionales yacientes en dicho espacio libre. Formalmente se definen como una retracción (Janich 84) con preservación de la continuidad. Si el conjunto C_l define las posiciones libres de obstáculos de un entorno, la función retracción RT construye un subconjunto C_v continuo de C_l .

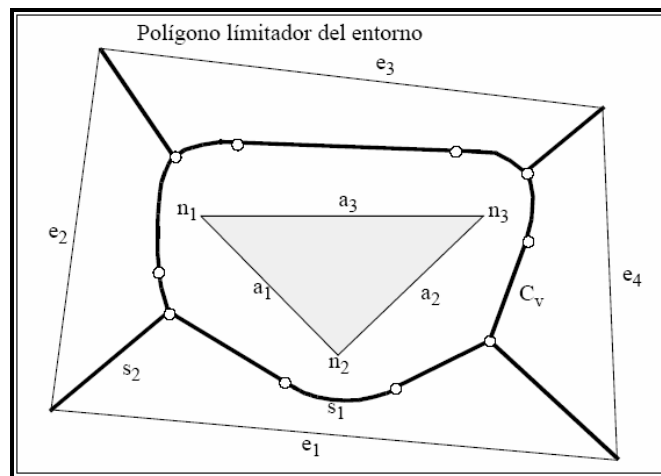
$$RT(q) : C_l \rightarrow C_v / C_v \subset C_l$$

De esta forma, se dice que existe un camino desde una configuración inicial q_a hasta otra final q_f , supuestas ambas libres de obstáculos, si y solo si existe una curva continua desde $RT(q_a)$ hasta $RT(q_f)$.

La definición de la función retracción RT implica la construcción del diagrama de Voronoi. La idea fundamental, es ampliar al máximo la distancia entre el camino del robot y los obstáculos. Por ello, el diagrama de Voronoi resulta el lugar geométrico de las configuraciones que se encuentran a igual distancia de los dos obstáculos más próximos del entorno. El diagrama estará formado por dos tipos de segmentos: rectilíneos y parabólicos. La elección de la modalidad de segmento corresponde con la clase de elementos de los obstáculos más cercanos que se encuentren enfrentados entre sí. (Rombaut 1991)

De esta forma, el lugar geométrico de las configuraciones que se hallan a igual distancia de dos aristas de dos obstáculos diferentes es una línea recta, mientras que en el caso de tratarse de un vértice y una arista resulta una parábola.

Retracción del espacio libre en un diagrama de Voronoi.

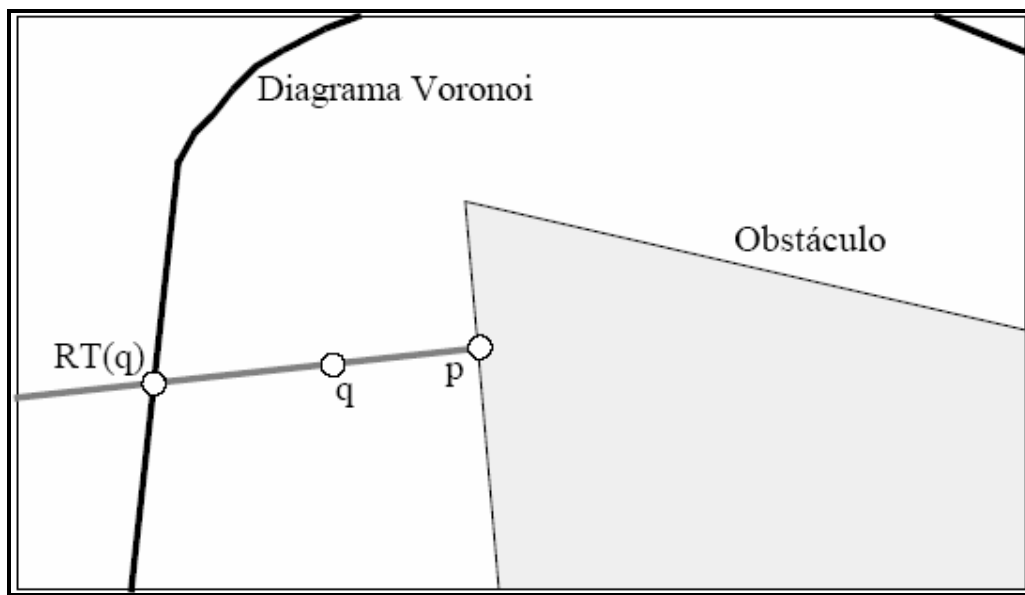


Fuente: OLLERO. Aníbal. Manipuladores y Robots móviles. Barcelona: Marcombo, 2001.p330

En la figura anterior se muestra un entorno delimitado por un polígono de aristas $\{e_1, e_2, e_3, e_4\}$ y un obstáculo triangular de vértices $\{n_1, n_2, n_3\}$ y aristas $\{a_1, a_2, a_3\}$. La retracción del espacio libre en una red continua de curvas es el diagrama de Voronoi C_v , representado mediante las líneas de trazo grueso. Los dos tipos de segmento utilizados en la construcción del diagrama pueden distinguirse en la mencionada figura, así, el segmento s_1 es el lugar geométrico de los puntos equidistantes entre la arista e_1 , y el vértice n_2 . Por otra parte, puede observarse como el segmento rectilíneo s_2 cumple la misma condición pero con respecto a las aristas e_1 y e_2 .

Dado una configuración q no perteneciente a C_v , existe un único punto p más cercano perteneciente a un vértice o arista de un obstáculo. La función $RT(q)$ se define como el primer corte con C_v de la línea que une p con q .

Imagen de una configuración q en el diagrama de Voronoi.



Fuente: OLLERO. Aníbal. Manipuladores y Robots móviles. Barcelona: Marcombo, 2001.p350

El algoritmo de planificación, en esencia, consiste en encontrar la secuencia de segmentos s_i del diagrama de Voronoi tal que conecten $RT(q_a)$ con $RT(q_f)$, Dicha secuencia conforma la ruta buscada. A continuación se describe el algoritmo:

- Calcular el diagrama de Voronoi.
- Calcular $RT(q_a)$ y $RT(q_f)$.

- Encontrar la secuencia de segmentos $\{s_1, \dots, s_p\}$ tal que $RT(qa)$ pertenece a s_1 y $RT(qf)$ pertenece a s_p .
- Si se encuentra dicha secuencia, devolver la ruta. Si no indicar condición de error.

Al igual que los grafos de visibilidad, este método también trabaja en entornos totalmente conocidos y con obstáculos modelados mediante polígonos. Sin embargo, también existen versiones para la utilización del mismo con obstáculos inesperados. (Meng 1988).

- **Planificación basada en modelado del espacio libre.**

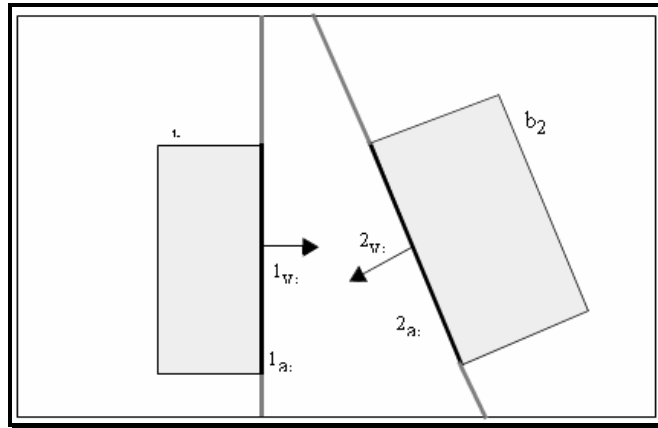
Se aplica a arquetipos de entornos con obstáculos poligonales, y la planificación en este caso se realiza mediante el modelado del espacio libre. Esta acción se lleva a cabo por los denominados cilindros rectilíneos generalizados (*CRG*). Al igual que los diagramas de Voronoi, con el uso de los *CRG* se pretende que el vehículo navegue lo más alejado de los obstáculos. De forma que la ruta que lleve al robot desde una configuración inicial hasta otra final estará compuesta por una serie de *CRG* interconectados, de tal modo que la configuración de partida se encuentre en el primer cilindro de la sucesión y la final en el último. (Brooks 86).

La construcción de un *CRG* se realiza a partir de las aristas de los distintos obstáculos que se encuentran en el entorno. Para que un par de aristas $1a_i$ y $2a_j$ pertenecientes a los obstáculos b_1 y b_2 respectivamente puedan formar un cilindro generalizado, deben cumplir las siguientes condiciones:

- i) La arista $1a_i$ está contenida en una recta que divide al plano en dos regiones. La arista $2a_j$ debe yacer por completo en la región opuesta en la que se encuentra situada b_1 . Este criterio es simétrico.
- ii) El producto escalar de los vectores normales con dirección hacia el exterior del obstáculo que contiene cada arista debe resultar negativo.

Si se cumplen estas condiciones significa que ambas aristas se encuentran enfrentadas, y por tanto se puede construir un *CRG* con ellas, como se observa en la siguiente figura:

Condiciones que deben cumplir dos aristas para construir un *CRG*.

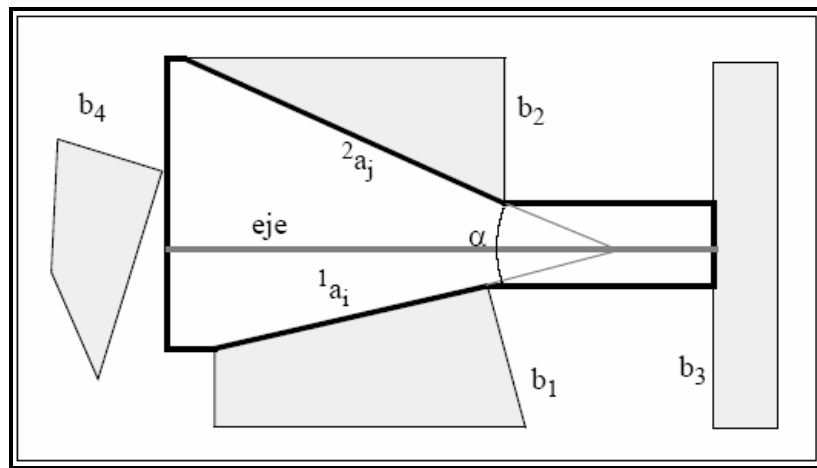


Fuente: OLLERO. Aníbal. Manipuladores y Robots móviles. Barcelona: Marcombo, 2001.p360

Una vez detectadas dos aristas que pueden formar un *CRG*, el siguiente paso será construirlo.

El proceso para alcanzar este cometido, se ve descrito a continuación:

Construcción de un *CRG*.



Fuente: OLLERO. Aníbal. Manipuladores y Robots móviles. Barcelona: Marcombo, 2001.p363

El primer paso es el cálculo del *eje* del *CRG*, el cual se define como la bisectriz del

ángulo formado por el corte de las rectas que contienen las aristas $1ai$ y $2aj$ que cumplen las condiciones i) y ii) expuestas más arriba. Por ambos lados de dichas aristas se construyen segmentos rectilíneos paralelos al *eje*, con origen en los vértices de las aristas implicadas y con extremo señalado por la proyección del primer obstáculo que corta el *eje*.

Repitiendo este proceso, se construye una red *CRG* en el entorno del robot que modela el espacio libre del mismo. El robot navegará por el eje del cilindro, en el cual se encuentran anotadas para cada punto el rango de orientaciones admisibles. El paso de un *CRG* a otro se produce siempre y cuando sus ejes intercepten y la intersección del rango de orientaciones admisibles en el punto de corte de ambos *ejes* no sea nula.

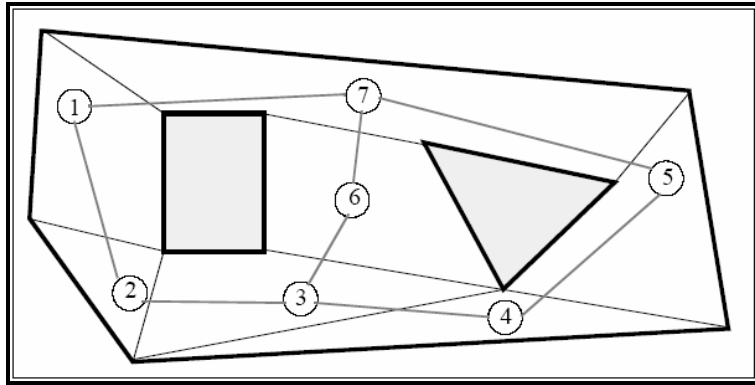
- **Planificación basada en la descomposición en celdas.**

Este tipo de métodos se fundamenta en una descomposición en celdas del espacio libre. Así, la búsqueda de una ruta desde una postura inicial qa hasta otra final qf , consiste en encontrar una sucesión de celdas que no presente discontinuidades, tal que la primera de ellas contenga a qa y la última a qf . Al contrario que los métodos expuestos a lo largo de este apartado, no encuentra una serie de segmentos que modele la ruta, sino una sucesión de celdas; por ello, se hace necesario un segundo paso de construcción de un *grafo de conectividad*, encargado de definir la ruta. (Torpe 84)

Para la planificación según el método de descomposición en celdas, se precisa la resolución de dos problemas: la descomposición del espacio libre en celdas y la construcción de un grafo de conectividad. El primero de ellos implica construir unas celdas con determinada forma geométrica tal que resulte fácil de calcular un camino entre dos configuraciones distintas pertenecientes a la celda, y la comprobación para averiguar si dos celdas son adyacentes debe disfrutar de la mayor simpleza posible. Aparte de estas características, la descomposición global del espacio libre implica que no deben existir solapamientos entre celdas y que la unión de todas ellas corresponde exactamente al espacio libre.

El grafo de conectividad es un grafo no dirigido, donde su construcción está asociada a la descomposición en celdas efectuada en el paso anterior, del tal forma, que los nodos van a ser cada una de las celdas, existiendo un arco entre dos celdas si y solo si son adyacentes

Descomposición en celdas y grafo de conectividad.

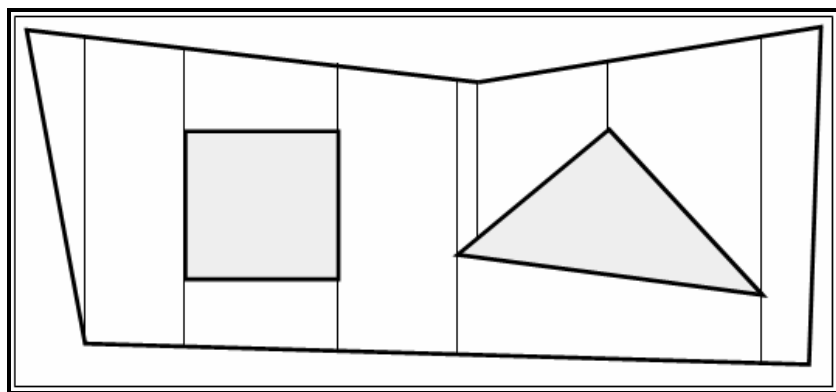


Fuente: OLLERO. Aníbal. Manipuladores y Robots móviles. Barcelona: Marcombo, 2001.p370

Una vez especificado el grafo de conectividad, sólo queda emplear un algoritmo de búsqueda en grafos, para la detección de la celda que contiene la postura a la cual se desea llegar, tomando como partida la que contiene la postura inicial.

Los distintos métodos basados en este principio, se distinguen por la forma en la cual realizan la descomposición en celdas y como se construye el grafo de conectividad. El método más sencillo de descomposición del espacio libre del entorno en celdas resulta de la denominada *descomposición trapezoidal*. (Latombe 91). Este método se basa en la construcción de segmentos rectilíneos paralelos al eje Y del sistema global F_g a partir de los vértices de cada uno de los elementos del entorno. El final del segmento queda delimitado por el primer corte de la línea con un elemento del entorno. Esta descomposición es la mostrada en la siguiente figura:

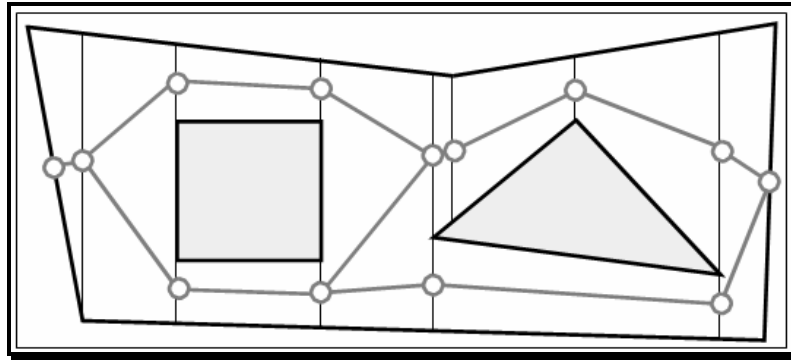
Descomposición trapezoidal del espacio libre.



Fuente: OLLERO. Aníbal. Manipuladores y Robots móviles. Barcelona: Marcombo, 2001.p371

El grafo de conectividad se construye por medio de la unión de los puntos medios (Figura siguiente) de los segmentos verticales definidos.

Grafo de conectividad de una descomposición trapezoidal.



Fuente: OLLERO. Aníbal. Manipuladores y Robots móviles. Barcelona: Marcombo, 2001.p375

Este tipo de enfoque se presta a muchas variantes, por ejemplo la utilización de varios niveles de resolución para una búsqueda jerarquizada (Kambhampati y Davis 86), o bien el uso de celdas en tres dimensiones para la planificación de caminos en espacios tridimensionales (Stentz, 90).

- **Planificación Basada En Campos Potenciales.**

Los métodos basados en campos potenciales poseen una concepción totalmente distinta a los expuestos más arriba al estar basados en técnicas reactivas de navegación. El ámbito de uso de esta técnica se centra en la planificación local en entornos desconocidos, como puede ser el sorteo en tiempo real de obstáculos o de los que no se tiene constancia. (Borenstein y Koren,89, 91)

La teoría de campos potenciales considera al robot como una partícula bajo la influencia de un campo potencial artificial, cuyas variaciones modelan el espacio libre. La función potencial U en un punto p del espacio euclídeo, se define sobre el espacio libre y consiste en la composición de un potencial atractivo $U_a(p)$, que atrae al robot hacia la posición destino, y otro repulsivo $U_r(p)$ que lo hace alejarse de los obstáculos, es decir:

$$U (p) = U _ a p + U _ r p$$

La fuerza artificial $F(p)$ a la que afecta el vehículo en la posición p , por el potencial artificial $U(p)$ resulta:

$$F(p) = -\nabla U(p)$$

Al igual que la función potencial, la fuerza artificial es el resultado de la suma de una fuerza de atracción $F_a(p)$, proveniente de la posición destino, y otra fuerza de repulsión $F_r(p)$ debidas a los obstáculos del entorno de trabajo:

$$F(p) = F_a(p) + F_r(p)$$

Así, la navegación basada en campos potenciales se basa en llevar a cabo la siguiente secuencia de acciones:

i) Calcular el potencial $U(p)$ que actúa sobre el vehículo en la posición actual p según la información recabada de los sensores.

ii) Determinar el vector fuerza artificial $F(p)$ según la expresión $F(p) = -\nabla U(p)$

iii) En virtud del vector calculado construir las consignas adecuadas para los actuadores del vehículo que hagan que éste se mueva según el sentido, dirección y aceleración especificadas por $F(p)$.

La iteración continua del ciclo expuesto proporciona una navegación reactiva basada en campos potenciales. El comportamiento del vehículo está muy ligado a la definición que se efectúe de los potenciales de atracción y repulsión. El potencial de atracción debe ir en función de la distancia euclídea a la posición destino, de forma, que a medida que el robot móvil se acerca, este disminuya su influencia. Por otra parte, el potencial repulsivo conviene que sólo influya en el movimiento del vehículo cuando éste se encuentre demasiado próximo a un obstáculo, de forma que la fuerza debida a este hecho tenga una dirección tal que lo aleje del mismo. En la posición destino es necesario que la suma de ambos potenciales resulte nula.

En el caso de conocer todo el entorno de trabajo y realizando una simulación del movimiento del robot a través del mismo, resulta posible construir una ruta que lleve al vehículo desde la posición inicial hasta la final. Dada la posición actual p_i , la próxima posición que debe alcanzar en un ciclo de simulación p_{i+1} , resulta:

$$p_{i+1} = p_i + \delta_i J(U(p))$$

donde δ_i es un factor de escalado y $J(U(p))$ representa al jacobiano de la función potencial en el punto p . El factor de escalado define la longitud del segmento con origen en p_i y final en p_{i+1} , y debe ser tal que dicho segmento esté libre de obstáculos.

El problema en este tipo de métodos deviene en la aparición de mínimos locales, es decir lugares que no son la posición destino en los cuales el potencial resulta nulo. Una situación de este tipo puede hacer que el robot quede atrapado en una posición que no sea la destino, o bien debido a la naturaleza discreta del método girar alrededor de ella. Solucionar este conflicto implica definir ciertas funciones potenciales que eviten la aparición de mínimos locales, lo cual resulta arduo, si bien existen soluciones que lo aseguran en entornos donde los obstáculos están modelados mediante círculos (Rimon y Koditschek, 1.988). Otra solución para evitar caer en un mínimo local se encuentra en el uso de un algoritmo de búsqueda en grafos. Para ello se divide el entorno mediante el uso de una rejilla. Cada celda tiene almacenado un valor que indica su potencial. Un algoritmo de búsqueda utilizable es usándose como función de coste la función potencial. La expansión de la celda elegida se realiza mediante el recurso a las celdas vecinas.

ANEXO C. ALGORITMOS SELECCIONADOS

ALGORITMOS SELECCIONADOS

Teniendo en cuenta que para poder realizar una comparación basada en los criterios escogidos, las alternativas deben presentarse en el mismo lenguaje de programación y sobre la misma plataforma de desarrollo, por tal razón los algoritmos seleccionados se presentan a continuación.

Voronoi.

```
%function PotentialPath
set(dista,'string','Dist=')
clear trace;
%close all;
%nFeatures = 7;
MapSize = 20;
%Map = MapSize*rand(2,nFeatures);
%Map=[5 5 5 5 5 5;1 2 3 4 5 6 7]
%a=round(rand(20,20));

[x,y]=ind2sub(size(matriz),find(matriz==1));
Map=[x';y'];
nFeatures=length(x);
nMaxSteps = 150;
%Map = [10;9.9];
%xGoal = [MapSize;MapSize];

%xStart = [1;1];

RadiusOfInfluence = 30;

KGoal= 1;
KObj = 1;
```

```

figure(1);%clf;
hold on;set(gcf,'doublebuffer','on');
mostrar
%plot(Map(1,:),Map(2,:),'rs');
axis([0 20 0 20]);
xStart = [1.1;1.1];
%xStart=ginput(1)';
plot(xStart(1),xStart(2),'rp');
set(estados,'string','Especifique punto de inicio')
xGoal=ginput(1)';
set(estados,'string','Hallando punto señalado')
plot(xGoal(1),xGoal(2),'rp');
xVehicle = xStart;
GoalError = xGoal - xVehicle;
trace(:,1)=xStart;
%Hr = DrawRobot([xVehicle;0],'r',[]);
k = 0;
while(norm(GoalError)>1 & k<nMaxSteps)

    %find distance to all entities
    Dp = repmat(xVehicle,1,nFeatures);
    Distance = sqrt(sum(Dp.^2));
    iInfluencial = find(Distance<RadiusOfInfluence);

    if(~isempty(iInfluencial))
        %vector sum of repulsions:
        rho = repmat(Distance(iInfluencial),2,1);
        V = Dp(:,iInfluencial);
        DrhoDx = -V./rho;
        F = (1./rho-1./RadiusOfInfluence)*1./(rho.^2).*DrhoDx;

```

```

    FObjects = KObj*sum(F,2);
else
    %nothing close
    FObjects = [0;0];
end;

FGoal = KGoal*(GoalError)/norm(GoalError);

FTotal = FGoal+FObjects;

Magnitude = min(1,norm(FTotal));

FTotal = FTotal/norm(FTotal)*Magnitude;

xVehicle = xVehicle+FTotal;

k = k+1;

Theta = atan2(FTotal(2),FTotal(1));
trace(:,k+1)=xVehicle;
plot(trace(1,:),trace(2:),'r')
%DrawRobot([xVehicle;Theta],'k',Hr);
pause(0.1);
drawnow;

GoalError = xGoal - xVehicle;

end;
if (k<nMaxSteps)
    set(estados,'string','Punto encontrado')

```

```

    long=longitud(trace');
        s=sprintf('Dist= %6.2f',long);
        set(dista,'string',s)
else
    set(estado,'string','Punto no encontrado')
end

```

Triangulación de Delaunay.

```

%function trace=findpathtri(matriz)
%clear x
%clear y
%x=[1 5 5 1 2 4 2 3 4];y=[5 5 1 1 2 3 3 3 3];
[x,y]=ind2sub(size(matriz),find(matriz==1));
ox=x;
oy=y;
mostrar
hold on
%plot(x,y,'*r')
%hold on
%[vx,vy]=voronoi(x,y);
%plot(vx,vy,'b')
%hace el hallazgo de la ruta optima
%clear
%x=[1 5 5 1 3 4 3 4];y=[5 5 1 1 3 3 4 4];
%function [pmx,pmy]=triangulos(x,y)
%plot(x,y,'*r')
tri=delaunay(x,y);
%trimesh(tri,x,y,zeros(size(x)))
%hold on
%plot3(x,y,ones(size(x)).1,'*r')
%mx=size(tri)

```

```

%trii=reshape(tri',1,3*mx(1,1));
%i=1;
%halla las distancias entre los puntos triangulados y deshace las distancias
menores de
%dos unidades
%for k=1:length(trii)-1
%  dista=sqrt((x(trii(k))-x(trii(k+1)))^2+(y(trii(k))-y(trii(k+1)))^2);
%  if dista>3
%    pmx(i)=(x(trii(k))+x(trii(k+1)))/2;
%    pmy(i)=(y(trii(k))+y(trii(k+1)))/2;
%    i=i+1;
%  end
%end

%plot(x,y,'*g')
%hold on
%plot3(pmx,pmy,ones(size(pmx))*1,'*g')

a=size(tri);
p1=(x(tri(:,1))+x(tri(:,2)))/2;
p2=(x(tri(:,2))+x(tri(:,3)))/2;
pmx=[p1 p2];
p1=(y(tri(:,1))+y(tri(:,2)))/2;
p2=(y(tri(:,2))+y(tri(:,3)))/2;
pmy=[p1 p2];
%trimesh(tri,x,y,zeros(size(x)))
[px,pi]=sortrows(pmy);
for i=1:length(pi)
    pmxo(i)=pmx(pi(i));
    pmyo(i)=pmy(pi(i));
end
k=1;

```

```

for i=1:length(pmx)-1
    d=sqrt((pmxo(i)-pmxo(i+1))^2+(pmyo(i)-pmyo(i+1))^2);
    if d>1.4142
        pmxd(k)=pmxo(i);
        pmyd(k)=pmyo(i);
        k=k+1;
    end
end
pmx=pmxd;
pmy=pmyd;
%plot(x,y,'*r')
hold on
%plot(pmx,pmy,'*b')
%plot(pmxd,pmyd,'*g')

vx=pmx;
vy=pmy;

xInicio=[1;1];
xFinal=ginput(1)';
plot(xInicio(1,1),xInicio(2,1),'gp')
plot(xFinal(1,1),xFinal(2,1),'gp')
%xFinal=[5;1];
xpoint=[vx(1,:) xInicio(1,1) xFinal(1,1)];
ypoint=[vy(1,:) xInicio(2,1) xFinal(2,1)];
%plot(xpoint,ypoint,'*')
a=[xpoint;ypoint];
%plot(vx,vy,'sg')
[m,b]=recta(xInicio(1,1),xInicio(2,1),xFinal(1,1),xFinal(2,1));

t=linspace(xInicio(1,1),xFinal(1,1),20);

```

```

py=m*t+b;

hold on
%plot(t,py,'r')
%plot(rx,ry,'r')

%plot(rx,ry,'r')
trace(:,1)=xInicio;
t=linspace(xInicio(1,1),xFinal(1,1),20);
k=1;
Pmovil=xInicio;
trace=xInicio;
romper=10;
j=1;
for i=1:length(t)
    if romper==1
        break
    end
    j=j+1;
    [m,b]=recta(Pmovil(1,1),Pmovil(2,1),xFinal(1,1),xFinal(2,1));
    t=linspace(Pmovil(1,1),xFinal(1,1),20);
    py=m*t+b;
    %t(j)
    %py(j)
    [pcx,pcy]=nearest(a,t(j),py(j));
    cerca([t(j) py(j)]',x,y)
    %limpiar
    %[t(j) py(j)]'
    %plot(t,py)
    %pause
    %plot(Pmovil,'*b')

```

```

if (Pmovil(1,1)~=pcx) | (Pmovil(2,1)~=pcy)
    k=k+1;
    trace(:,k)=[pcx;pcy];
    Pmovil=[pcx;pcy];
    j=1;
    if Pmovil==xFinal
        romper=1;
        %trace(:,k)=xFinal;
    end

end

end
%pause
end
trace(:,k+1)=xFinal;
%trace;
%plot(xpoint,ypoint,'*')
%hold on
plot(trace(1,:),trace(2:,:),'g')

Campos de Potencial.

% Taller de Robotica
% Unidad Didactica 6. Control de Robots Moviles
% Generacion de campos de potencial
% A. Ortiz

warning off MATLAB:divideByZero

% tamaño de la rejilla
n = 15;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% comportamiento GO-TO-GOAL %

```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
go2goal_u = zeros(n,n);  
go2goal_v = zeros(n,n);  
goal = [n-3 n-2];  
m = 0.75;  
for j=1:n  
    for i=1:n  
        aux = sqrt((j-goal(1))^2 + (i-goal(2))^2);  
        go2goal_v(j,i) = -m*(j-goal(1))/aux;  
        go2goal_u(j,i) = -m*(i-goal(2))/aux;  
    end  
end
```

```
[x,y] = meshgrid(1:n,1:n);  
figure(1); clf; hold on;  
fill([goal(2) goal(2) goal(2)-1 goal(2)-1 goal(2)], [goal(1) goal(1)-1 goal(1)-1 goal(1)  
goal(1)], 'g');  
quiver(x-0.5,y-0.5,go2goal_u,go2goal_v,0);  
grid on; axis equal; axis([0 n 0 n]);  
set(gca,'XTick',1:n,'YTick',1:n);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% comportamiento AVOID-OBSTACLE %
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
avoid_u = zeros(n,n);  
avoid_v = zeros(n,n);  
obst = [n-10 n-8]; soi = n/3;  
for j=1:n  
    for i=1:n  
        aux = sqrt((j-obst(1))^2 + (i-obst(2))^2);
```

```

d = aux;
m = 1/d;
if d > soi,
    avoid_v(j,i) = 0;
    avoid_u(j,i) = 0;
else
    avoid_v(j,i) = m*(j-obst(1))/aux; %(1/d - 1/soi)*(j-obst(1))/d^3;
    avoid_u(j,i) = m*(i-obst(2))/aux; %(1/d - 1/soi)*(i-obst(2))/d^3;
end
end
end

[x,y] = meshgrid(1:n,1:n);
figure(2); clf; hold on;
fill([obst(2) obst(2) obst(2)-1 obst(2)-1 obst(2)], [obst(1) obst(1)-1 obst(1)-1 obst(1)
obst(1)], 'r');
quiver(x-0.5,y-0.5,avoid_u,avoid_v,0);
grid on; axis equal; axis([0 n 0 n]);
set(gca,'XTick',1:n,'YTick',1:n);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% comportamiento STAY-ON-PATH %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

stay_u = zeros(n,n);
stay_v = zeros(n,n);
cpath = n/2; W = n/4;
for j=1:n
    for i=1:n
        stay_u(j,i) = 0;
        stay_v(j,i) = -(j-cpath)/W/2;
    end
end

```

```
end
```

```
[x,y] = meshgrid(1:n,1:n);  
figure(3); clf; hold on;  
plot([0 n],[cpath-0.5 cpath-0.5],'r',[0 n],[cpath-0.5-W/2 cpath-0.5-W/2],'g',[0  
n],[cpath-0.5+W/2 cpath-0.5+W/2],'g');  
quiver(x-0.5,y-0.5,stay_u,stay_v,0);  
grid on; axis equal; axis([0 n 0 n]);  
set(gca,'XTick',1:n,'YTick',1:n);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
% comportamiento MOVE-AHEAD %  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
ahead_u = zeros(n,n);  
ahead_v = zeros(n,n);  
dir = [1 1]/sqrt(2);  
for j=1:n  
    for i=1:n  
        ahead_u(j,i) = dir(2);  
        ahead_v(j,i) = dir(1);  
    end  
end
```

```
[x,y] = meshgrid(1:n,1:n);  
figure(4); clf; hold on;  
quiver(x-0.5,y-0.5,ahead_u,ahead_v,0);  
grid on; axis equal; axis([0 n 0 n]);  
set(gca,'XTick',1:n,'YTick',1:n);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```

% comportamiento COMBINADO: GO-TO-GOAL + AVOID-OBSTACLE %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[x,y] = meshgrid(1:n,1:n);
figure(5); clf; hold on;
fill([goal(2) goal(2) goal(2)-1 goal(2)-1 goal(2)], [goal(1) goal(1)-1 goal(1)-1 goal(1)
goal(1)], 'g');
fill([obst(2) obst(2) obst(2)-1 obst(2)-1 obst(2)], [obst(1) obst(1)-1 obst(1)-1 obst(1)
obst(1)], 'r');
quiver(x-0.5,y-0.5,go2goal_u+avoid_u,go2goal_v+avoid_v,0);
grid on; axis equal; axis([0 n 0 n]);
set(gca,'XTick',1:n,'YTick',1:n);

```

Redes Neuronales.

```

%encuentra una camino libre de obstáculos para un vehiculo en un laberinto de
20x20
%cuadros
%matriz=zeros(20,20);
%matriz(10,10)=1;
%matriz(9,10)=1;
%matriz(11,10)=1;
clear trace;
matrizt=matriz;
matrizt(1,1)=0;
pInicio=[1;2];
set(estados,'string','Señale el punto de llegada')
pFinal=round(ginput(1));
hold on
plot(pFinal(1),pFinal(2),'sg')
set(estados,'string','Hallando punto señalado')
pCarro=pInicio;

```

```

x=plnicio(1,1);y=plnicio(2,1);
veces=0;
while pFinal(1,1)~=pCarro(1,1) | pFinal(2,1)~=pCarro(2,1)
    [m,b]=recta(pCarro(1,1),pCarro(2,1),pFinal(1,1),pFinal(2,1));
    if x>1
        patron=matrizt(x-1:x+1,y-1:y+1)'
        pCons=reshape(patron',1,9)
    else
        patron=matrizt(x:x+1,y-1:y+1)'
        pCons=reshape(patron',1,6);
        pCons=[0 0 0 pCons];
    end
    a=round(sim(net,pCons'))
    %fig=matrizt';
    %fig(y,x)=1;
    %pcolor(fig)
    %axis off
    txt=[sprintf('x=%d ',x) sprintf('y=%d ',y) sprintf('px=%d ',pFinal(1,1)) sprintf('py=%d ',pFinal(2,1)) sprintf('m=%d ',m)];
    disp(txt)
    veces=veces+1
    if veces>60
        set(estado,'string','Punto no hallado')
        break
    end
    trace(veces,:)= [y,x];
    hold on
    plot(trace(:,2)+0.5,trace(:,1)+0.5)
    pause
switch a
case 1,

```

```

if abs(m)<3,
    if x>pFinal(1)
        x=x-1;
    else
        x=x+1;
    end
    y=round(m*x+b);
else
    x=x;
    y=y+1;
end
case 2,
    x=x+1;
    y=y;
case 3,
    x=x;
    y=y+1;
case 4,
    x=x;
    y=y-1;
end
pCarro(1,1)=x;
pCarro(2,1)=y;
if pCarro==pFinal
    trace(veces,:)= [y,x];
    limpiar
    plot(trace(:,2),trace(:,1))
    plot(pFinal(1),pFinal(2),'sg')
    set(estado,'string','Punto encontrado')
    long=longitud(trace);
    s=sprintf('Dist= %6.2f',long);

```

```
        set(dista,'string',s)
    break
end
    %pause
end
```

ANEXO D. VALORACIÓN DE CRITERIOS

VALORACIÓN DE CRITERIOS

Para la selección de la más adecuada metodología se hizo una evaluación ponderada de acuerdo a tres criterios A, B y C, de cada una de las alternativas de planificación seleccionadas.

La valoración se desarrolla con base a las preguntas que se enuncian a continuación, las cuales fueron respondidas teniendo como respaldo toda la investigación desarrollada hasta el momento. La valoración detallada de la forma en que se selecciono la más adecuada metodología de planeación de trayectorias está documentada en el Anexo D.

Criterio A: Percepción. (Peso: 4/10)

La necesidad de medición de las variaciones físicas del entorno para cada método, identifica los requerimientos en puertos del sistema?

Aunque no parece una característica relevante a la hora de evaluar, la información que se recibe a través de los sensores es la base fundamental para el planeamiento de cualquier tipo de trayectoria, si llegara a faltar la información de algún sensor la planeación sería errada, por esta razón debe tener un número adecuado de entradas y salidas, el cual permita manejar todos los posibles módulos del sistema.

En este caso se decidió realizar una valoración según los requerimientos de parámetros de entrada en cada sistema evaluando así el nivel de percepción según la siguiente escala.

Escala de valor: rango 1 - 5

Posición de inicio	- 1
Posición de objetivo	- 1
Matriz del entorno	- 5
Coordenadas de obstáculos	- 4
Estado de sensores	- 2

Metodología	Nivel de Percepción	Valor N.
Voronoi	11	1
Triangulación de Delaunay	6	0.54
Campos de potencial	11	1
Redes Neuronales	4	0.36

Criterio B: Memoria. (Peso: 4/10)

Los requerimientos de memoria podrían convertirse en un aspecto diferenciador entre las propuestas presentadas?

Para poder implementar el algoritmo o para realizar operaciones entre cualquier tipo de dato escogido es claro suponer la necesidad de un espacio que me permita el desarrollo de estas acciones, es por eso que los sistemas presentan dos tipos de memoria, una memoria de programa (ROM) y una de datos (RAM) o variables, lo que indica una distribución igual del peso asignado para las dos opciones.

Para poder realizar la evaluación se tomo cada uno de los cuatro algoritmos escogidos en los cuales se identificaron sus variables para poder sumar la magnitud en bits de cada una de ellas y obtener así una limitante de memoria RAM.

Metodología	Memoria RAM (Mbits)	Valor N.
Voronoi	38.784	0.19
Triangulación de Delaunay	83.486	0.40
Campos de potencial	205.632	1
Redes Neuronales	33.664	0.16

Para el caso de la memoria ROM se determino el conteo del numero de líneas de código, aclarando que este numero no es el equivalente a las necesarias en lenguaje de maquina, sino que es una proporción escogida para poder evaluar las metodologías sobre su mismo lenguaje

Metodología	Líneas de Código	Valor N.
Voronoi	50	0.63
Triangulación de Delaunay	71	0.90
Campos de potencial	79	1
Redes Neuronales	71	0.90

Criterio C: Procesamiento. (Peso: 2/10)

La cantidad y calidad de operaciones depende directamente de la metodología escogida para la solución del problema?

Aunque este criterio recibió un peso menor que los dos anteriores es de suma importancia a la hora de plantear estrategias con necesidades altas de velocidad de procesamiento ya que poseer dificultades con este criterio limita alternativas como la posibilidad de alcanzar procesamiento en tiempo real o aumentar el nivel de complejidad del algoritmo que derivan para procesadores no adecuados en retardos en el tiempo o la imposibilidad de la ejecución del programa.

En este caso se dividió en dos la evaluación de este criterio y simultáneamente el peso asignado, según las funciones utilizadas en el algoritmo y según las operaciones requeridas en el programa

Para la primera evaluación se toman las funciones más comunes utilizadas en los algoritmos y se procede dar una escala de valor según el nivel de dificultad que cada una de ellas representa en su procesamiento, para posteriormente realizar una sumatoria según su cantidad de apariciones en el programa

Escala de valor: rango 1 - 5

for - 4 break - 1 else - 2 case -1

If - 3 while - 4 switch - 3

Metodología	Total de sumatoria según funciones utilizadas	Valor N.
Voronoi	9	0.001
Triangulación de Delaunay	25	0.003
Campos de potencial	6490	1
Redes Neuronales	34	0.005

En la segunda se evalúa según las operaciones requeridas en cada programa asignando un valor según su nivel de complejidad en procesamiento, realizando posteriormente una sumatoria según sean requeridas cada una de ellas en el programa.

Escala de valor: rango 1 - 10

+ - 1 * - 2 ^ - 4 atan -8
 - - 1 / - 3 sqrt - 5

Metodología	Total de sumatoria según operaciones requeridas	Valor N.
Voronoi	808	0.58
Triangulación de Delaunay	979	0.70
Campos de potencial	1405	1
Redes Neuronales	910	0.65

RESULTADOS

Criterio	Peso (P)	Metodología											
		Voronoi			Delaunay			Campos			Redes		
		V	VN	P*VN	V	VN	P*VN	V	VN	P*VN	V	VN	P*VN
A	4	11	1	4	6	0.54	2.16	11	1	4	4	0.36	1.44
B1	2	38.8	0.19	0.38	83.5	0.40	0.8	205.6	1	2	33.7	0.16	0.32
B2	2	50	0.63	1.26	71	0.90	1.8	79	1	2	71	0.90	1.8
C2	1	9	0.00	0.00	25	0.003	0.00	6490	1	1	34	0.005	0.00
C2	1	808	0.58	0.58	979	0.70	0.70	1405	1	1	910	0.65	0.65
Total	10	-	-	6.22	-	-	5.46	-	-	10	-	-	4.21

De acuerdo a la anterior evaluación, la prioridad para seleccionar alguna de las respectivas metodologías es la siguiente:

<i>Metodología</i>	<i>Total</i>
<i>Redes</i>	4.21
<i>Delaunay</i>	5.46
<i>Voronoi</i>	6.22
<i>Redes Campos</i>	10

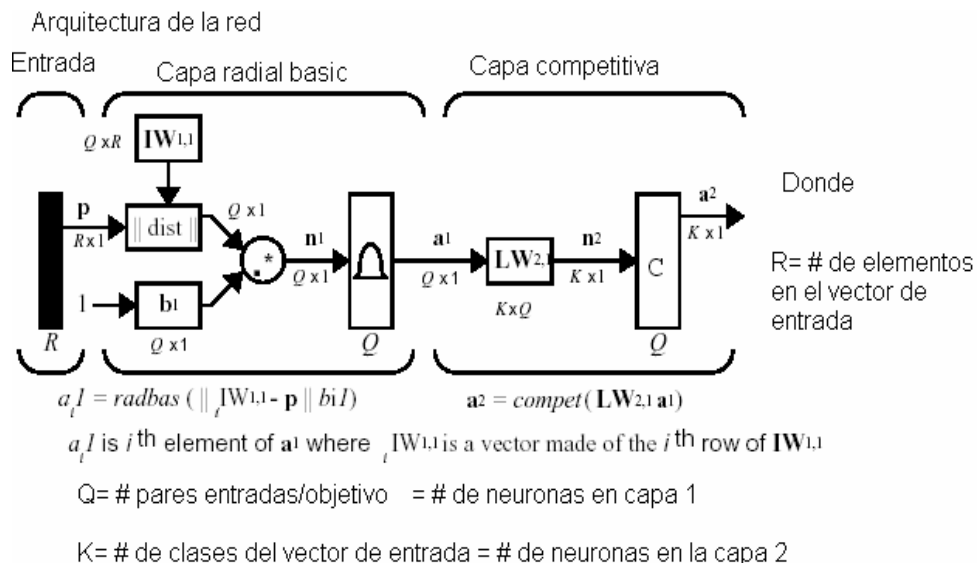
ANEXO E. RESUMEN RED PNN

RED PNN.

Esta red es muy utilizada en problemas de clasificación. La red consta de dos capas, una red de neuronas de base radial con un número de neuronas igual al número de vectores de entrenamiento y una capa competitiva de neuronas cuyo número es igual al número de categorías consideradas en el problema de clasificación. En la figura E.1 se muestra un esquema general de esta arquitectura de red.

Cada neurona de base radial (de la capa de entrada) almacena como pesos uno de los vectores de entrenamiento. En la fase de clasificación o reconocimiento, se presenta un nuevo vector de entrada. En la primera capa se calculan las distancias Euclídeas entre el vector de entrada y los pesos de cada neurona. En general, en la entrada de cada neurona de base radial se obtiene un escalar indicativo del parecido entre el vector a clasificar y los pesos de dicha neurona. Si para una neurona el escalar es cero, entonces el vector a clasificar ha resultado ser idéntico al vector de pesos de dicha neurona. Los escalares resultantes del cálculo de la distancia son multiplicados por un escalar denominado 'spread' que es idéntico para todas las neuronas de base radial de la red e introducidos como argumento en una función de tipo Gaussiana denominada 'radbas'.

Figura E.1 Esquema de la red PNN.



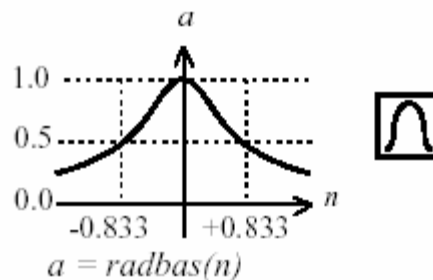
Fuente: Demuth, Howard, y Mark Beale. **Neural Network Toolbox**. Version 3.0. U.S.: The Math Works, 1998.

La expresión de dicha función (donde se ha denominado n al argumento) es:

$$radbas(n) = e^{-n^2} \quad (1)$$

La figura E.2 muestra el aspecto de dicha función que adquiere su valor máximo (igual a la unidad) cuando su argumento vale 0. Por lo tanto, dado un vector nuevo que deba ser clasificado, la neurona de base radial que produzca un máximo a su salida será aquella cuyos pesos más se parezcan al vector de entrada. La función del parámetro 'spread' es determinar la anchura de la campana de Gauss.

Figura E.2. Función de transferencia de una neurona de base radial



Fuente: Demuth, Howard, y Mark Beale. **Neural Network Toolbox**. Version 3.0. U.S.: The Math Works, 1998.

La segunda capa suma las contribuciones de las diferentes neuronas para generar un vector de dimensión igual al número de categorías. Este proceso es muy sencillo. Imaginemos que cuando resulta ganadora la neurona de base radial i esto implique que el vector de entrada pertenece a la categoría j . Entonces el peso del enlace entre la neurona de base radial i y la neurona competitiva j es igual a la unidad mientras que los pesos que enlazan a la neurona i con las demás neuronas competitivas (las otras categorías) son cero. El elemento de la posición i -ésima del vector resultante de sumar las contribuciones de las neuronas de base radial puede ser interpretado como la probabilidad de que el vector de entrada pertenezca a la categoría i -ésima. Finalmente, una función de transferencia "competitiva" sobre la salida de la segunda capa escoge el máximo entre todas las probabilidades, y produce un uno para esa clase y un cero para todas las demás clases.

La red probabilística (PNN) realiza un entrenamiento sumamente rápido puesto

que el proceso de ajuste de los pesos no es iterativo. Se ha demostrado que esta red se comporta como el algoritmo k-NN (k-nearest neighbour), si se escoge un valor para el 'spread' suficientemente bajo (típicamente de 0.1).

ANEXO F. PROGRAMA

PROGRAMA

En el Anexo F se presenta un análisis de valoración de criterios utilizando la misma metodología desarrollada en el Anexo D con la alternativas de planificación seleccionadas; primero se aplica sobre el algoritmo utilizado en el simulador (PNN 1), y simultáneamente sobre el mismo algoritmo de control propuesto pero limitando las funciones enfocadas al desarrollo de proyectos con redes neuronales que presenta Matlab (PNN 2), las cuales se convierten en una gran ayuda para su implementación; esta acción con el objetivo de aproximar la programación al lenguaje de alto nivel (C), teniendo como proyección la implementación de este algoritmo en un robot móvil, donde se debe tener en cuenta las limitaciones computacionales.

Criterio A: Percepción. (Peso: 4/10)

<i>Metodología</i>	<i>Nivel de Percepción</i>	<i>Valor N.</i>
<i>PNN 1</i>	4	0.36
<i>PNN 2</i>	4	0.36

Criterio B: Memoria. (Peso: 4/10)

<i>Metodología</i>	<i>Memoria RAM (Mbits)</i>	<i>Valor N.</i>
<i>PNN 1</i>	9.442	0.04
<i>PNN 2</i>	158.400	0.7

<i>Metodología</i>	<i>Líneas de Código</i>	<i>Valor N.</i>
<i>PNN 1</i>	72	0.91
<i>PNN 2</i>	1693	21.4

Criterio C: Procesamiento. (Peso: 2/10)

<i>Metodología</i>	<i>Total de sumatoria según funciones utilizadas</i>	<i>Valor N.</i>
<i>PNN 1</i>	44	0.00
<i>PNN 2</i>	1264	0.19

<i>Metodología</i>	<i>Total de sumatoria según operaciones requeridas</i>	<i>Valor N.</i>
<i>PNN 1</i>	1080	0.77
<i>PNN 2</i>	1500	1.07

RESULTADOS

<i>Criterio</i>	<i>Peso (P)</i>	<i>Metodología</i>					
		<i>PNN 1</i>			<i>PNN 2</i>		
		<i>V</i>	<i>VN</i>	<i>P*VN</i>	<i>V</i>	<i>VN</i>	<i>P*VN</i>
<i>A</i>	4	4	0.36	1.44	4	0.36	1.44
<i>B1</i>	2	9.442	0.04	0.08	158.4	0.70	1.40
<i>B2</i>	2	72	0.91	1.82	1693	21.4	42.8
<i>C1</i>	1	44	0.00	0	1264	0.19	0.19
<i>C2</i>	1	1080	0.77	0.77	1500	1.07	1.07
<i>Total</i>	10	-	-	4.11	-	-	46.9

<i>Metodología</i>	<i>Total</i>
<i>PNN 1</i>	4.11
<i>PNN 2</i>	-

Como se puede mostrar gracias a la valoración previa, el algoritmo de control propuesto que utiliza la red PNN para su desarrollo tiene un desempeño muy cercano (solo 10 puntos de diferencia) sobre la mejor alternativa de control analizada en el Anexo D.

Se debe tener en cuenta que los valores obtenidos para PNN2 solo son presentados para tener un acercamiento hacia las necesidades relacionadas con la implementación del programa sobre un dispositivo embebido y estos valores no pueden ser comparados con los resultados de otras propuestas debido a que esta propuesto sobre otro lenguaje de programación.

En el programa anexof.m presentamos el algoritmo de PNN2.

ANEXO G. COMPROBACIÓN TÉCNICA

La prueba del desempeño del algoritmo de control se obtiene por medio de 8 casos sobre los cuales para cada uno se ubicara un punto objetivo fijo y de manera aleatoria se escoge el punto de origen tabulando los resultados y repitiendo el proceso en 10 ocasiones.

Los tres resultados posibles en una simulación son:

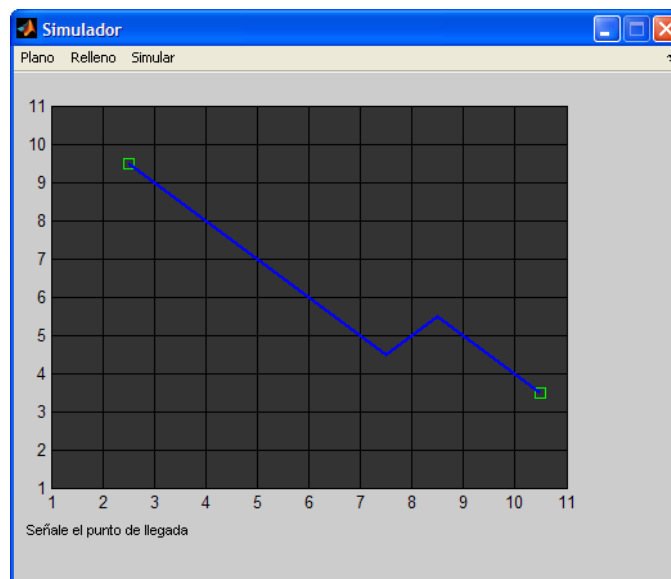
Correcto: se traza de manera clara un camino que une el punto objetivo con el punto de origen sin pasar por recorridos físicamente imposible para un móvil.

Incorrecto: el trayecto propuesto plantea un movimiento incorrecto al ignorar los obstáculos ubicados en el plano atravesándolos.

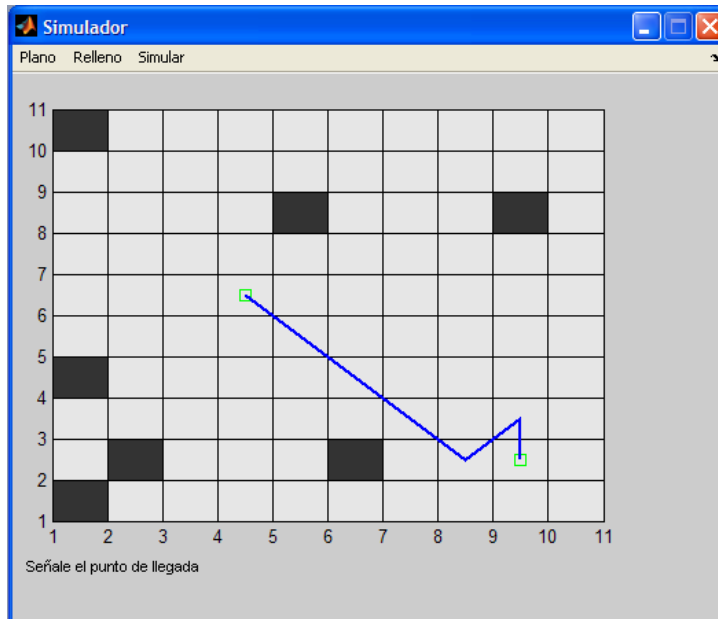
No realizado: el camino propuesto en el simulador queda inconcluso.

Los ocho casos seleccionados para la prueba son:

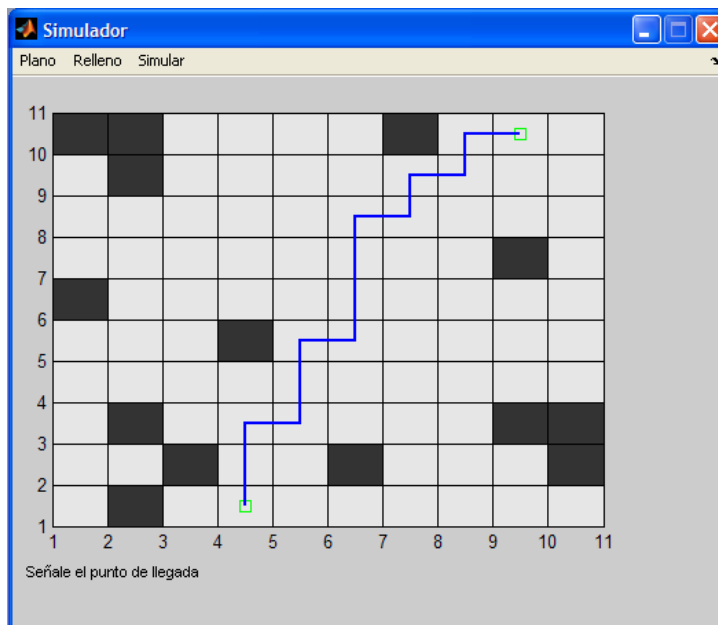
Sin obstáculos: El plano no posee ningún obstáculo.



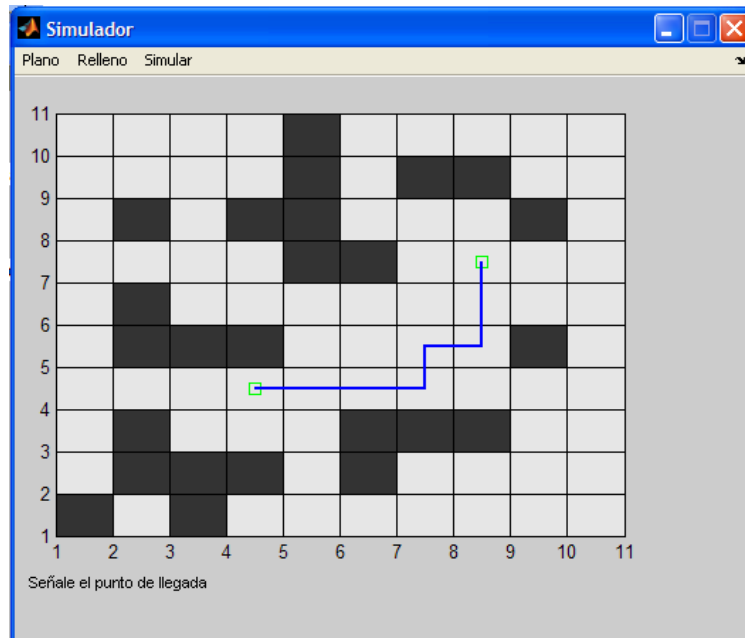
Baja densidad de obstáculos aleatorios: se toma la opción aleatorio en el simulador y se inicia el proceso.



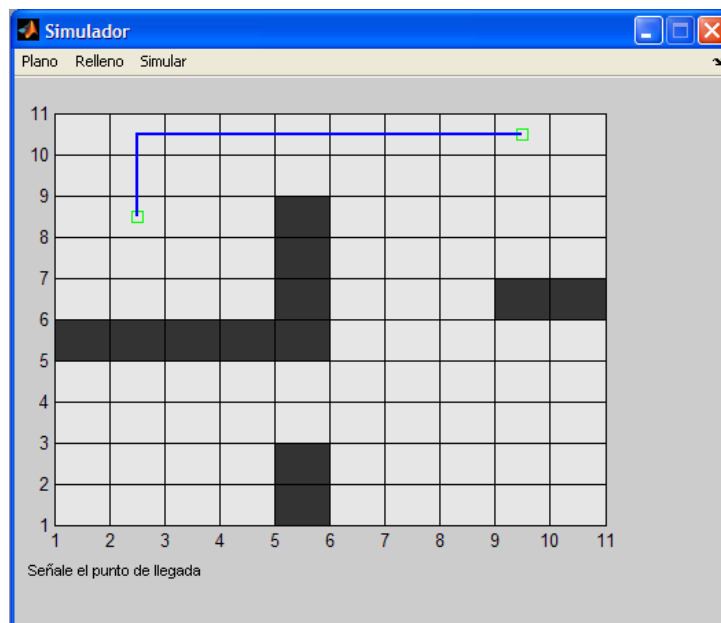
Media densidad de obstáculos aleatorios: Además de seleccionar la opción aleatorio en el simulador se añaden de manera manual algunos obstáculos al azar.



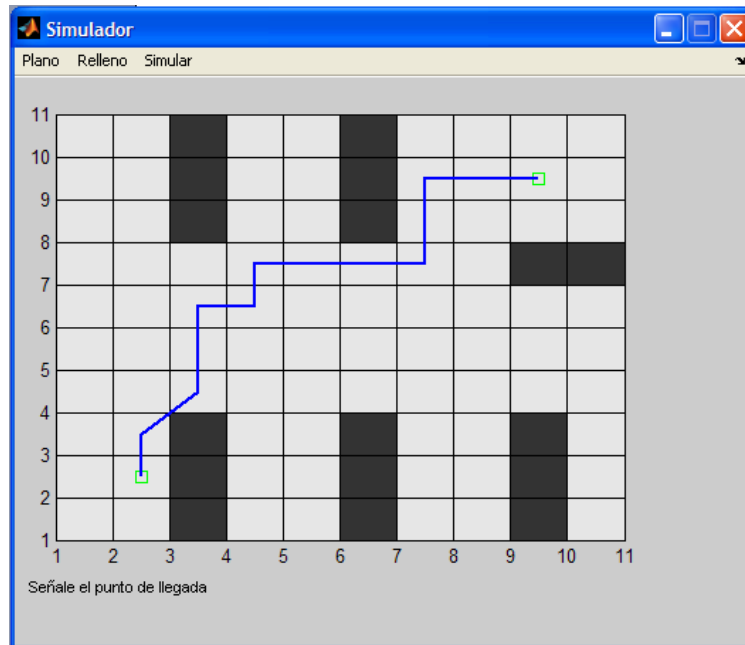
Alta densidad de obstáculos aleatorios: El número de obstáculos debe ser mucho mayor que las anteriores opciones.



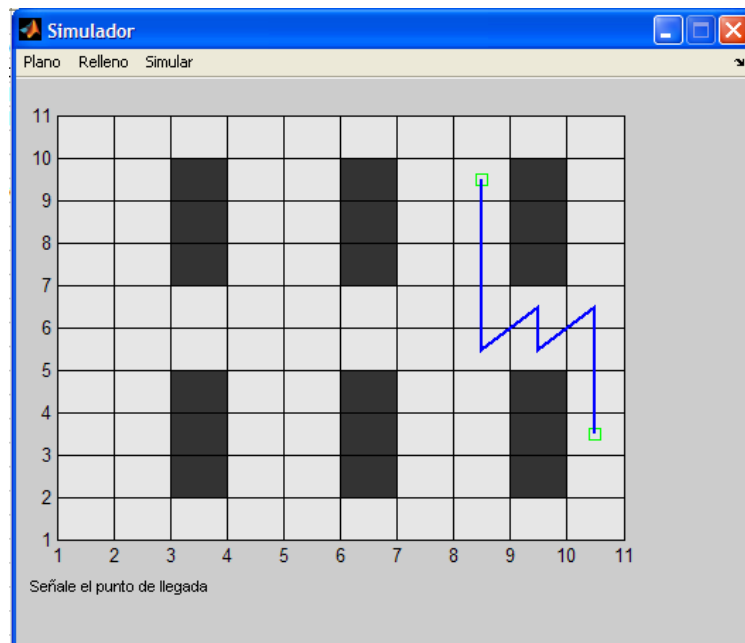
Alcobas: Se determina una distribución uniforme de los obstáculos simulando un pequeño apartamento.



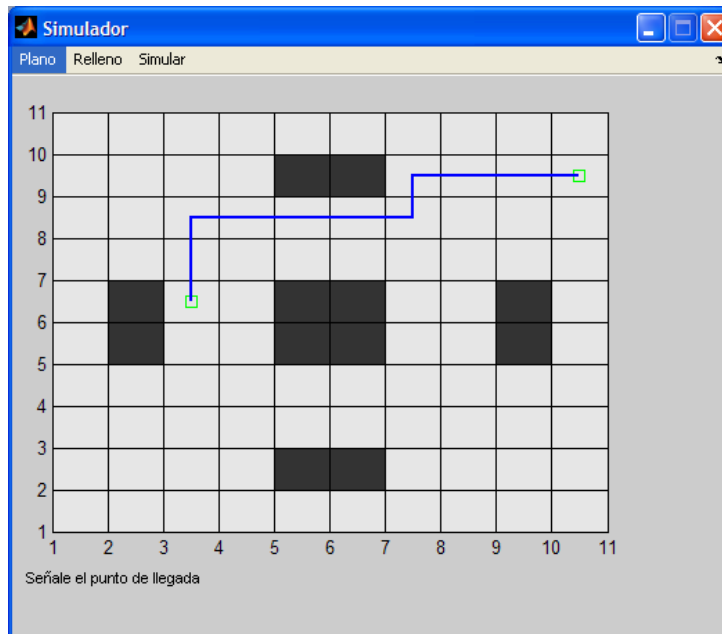
Parqueadero: Se organizan obstáculos en líneas rectas aparentando lugares de estacionamiento.



Centro comercial: La distribución rectangular de los obstáculos reubica hacia el interior del plano.



Parque: El obstáculo en el centro del plano representa la fuente de un parque.



El resultado para cada uno de los anteriores casos es condensado en la siguiente gráfica de resultados de desempeño.

