

**REPRESENTACIÓN VISUAL DE MODELOS ROBÓTICOS PARA
APLICACIONES DIDÁCTICAS**



**Autor:
ORLANDO GÓMEZ DURÁN**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
BUCARAMANGA
AGOSTO DE 2005**

**REPRESENTACIÓN VISUAL DE MODELOS ROBÓTICOS PARA
APLICACIONES DIDÁCTICAS**



**Autor:
ORLANDO GÓMEZ DURÁN**

TRABAJO DE GRADO

**Director:
Dr. Techn. ROBERTO MARTÍNEZ ÁNGEL**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
BUCARAMANGA
AGOSTO DE 2005**

TABLA DE CONTENIDO

	Pag.
1. INTRODUCCIÓN	1
2. GENERALIDADES DE ROBÓTICA.....	5
2.1 GENERALIDADES DE LOS ROBOTS	6
2.2 DEFINICIONES	6
2.2.1 Robot.....	7
2.2.2 Robot Manipulador.....	7
2.2.3 Eslabón.....	7
2.2.4 Articulación	7
2.2.5 Variable Articular	8
2.2.6 Grado de Libertad	8
2.2.7 Volumen de Trabajo.....	8
2.2.8 Configuraciones de los Robots.....	9
2.2.9 Sistema de Referencia	11
2.2.10 Efecto Final	11
2.2.11 Cinemática	11
2.2.12 Modelo Cinemático	12
2.2.13 Matriz de Transformación Homogénea	12
2.2.14 Matriz de Brazo.....	13
3. DESARROLLO DEL MODELO CINEMATICO DE UN ROBOT	14
3.1 REPRESENTACIÓN DE LA POSICIÓN	14
3.2 REPRESENTACIÓN DE LA ORIENTACIÓN	16

3.2.1 La Matriz de Rotación.....	16
3.2.2 Operaciones con Matrices de Rotación	18
3.3 UBICACIÓN EN EL ESPACIO TRIDIMENSIONAL.....	20
3.3.1 Matrices de Transformación Homogénea.....	20
3.3.2 Operaciones con Matrices de Transformación Homogénea.....	21
3.4 LA REPRESENTACIÓN DENAVIT-HARTENBERG	22
3.4.1 Algoritmo Denavit-Hartenberg.....	23
3.4.2 Significado de la Representación Denavit-Hartenberg	24
3.4.3 Matriz de Transformación D-H.....	25
3.5 LA CADENA CINEMÁTICA.....	26
3.5.1 Productoria de Matrices D-H.....	26
3.5.2 La Matriz de Brazo.....	27
3.6 EL PROBLEMA CINEMÁTICO DIRECTO.....	28
3.6.1 Solución del Problema Cinemático Directo	28
3.7 EL PROBLEMA CINEMÁTICO INVERSO	29
3.7.1 Solución del Problema Cinemático Inverso	29
4. HERRAMIENTAS SOFTWARE PARA LA REPRESENTACION DE ROBOTS	31
4.1 MATLAB EL SOFTWARE BASE	31
4.2 REPRESENTACIÓN DE LA ORIENTACIÓN Y LA POSICIÓN	32
4.2.1 Determinación del Origen de Coordenadas.....	33
4.2.2 Ubicación de Nuevos Sistemas	33
4.2.3 Visualización del Entorno Tridimensional Virtual	33
4.3 VARIACIÓN DEL SISTEMA DE REFERENCIA	34
4.3.1 Producto entre Matrices de Transformación Homogénea	34

4.3.2 La Transformada Inversa	35
4.4 MODELOS CINEMÁTICOS DE ROBOTS.....	36
4.4.1 La Representación Denavit-Hartenberg.....	36
4.4.2 Visualización del Modelo de Robot	37
4.4.3 Generación de Animaciones.....	37
4.5 LA CADENA CINEMÁTICA.....	38
4.5.1 La Matriz D-H.....	38
4.5.2 La Matriz Tn	39
4.6 CINEMÁTICA DIRECTA	39
4.6.1 Obtención de la Matriz Tn.....	40
4.7 CINEMÁTICA INVERSA	40
4.7.1 Obtención de los Valores de las Variables Articulares.....	40
4.8 ROBOWORKS 2.0 EL SOFTWARE DE VISUALIZACIÓN	41
4.9 INTRODUCCIÓN A ROBOWORKS 2.0.....	42
4.9.1 Recuperar un Ejemplo	42
4.9.2 Acondicionar el Entorno	43
4.9.3 Animación Mediante Teclado y Archivo de tipo .dat.....	44
4.10 ANIMACIÓN EN ROBOWORKS 2.0	44
4.10.1 Animación desde Matlab a RoboWorks 2.0	45
4.10.2 Identificación de Variables.....	45
4.10.3 Generación de Archivos .dat.....	45
4.11 MODELANDO EN ROBOWORKS 2.0	46
4.11.1 Insertar Elementos	46
4.11.2 Transformaciones	47

4.11.3 Opciones.....	49
5. PROCEDIMIENTO (Elaboración de los Gurús).....	51
5.1 INTRODUCCIÓN A ROBOWORKS 2.0 (gurú1)	52
5.2 REPRESENTACIÓN DE LA ORIENTACIÓN Y LA POSICIÓN (gurú2)	53
5.2.1 Visualización en el Entorno Matlab	53
5.2.2 Traslación.....	54
5.2.3 Rotación.....	56
5.2.4 Rotación y Traslación.....	57
5.2.5 Movimientos Combinados.....	58
5.3 VARIACIÓN DEL SISTEMA DE REFERENCIA (gurú3).....	59
5.3.1 Transferencia de la Referencia Mediante Multiplicación por MTH	60
5.3.2 Intercambio de Referencia Mediante Transformada inversa	60
5.3.3 Operaciones Consecutivas en la Misma Instrucción.....	61
5.4 MODELOS CINEMÁTICOS DE ROBOTS (gurú4)	61
5.4.1 Elementos de un Robot.....	61
5.4.2 Representación Denavit -Hartenberg	62
5.4.3 Visualización en el Entorno Matlab	62
5.4.4 Desarrollo del Algoritmo D-H para un Robot Torre	63
5.5 LA CADENA CINEMÁTICA (gurú5)	66
5.5.1 Incremento de Elementos de un Modelo.....	67
5.5.2 Obtención de la Matriz D-H	69
5.5.3 Referenciar la Matriz D-H al Origen.....	70
5.6 ANIMACIÓN EN ROBOWORKS 2.0 (gurú6)	70
5.6.1 Creación de los Modelos en Matlab	71

5.6.2 Creación de los modelos en RoboWorks 2.0	71
5.6.3 Creación del Programa creadat.m.....	72
5.6.4 Sincronización de los modelos Matlab-RoboWorks 2.0.....	72
5.6.5 Animación Desde el Teclado	73
5.6.6 Animación desde Matlab a RoboWorks 2.0	73
5.6.7 Creación de un Archivo tipo .dat para Animación.....	75
5.7 CINEMÁTICA DIRECTA (gurú7)	75
5.7.1 Significado de la Matriz de Brazo	76
5.7.2 Cinemática Directa de los Modelos Planteados.....	76
5.7.3 Animación de la Cinemática Directa.....	77
5.8 CINEMÁTICA INVERSA (gurú8)	77
5.8.1 La Matriz Objeto	77
5.8.2 Las Condiciones de Iteración	78
5.8.3 La Máscara de Restricciones	78
5.8.4 Cinemática Inversa de los Modelos Planteados	79
5.8.5 Animación de la Cinemática Inversa	79
5.9 MODELANDO EN ROBOWORKS 2.0 (gurú9).....	80
5.9.1 Iniciando la Construcción	80
5.9.2 Inserción de Elementos.....	80
5.9.3 Transformaciones	82
6. DISEÑO DE LAS GUÍAS DE LABORATORIO DE ROBÓTICA.....	84
6.1 PARTIR DEL DESARROLLO DEL ALUMNO.....	84
6.2 ASEGURAR LA CONSTRUCCIÓN DE APRENDIZAJES SIGNIFICATIVOS.	84

6.3 POSIBILITAR QUE LOS ALUMNOS REALICEN APRENDIZAJES SIGNIFICATIVOS POR SÍ SOLOS.....	85
6.4 PROCURAR QUE LOS ALUMNOS MODIFIQUEN SUS ESQUEMAS DE CONOCIMIENTO.....	85
6.5 ESTABLECER RELACIONES RICAS ENTRE EL NUEVO CONOCIMIENTO Y LOS ESQUEMAS DE CONOCIMIENTO YA EXISTENTES.....	86
7. CONCLUSIONES.....	87
8. RECOMENDACIONES.....	90
9. REFERENCIAS BIBLIOGRAFICAS.....	91

INDICE DE FIGURAS

	Pag.
Figura 1. Configuraciones de Robots.....	11
Figura 2. Diagrama de relación entre cinemática directa e inversa.	12
Figura 3. Representación de un punto P en el espacio.	15
Figura 4. Representación de un sistema OUVW con su origen en el punto P..	15
Figura 5. Representación del sistema OUVW rotado con respecto al sistema OXYZ.	16
Figura 6. Modelo de Robot ejes.	52
Figura 7. Representación en el espacio Matlab del sistema origen.	54
Figura 8. Representación de la traslación en el espacio Matlab.....	55
Figura 9. Representación de una rotación (rot z) en el espacio Matlab.....	57
Figura 10. Representación de una rotación y una traslación en el espacio Matlab.	58
Figura 11. Representación de movimientos combinados en el espacio Matlab.	59
Figura 12. Modelo de un Robot torre.	63
Figura 13. Representación del primer grado de libertad en el espacio de Matlab y en el espacio de RoboWorks.....	64
Figura 14. Representación del modelo con dos grados de libertad en el espacio Matlab y en el espacio RoboWorks.....	65

Figura 15. Representación del modelo con tres grados de libertad en le espacio Matlab y en le espacio RoboWorks.....	66
Figura 16. Representación del modelo con cuatro grados de libertad en el espacio de Matlab y en el espacio de RoboWorks.	68
Figura 17. Representación del modelo con cinco grados de libertad en el espacio de Matlab y en el espacio de RoboWorks.	69

INDICE DE TABLAS

Tabla 1. Representación Denavit-Hartenberg	24
Tabla 2. Representación D-H del grado de libertad.	64
Tabla 3. Representación D-H para dos grados de libertad.	64
Tabla 4. Representación D-H para tres grados de libertad.	65
Tabla 5. Representación D-H para cuatro grados de libertad.	67
Tabla 6. Representación D-H para cinco grados de libertad.....	68

LISTA DE ANEXOS

	Pag.
ANEXO A GUÍAS PROPUESTAS PREVIAMENTE POR ESTUDIANTES DE ROBÓTICA	94
ANEXO B GUÍAS DISEÑADAS PARA EL USO DE LA HERRAMIENTA ROBOTICA PROPUESTA CON MH Y RW.	103
ANEXO C. EL CONSTRUCTIVISMO, UN MODELO PEDAGÓGICO	176

RESUMEN

TÍTULO: REPRESENTACIÓN VISUAL DE MODELOS ROBÓTICOS PARA APLICACIONES DIDÁCTICAS*

AUTORES: GÓMEZ DURÁN, ORLANDO**

PALABRAS CLAVES: Robot, Robótica, Laboratorio de Robótica, Matriz de Transformación Homogénea, Algoritmo Denavit-Hartenberg, cinemática Directa, Cinemática Inversa, Animación de Robots, Simulación de Robots.

DESCRIPCIÓN:

En este documento se describe como se desarrolla una herramienta computacional basada en el programa Matlab, la toolbox Hemero y el programa RoboWorks 2.0, por medio de la cual se presenta el tema de la cinemática de Robots manipuladores; la herramienta está separada en sesiones interactivas de nivel de dificultad creciente, cada una de las cuales se acompaña por una guía de laboratorio.

Inicialmente se exponen las generalidades de la Robótica con las definiciones de los términos utilizados en el texto. El siguiente capítulo explica las herramientas matemáticas que se requieren para lograr el desarrollo cinemático de un Robot. En el capítulo cuatro se presentan las herramientas de software con los comandos y procedimientos para desarrollar el modelo cinemático de un Robot. En el capítulo siguiente se trata la metodología para el desarrollo de las sesiones interactivas (programas gurú) para la enseñanza de la Robótica, presentando ejemplos y paquetes de instrucciones completos con el fin de familiarizar al usuario con los conceptos de Robótica y las herramientas software utilizadas. Por último se presenta el modelo pedagógico basado en las teorías del constructivismo, que se utilizó para diseñar las prácticas de laboratorio. Las conclusiones y recomendaciones se describen en los capítulos siete y ocho.

Las prácticas de laboratorio se realizaron acorde con las sesiones interactivas y se basaron en el constructivismo como soporte para sustentar que el componente práctico de las experiencias es un factor fundamental en el aprendizaje como construcción de conocimiento.

* Trabajo de Grado.

** Facultad de Ingenierías Físico-mecánicas. Ingeniería Electrónica. Roberto Martínez Ángel.

ABSTRACT

TITLE: ROBOTIC MODEL'S VISUAL REPRESENTATION FOR DIDACTICAL APPLICATIONS.*

AUTHORS: GÓMEZ DURÁN, ORALANDO**

KEY WORDS: Robot, Robotics, Robotics Laboratory, Homogeneous Transform Matrix, Denavit-Hartenberg Algorithm, Directa Kinematics, Inverse Kinematics, Robots Animation, Robots Simulation.

DESCRIPTION:

In this document is described the developing of a computational tool based on program Matlab, the toolbox Hemero and program Roboworks 2.0 in order to present the theme manipulator Robot's kinematics; this tool appears separated in increasing difficult level interactive sessions, each one is accompanied by a laboratory guide.

Initially definitions of Robotic's terms used in this paper are exposed. The next chapter explains the mathematical tools required to achieve a Robot's kinematic development. The fourth chapter presents the software tools with their commands and proceedings required to develop a kinematic's Robot model. The next chapter deals with the methodology used in the developing of the interactive sessions (guru programs) designed to teaching Robotics, which presents examples and instructions packets intended to introduce the user to the Robotic's concepts and the software tools used in that process. The last chapter presents the pedagogical model based on constructivism theoretics, used in this project to design the laboratory practices. Conclusions and recommendations are described in chapters seven and eight.

The laboratory practices were made conformable to the interactive sessions and they were based on constructivism as a support to sustain that experiences' practical component is a fundamental factor in learning and knowledge construction.

* Work of Grade.

** Faculty of Engineering Physical-mechanics. Engineering Electronic. Roberto Martínez Ángel.

1. INTRODUCCIÓN

Se entiende por herramienta cualquier objeto ajeno al cuerpo y que sea utilizado como una extensión del mismo para realizar tareas que escapan a las capacidades de fuerza, precisión, tamaño, etc. de las extremidades naturales o que puedan poner en riesgo la salud e integridad del individuo. Las herramientas han sido un indicador del desarrollo cultural y tecnológico a lo largo de la evolución humana. El primer paso fue aprender a manejar exitosamente la mano, entonces con las manos se fabricaron herramientas, posteriormente se desarrollaron herramientas para fabricar herramientas más elaboradas, luego vino la revolución industrial en la cual se crearon las máquinas operadas por humanos para fabricar las herramientas de uso cotidiano, ahora estas máquinas en muchas fábricas están automatizándose para mejorar la producción y reducir el riesgo de los operarios. En este punto de la automatización es donde entran los Robots a realizar las tareas que otrora fueran realizadas por humanos y que debido a las características de dichas tareas exigen el desarrollo de dispositivos de control, inteligencia y aprendizaje totalmente artificiales.

La Robótica es la ciencia que envuelve todo lo referente a los Robots, aglutina conocimientos especializados de álgebra, física, electrónica, informática, diseño entre otros. Sin embargo, su característica más importante es servir de puente para comunicar la inteligencia artificial con el mundo real, sin importar el nivel de complejidad de la inteligencia artificial involucrada.

Modelar el mundo real de forma que el Robot lo entienda, es una actividad compleja que involucra manejo matricial del espacio tridimensional cuyas variables deben ser articuladas con las características morfológicas y las variables internas del Robot. La complejidad de las operaciones matemáticas involucradas se va incrementando con el número de articulaciones del Robot de forma que ya para un reducido número de ellas resulta imposible manejar mentalmente su significado y el efecto neto que el cambio de una variable

articular produce en el mundo real. Esto muestra que para llevar a cabo la enseñanza de la Robótica se requiere de herramientas que por un lado permitan optimizar y agilizar los procesos matemáticos y de otro lado permitan observar el significado de los mismos en un entorno visual tridimensional de fácil comprensión.

El desarrollo de un Robot implica la realización de pruebas que de utilizar modelos reales resultarían demasiado costosas. Por esta razón la simulación por computador es una alternativa que presenta enormes ventajas no solamente desde el punto de vista de costos sino también de su flexibilidad, rapidez de implementación, seguridad y por las opciones de representación del entorno que ofrece. Todas estas ventajas han sido integradas en el presente proyecto pensado como una herramienta para realizar la simulación de modelos de Robots y complementar la enseñanza de la Robótica usando software capaz de realizar las operaciones matemáticas y mostrar la simulación de modelos Robóticos en un entorno tridimensional virtual.

Como primera etapa en el desarrollo de esta herramienta se presenta un conjunto de conceptos y definiciones que deben quedar claros desde el principio, pues son los términos, cuyos significados son muy específicos, con los cuales se va a trabajar a lo largo de todo el proyecto.

Posteriormente en el tercer capítulo se presenta toda la teoría que explica el desarrollo cinemático de un Robot, lo cual exige la descripción matemática del espacio tridimensional, representación de la posición, representación de la orientación, la descripción y significado de la matriz de rotación y sus operaciones, la descripción de las matrices de transformación homogénea y su importancia en la determinación de sistemas de ejes coordenados, las operaciones con éstas que permiten describir el entorno tridimensional, el manejo de marcos de referencia y los movimientos en el espacio; luego se determina la aplicación de las matrices de transformación homogénea para la descripción de un Robot aplicando el algoritmo de Denavit-Hartenberg, la

solución de la cadena cinemática para manipular el modelo matemático del Robot finalmente el planteamiento de los problemas de la cinemática directa y la cinemática inversa.

En el capítulo siguiente se presentan los paquetes de software que fueron seleccionados para el desarrollo de la herramienta. Para el software base se describe su funcionamiento para realizar las diversas tareas que involucra el modelado matemático del espacio tridimensional, las operaciones con matrices de transformación homogénea, el desarrollo del modelo cinemático de un Robot siguiendo el algoritmo de Denavit-Hartenberg y la simulación del mismo en el entorno tridimensional virtual con la generación de animaciones para mostrar sus movimientos. Igualmente se presenta el software de visualización, se describe el proceso de recuperación de los ejemplos, la ejecución y generación de animaciones y por último la construcción de modelos Robóticos.

En el quinto capítulo se describen las etapas en las cuales quedo dividida la herramienta como un conjunto de sesiones interactivas y su funcionamiento mediante los programas gurú que se diseñaron para presentar el desarrollo cinemático de los Robots y cómo se utilizan los paquetes de software para entender estos conceptos, incluyendo comandos, sintaxis y las imágenes obtenidas en el proceso.

El capítulo siguiente describe las pautas que se siguieron con base en el modelo pedagógico planteado por el Constructivismo para diseñar un conjunto de practicas de laboratorio.

Complementario al desarrollo de estas sesiones, se presentan en el anexo A algunas de las guías que los estudiantes de Robótica propusieron en el pasado como complemento de la clase; en anexo B se encuentra el conjunto de guías planteadas en las que se proponen unas actividades para aplicar los conceptos de Robótica, experimentar con los paquetes software y comprobar las

situaciones planteadas en la herramienta desarrollada en este proyecto y en el anexo C se realiza una breve descripción del modelo pedagógico con base en el cual se diseñó el conjunto de guías de laboratorio.

2. GENERALIDADES DE ROBÓTICA

La Robótica es una compleja actividad que abarca varias ramas de la Ingeniería, ya que requiere de amplios conocimientos en Mecánica, Electrónica, Informática, Accionamiento, Control y Visualización. Su campo de acción incluye etapas desde el diseño de componentes mecánicos y eléctricos hasta tecnologías de sensores, sistemas de computación e inteligencia artificial

La Robótica es un área multidisciplinaria que debe resolver, principalmente, el siguiente conjunto de problemas para desarrollar un Robot real.

- Identificar las tareas que el Robot va a realizar.
- Describir el entorno de trabajo del Robot.
- Definir la morfología del Robot.
- Seleccionar los actuadores.
- Seleccionar los sensores y la realimentación.
- Desarrollar el sistema de control.

El paso desde la concepción de un Robot y el desarrollo de la matemática que esto implica hasta el objeto físico en el mundo real, es un proceso largo, tedioso y costoso desde varios puntos de vista, por eso es de enorme utilidad disponer de una herramienta que permita visualizar el producto final antes de iniciarse la construcción. La posibilidad de tener una vista previa de un Robot con sus características físicas en un entorno tridimensional, es el objetivo del presente proyecto, aunque para desarrollar la etapa de visualización se debe haber resuelto toda la matemática que soporta la solución del problema cinemático del Robot.

2.1 GENERALIDADES DE LOS ROBOTS

Los Robots están diseñados de modo que se puedan programar para llevar a cabo un conjunto de tareas que consiste básicamente en manipular piezas o herramientas y ubicarlas en un sitio preciso dentro su volumen de trabajo, para ser procesadas posteriormente. El programa puede consistir en ejecutar tales tareas específicas de manera repetitiva durante largos periodos de operación. Esto requiere un alto nivel de precisión, el cual puede lograrse si se dispone de un conocimiento exacto de la configuración física del Robot y de la posición y orientación que el efector final adquiere cuando los actuadores del Robot posicionan a cada uno de los eslabones; además, requiere conocer la ubicación de posibles obstáculos en del volumen de trabajo.

El conocimiento exacto de la configuración física de un Robot se puede lograr desarrollando un modelo cinemático del mismo y la utilización de herramientas matemáticas que permitan especificar información referente a las características físicas de los diferentes elementos del Robot y las posiciones relativas entre estos para determinar la relación de posición y orientación del efector final con respecto a la base del Robot. Para expresar de manera precisa y sencilla estas relaciones espaciales se presentará progresivamente la aplicación de tales herramientas matemáticas.

2.2 DEFINICIONES

Es conveniente abordar el proceso definiendo los términos que se van a utilizar para tener un concepto claro de los conceptos que se irán presentando y lograr un seguimiento más cómodo del desarrollo del proyecto.

2.2.1 Robot

Según el Robot Institute Of America, un Robot es un manipulador multifuncional reprogramable, diseñado para mover materiales, elementos, herramientas o dispositivos especiales mediante movimientos variables programables con el fin de desempeñar una variedad de tareas [18].

Los Robots pueden clasificarse en dos grandes grupos básicos. Los Robots manipuladores, que se presentan como un brazo Robot de base fija, y los Robots móviles, que se mueven por el espacio mediante ruedas, patas, hélices, etc. [20].

2.2.2 Robot Manipulador

Se puede describir un Robot manipulador como un conjunto de eslabones articulados que le permiten adoptar un número de posiciones diferentes para alcanzar ciertos objetivos en el espacio tridimensional. En este estudio se utiliza el termino Robot para referirse a los Robots manipuladores de base fija.

2.2.3 Eslabón

Eslabón es cada uno de los segmentos rígidos que componen un Robot.

2.2.4 Articulación

La unión entre dos eslabones con la particularidad de permitir cierto movimiento relativo entre ellos se llama articulación. En Robótica se usan principalmente dos tipos de articulaciones:

- Articulación rotacional.

En la cual los dos eslabones se encuentran atravesados por un eje imaginario que pasa justo por la unión y es alrededor de este eje que uno de los dos eslabones rota mientras el otro permanece fijo.

- Articulación prismática.

En la cual un eslabón yace sobre casi la totalidad de la longitud del otro o se encuentra dentro del otro; la dirección del desplazamiento relativo determina el eje de articulación, que permite que uno de los eslabones se deslice telescópicamente en su dirección mientras el otro eslabón permanece fijo.

2.2.5 Variable Articular

Variable articular hace referencia a la particularidad de la articulación de tomar cualquier valor, ya sea en grados o en distancia y este valor determinará la posición específica de un eslabón con respecto al otro.

2.2.6 Grado de Libertad

El conjunto articulación eslabón le otorga al Robot un grado de libertad, por lo cual el número de grados de libertad coincide con el número de articulaciones y es común describir un Robot usando el número de grados de libertad que éste posee.

2.2.7 Volumen de Trabajo

Es el espacio dentro del cual se mueve el efector final; está determinado por la configuración del Robot, el número de grados de libertad, el tamaño de los eslabones, el tipo de articulaciones y la forma como se combinan esas

articulaciones. Este espacio se puede dividir en dos espacios claramente diferenciables:

- Espacio alcanzable, o volumen que puede ser alcanzado por el efector final en por lo menos una orientación.
- Espacio de manipulación con destreza, o volumen dentro del cual el efector final puede adoptar todas las orientaciones posibles.

Un volumen de trabajo con destreza requiere que el Robot cuente por lo menos con seis grados de libertad, tres grados de libertad para posicionar el efector final en un punto determinado del espacio y tres grados para obtener la orientación adecuada. En algunas situaciones se llega a posiciones degeneradas, que consiste en la pérdida de un grado de libertad porque quedan alineadas dos articulaciones de manera que el movimiento de cualquiera de las dos tiene el mismo efecto en la posición del efector final; en otros casos se requiere introducir grados de libertad adicionales, que permiten obtener un volumen de trabajo con destreza aun con la presencia de obstáculos, en este caso se denominan grados de libertad redundantes. Algunas tareas requieren de menos de seis grados de libertad, por lo tanto los Robots diseñados específicamente para tales tareas presentan restricciones de movimiento cuando se desean implementar para una función diferente.

2.2.8 Configuraciones de los Robots

A continuación se presentan las configuraciones más comunes para la construcción de Robots las cuales se pueden observar en la figura 1.

2.2.8.1 Configuración Cartesiana

Es un Robot de tres grados de libertad cuyas articulaciones son prismáticas distribuidas perpendicularmente entre sí paralelas a los tres ejes principales del sistema de coordenadas.

2.2.8.2 Configuración Cilíndrica

Es un Robot con tres grados de libertad cuya primera articulación es de tipo rotacional y las dos restantes son prismáticas.

2.2.8.3 Configuración Esférica o Polar

Es un Robot con tres grados de libertad, conformados por dos primeras articulaciones rotacionales y una prismática.

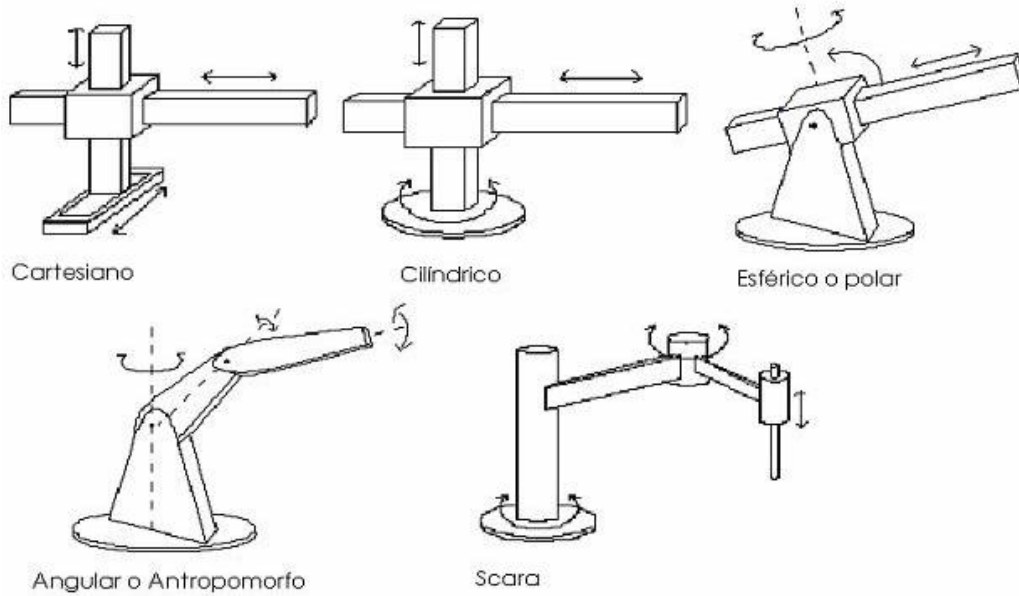
2.2.8.4 Configuración Angular o Antropomórfica

Es el grupo de Robots conformados solamente de articulaciones rotacionales, y sus movimientos semejan los del brazo humano.

2.2.8.5 Configuración Scara

Este Robot es una configuración especial con tres grados de libertad determinados por dos articulaciones rotacionales coplanares y una articulación prismática.

Figura 1. Configuraciones de Robots.



2.2.9 Sistema de Referencia

El extremo inicial del Robot se asume fijo a una base y está asociado a un sistema de ejes coordenados que va a ser el sistema de referencia cero.

2.2.10 Efecto Final

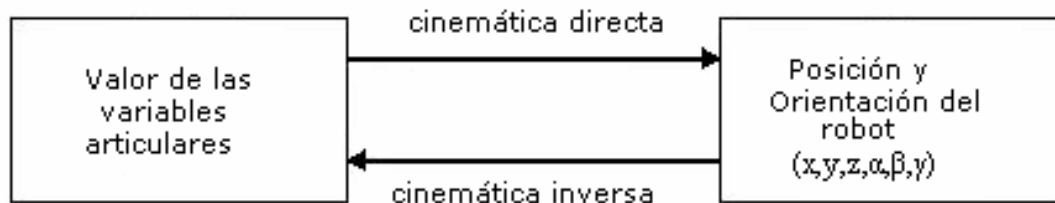
El extremo final del Robot es libre, se denomina efecto final y tiene asociado un sistema de coordenadas que representa la herramienta final del Robot, ya sea una pinza, un soldador, etc.

2.2.11 Cinemática

Cinemática es la parte de la física que estudia el movimiento de los cuerpos sin tener en cuenta su causa.

En Robótica el estudio cinemático se centra en obtener la relación entre la posición y orientación del efector final con los valores de las variables articulares como una función del tiempo. En la figura 2 se puede apreciar un diagrama que muestra esta relación.

Figura 2. Diagrama de relación entre cinemática directa e inversa.



BARRIENTOS, Antonio, Fundamentos de Robótica McGraw Hill. 1995

2.2.12 Modelo Cinemático

Para elaborar un modelo cinemático se aplica el algoritmo de Denavit-Hartenberg, mediante el cual es posible representar la totalidad del Robot utilizando matrices de transformación homogénea.

2.2.13 Matriz de Transformación Homogénea

Una matriz de transformación homogénea (*MTH*) es una matriz 4x4 que permite dar información de la ubicación, tanto de la posición como de la orientación, de un sistema de ejes coordenados **OUVW** con respecto a un sistema de ejes coordenados de referencia fijo **OXYZ**.

2.2.14 Matriz de Brazo

La matriz de brazo (matriz **T_n**) es la *MTH* que representa al efector final con respecto al origen del Robot.

3. DESARROLLO DEL MODELO CINEMATICO DE UN ROBOT

Para estudiar el problema cinemático es necesario desarrollar el modelo cinemático que permita tener conocimiento del número de eslabones, número y tipo de articulaciones, las dimensiones de los eslabones, las distancias entre las articulaciones y la posición relativa entre ellas; es decir, cómo se encuentra orientada cada articulación con respecto a la anterior.

Como ya se ha mencionado, ese proceso requiere de una cantidad de herramientas matemáticas de notable complejidad, las cuales se presentan a continuación.

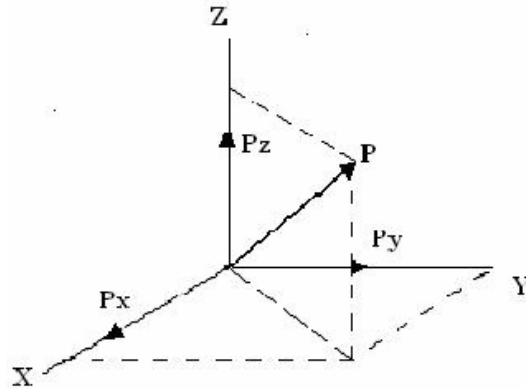
3.1 REPRESENTACIÓN DE LA POSICIÓN

Existen varios sistemas que permiten ubicar un punto en el espacio, coordenadas cartesianas, cilíndricas y esféricas. En el presente trabajo se va a utilizar el sistema de coordenadas cartesianas.

$$P = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} [1]$$

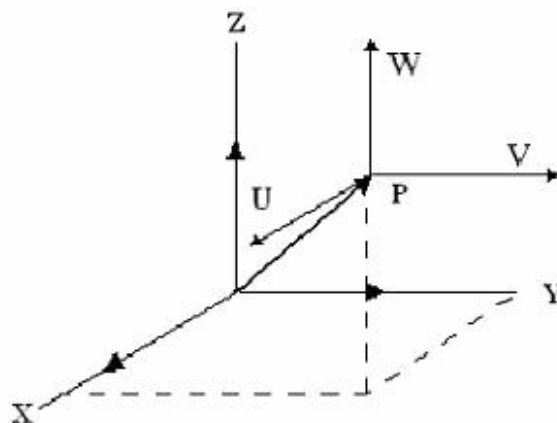
Este sistema de referencia **OXYZ** permite representar cualquier punto en el espacio mediante el vector de posición **P**, tal y como se aprecia en la ecuación 1, que consiste en un vector columna de tres elementos; $\mathbf{P} = (P_x, P_y, P_z)^T$. La figura 3 muestra como se ubica un punto **P** en el espacio utilizando las coordenadas cartesianas.

Figura 3. Representación de un punto P en el espacio.



Igualmente puede representarse por medio de otro sistema de coordenadas **OUVW** y la ubicación del punto P en el nuevo sistema de coordenadas depende de la ubicación del origen respecto al sistema de referencia **OXYZ** y de la orientación de sus ejes coordenados. La figura 4 representa un sistema cartesiano **OUVW** cuyo origen se encuentra ubicado en el punto **P** del sistema **OXYZ**.

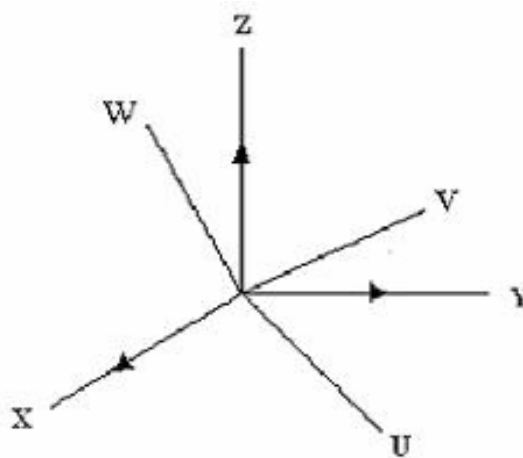
Figura 4. Representación de un sistema OUVW con su origen en el punto P.



3.2 REPRESENTACIÓN DE LA ORIENTACIÓN

Un punto queda totalmente descrito mediante las coordenadas tridimensionales, pero para un cuerpo sólido se requiere conocer como es la orientación de éste con respecto al sistema de referencia.

Figura 5. Representación del sistema OUVW rotado con respecto al sistema OXYZ.



Es conveniente entonces asociar al cuerpo un sistema de ejes coordenados **OUVW** que permita describirlo en el espacio; así, el problema de la orientación se refiere a la dirección de los vectores unitarios del sistema coordenado asociado al cuerpo en relación a los respectivos ejes coordenados del origen **OXYZ**. Como se observa en la figura 5, el sistema **OUVW** comparte el origen con el sistema **OXYZ**, pero sus vectores unitarios tienen una orientación diferente.

3.2.1 La Matriz de Rotación

Las matrices de rotación son el método más extendido para resolver el problema de la orientación, puesto que permite utilizar las herramientas de

álgebra matricial para indicar la rotación relativa entre los ejes principales de dos sistemas coordenados que comparten el origen.

Una matriz de rotación es una matriz \mathbf{R} 3x3 que se utiliza como operador, como muestra la ecuación 2, para realizar una transformación sobre un vector posición \mathbf{P} en un espacio tridimensional, permite convertir sus coordenadas expresadas en un sistema de ejes coordenados rotado \mathbf{OUVW} $\mathbf{P}_{uvw} = (P_u, P_v, P_w)^T$ (sistema solidario al objeto) a un sistema de referencia \mathbf{OXYZ} (sistema de coordenadas del origen) fijo en el espacio tridimensional $\mathbf{P}_{xyz} = (P_x, P_y, P_z)^T$ cuyos orígenes son comunes. Su orientación está determinada por ángulos de rotación con respecto a los ejes principales del sistema de referencia fijo.

$$\mathbf{P}_{xyz} = \mathbf{R}\mathbf{P}_{uvw} = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = R \begin{bmatrix} P_u \\ P_v \\ P_w \end{bmatrix} \quad [2]$$

La matriz de rotación se construye mediante operaciones con funciones trigonométricas del ángulo de rotación que indican cómo es la relación de orientación entre los dos sistemas; cualquier rotación se puede descomponer en rotaciones independientes alrededor de los ejes principales del sistema de referencia \mathbf{OXYZ} , donde el ángulo a corresponde a una rotación alrededor del eje \mathbf{OX} , el ángulo f alrededor del eje \mathbf{OY} y un ángulo γ alrededor del eje \mathbf{OZ} ; y las matrices de rotación respectivas, presentadas por las ecuaciones 3, 4 y 5, corresponden a $\mathbf{R}_{x,a}$, $\mathbf{R}_{y,f}$, $\mathbf{R}_{z,\gamma}$ [11].

$$R_{x,a} = \begin{vmatrix} 1 & 0 & 0 \\ 0 & \cos a & -\operatorname{sen} a \\ 0 & \operatorname{sen} a & \cos a \end{vmatrix} \quad [3]$$

$$R_{y,f} = \begin{bmatrix} \cos f & 0 & \operatorname{sen} f \\ 0 & 1 & 0 \\ -\operatorname{sen} f & 0 & \cos f \end{bmatrix} \quad [4]$$

$$R_{z,q} = \begin{bmatrix} \cos q & -\operatorname{sen} q & 0 \\ \operatorname{sen} q & \cos q & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad [5]$$

3.2.2 Operaciones con Matrices de Rotación

3.2.2.1 Producto

En muchas circunstancias el sistema de coordenadas solidario a un cuerpo experimenta más de una rotación con respecto a distintos ejes del sistema origen. El resultado final de tales rotaciones se obtiene multiplicando las matrices de rotación que representan cada una de las rotaciones individuales, obteniendo lo que se denomina una matriz de rotación compuesta.

El producto entre matrices no es una operación conmutativa, por lo cual las matrices de rotación deben ser multiplicadas en el mismo orden en el que ocurren las rotaciones del sistema en el espacio.

R1, en las ecuaciones 6 y 7, representa la matriz de rotación compuesta que resulta de una primera rotación alrededor del eje **OY**, seguida de una rotación alrededor del eje **OZ** y finaliza con una rotación alrededor del eje **OX** [11].

$$R1 = R_{y,f} R_{z,q} R_{x,a} = \begin{bmatrix} C_f & 0 & S_f \\ 0 & 1 & 0 \\ -S_f & 0 & C_f \end{bmatrix} \begin{bmatrix} C_q & -S_q & 0 \\ S_q & C_q & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_a & -S_a \\ 0 & S_a & C_a \end{bmatrix} \quad [6]$$

$$R1 = \begin{vmatrix} Cfcq & Sfsa - CfsqCa & CfsqSa + SfcCa \\ Sq & CqCa & -CqSa \\ -Sfcq & SfsqCa + Cfsa & CfcCa - SfsqSa \end{vmatrix} \quad [7]$$

R2, en las ecuaciones 8 y 9, es la matriz de rotación compuesta para giros alrededor de los mismos ejes, pero el orden es distinto **OX, OZ, OY** [11].

$$R2 = R_{x,a} R_{z,q} R_{y,f} = \begin{vmatrix} 1 & 0 & 0 \\ 0 & Ca & -Sa \\ 0 & Sa & Ca \end{vmatrix} \begin{vmatrix} Cq & -Sq & 0 \\ Sq & Cq & 0 \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} Cf & 0 & Sf \\ 0 & 1 & 0 \\ -Sf & 0 & Cf \end{vmatrix} \quad [8]$$

$$R2 = \begin{vmatrix} CqCf & -Sq & CqSf \\ CaSqCf + SaSf & CaCq & CaSqSf - SaCf \\ SaSqCf - CaSf & SaCq & SaSqSf + CaCf \end{vmatrix} \quad [9]$$

3.2.2.2 Inversa

En una matriz de rotación T como la mostrada en la ecuación 10, cada columna representa los principales ejes del sistema de coordenadas rotante con respecto al sistema de referencia fijo.

$$R = \begin{vmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{vmatrix} \quad [10]$$

La inversa de una matriz de rotación T^{-1} es equivalente a su transpuesta T^T ; los vectores fila de la matriz T representan los ejes principales del sistema de referencia fijo con respecto al sistema de coordenadas rotado [11]. La

ecuación 11 muestra los diferentes vectores unitarios que componen una matriz de rotación.

$$i_u = \begin{bmatrix} r_{11} \\ r_{21} \\ r_{31} \end{bmatrix} \quad i_v = \begin{bmatrix} r_{12} \\ r_{22} \\ r_{32} \end{bmatrix} \quad i_w = \begin{bmatrix} r_{13} \\ r_{23} \\ r_{33} \end{bmatrix} \quad i_x = \begin{bmatrix} r_{11} \\ r_{12} \\ r_{13} \end{bmatrix} \quad i_y = \begin{bmatrix} r_{21} \\ r_{22} \\ r_{23} \end{bmatrix} \quad i_z = \begin{bmatrix} r_{31} \\ r_{32} \\ r_{33} \end{bmatrix} \quad [11]$$

3.3 UBICACIÓN EN EL ESPACIO TRIDIMENSIONAL

Un cuerpo en el espacio tridimensional puede tener movimientos de rotación y traslación indistintamente, esto significa que el sistema coordenado asociado al cuerpo necesita de un vector de traslación para determinar su posición, y una matriz de rotación que indique su orientación; ambos respecto al sistema de referencia fijo. Estos dos conceptos quedan condensados en una matriz 4x4 llamada matriz de transformación homogénea (*MTH*).

3.3.1 Matrices de Transformación Homogénea

La *MTH* es una matriz 4 x 4 que representa la posición y la orientación de un sistema tridimensional móvil referenciado a un sistema tridimensional fijo.

$$T = \begin{bmatrix} R_{3x3} & P_{3x1} \\ F_{1x3} & W_{1x3} \end{bmatrix} = \begin{bmatrix} Rotación & Traslación \\ Perspectiva & Escalado \end{bmatrix} \quad [12]$$

Como se aprecia en la ecuación 12, puede decirse que una *MTH* está formada por cuatro submatrices de distinto tamaño:

Una submatriz \mathbf{R}_{3x3} que corresponde a una matriz de rotación.

Una submatriz \mathbf{P}_{3x1} que corresponde al vector de traslación.

Una submatriz \mathbf{F}_{1x3} que representa una transformación de perspectiva.

Una submatriz $\mathbf{W}_{1 \times 1}$ que corresponde a un escalado global [3].

En general se asumen las componentes de $\mathbf{F}_{1 \times 3}$ como nulas y el valor de $\mathbf{W}_{1 \times 1}$ la unidad, quedando entonces la *MTH* como muestra la ecuación 13.

$$T = \left| \begin{array}{cc|cc} R_{3 \times 3} & P_{3 \times 1} & \text{Rotación} & \text{Traslación} \\ 0 & 1 & 0 & 1 \end{array} \right| \quad [13]$$

3.3.2 Operaciones con Matrices de Transformación Homogénea

3.3.2.1 Producto

Es posible multiplicar varias *MTH* para presentar una secuencia de posiciones, lo que se puede traducir en una trayectoria paso a paso. El resultado se llama matriz de transformación homogénea compuesta, la cual depende totalmente del orden en que se realizan los productos.

3.3.2.2 Transformada Inversa

Las *MTH* están formadas por cuatro submatrices dos de las cuales deben permanecer constantes, la de perspectiva y la de escalado, por lo cual no se puede aplicar la transpuesta.

$$T = \left| \begin{array}{cc|cc} R_{3 \times 3} & P_{3 \times 1} & \text{Rotación} & \text{Traslación} \\ F_{1 \times 3} & W_{1 \times 1} & \text{Perspectiva} & \text{Escalado} \end{array} \right| = \left| \begin{array}{cccc} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{array} \right| \quad [14]$$

Para la matriz T mostrada en la ecuación 14 se tiene:

- El vector **n** representa las componentes del eje **OU** referenciado al sistema **OXYZ**.
- El vector **s** representa las componentes del eje **OV** referenciado al sistema **OXYZ**.
- El vector **a** representa las componentes del eje **OW** referenciado al sistema **OXYZ**.
- El vector **p** representa la posición del origen del sistema **OUVW** referenciado al sistema **OXYZ** [11].

Para expresar el sistema **OXYZ** referenciado al sistema **OUVW** se aplica la técnica de la transformada inversa a la matriz **T**, cuya operación se describe en la ecuación 15.

$$T^{-1} = \begin{vmatrix} n_x & n_y & n_z & -n^T p \\ s_x & s_y & s_z & -s^T p \\ a_x & a_y & a_z & -a^T p \\ 0 & 0 & 0 & 1 \end{vmatrix} = \begin{vmatrix} R_{3 \times 3}^T & \begin{matrix} -n^T p \\ -s^T p \\ -a^T p \end{matrix} \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad [15]$$

La matriz T-1 es la transformada inversa de la matriz T [11].

3.4 LA REPRESENTACIÓN DENAVIT-HARTENBERG

La representación Denavit-Hartenberg es un método que aplica las propiedades de las *MTH* para representar las relaciones de traslación y rotación entre elementos adyacentes de un Robot. En 1955 Denavit y Hartenberg propusieron un algoritmo para establecer de manera sistemática un sistema de coordenadas **S_i** ligado a cada eslabón *i* de una cadena articulada, esto permite pasar de un eslabón al siguiente mediante 4 transformaciones básicas que dependen exclusivamente de las características geométricas del eslabón [3].

3.4.1 Algoritmo Denavit-Hartenberg

3.4.1.1 Ubicación de los Ejes Solidarios a cada Articulación

Se hace coincidir el eje Z_i con el eje de la articulación en la dirección de desplazamiento para articulaciones prismáticas y en la dirección del eje de rotación para articulaciones rotacionales. El eje X_i se ubica en la dirección de la perpendicular común entre el eje Z_i y el eje Z_{i+1} . El eje Y_i se define completando un sistema dextrógiro.

3.4.1.2 Determinación de los Parámetros Denavit-Hartenberg

Los parámetros Denavit-Hartenberg definen las relaciones dimensionales entre los eslabones consecutivos y las variables articulares. En la tabla 1 se muestra la determinación y posición de los parámetros D-H obtenidos según Craig, para ser ingresados en la **MH**.

- θ_i es el ángulo de la variable articular para una articulación de rotación.
- d_i es la distancia de la variable articular para una articulación prismática.
- a_{i-1} es la distancia desde el eje de la articulación $i-1$ hasta el eje de la articulación i medida a lo largo de la línea perpendicular común a estos ejes. Cuando los ejes $i-1$ e i se cortan a_{i-1} vale cero.
- α_{i-1} es el ángulo entre el eje de la articulación $i-1$ y el eje de la articulación i ; medido como una rotación del eje $i-1$ alrededor de la perpendicular común hasta hacerlo coincidir con la dirección del eje i .
- d_i indica el tipo de articulación. Se le asigna valor cero para articulaciones rotacionales y valor uno para las articulaciones prismáticas [20].

Tabla 1. Representación Denavit-Hartenberg

ARTICULACIÓN i	a_{i-1}	a_{i-1}	θ_i	d_i	d_i
1	a_0 se asume cero	a_0 se asume cero	θ_1 será cero si la articulación es prismática.	d_1 será cero si la articulación es de rotación.	d_1 toma el valor de la articulación 1.
i	a_{i-1} el ángulo entre el eje Z_{i-1} y Z_i medido según la regla de la mano derecha con respecto a X_{i-1} .	a_{i-1} la distancia que separa los ejes Z_{i-1} y Z_i medido a lo largo de X_{i-1} , si Z_{i-1} y Z_i se cortan, $a_{i-1} = 0$	El ángulo θ_i se mide según la regla de la mano derecha entre el eje X_{i-1} y el eje X_i con respecto al eje Z_i .	La distancia d_i se mide entre el eje X_{i-1} y el eje X_i a lo largo del eje Z_i .	d_i vale cero (0) para articulaciones rotacionales y uno (1) para las prismáticas
La última articulación, n , si es de rotación, X_n está en línea con X_{n-1} cuando $\theta_n = 0$. Si es prismática, el origen de n está en la intersección de Z_n con X_{n-1} cuando $d_n = 0$	a_{n-1}	a_{n-1}	θ_n	d_n	d_n

3.4.2 Significado de la Representación Denavit-Hartenberg

La representación D-H de un Robot permite conocer el número y tipo de articulaciones que éste posee, las dimensiones de los eslabones, las relaciones angulares que existen entre las articulaciones consecutivas y la forma como están separadas entre sí. Permite también determinar las variables articulares del Robot. Estos datos se pueden presentar de manera organizada en una tabla para un rápido acceso y una fácil comprensión de las características físicas del Robot.

3.4.3 Matriz de Transformación D-H

Luego de tener la representación D-H se pueden realizar las siguientes transformaciones sucesivas que relacionan el sistema de coordenadas i -ésimo con el sistema de coordenadas $(i-1)$ -ésimo.

Estas operaciones se realizan mediante *MTH* básicas obteniéndose una *MTH* compuesta ${}^i\mathbf{A}_{i-1}$, llamada matriz de transformación D-H o en forma corta matriz **D-H** [11]. La ecuación 16 presenta el papel de cada uno de los parámetros D-H en la obtención de dicha matriz.

$$\begin{aligned}
 {}^{i-1}\mathbf{A}_i &= T_{Z,d} T_{Z,q} T_{X,a} T_{X,a} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \mathbf{q}_i & -\text{sen} \mathbf{q}_i & 0 & 0 \\ \text{sen} \mathbf{q}_i & \cos \mathbf{q}_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 & * \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \mathbf{a}_i & -\text{sen} \mathbf{a}_i & 0 \\ 0 & \text{sen} \mathbf{a}_i & \cos \mathbf{a}_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 & = \begin{bmatrix} \cos \mathbf{q}_i & -\cos \mathbf{a}_i \text{sen} \mathbf{q}_i & \text{sen} \mathbf{a}_i \text{sen} \mathbf{q}_i & a_i \cos \mathbf{q}_i \\ \text{sen} \mathbf{q}_i & \cos \mathbf{a}_i \cos \mathbf{q}_i & -\text{sen} \mathbf{a}_i \cos \mathbf{q}_i & a_i \text{sen} \mathbf{q}_i \\ 0 & \text{sen} \mathbf{a}_i & \cos \mathbf{a}_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [16]
 \end{aligned}$$

3.4.3.1 Interpretación de la Matriz de Transformación D-H

Esta matriz ${}^{i-1}\mathbf{A}_i$ está indicando que \mathbf{A} corresponde a una *MTH* que representa al sistema de ejes coordenados (i) asociado a la articulación (i) referenciado al sistema de ejes coordenados $(i-1)$ solidario a la articulación $(i-1)$. Esta matriz está determinada para un grado de libertad.

3.5 LA CADENA CINEMÁTICA

La cadena cinemática es el resultado de articular los eslabones para formar el brazo. La ubicación del efector final respecto del sistema de referencia de la base se obtiene del producto de las matrices que representan el sistema de ejes coordinados solidario con cada articulación.

3.5.1 Productoria de Matrices D-H

A cada grado de libertad (i) le corresponde una matriz **D-H** del tipo ${}^{i-1}\mathbf{A}_i$; en un Robot con n grados de libertad se tendrá un conjunto de n matrices **D-H** cada una de las cuales presenta al sistema de referencia asociado a la articulación (i) con respecto al sistema de referencia de la articulación anterior. Para la primera articulación su matriz **D-H** está referenciada al origen ${}^0\mathbf{A}_1$, la segunda está referenciada a la primera articulación ${}^1\mathbf{A}_2$ y así sucesivamente hasta llegar a la articulación n, cuya matriz **D-H** sería ${}^{n-1}\mathbf{A}_n$.

En el caso de matrices **D-H** el producto entre ellas se interpreta como un cambio de punto de referencia, presentándose una concatenación de sistemas referenciados consecutivamente.

$${}^0\mathbf{A}_1 \times {}^1\mathbf{A}_2 = {}^0\mathbf{A}_2 \quad [17]$$

En la ecuación 17 ${}^0\mathbf{A}_2$ es la matriz de transformación homogénea que representa al sistema coordinado asociado a la segunda articulación referenciado al origen.

3.5.2 La Matriz de Brazo

Para el caso de un Robot con n grados de libertad, la ecuación 18 muestra que las n matrices **D-H** se multiplican consecutivamente para obtener ${}^0\mathbf{T}_n$, la *MTH* del efector final referenciada al sistema de origen en la base del Robot:

$${}^0\mathbf{A}_1 * {}^1\mathbf{A}_2 * {}^2\mathbf{A}_3 * {}^3\mathbf{A}_4 * \dots * {}^{n-1}\mathbf{A}_n = {}^0\mathbf{A}_n = {}^0\mathbf{T}_n \quad [18]$$

La matriz ${}^0\mathbf{T}_n$ es la representación del extremo de la cadena cinemática y resume la totalidad de las operaciones que experimenta todo el brazo Robot para llevar un sistema de coordenadas desde el extremo libre hasta el origen. Por esta razón recibe el nombre de matriz de brazo o en forma corta matriz **Tn**, como se observa en la ecuación 19.

$${}^0T_n = \begin{bmatrix} x_n & y_n & z_n & p_n \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^0R_n & {}^0p_n \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} n & s & a & p \\ 0 & 0 & o & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad [19]$$

El significado de esta matriz **Tn** lo determinan los diferentes vectores columna que la componen [11].

- **n** = vector normal de la mano. Suponiendo una mano del tipo pinza con dedos paralelos, el vector **n** apunta en dirección perpendicular a los vectores **a** y **s** completando un sistema dextrógiro determinado por **sxa**.

- **s** = vector deslizamiento de la mano. Apunta en la dirección en la que se deslizan los dedos cuando se abre y se cierra la mano.
- **a** = vector de aproximación de la mano. Apunta en la dirección normal a la palma de la mano, es normal a la placa de montaje de los dedos y paralelo a la longitud de los mismos.
- **p** = vector posición de la mano. Apunta desde el origen del sistema de coordenadas de la base hasta el origen del sistema de coordenadas de la mano, el cual se ubica en el punto donde hacen contacto los dedos al cerrar la mano.

3.6 EL PROBLEMA CINEMÁTICO DIRECTO

El problema de la cinemática directa consiste en hallar la posición y orientación del efector final del Robot con respecto al sistema de referencia origen cuando se conocen los valores de las variables articulares y los parámetros geométricos de los elementos del Robot.

3.6.1 Solución del Problema Cinemático Directo

El problema de la cinemática directa se puede resumir como la obtención de la matriz **T_n**, cuando se conoce el valor de todas las variables articulares. Esta solución se obtiene resolviendo la ecuación 20, en donde se completa la cadena cinemática del Robot para los valores que se han asignado a las respectivas variables articulares.

$${}^0\mathbf{A}_1 * {}^1\mathbf{A}_2 * {}^2\mathbf{A}_3 * {}^3\mathbf{A}_4 * \dots * {}^{n-1}\mathbf{A}_n = {}^0\mathbf{T}_n \quad [20]$$

3.7 EL PROBLEMA CINEMÁTICO INVERSO

El problema de la cinemática inversa consiste en hallar los valores que se deben asignar a las variables articulares para posicionar el efector final del Robot de modo que su matriz T_n coincida con una MTH , llamada matriz T , que determina un sistema de coordenadas arbitrariamente ubicado en el espacio.

El proceso de hallar los valores de las variables articulares se convierte en resolver el sistema de ecuaciones determinado por la matriz dada. Para el efecto existen varias técnicas, algunos son métodos retamente geométricos para Robots de pocos grados de libertad, pero con el incremento del número de articulaciones se aumenta la complejidad de la solución, de manera que se hace necesario recurrir a métodos numéricos y técnicas computacionales iterativas.

3.7.1 Solución del Problema Cinemático Inverso

Este problema parte de la matriz T que está compuesta por los vectores n , o , a y p , y busca el valor para cada una de las variables articulares que posiciona el efector en las coordenadas dadas, es decir, resolver la ecuación $T_n = T$.

3.7.1.1 Método Geométrico

Este procedimiento es adecuado para Robots con pocos grados de libertad, o simplemente para solucionar la etapa de los tres primeros grados de libertad, que consiste en posicionar al efector final sin tener en cuenta su orientación. El método consiste en encontrar un número de relaciones geométricas que involucran las coordenadas del efector final, las coordenadas articulares y las dimensiones de los elementos [20].

3.7.1.2 Solución de la Matriz de Transformación Homogénea

Como se parte de una matriz \mathbf{T} , entonces se tendrían dieciséis incógnitas. La última fila de esta matriz es conocida, $[0 \ 0 \ 0 \ 1]$, por lo que se reduce el problema a doce incógnitas, tres de traslación, determinadas por la cuarta columna $[p_x \ p_y \ p_z]^T$ y nueve de rotación, de las cuales solo tres son independientes. Esto reduce el problema a seis ecuaciones con seis incógnitas no lineales e independientes con senos y cosenos [20].

3.7.1.3 Desacoplo Cinemático

Una técnica muy útil consiste en el desacoplo cinemático, que separa el proceso en dos etapas a completar.

La primera etapa consiste en resolver el problema de posicionar el extremo del Robot localizando el origen del sistema de coordenadas de la mano en las coordenadas determinadas por el vector \mathbf{p} de la matriz \mathbf{T} que se va a resolver, obteniéndose de esta forma los valores de las primeras $n-3$ variables articulares q_1, q_2, \dots, q_{n-3} .

La segunda etapa se concentra en orientar el extremo del Robot. Con base en los vectores \mathbf{n} , \mathbf{s} , \mathbf{a} y los resultados obtenidos de la etapa anterior se calculan los valores de las tres últimas variables articulares q_{n-2}, q_{n-1}, q_n .

Cabe mencionar que para Robots con menos de seis grados de libertad existen posiciones que sólo pueden ser alcanzadas mediante una única orientación del efector final debido, a restricciones de movimiento que determinan orientaciones prohibidas del mismo. Con seis o más grados de libertad el efector final puede orientarse de cualquier forma dentro del volumen de trabajo [20].

4. HERRAMIENTAS SOFTWARE PARA LA REPRESENTACION DE ROBOTS

El desarrollo de una herramienta para la representación visual de modelos Robóticos requiere de software que permita ejecutar todas las operaciones algebraicas contenidas en la construcción del modelo de un Robot y que además permita observar el Robot con sus características físicas y sus propiedades de movimiento en un entorno de simulación.

Existen programas que cumplen con tales requerimientos, pero hay factores de presupuesto, tiempo de entrenamiento, etc. que obligan a pensar en alternativas diferentes y a realizar adaptaciones con base en elementos que ya se tienen o que permitan reducir exigencias en diferentes aspectos.

Para el presente trabajo se consideró conveniente aprovechar los conocimientos de Matlab de que disponen todos los estudiantes de la Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones en combinación con la Toolbox Hemero y el libro de Ollero, que es el texto básico de la asignatura, potenciándolo con RoboWorks 2.0 que es un paquete de bajo costo y óptimas cualidades de visualización y animación.

4.1 MATLAB EL SOFTWARE BASE

El programa Matlab cuenta con varias cualidades que lo convierten en el soporte matemático ideal para el desarrollo de un proyecto de este tipo.

Los principales factores que determinaron su elección son:

- Facilidad para el desarrollo del álgebra de matrices.
- La existencia de las Toolbox Robotics y Hemero.
- El licenciamiento en la Escuela de Ingenierías Eléctrica Electrónica y de Telecomunicaciones.

- La experiencia que con él se tiene en el desarrollo de las prácticas de la asignatura Sistemas de Control Clásico y Digital.

La Toolbox Hemero (**MH**), de uso libre y diseñada específicamente como complemento del texto Robótica manipuladores y Robots móviles del Profesor A. Ollero Baturone, utiliza como plataforma las instrucciones de la Toolbox Robotics, desarrollada por Peter Corke, a partir de la cual se mejoran y se implementan muchas funciones que son de gran valor para el estudio de los modelos cinemáticos de Robots.

El trabajo realizado en **MH** presenta un conjunto de comandos de fácil comprensión y de una sintaxis lógica, en los cuales se compila una gran cantidad de operaciones y manipulaciones matemáticas que escapan a los objetivos de este proyecto pero sin las cuales no se habría podido llevar a cabo.

4.2 REPRESENTACIÓN DE LA ORIENTACIÓN Y LA POSICIÓN

La representación de la posición y orientación de cuerpos en el espacio demanda gran cantidad de cálculos matemáticos de sumo cuidado, **MH** trabaja directamente con *MTH*, de manera que no hay instrucciones para operar exclusivamente con los vectores de posición ni las matrices de Rotación.

La calidad de la visualización en esta etapa es suficiente con los resultados gráficos de **MH**, ya que permite ubicar sistemas coordenados en el espacio tridimensional virtual y asignar diferentes colores a los mismos para diferenciarlos entre sí.

4.2.1 Determinación del Origen de Coordenadas

Es importante para la ejecución de operaciones con *MTH* tener siempre presente la localización del origen, ya que es a éste punto al que **MH** referenciará todos los sistemas que se vayan ubicando en el espacio tridimensional virtual.

Para el desarrollo del proyecto es conveniente determinar este punto mediante un sistema de coordenadas llamado **SO** representado en la ecuación 21 por una *MTH* del mismo nombre.

$$S0 = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} [21]$$

4.2.2 Ubicación de Nuevos Sistemas

Para definir nuevos sistemas de coordenadas en el espacio tridimensional virtual de **MH**, cada nuevo origen se ubica mediante operaciones de rotación y traslación cuando se conocen las relaciones de movimiento respecto del sistema de coordenadas origen **SO** o cualquier otro sistema que ya esté identificado por el programa. También se puede simplemente introducir la *MTH* que lo representa.

4.2.3 Visualización del Entorno Tridimensional Virtual

La instrucción `frame` visualiza una ventana en la que se presentan los diferentes sistemas que se están trabajando, con la posibilidad de asignar un color específico a cada uno de ellos.

Esta ventana permite realizar acercamientos y rotaciones de la cámara para poder observar diferentes perspectivas de los sistemas que se están mostrando en el espacio tridimensional.

4.3 VARIACIÓN DEL SISTEMA DE REFERENCIA

Variar el sistema de referencia significa cambiar el marco de referencia con respecto al cual se está describiendo un cuerpo.

MH referencia todos los sistemas al origen, el cual fue previamente determinado con la matriz **S0**; pero en algunos casos resulta conveniente posicionar un objeto utilizando como referencia un sistema de ejes coordenados diferente al origen. El cambio de sistema de referencia requiere seguir una secuencia de operaciones algebraicas que toman el sistema de coordenadas solidario a tal objeto y lo transfiere al sistema de coordenadas respecto al cual se desea referenciar.

4.3.1 Producto entre Matrices de Transformación Homogénea

Un objeto en el espacio se puede referenciar a cualquier sistema de ejes coordenados, siempre y cuando sea éste un sistema dextrógiro de ejes ortonormales.

Un sistema de coordenadas **S1** cuya *MTH* referenciada al sistema **S0** está determinada por la matriz ${}^0\mathbf{A}_1$, sirve como sistema de referencia para un objeto con un sistema de coordenadas solidario **S2**. Su posición con respecto al sistema **S1** la determina la $MTH^1\mathbf{A}_2$. Es posible conocer la posición del sistema solidario al objeto **S2** con referencia al sistema origen **S0** mediante una operación producto matricial, y su resultado es la $MTH^0\mathbf{A}_2$, que indica el sistema **S2** referenciado al sistema **S0**.

$${}^0\mathbf{A}_2 = {}^0\mathbf{A}_1 * {}^1\mathbf{A}_2 \text{ [22]}$$

Esto significa que un objeto en el espacio de **MH** se puede referenciar a cualquier sistema de ejes coordenados siempre y cuando se conozca la relación que existe entre el sistema original de referencia del objeto y el sistema al cual se quiere referir.

4.3.2 La Transformada Inversa

Para las matrices de rotación se había visto que la transpuesta representa un cambio en el sistema de referencia con respecto al cual se están describiendo dos sistemas coordenados diferentes que comparten el punto origen. Pero cuando se trabaja con *MTH* se tiene que utilizar una herramienta matemática mas complicada, debido a la distribución de las submatrices que conforman una *MTH*.

En síntesis, la operación de transformada inversa permite invertir el punto de vista desde el cual se están observando dos sistemas, operación que se realiza en **MH** con la instrucción *trinv*.

Si la matriz ${}^0\mathbf{A}_1$ indica como esta el sistema **S1** con respecto al sistema **S0**, entonces la transformada inversa indica como está el sistema **S0** referenciado al sistema **S1**.

Transformada inversa de ${}^0\mathbf{A}_1 = ({}^0\mathbf{A}_1)^{-1} = {}^1\mathbf{A}_0 \text{ [23]}$

Indica como está el sistema **S0** referenciado al sistema **S1**.

4.4 MODELOS CINEMÁTICOS DE ROBOTS

Un Robot está constituido por un conjunto de eslabones que se encuentran unidos entre sí mediante articulaciones, cada una de las cuales le confiere al Robot un grado de libertad y determina una variable articular.

Para representar un Robot en el espacio de trabajo de **MH** es necesario realizar un modelo del mismo con base en las dimensiones de los eslabones, la relación espacial entre las articulaciones consecutivas y el tipo de articulaciones que lo conforman, para lo cual se sigue el algoritmo de Denavit-Hartenberg. De esta manera se reduce un Robot del espacio físico a una matriz de parámetros en el espacio matemático y se deja a **MH** toda la manipulación matemática que se requiere para generar el modelo en un entorno virtual.

4.4.1 La Representación Denavit-Hartenberg

Luego de aplicar el algoritmo D-H se tiene la matriz **D-H**, la cual es introducida en **MH**, generalmente bajo el nombre de **dh**.

La **MH** lee esta matriz fijándose en el parámetro s_i para determinar a qué tipo de articulación corresponde para luego identificar la variable articular e ignorar el valor correspondiente en dicha matriz. Esto se debe a que **MH** lee los valores articulares de otra matriz que debe ser especificada por el usuario y generalmente se llama matriz **q**.

La matriz **q** es una matriz fila que posee un número de columnas igual al número de variables articulares asignadas de manera ascendente de acuerdo con el orden de alejamiento de la articulación correspondiente con respecto del origen.

Los valores de estas variables deben ser coherentes con el tipo de articulación al que serán asignados.

La matriz \mathbf{q} puede ser también una matriz de animación, donde el movimiento se determina por una secuencia de posiciones, en cuyo caso el número de filas es igual al número de pasos a través de los cuales se genera el movimiento

4.4.2 Visualización del Modelo de Robot

Cuando se tiene definida la matriz \mathbf{dh} y se desea observar el modelo del Robot en el espacio tridimensional de \mathbf{MH} , es necesario asignar un valor coherente a cada una de las variables articulares que fueron definidas para la matriz \mathbf{dh} a través de la matriz \mathbf{q} , la cual por consiguiente debe tener tantas columnas como variables articulares tiene el Robot, y preferiblemente para esta etapa una sola fila.

La instrucción plotbot se ejecuta para generar una imagen del modelo del Robot posicionado en la configuración determinada por los valores de las variables articulares asignados en la matriz \mathbf{q} .

Esta instrucción presenta un dibujo en perspectiva del modelo del Robot donde los eslabones son representados por barras. La visualización se puede mejorar con algunas instrucciones que facilitan el entendimiento del modelo en observación.

4.4.3 Generación de Animaciones

\mathbf{MH} permite crear animaciones sencillas de los modelos de Robots, esta posibilidad facilita la identificación de los sistemas de referencia asociados a cada articulación y determinar si la ubicación de los mismos corresponde a la distribución espacial de las articulaciones del Robot real.

Como ya se dijo, \mathbf{MH} asigna a las variables articulares del modelo los valores que fueron determinados en la matriz \mathbf{q} , sin embargo, esta matriz puede estar

formada por varias filas, donde cada una de las mismas corresponde a una instantánea de las posiciones de cada articulación del Robot, de esta manera se pueden generar secuencias especiales de posiciones para observar el comportamiento del Robot en movimiento. Esta modalidad se aprovecha usando la opción *r* del comando plotbot, para mover la cámara alrededor del modelo y observarlo desde diferentes ángulos mientras se repite el movimiento programado.

4.5 LA CADENA CINEMÁTICA

Un brazo Robot está conformado por varios eslabones los cuales se unen entre sí mediante articulaciones, a cada una de las cuales se ha asociado un sistema de coordenadas cuyo eje **OZ** apunta en la dirección de la articulación según el algoritmo D-H.

En resumen, la cinemática de un Robot se dedica a estudiar la relación que existe entre los valores que toman las variables articulares y la posición del efector final referenciado al sistema origen, el cual se asume ubicado en el centro de la base del Robot.

La posición del efector final con respecto al origen corresponde a la matriz **T_n**, la cual se obtiene mediante la transferencia consecutiva de referencia de los sistemas asociados a cada articulación, partiendo desde el origen y pasando por todos y cada uno de los sistemas solidarios a cada articulación hasta terminar en el sistema solidario con el efector final.

4.5.1 La Matriz D-H

La matriz **D-H** es una *MTH* que indica cómo está en ese instante posicionado el sistema de coordenadas solidario a la articulación *i* con respecto al sistema de coordenadas asociado a la articulación *i-1*, esto significa que se necesita

conocer el valor que toma la variable articular en la posición actual de la articulación.

El comando linktrans de Matlab devuelve la matriz **D-H** de la articulación i , para lo cual necesita asignarle un valor numérico a la correspondiente variable articular.

4.5.2 La Matriz T_n

La matriz **T n** indica como está posicionado el sistema de coordenadas n solidario al efector final con respecto al sistema de referencia origen **S 0** .

Esta matriz es el resultado final del desarrollo de la cadena cinemática, la cual se calcula con la multiplicación consecutiva de cada una de las n matrices **D-H** de todas las articulaciones del Robot tal y como éste último quedó posicionado por los valores de las variables articulares que fueron asignados a cada articulación en cada instrucción linktrans ejecutada.

$$T_n = T_1 * T_2 * \dots * T_N \quad [24]$$

4.6 CINEMÁTICA DIRECTA

La cinemática directa estudia la posición y orientación del efector final como consecuencia de asignar un valor determinado a cada una de las variables articulares y de resolver la cadena cinemática con base en tales valores.

La cadena cinemática se puede completar de la manera antes descrita utilizando el comando linktrans, sin embargo existe una manera más ágil de obtenerla utilizando una sola instrucción que toma en cuenta la matriz **dh** y los valores articulares guardados en la matriz **q**.

4.6.1 Obtención de la Matriz T_n

En **MH** el comando `fkine` realiza toda la cadena cinemática y su resultado final es la *MTH* del efector final generalmente llamada matriz **T**. Para obtener la matriz **T**, el comando `fkine` necesita conocer los valores de las variables articulares, que deben haberse ingresado previamente a través de la matriz **q**.

La matriz **T** puede ser una posición expresada como una *MTH* cuando la matriz **q** tiene solo una fila, o puede ser una matriz tridimensional $4 \times 4 \times m$, que representa una trayectoria de *MTH* en la que cada fila m es una *MTH* que representa un paso en la secuencia de posiciones de la trayectoria cuando la matriz **q** está conformada por varias filas.

4.7 CINEMÁTICA INVERSA

La cinemática inversa se dedica a encontrar los valores que deben ser asignados a las variables articulares para posicionar el efector final del Robot de modo que su matriz **T_n** coincida con una *MTH* dada.

En la práctica se tiene ubicado en el volumen de trabajo del Robot un objeto x el cual tiene asociado un sistema de coordenadas referenciado al origen y representado por una *MTH*. Se busca que el Robot alcance este objeto con su efector final, para lo cual se debe permitir a cada una de las articulaciones un movimiento que debe ser calculado en este proceso.

4.7.1 Obtención de los Valores de las Variables Articulares

Para obtener los valores de las variables articulares en **MH**, se utiliza la instrucción `ikine`, la cual resuelve mediante métodos numéricos las ecuaciones que relacionan al efector final con el origen del sistema de coordenadas y

arroja como resultado la matriz \mathbf{q} con los valores de las variables articulares que corresponden a la posición determinada por la *MTH* dada.

Es necesario limitar el número de iteraciones que el programa va a ejecutar para encontrar los valores de las variables articulares e igualmente se requiere definir una tolerancia máxima entre la *MTH* correspondiente a las variables articulares solución y la *MTH* dada.

Cuando la *MTH* dada corresponde a una posición, la respuesta que arroja ikine será un vector fila con un único conjunto de valores de articulación para posicionar el Robot. Si la matriz dada es una trayectoria dada por una matriz de tres dimensiones entonces la respuesta corresponde a una matriz de animación.

Para Robots con menos de seis grados de libertad existen algunas restricciones, las cuales generarían no convergencia por parte de ikine; en esos casos es necesario introducir una matriz \mathbf{M} que actúe como una máscara.

4.8 ROBOWORKS 2.0 EL SOFTWARE DE VISUALIZACIÓN

La visualización es particularmente importante para el entendimiento de la Robótica y existen varios paquetes de software que ofrecen una visualización agradable de los modelos de Robots y su entorno en un ambiente tridimensional.

El programa RoboWorks 2.0 (**RW**) es una interfase gráfica que permite construir y visualizar con gran claridad los modelos de Robots y además ofrece varias opciones de animación de los mismos en un entorno tridimensional virtual, entre sus cualidades más destacadas se pueden citar las siguientes:

- Facilidad y agilidad para la construcción de modelos.
- Poco consumo de recursos de computador.

- Posibilidad de comunicación con Matlab, LabView y otros.
- Bajo costo.

4.9 INTRODUCCIÓN A ROBOWORKS 2.0

La interfase gráfica de **RW** consta de las siguientes partes:

Vista en 3D: En esta vista se puede ver en tres dimensiones el estado en que se encuentra un modelo tridimensional. Desde esta ventana no se puede modificar el modelo.

Vista de Árbol: En esta vista se crearán todos los modelos. Cada forma visible en la vista en 3D se verá referenciada en esta vista; y sólo desde esta ventana se pueden modificar sus parámetros.

Barra de Tareas: Ofrece las clásicas tareas de todo programa basado en Windows: Archivo, Editar, Ver, Ventana y Ayuda, además de la opción de Animación, propia de RoboWorks.

Barra de Íconos: En la barra de Íconos sólo se encuentran los íconos que pertenecen a las aplicaciones generales de Windows, estos son: Nuevo, Abrir, Guardar, Cortar, Copiar, Pegar, Acerca del RoboWorks y Ayuda.

4.9.1 Recuperar un Ejemplo

Los modelos de **RW** están guardados en la carpeta examples, en la cual viene incorporado un conjunto de ejemplos desarrollados con el fin de presentar las propiedades y los alcances del programa, algunos de los cuales tienen animaciones generadas por archivos tipo .dat

Hay dos animaciones creadas en lenguaje C que actualizan los valores de las variables articulares a través del protocolo TCP/IP, una opción muy interesante para posteriores trabajos de animación de prototipos en tiempo real.

Desde la barra de iconos o desde la barra de tareas se puede tener acceso al contenido de la carpeta examples y seleccionar los modelos allí guardados.

Cuando se abre un ejemplo, **RW** presenta el modelo en el espacio tridimensional en la vista 3D y presenta la vista de árbol del mismo.

4.9.2 Acondicionar el Entorno

En la vista 3D se tiene el modelo y su ambiente tridimensional. Con la opciones del menú edit es posible realizar cambios en esta visualización con el fin de mejorar la presentación y adecuarla al gusto e interés de la persona que está observando.

Opciones del menú edit:

1. Background: Asigna un color de fondo.
2. Autorotation: Pone el modelo a rotar automáticamente.
3. Grid: Activa o desactiva la visualización de la rejilla.
4. Coordinate frames: Presenta los ejes coordenados de todos los elementos.
5. Antialias: Mejora la calidad de la textura del modelo.
6. Interactive mode: Presenta varias opciones para la visualización.
7. Rendering: Presenta opciones para la textura del modelo.
8. Projection: Permite seleccionar un modo de perspectiva.
9. View: Permite seleccionar las vistas del modelo.
10. No Clear: No borra los pasos del modelo en la animación.

4.9.3 Animación Mediante Teclado y Archivo de tipo .dat

Todos los ejemplos de **RW** pueden moverse a través del teclado, pues en el diseño de cada modelo se ha asignado una tecla a cada articulación o conjunto de piezas articuladas, lo que permite realizar un estudio inicial de las características físicas del modelo, el número y tipo de articulaciones que lo componen y las posibles aplicaciones del mismo.

En la carpeta examples también se guardan unos archivos tipo .dat, que se pueden observar al seleccionar la opción Animation de la barra de tareas y escogiendo From File. Cada uno de estos archivos corresponde a una animación que fue generada previamente y generalmente tiene el mismo nombre o uno relacionado con el modelo que se tiene en pantalla y que se desea animar.

La ventana de animación permite controlar las propiedades de la misma, tales como velocidad, repetición continua, entre otras.

4.10 ANIMACIÓN EN ROBOWORKS 2.0

RW presenta una imagen en tres dimensiones de los modelos que se diseñan en **MH** y permite la reproducción de los movimientos que se generaron en el ambiente **MH**; con la ventaja de ofrecer opciones de manipulación de perspectiva, velocidad y demás opciones de video.

Para lograr una mejor comprensión de las características cinemáticas de los modelos de Robots creados en **RW**, estos pueden animarse mediante los archivos tipo .dat. Durante la ejecución de la animación también se puede monitorear el estado de las variables articulares, lo que facilita conocer la relación entre las variables articulares y la posición del efector final en particular y de todo el cuerpo del Robot en general, en el entorno tridimensional.

4.10.1 Animación desde Matlab a RoboWorks 2.0

Cuando se genera en **MH** la animación de los modelos de Robot diseñados, se ejecuta una instrucción que genera un archivo tipo .dat en el cual se guardan las diferentes posiciones asumidas por las articulaciones durante el proceso de la animación. Este archivo se guarda en la carpeta examples de **RW** y puede usarse para repetir la misma animación en **RW** con la consiguiente mejora en el aspecto físico del Robot y el control sobre la visualización del modelo en movimiento.

4.10.2 Identificación de Variables

En **RW**, cada articulación o conjunto de piezas móviles corresponde a una variable articular a la cual se le ha asignado un nombre, una tecla de comando y un valor de incremento.

RW tiene una ventana monitor la cual es de gran valor porque presenta una tabla en la que se encuentra organizada la información correspondiente a las articulaciones del Robot. Esta ventana permite reconocer el nombre de la variable articular asignada a cada articulación, la tecla de comando que la controla y en ella se tiene una actualización de los valores que van adquiriendo las variables articulares a medida que se mueve el modelo.

4.10.3 Generación de Archivos .dat

Cuando ya se conocen los nombres de las variables articulares, es posible generar una animación del Robot escribiendo un archivo tipo .dat.

Los archivos de animación tipo .dat se generan en el block de notas, teniendo en cuenta que la primera fila de dicho archivo contiene los nombres de las

variables que se identificaron en el modelo, separando cada una por un TAB, lo cual determina un numero de columnas claramente diferenciables. En las filas siguientes se asignan los valores que cada variable articular irá tomando durante la animación. En realidad este es un método de animación cuadro por cuadro, en el cual cada fotograma de la animación corresponde a una instantánea determinada por el valor que se asigne a las variables articulares en esa fila, y la sucesión de filas indica un cambio de posiciones que va a determinar el efecto de movimiento.

4.11 MODELANDO EN ROBOWORKS 2.0

RW es un programa de modelado tridimensional. Esto significa que además de los ejemplos existentes, se pueden crear nuevos modelos para realizar simulación del funcionamiento de los mismos mediante las animaciones.

El entorno tridimensional de **RW** está gobernado por coordenadas cartesianas, donde el plano horizontal esta formado por los ejes **OX** y **OZ**, y la coordenada vertical es el eje **OY**. Todos los objetos estarán referenciados a este sistema de coordenadas. Sin embargo el programa permite establecer nuevos sistemas de referencia, lo que hace más sencillo referenciar los elementos que componen un eslabón y facilita la configuración de los ejes de las articulaciones.

4.11.1 Insertar Elementos

La construcción del modelo se inicia en la vista de árbol, donde se van a insertar los diversos elementos y van a quedar organizados en estructuras jerárquicamente diferenciadas para tener un mejor entendimiento del modelo final.

La inserción de elementos se realiza mientras la vista de árbol se encuentre activada, seleccionando la opción edit de la barra de herramientas. Existen dos opciones para insertar elementos, insert child e insert after.

La opción insert child inserta el nuevo elemento en un nivel inferior de la posición de origen. La opción insert after inserta el nuevo elemento en un nivel paralelo a la posición de origen.

Dentro de los elementos a insertar existen tres tipos de elementos: Formas, transformaciones y otros.

- Shape: (Formas) Permite seleccionar una figura tridimensional básica. Una vez seleccionada la figura el programa le presenta una ventana en la que se asignan un nombre, dimensiones, orientación respecto del sistema de referencia, color y textura.
- Transformation: (Transformaciones) Las transformaciones son cambios que pueden ser rotaciones o traslaciones, hay dos tipos, las estáticas que se usan durante la construcción para mover el sistema de referencia y las dinámicas que configuran las articulaciones.
- Others: Define características de tipo grupal, es ideal para separar diferentes segmentos del modelo y permitir una construcción organizada separando los eslabones tanto por nombre como por color.

4.11.2 Transformaciones

La parte más exigente en la construcción de los modelos es el manejo de las transformaciones, en donde recae la conformación tridimensional del Robot, que permite a los diferentes sólidos de construcción ubicarse en la posición adecuada dentro del elemento que conforman. También determina las articulaciones definiendo el tipo de articulación, el eje de articulación, la tecla

de control, el nombre asignado a la variable articular, los límites o topes de articulación, etc.

- Transformaciones estáticas. Estas transformaciones permiten mover el sistema de referencia para hacer más sencillo el modelado y facilitar la animación. A cada transformación debe asignársele un nombre, el eje respecto del cual va a realizarse la transformación y el valor del cambio en grados para rotaciones teniendo en cuenta la regla de la mano derecha, y en centímetros para traslaciones teniendo en cuenta el signo de los ejes.

- Translation sirve para desplazar el punto de referencia una distancia determinada, ya sea en el eje x, el eje y, o el eje z exclusivamente. Así cualquier objeto que se inserte después de la transformación, se encontrará desplazado según lo especificado por la traslación.

- Rotation sirve para rotar un ángulo determinado alrededor de uno de los ejes del sistema de referencia que rige al cuerpo que se desea rotar. Esta rotación aplica a todos los cuerpos que se construyan a partir de ella.

- Transformation Start/Stop limita la transformación indicada a los objetos contenidos entre estas dos instrucciones. Transformation start actúa como un paréntesis que se abre separando el resto del modelo de los objetos que serán insertados posteriormente; Transformation stop cierra el paréntesis para que la transformación definida a continuación del transformation start no afecte el resto del modelo.

- Transformaciones dinámicas. En este espacio se configura cada articulación que posee el modelo. Estas transformaciones no tienen un valor específico, su magnitud es una variable que depende de las indicaciones dadas por la animación de los modelos, ya sea por medio de otro programa, el teclado o un archivo de datos generado por el usuario. Se debe determinar un conjunto de opciones de control que describen la articulación.

- Key selection asigna una tecla a la variable articular para activar la articulación mediante el teclado.
- Increment determina el paso de cambio en la articulación por cada vez que se presiona la tecla, tanto en grados para rotaciones como en cm. para desplazamientos.
- Tag name etiqueta la variable articular con un nombre, el cual se leerá en la tabla del archivo de tipo .dat para asignar el valor a la articulación.
- Límits determina los valores máximo y mínimo para la articulación y permite decidir si el programa informa cuando se ha alcanzado uno de ellos.

4.11.3 Opciones

En el proceso de construcción de modelos es posible que se requiera elementos, eslabones o conjuntos móviles que ya fueron elaborados en otro modelo. Para ahorrar tiempo y esfuerzo repitiendo el proceso de construcción, RoboWorks ofrece cortar, copiar y pegar como opciones que optimizan la construcción de los Robots.

- Cut. Cortar, cuando se desea retirar el elemento seleccionado para luego ubicarlo en otra sección del árbol. El elemento desaparece de la ubicación actual y queda en memoria mientras se busca la nueva ubicación donde será insertado ya sea mediante paste o paste child.
- Copy. Copia el elemento seleccionado en memoria para luego ser clonado mediante paste o paste child en una parte del árbol seleccionada. Durante el proceso el elemento no desaparece de su posición original porque no va a ser removido.

- Paste. Pega el elemento que se encuentra en memoria inmediatamente después del elemento que ha sido escogido como nueva locación del elemento cargado, pero lo pega en el mismo nivel jerárquico.

- Paste Child. Pega el elemento residente en memoria en un nivel inferior al del elemento que se ha elegido como redireccionamiento.

5. PROCEDIMIENTO (Elaboración de los Gurús)

Durante la trayectoria de la asignatura Robótica se ha visto la necesidad de una herramienta que permita complementar ciertos temas de la teoría con el desarrollo de actividades prácticas en las cuales se pueda visualizar el significado de los diferentes y complejos procedimientos matemáticos y su efecto en el mundo real.

La Toolbox Hemero-Matlab (**MH**) y el programa RoboWorks 2.0 (**RW**) ofrecen las capacidades necesarias para realizar una enseñanza basada en experiencias que complementen los diferentes tópicos presentes en la asignatura Robótica.

Con **MH** se desarrolló un conjunto de programas llamados gurú (**guía para el uso de la herramienta Robótica**), que a través de sesiones interactivas desarrollan y presentan ejemplos de las operaciones matemáticas involucradas en el estudio de la cinemática en Robots.

Con **RW** se construyeron modelos de las configuraciones de Robots más representativas con sus respectivas animaciones.

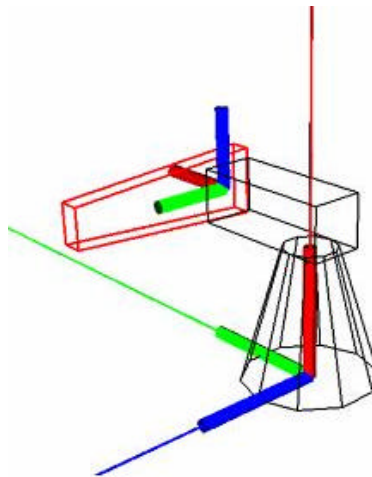
Finalmente se desarrollaron programas en **MH** para generar animaciones en el espacio de **MH** y que a su vez generan los archivos de animación para mostrar esas animaciones en el espacio de **RW**.

En este capítulo se presenta cada una de las etapas que componen la herramienta computacional con los respectivos comandos y sentencias para utilizarla.

5.1 INTRODUCCIÓN A ROBOWORKS 2.0 (gurú1)

El primer encuentro con este software brinda la oportunidad de experimentar con el programa, observando los ejemplos propuestos y sus animaciones. En la siguiente etapa se trabaja con un modelo de Robot llamado ejes que posee un grado de libertad. La visualización de sus elementos está desplegada en estructura alámbrica para mostrar los ejes coordenados asociados al origen y a la articulación.

Figura 6. Modelo de Robot ejes.



La opción de movimiento por teclado reproduce el comportamiento del eslabón móvil y del sistema de ejes solidario a la articulación, usando la tecla q . También se generó un archivo tipo `.dat` de animación que muestra la secuencia de movimientos que se requiere para hacer coincidir el sistema coordenado solidario al eslabón con el sistema origen ubicado en la base del Robot.

Se realiza una breve presentación de **MH**, en la cual se muestran las *MTH* que definen sistemas coordenados y la representación de los mismos en el espacio

tridimensional. Posteriormente se muestra el modelo en **MH** del Robot ejes con una animación del movimiento de su articulación.

5.2 REPRESENTACIÓN DE LA ORIENTACIÓN Y LA POSICIÓN (gurú2)

La **MH** tiene definido un origen de coordenadas en el espacio de trabajo respecto del cual se van a referenciar todos los sistemas que se vayan introduciendo; para referirse a él es necesario definirlo por medio de una *MTH* que para efectos del presente proyecto se llamará **SO**. Con esta matriz se van a desarrollar las operaciones y los procesos matemáticos posteriores. Cabe recordar que la **MH** trabaja toda el álgebra del espacio tridimensional con *MTH*, de manera que las operaciones de traslación y rotación de sistemas no pueden expresarse por medio de vectores de posición ni matrices de rotación como operadores.

5.2.1 Visualización en el Entorno Matlab

En el espacio tridimensional todo cuerpo tiene asociado un sistema de ejes coordenados el cual describe la orientación y la posición del mismo con respecto del origen **SO**.

El comando `frame` dibuja en el espacio tridimensional el sistema designado y permite definir ciertos atributos como el color y el espesor de las líneas correspondientes a sus vectores unitarios.

Sintaxis del comando `frame` para visualizar en flechas negras el sistema origen:

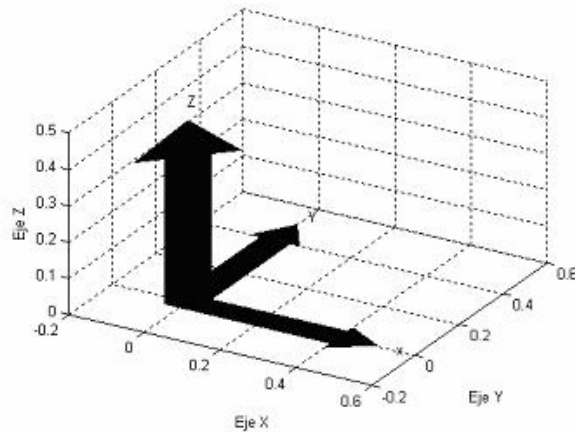
```
>>S0 = [1 0 0 0;0 1 0 0;0 0 1 0;0 0 0 1]
>>frame (S0, 'k', 1)
>>grid on
```

Esta instrucción grafica los ejes ortonormales del origen determinado por la *MTH SO*. La letra k indica el color para las flechas, el número 1 determina el ancho de las mismas. La instrucción grid on dibuja la reglilla.

Colores que se pueden asignar a las flechas:

'c' cyan	'm' magenta
'y' amarillo	'r' rojo
'g' verde	'b' azul
'w' blanco	'k' negro

Figura 7. Representación en el espacio Matlab del sistema origen.



5.2.2 Traslación

Trasladar un sistema en el espacio significa deslizarlo por una trayectoria sin producir cambios en la orientación de sus ejes coordenados. Esta maniobra se desarrolla matemáticamente multiplicando la *MTH* que representa el sistema a mover por una *MTH* que representa el movimiento a realizar.

Como el movimiento es un desplazamiento, la submatriz de rotación será una matriz de ceros y la submatriz de traslación tendrá como componentes las coordenadas del punto al cual se movió el sistema.

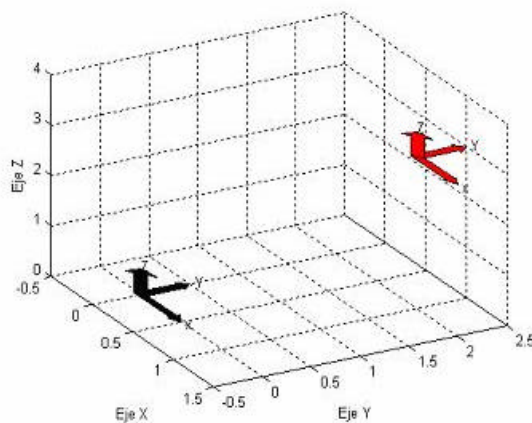
Para esta operación se utiliza la instrucción `transl` en la que se puede indicar las coordenadas del punto destino o indicar un vector cuyas componentes determinan las coordenadas de este punto; en ambos casos se debe multiplicar por la *MTH* que representa el sistema.

Sintaxis para el uso de la instrucción `transl`:

```
>>S2=transl(x, y, z)*S0
```

```
>>S2=transl(V)*S0
```

Figura 8. Representación de la traslación en el espacio Matlab.



En la primera forma (x,y,z) es el conjunto de coordenadas destino; en la segunda forma \mathbf{V} es un vector columna cuyas tres componentes corresponden a las coordenadas del mismo punto. $\mathbf{S2}$ es la *MTH* que representa el sistema movido con respecto al sistema origen.

5.2.3 Rotación

Rotar un sistema significa hacerlo girar alrededor de un eje. Esta operación resulta de multiplicar la *MTH* que representa el sistema por una *MTH* que determina la rotación, por lo cual su submatriz de rotación describe el giro aplicado al sistema.

La instrucción `rot` tiene un conjunto de posibilidades que facilita la especificación de las características de giro, las cuales agilizan algunos procesos. Estas posibilidades son la opción de rotar alrededor de uno de los ejes principales, `rotx`, `roty`, `rotz`, y rotar alrededor de un vector auxiliar, `rotvec1`. La rotación alrededor tanto de ejes principales como de vectores auxiliares se realiza definiendo el ángulo en radianes.

Para realizar cualquier rotación debe tenerse en cuenta que todo eje de giro pasa siempre por el origen del sistema de coordenadas **S0**. En esta sesión el sistema que gira será **S0**, de manera que el eje de rotación pasa directamente por el origen del sistema que gira, de forma que el resultado final de la operación no genera desplazamientos, ya que el sistema **S0** y el sistema rotado **S2** comparten el origen.

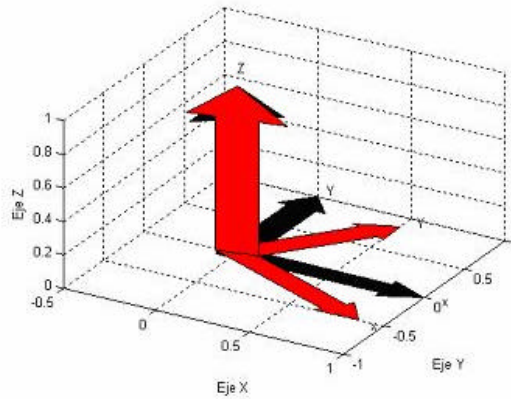
Sintaxis para el uso de las instrucciones `rot` y `rotvec1`:

```
>>S2=rotz(?)*S0
```

```
>>S2=rotvec1(V, ?)*S0
```

Donde `?` es el ángulo expresado en radianes que se desea rotar y **V** es el vector alrededor del cual se desea realizar la rotación.

Figura 9. Representación de una rotación (rot z) en el espacio Matlab.



5.2.4 Rotación y Traslación

Para realizar rotaciones y traslaciones las operaciones que resultan de las multiplicaciones consecutivas de las *MTH* involucradas dan como resultado una matriz cuyos elementos son un conjunto de operaciones algebraicas de senos y cosenos de los ángulos de rotación con los valores de traslación.

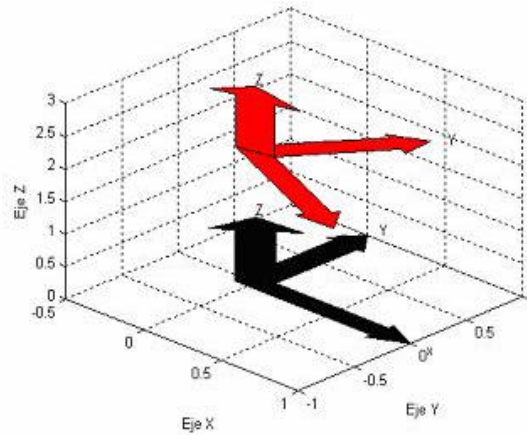
La complejidad en los elementos de una *MTH* aumenta en la medida en que involucra mayor número de operaciones para su determinación. Esto puede observarse fácilmente gracias a la capacidad de **MH** para desarrollar ecuaciones simbólicas. Esta capacidad permite también realizar el seguimiento de las variables que determinan los movimientos. Para el efecto se definen las variables en la instrucción `syms` y se antepone la instrucción `simple` a cada una de las operaciones que se va a ejecutar con las *MTH*.

Sintaxis de una operación sobre el sistema origen definida por una rotación un ángulo t alrededor del eje OZ y luego un desplazamiento una distancia d a lo largo del eje OZ :

```
>>syms t d real
```

```
>>S1=simple(rotz(t)*S0)
>>S2=simple(transl(0,0,d)*S1)
```

Figura 10. Representación de una rotación y una traslación en el espacio Matlab.



5.2.5 Movimientos Combinados

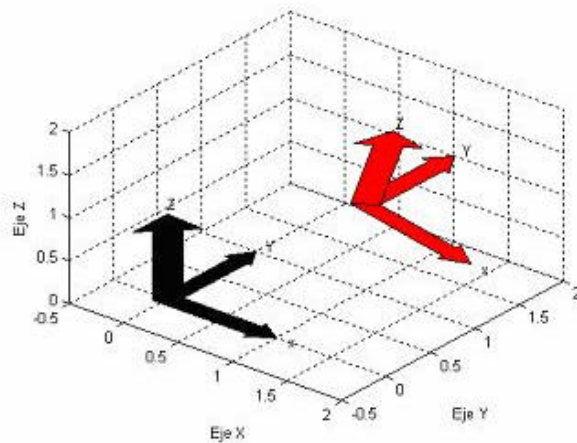
En la manipulación de sistemas de referencia dentro del espacio tridimensional, es muy común que éstos experimenten movimientos de rotación y traslación simultáneamente, además de rotaciones alrededor de sus propios ejes.

Cuando se desea rotar sobre sí mismo un sistema cuyo origen no coincide con S_0 , debe tenerse en cuenta que en **MH** todas las operaciones que involucren x , y , z están referenciadas al sistema de origen **S_0** y que los vectores utilizados en la rotación de sistemas pasan siempre por el origen. El procedimiento a utilizar en estos casos consiste en trasladar el origen del sistema que se quiere rotar haciéndolo coincidir con S_0 . Seguidamente se realiza la rotación y se desplaza nuevamente a su posición original. Si se desea realizar una operación combinada de rotación y traslación, debe realizarse primero la operación de rotación con el sistema móvil ubicado en el origen y luego la de traslación.

Esta sucesión de transformaciones se traduce como la multiplicación consecutiva de las *MTH* que representan el movimiento. El producto entre matrices no es una operación conmutativa, luego es necesario realizar cuidadosamente la separación de los movimientos y la posterior organización de la secuencia de los mismos.

El comando para la rotación alrededor de un vector auxiliar es *rotvec*, pero ese comando es sensible a las dimensiones del vector. Por tanto si el vector no es un vector unitario entonces los vectores internos de la submatriz de rotación dentro de la *MTH* que resulta no son vectores unitarios; por consiguiente esa matriz no cumple con las características de *MTH*. Para hacer de esta una *MTH*, se desarrolló la instrucción *rotvec1*, la cual normaliza siempre el vector de rotación como paso previo a la rotación.

Figura 11. Representación de movimientos combinados en el espacio Matlab.



5.3 VARIACIÓN DEL SISTEMA DE REFERENCIA (gurú3)

Una *MTH* es un operador que actúa modificando la posición y la orientación del sistema de referencia al cual es aplicado; se puede pensar de ella como un conjunto de instrucciones que describe el movimiento de un sistema móvil con

respecto a uno fijo. La ventaja de esta configuración radica en que se puede referenciar entre sí dos sistemas coordinados, cuando se conoce la *MTH* que los relaciona.

5.3.1 Transferencia de la Referencia Mediante Multiplicación por MTH

En el caso en que se conoce la posición de un sistema respecto al origen **S0** mediante la matriz **S10** y también se conoce la posición del sistema solidario con un objeto respecto al sistema definido anteriormente mediante la matriz **S21**, es posible conocer la posición del objeto con respecto al origen, definido por la matriz **S20**, aplicando la secuencia de productos matriciales del origen al primer sistema y de éste al objeto:

$$\mathbf{S10} = {}^0A_1 \text{ (Del origen al primer sistema) [25]}$$

$$\mathbf{S21} = {}^1A_2 \text{ (Del primer sistema al objeto) [26]}$$

$$\mathbf{S20} = {}^0A_2 = {}^0A_1 * {}^1A_2 \text{ (Del origen al objeto) [27]}$$

La **MH** resuelve el producto digitando la operación:

```
>>S20=S10*S20
```

5.3.2 Intercambio de Referencia Mediante Transformada inversa

La matriz **S10** del ejemplo anterior indica la posición de un sistema referenciado al sistema origen **S0**. Igualmente el sistema origen tiene una posición y una orientación con respecto a **S10** que vendría a ser **S01**. La operación de cambio de referencia se realiza mediante la instrucción *trinv*.

$$\mathbf{S10} = {}^0A_1 \text{ (Del origen al sistema) [28]}$$

$$\mathbf{S01} = \mathbf{S10}^{-1} = {}^1A_0 \text{ (Del sistema al origen) [29]}$$

Sintaxis para el uso de la instrucción trinv:

```
>>S01=trinv(S1)
```

5.3.3 Operaciones Consecutivas en la Misma Instrucción

Es posible realizar traslaciones y rotaciones en una misma instrucción. Si las rotaciones se realizan sobre un eje coordinado propio del sistema que gira las operaciones se indican como productos consecutivos colocando primero las traslaciones, luego las rotaciones y por último el sistema a rotar.

Sintaxis para posicionar un sistema en las coordenadas (1,-1,3) y rotarlo 180° alrededor del eje **OW** y -90° alrededor del eje **OU**.

```
>> S1=transl(1,-1,3)*rotz(pi)*rotx(-pi/2)*S0
```

5.4 MODELOS CINEMÁTICOS DE ROBOTS (gurú4)

Para lograr una comprensión cabal de la cinemática de Robots es necesario identificar las articulaciones y aplicar el algoritmo Denavit-Hartenberg. Estos dos procesos deben ser entendidos plenamente

5.4.1 Elementos de un Robot

Un Robot manipulador es un conjunto de elementos rígidos llamados eslabones, los cuales se mantienen unidos mediante articulaciones, cuyo número determina el número de grados de libertad del Robot.

La arquitectura del Robot depende de la forma como están relacionados los eslabones y el tipo de las articulaciones; la configuración del mismo se puede describir matemáticamente usando la metodología de Denavit y Hartenberg.

5.4.2 Representación Denavit-Hartenberg

El algoritmo Denavit y Hartenberg permite definir una serie de parámetros que se agrupan en la matriz **dh** la cual describe las relaciones espaciales intrínsecas de un Robot y sus variables articulares.

5.4.3 Visualización en el Entorno Matlab

Para obtener una imagen del Robot en el espacio **MH** es necesario asignar valores a las variables articulares para que el Robot adopte una posición. Estos valores son asignados en la matriz **q**.

El comando `plotbot` determina la visualización de los Robots en la **MH** graficando los eslabones como barras, las cuales unen el origen de los sistemas de coordenadas asignados a articulaciones consecutivas.

Sintaxis de la instrucción `plotbot`

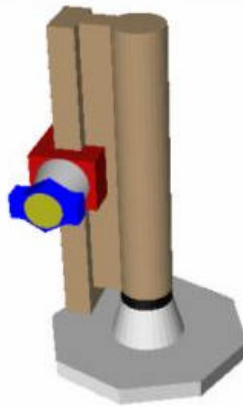
```
>>plotbot(dh,q,'d,f,w,l,r')
```

Cuyos parámetros son la matriz de los parámetros D-H, la matriz de los valores articulares **q** y entre las comillas se designan los accesorios para mejorar la visualización.

- La opción `d` se utiliza para una representación en 2 dimensiones.
- La opción `f` dibuja los ejes de los sistemas de coordenadas asociados a cada articulación excepto el sistema del efector final.

- La opción w visualiza el sistema de coordenadas del efector final.
- La opción l no borra el Robot en cada paso durante la animación.
- La opción r repite 50 veces la secuencia de movimientos.

Figura 12. Modelo de un Robot torre.



5.4.4 Desarrollo del Algoritmo D-H para un Robot Torre

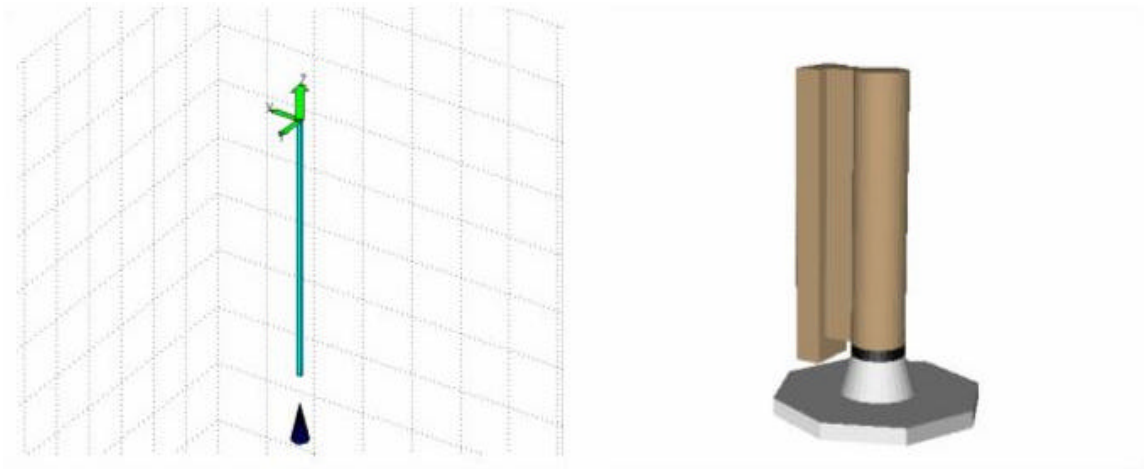
El desarrollo del modelo cinemático de un Robot como el mostrado en la figura 12 implica conocer primero sus dimensiones, para luego aplicar el algoritmo D-H.

La **MH** ubica el origen **SO** en la base del Robot. En el presente caso, el primer eslabón del Robot se encuentra a 5 cm. por encima del origen; ese primer segmento rota alrededor del eje Z_1 , el cual está alineado con el eje Z_0 . El parámetro d_1 vale 0.05 y el parámetro s_1 vale cero por ser una articulación de rotación, como los ejes del sistema 1 se encuentran alineados con los del sistema origen los demás parámetros son cero.

Tabla 2. Representación D-H del grado de libertad.

ARTICULACIÓN i	a_{i-1}	a_{i-1}	θ_i	d_i	s_i
1	0	0	0	0.05	0

Figura 13. Representación del primer grado de libertad en el espacio de Matlab y en el espacio de RoboWorks.

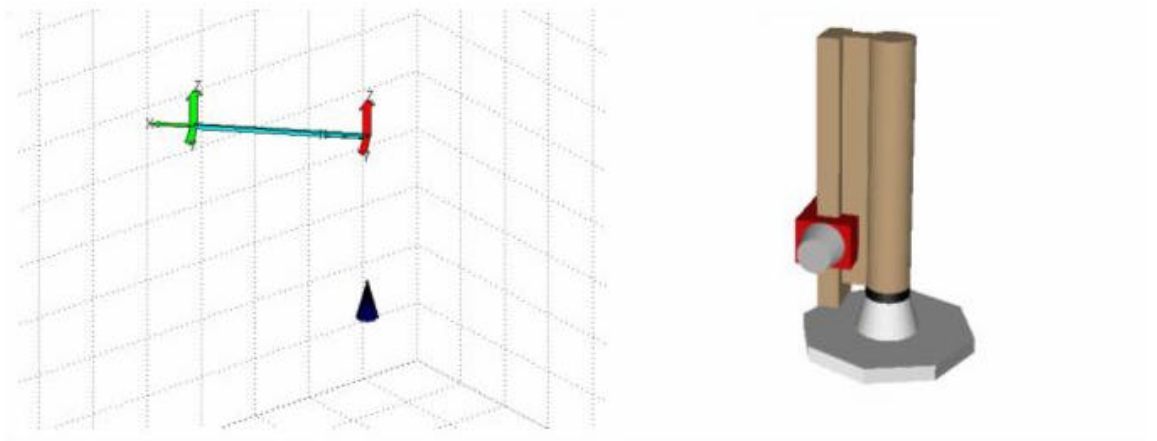


La segunda articulación desliza el segundo eslabón a lo largo de un eje paralelo a Z_0 y está desplazado 5 cm. en el sentido del eje X_1 ; por ser una articulación prismática $a_1=0.05$ y $s_2= 1$, los demás parámetros valen cero porque el sistema solidario con la segunda articulación tiene la misma orientación de ejes coordenados que el primero.

Tabla 3. Representación D-H para dos grados de libertad.

ARTICULACIÓN i	a_{i-1}	a_{i-1}	θ_i	d_i	s_i
1	0	0	0	0.05	0
2	0	0.05	0	0	1

Figura 14. Representación del modelo con dos grados de libertad en el espacio Matlab y en el espacio RoboWorks.

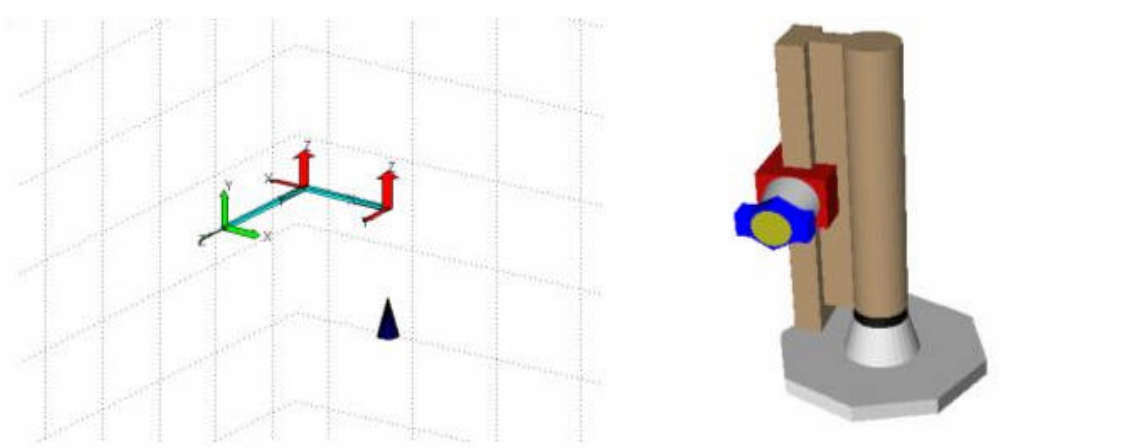


La tercera articulación del Robot torre es de rotación y gira el tercer eslabón alrededor de un eje perpendicular al eje Z_0 , por consiguiente s_3 vale cero. Como los ejes Z_0 , Z_1 y Z_2 son paralelos, el eje de articulación Z_3 está rotado 90° en sentido negativo respecto de X_2 por lo que a_2 vale $-\pi/2$; esta articulación está siempre a la misma altura que la segunda articulación, pero se encuentra desplazada 7 cm. sobre el eje Y_2 ; entonces d_3 vale 0.07.

Tabla 4. Representación D-H para tres grados de libertad.

ARTICULACIÓN i	a_{i-1}	a_{i-1}	θ_i	d_i	s_i
1	0	0	0	0.05	0
2	0	0.05	0	0	1
3	$-\pi/2$	0	0	0.07	1

Figura 15. Representación del modelo con tres grados de libertad en el espacio Matlab y en el espacio RoboWorks.



Estas distancias representan el espacio que debe dejarse entre los elementos que componen los diferentes eslabones para alojar los componentes físicos tales como actuadores, engranajes, correas, entre otros, cuyo volumen debe tenerse en cuenta para la construcción.

5.5 LA CADENA CINEMÁTICA (gurú5)

La importancia del algoritmo D-H radica en que permite referenciar el sistema solidario con cada articulación al sistema de la articulación inmediatamente anterior; además, con un poco de procesamiento de las *MTH* es posible referenciar cada articulación directamente con el origen, para tal fin se desarrolla una cadena cinemática que inicia en el origen del Robot y termina en la articulación determinada o en el efector final.

En el desarrollo de modelos de Robot, es frecuente adicionar nuevos eslabones con nuevos grados de libertad a modelos ya existentes, este proceso es relativamente sencillo en la medida en que se tenga destreza para manipular la *MTH* y aplicar el algoritmo D-H.

Asociado a la arquitectura se encuentra un conjunto de características propias de cada configuración; tales características incluyen alcance, volumen de trabajo, destreza, restricciones, redundancias y posiciones degeneradas.

5.5.1 Incremento de Elementos de un Modelo

Se va a incrementar en dos grados de libertad el modelo de Robot torre para obtener una serie de posiciones que muestren algunos estados característicos de un Robot.

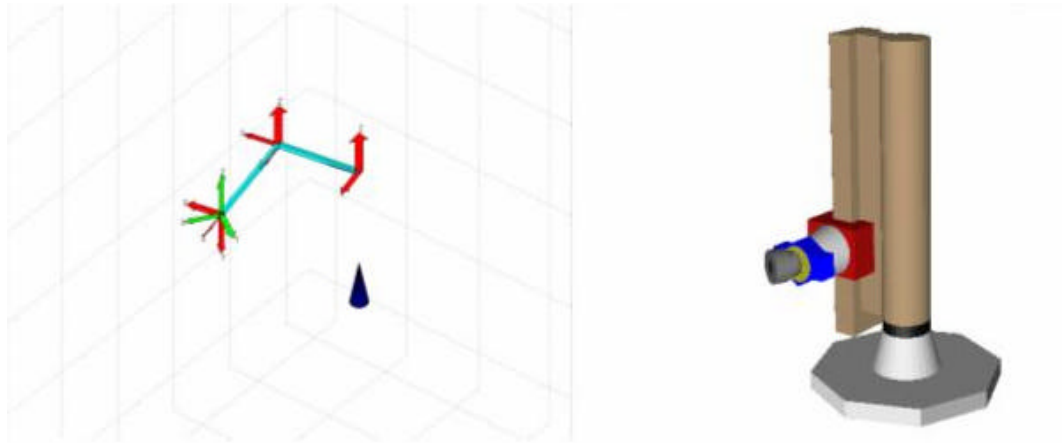
En primer lugar se adicionó una articulación de rotación, con su eje perpendicular al eje de articulación anterior. La matriz **dh** se incrementa con un nuevo grado de libertad.

En este punto se presenta una deficiencia de la **MH** para asignar los parámetros D-H. Estas dos articulaciones se pueden acoplar únicamente si los sistemas de coordenados asociados a ambas comparten el origen. Por esta razón los demás parámetros para este grado de libertad tienen un valor de cero.

Tabla 5. Representación D-H para cuatro grados de libertad.

ARTICULACIÓN i	a_{i-1}	a_{i-1}	θ_i	d_i	s_i
1	0	0	0	0.05	0
2	0	0.05	0	0	1
3	$-\pi/2$	0	0	0.07	1
4	$\pi/2$	0	0	0	0

Figura 16. Representación del modelo con cuatro grados de libertad en el espacio de Matlab y en el espacio de RoboWorks.



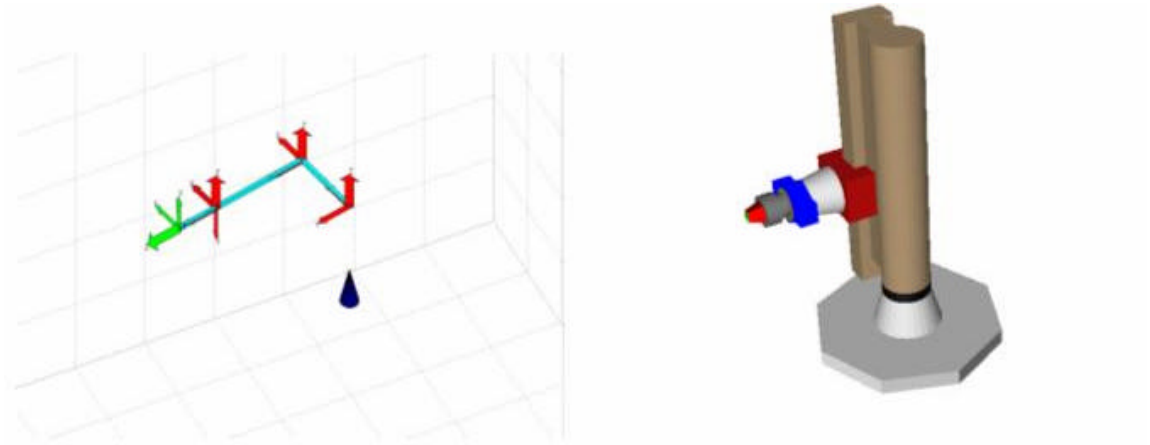
El último grado de libertad se escoge tal que el eje de articulación determine una rotación alrededor del eje de aproximación del efector final. El eje de esta articulación resulta entonces perpendicular al eje de la articulación anterior. Esta configuración permite al efector final un acceso más cómodo a la geometría del objeto a tomar, cuando el efector final correspondiera a unas pinzas.

Asumiendo que el actuador de la parte móvil tiene una longitud de 3 cm. la matriz dh será:

Tabla 6. Representación D-H para cinco grados de libertad.

ARTICULACIÓN i	a_{i-1}	a_{i-1}	θ_i	d_i	s_i
1	0	0	0	0.05	0
2	0	0.05	0	0	1
3	$-\pi/2$	0	0	0.07	1
4	$\pi/2$	0	0	0	0
5	$-\pi/2$	0	0	0.03	0

Figura 17. Representación del modelo con cinco grados de libertad en el espacio de Matlab y en el espacio de RoboWorks.



5.5.2 Obtención de la Matriz D-H

Para hallar la matriz **D-H** de la articulación i se requiere que la variable articular i tenga un valor, es decir que se conozca la posición relativa entre los dos eslabones involucrados.

El comando `linktrans` se especifica para cada articulación con base en el valor de su variable articular y la matriz **dh**, para encontrar la matriz **D-H**.

Sintaxis para el uso del comando `linktrans` en la articulación i :

```
>>Ti = linktrans(dh(:,i),ti)
```

Donde `dh` se refiere a la matriz de los parámetros D-H, el término i entre paréntesis es el número de articulación que se está procesando y t_i representa el valor que se le asigna a la variable articular.

5.5.3 Referenciar la Matriz D-H al Origen

El comando linktrans devuelve la matriz **D-H** que muestra la relación entre dos articulaciones consecutivas, esto quiere decir que cuando i vale uno (1) la matriz **D-H** obtenida corresponde a ${}^0\mathbf{A}_1$. Cuando i vale dos (2) se obtiene la matriz que representa ${}^1\mathbf{A}_2$. Aplicando las técnicas de variación del sistema de referencia se puede expresar el sistema de la segunda articulación referenciado al origen.

$${}^0\mathbf{A}_1 * {}^1\mathbf{A}_2 = {}^0\mathbf{A}_2 \quad [30]$$

En el lenguaje de la **MH** este procedimiento se obtiene ejecutando la siguiente cadena de instrucciones:

```
>>T1=linktrans(dh(:,1),t1)
>>T2=linktrans(dh(:,2),t2)
>>S20=T1*T2
```

Cuando se busca hallar la matriz **Tn** para un Robot con N grados de libertad, hay que hallar las N matrices **D-H**, desde **T1** hasta **TN** y la matriz **Tn** se obtiene como el producto de las N matrices **D-H**.

```
>>Tn=T1*T2*...*TN
```

5.6 ANIMACIÓN EN ROBOWORKS 2.0 (gurú6)

RW presenta una composición en 3-D de los modelos que se diseñan en **MH**, que permite la reproducción de los movimientos que ésta genera, con la posibilidad de mejorar la perspectiva y monitorear los valores de las variables articulares para las posiciones del Robot durante la ejecución de la animación.

5.6.1 Creación de los Modelos en Matlab

Con el fin de presentar la aplicación de la teoría de Robótica de una manera cómoda, con un aumento gradual en la complejidad de los modelos, se escogió cinco configuraciones que presentan los diferentes tipos de articulaciones y muestran cómo el número y la combinación de las mismas determinan la arquitectura y sus características particulares.

Se seleccionó los Robots más utilizados y se los presenta en orden creciente de complejidad.

- Scara
- Cartesiano
- Humano
- Puma GYM
- Stanford

5.6.2 Creación de los modelos en RoboWorks 2.0

El proceso de elaboración de modelos en **RW** es fácil, rápido y lógico. Hasta la fecha el proveedor no ha desarrollado una guía de usuario o manual de comandos aunque el programa cuenta con una ayuda a través de la opción help que resuelve en gran medida las dudas que puedan presentarse. También existe un archivo pdf en español que presenta pistas y ejemplos.

Se construyeron los modelos en **RW** con las dimensiones de los modelos que ya habían sido elaborados en **MH**, poniendo especial cuidado al asignar los ejes de rotación ya que el sistema de referencia de **RW** esta rotado 90° alrededor del eje **OX** con respecto al sistema de referencia de **MH**.

5.6.3 Creación del Programa creadat.m

Uno de los objetivos más importantes de este proyecto es la presentación adecuada de los modelos y la animación de los mismos tanto en el entorno **MH**, en cuyo caso puede llamarse modelo de barras y la animación del modelo tal y como se vería en la vida real presentado en el espacio de **RW**.

Dentro de las posibilidades de comunicación que ofrece el programa RoboWorks se encuentra un contacto directo con archivos tipo .vi de LabView, comunicación directa con programas elaborados en lenguaje C, comunicación a través del protocolo TCP/IP y lectura de archivos de datos del tipo .dat.

Con el fin de optimizar el proceso de animación y reducir el número de pasos intermedios para pasar de un espacio a otro, se optó por realizar todas las animaciones a través de los archivos tipo .dat. Aunque este tipo de archivos es relativamente sencillo, se encontró muchas dificultades para generarlos desde **MH**.

Después de varios intentos, se desarrolló un programa en **MH** que toma los valores asignados a cada variable articular, convierte las dimensiones, genera la tabla en formato .dat y la guarda directamente en la carpeta Examples de **RW**.

5.6.4 Sincronización de los modelos Matlab-RoboWorks 2.0

La construcción se hizo siguiendo el modelo elaborado en **MH**, algunos ejes de rotación quedaron invertidos; esto significa que un ángulo positivo para el espacio **MH** produce rotaciones en sentido negativo en el espacio **RW**.

Afortunadamente el diseño de **RW** permite introducir modificaciones dentro de modelos terminados, además de la posibilidad de aislar secciones del modelo para introducir cambios que no modifiquen en absoluto el resto del modelo.

Finalmente se establecieron las posiciones de cero para que tanto el modelo en **MH**, como el modelo en **RW** inicien en la misma posición y así los incrementos de posición ya sea con valores positivos o negativos generados en **MH** sean reproducidos tal cual en el espacio **RW**.

5.6.5 Animación Desde el Teclado

En el menú Animation de la barra de tareas se encuentra la opción monitor, que permite identificar las articulaciones del modelo. Cuando se invoca el comando Monitor, se despliega una ventana que presenta un tabla en la cual describe el modelo de Robot que se está observando. Está formada por tres columnas cuyas propiedades se explican a continuación.

- Tag Name indica la columna con los nombres de las variables articulares que componen el modelo.
- Key Name es la columna que muestra las teclas que han sido asignadas para controlar cada variable articular o conjunto de piezas móviles.
- Tag Value Es la columna que muestra el valor de la variable articular y se actualiza cada vez que se presiona una de las teclas de control.

5.6.6 Animación desde Matlab a RoboWorks 2.0

La instrucción creardat esta diseñada para que durante la ejecución de un programa en **MH**, se escriba un archivo tipo .dat de animación.

Sintaxis de la instrucción creardat.

```
>>creardat('nombre',va,Q)
```

En la cual 'nombre' se refiere al nombre completo del archivo. Si no se incluye una dirección, el programa por defecto guarda el archivo en la carpeta Work de Matlab; para este proyecto los archivos se guardan inmediatamente en la carpeta Examples de **RW**. En la misma instrucción el elemento *va* se refiere a una cadena de caracteres conformada por los nombres de las variables articulares que fueron asignadas en **RW**; cada nombre debe estar separado del siguiente mediante un tab., la expresión Q se refiere a la matriz donde han de estar los valores que serán asignados paso a paso en la animación. Es importante tener en cuenta que **MH** trabaja en metros y radianes, mientras **RW** trabaja con centímetros y grados.

Ejemplo de una animación creada desde Matlab para RoboWorks:

```
>> sc=[0 0 0 0.05 0;0 0.1 0 0.02 0;pi 0.08 0 0.1 1];
>>Q=[ 0 pi/2 0; 0 0 0.07; (3*pi)/4 0 0.0];
>> [F,C]=size(Q);
>>figure
>>for k=1:F
>>Q(k,:)
>>plotbot(sc,Q(k,:), 'w,f');
>>pause
>>end
>>va='rot1  rot2  palo';
>>Q(:,1:2)=Q(:,1:2)*180/pi;
>>Q(:,3)=Q(:,3)*100;
>>creardat('C:\Archivosde programa\Newtonium\RoboWorks\Examples\
scara.dat',va,Q);
```

5.6.7 Creación de un Archivo tipo .dat para Animación

Cuando se conocen los nombres de las variables articulares del modelo en **RW** y las características de las respectivas articulaciones, se puede proceder a crear un archivo de animación.

El block de notas de Windows es el espacio más recomendable para escribir este archivo. En la primera fila se escriben los nombres de las variables articulares, determinando así una columna para cada variable articular; en las filas siguientes se escriben los valores correspondientes a cada variable articular que serán usados para generar la animación.

Luego de escribir los valores, se guarda el archivo, confirmando que sea guardado como un archivo tipo .dat; en ese proceso se puede determinar el direccionamiento del mismo.

Entre los ejemplos cabe destacar que el archivo .dat del ejemplo 6 DOF Robot está mal escrito, la primera fila de valores pertenece a la fila de las variables; hay que modificarlo para poder ejecutar la animación.

5.7 CINEMÁTICA DIRECTA (gurú7)

El problema cinemático directo consiste en hallar la matriz **T_n** con base en la matriz **dh** y un conjunto de valores asignado a cada variable articular en la matriz **q**.

El comando `fkine` resuelve el problema cinemático directo con base en los datos suministrados por la matriz **dh** y la matriz **q**.

5.7.1 Significado de la Matriz de Brazo

La matriz de brazo, llamada matriz **T_n**, es la *MTH* que representa al sistema de coordenadas solidario con el efector final referenciado al sistema origen ubicado en la base del Robot.

El papel de las variables articulares en el proceso de resolver el problema cinemático directo, se puede observar aplicando ecuaciones simbólicas.

```
>> syms t1 t2 t3 t4 t5 real ;  
>>q1= [t1 t2 t3 t4 t5] ;  
>>T=simple(fkine(dh,q1))
```

La matriz **T** es una *MTH* cuyos elementos presentan las operaciones matemáticas que protagonizan las variables articulares para referenciar el efector final al origen del Robot.

5.7.2 Cinemática Directa de los Modelos Planteados

Se presenta el desarrollo de la cinemática directa de los modelos que fueron creados en **MH** y **RW**.

Dentro de la información que se muestra en la ventana de diálogo en la **MH** se indica el número de columnas que debe tener la matriz **q** que va a digitarse; también se muestra la matriz **dh** del modelo para, de ser necesario, consultar la arquitectura del Robot e identificar el tipo de articulaciones.

Posteriormente se presenta una gráfica en el espacio **MH** de la posición en que quedó configurado el Robot con los valores introducidos y la matriz de brazo correspondiente.

5.7.3 Animación de la Cinemática Directa

Cuando el programa resuelve el problema cinemático directo, simultáneamente genera un archivo tipo .dat de animación para presentar en el espacio de **RW** el recorrido del Robot desde la posición de cero hasta la posición determinada por los valores de articulación dados por la matriz **q**.

5.8 CINEMÁTICA INVERSA (gurú8)

Para solucionar el problema cinemático inverso es necesario ante todo plantear un problema que sea coherente, es decir, que todos los factores involucrados en este proceso estén debidamente identificados y determinados; implica un conocimiento completo del Robot, esto es, la matriz de parámetros, el volumen de trabajo, el alcance y las posibles restricciones del mismo.

La cinemática inversa se obtiene de un proceso matemático iterativo que puede resultar muy largo por lo cual se debe limitar tanto el número de iteraciones como la tolerancia o el error entre la matriz objeto y la matriz que se obtiene como resultado.

El comando ikine resuelve el problema cinemático inverso, y para ello depende de la matriz **dh**, la matriz objeto, las condiciones de iteración, una posición inicial del Robot y una máscara de restricciones.

5.8.1 La Matriz Objeto

La matriz objeto es una *MTH* que contiene los datos de posición y orientación con respecto al origen del efector final del Robot que se desea alcanzar. Esta matriz se llamará matriz **T**.

5.8.2 Las Condiciones de Iteración

El número máximo de iteraciones y la tolerancia son dos parámetros que se indican en los espacios stol e ilimit dentro de la instrucción ikine.

- Stol se refiere a la máxima diferencia que será admitida entre la *MTH* calculada y la *MTH* que fue ingresada en la matriz T. Se indica en notación científica; 1e-3 es recomendable.
- Ilimit es el número máximo de iteraciones que se ejecutarán para calcular la solución. Se indica en notación científica; 1e+3 es recomendable.

También se necesita saber desde qué posición parte el Robot para resolver el problema. Esta posición inicial se determina mediante una matriz fila con los valores de articulación llamada matriz **q0**.

5.8.3 La Máscara de Restricciones

Para Robots con menos de seis grados de libertad existen posiciones imposibles de alcanzar por el efector final, denominadas restricciones.

Las restricciones que presenta la arquitectura del Robot deben especificarse como una matriz máscara en la que se designan las características móviles del Robot. Esta matriz, llamada matriz **M**, presenta la siguiente estructura: Los primeros tres elementos se refieren a la movilidad en los ejes x y z y los tres últimos a la capacidad de Rotación alrededor de los mismos ejes. Se designa con valor uno (1) los movimientos habilitados y con valor cero (0) las restricciones.

$$\mathbf{M}=[T_x \ T_y \ T_z \ R_x \ R_y \ R_z] [26]$$

Sintaxis para ejecutar el comando ikine:

```
>>q=ikine(dh,T,stol, ilimit,q0,M)
```

Donde dh se refiere a la matriz **dh** del Robot, T es la matriz objeto, $stol$ es la tolerancia de error, $ilimit$ es el número de iteraciones que se van a realizar, $q0$ es la matriz de posición de partida y M es la matriz máscara.

5.8.4 Cinemática Inversa de los Modelos Planteados

Gurú 8 presenta el desarrollo de la cinemática inversa de los modelos que fueron creados en **MH** y **RW**.

Dentro de la información que se muestra en la ventana de diálogo se indica que la matriz a introducir corresponde a una MTH la cual determina la posición y orientación del efector final, también se incluye la matriz **dh** del modelo para, de ser necesario, consultar la arquitectura del Robot e identificar el tipo de las articulaciones.

Finalmente se presenta una gráfica en el espacio **MH** de la posición en que quedó configurado el Robot con los valores introducidos en la matriz **T** y la matriz **Tn** correspondiente.

5.8.5 Animación de la Cinemática Inversa

Con el fin de mostrar el recorrido desde la posición de cero del Robot hasta la posición determinada por los valores de articulación obtenidos en la solución dada por la matriz **q**, el programa genera un archivo tipo **.dat** de animación para mostrar el movimiento en **RW**.

5.9 MODELANDO EN ROBOWORKS 2.0 (gurú9)

Elaborar el modelo de un Robot en **RW** es un proceso delicado y de mucha atención, pues en este programa los objetos deben descomponerse en sus figuras geométricas básicas, las cuales han de estar perfectamente dimensionadas y orientadas respecto al sistema de referencia que se está utilizando.

Los eslabones del Robot pueden separarse en grupos de elementos identificados con un color específico para facilitar la construcción y el estudio del modelo durante la simulación de movimientos.

Las transformaciones permiten configurar las articulaciones, trasladar el sistema de referencia durante la construcción y aislar elementos o grupos de elementos para asignar un sistema de referencia o una articulación especial.

5.9.1 Iniciando la Construcción

Cuando inicia el programa **RW**, se abren las dos ventanas principales, la de vista 3D y la vista de árbol.

En la barra de herramientas de **RW** se encuentran la barra de tareas y la barra de íconos, con las opciones típicas de Windows para manejo de archivos.

5.9.2 Inserción de Elementos

En la posición inicial de la estructura que va a construirse, aparece activa la raíz (root). Dando doble click en ella se abre una ventana en la cual se puede asignar un nombre a la estructura.

En el menú edit se selecciona la opción insert child, con lo cual se abre una ventana de selección del objeto a insertar.

Cuando se está iniciando un modelo es conveniente insertar primero un grupo en la opción others y asignarle un nombre. Generalmente los elementos que conforman el primer grupo corresponden al primer eslabón, que por conveniencia se llamará base.

Luego de insertado el grupo de nombre base, se selecciona insert child para insertar color, que se obtiene en la opción others seleccionando el ítem material; allí se puede elegir cualquiera de las opciones de color presentadas. En este momento se tiene identificado un grupo por nombre y por color, el paso siguiente es insertar los elementos que conforman ese primer eslabón.

Para construir la base es necesario descomponer la totalidad del eslabón en las formas geométricas básicas, debido a que **RW** sólo ofrece las formas cilindro, cono, disco, arandela, esfera, cubo y cuña, las cuales adecuadamente combinadas pueden reproducir casi cualquier forma tridimensional imaginable.

La primera figura componente del grupo se inserta con insert child, los demás elementos de la figura se insertaran con insert after para que todos los componentes del eslabón queden en el mismo nivel jerárquico. Se sugiere igualmente que todos los grupos que representan los eslabones de un Robot pertenezcan a un mismo nivel jerárquico.

Además de grupos, color y figuras, se pueden insertar transformaciones, las cuales constituyen el componente dinámico del modelo y por ser de gran importancia se explicarán con detalle en el apartado siguiente.

5.9.3 Transformaciones

La opción transformation de la ventana de inserciones permite escoger entre varias opciones estas son: Transformation stop, transformation start, scaling, translation y rotation.

- Cuando se selecciona rotation o translation se abre una ventana la cual por defecto inicia en la opción name, que permite asignar un nombre a la transformación, un eje al cual se va a aplicar y el valor de la transformación que se desea ingresar.
- La opción control abre una ventana en la cual se va a determinar el tipo de transformación, si es estática o dinámica. La ventana por defecto está activa en la opción estática. Si la transformación a insertar es una articulación, entonces hay que activar dynamic; inmediatamente se abre una ventana donde se definirán las propiedades de la articulación.

Las transformaciones estáticas se usan en la construcción de los modelos y sirve para desplazar el sistema de referencia. Usualmente en el extremo final de un eslabón se encuentra la respectiva articulación que lo une con el eslabón siguiente. Para la construcción resulta conveniente referenciar el eslabón en construcción a su origen real en el extremo del eslabón anterior; esto evita el complicado seguimiento que significa referenciarlo directamente al origen en la base del Robot. Además, este procedimiento es de gran importancia para la configuración de las articulaciones.

Como el eje vertical en el espacio de **RW** es el eje **OY**, hay situaciones en las cuales es necesario introducir transformaciones estáticas de rotación que corrijan la diferencia en el sentido de giro de los ejes, para que las rotaciones generadas por **MH** puedan reproducirse exactamente en **RW**.

En **RW** la construcción es secuencial de forma que los cambios introducidos en cualquier parte del árbol modifican a todos los elementos que sean insertados

después; los eslabones insertados entre las instrucciones transformation start y transformation stop forman un conjunto interdependiente de forma que el movimiento de uno de ellos afecta a los demás del mismo conjunto pero deja inmodificado a todos los demás. Los elementos que se inserten posteriores al transformation stop seguirán con las condiciones que existían antes de la instrucción transformation start.

Uno de los casos más comunes en que se utiliza transformation start/stop es en el diseño de las pinzas de un manipulador, cuyos dedos deben moverse con la misma tecla de control pero en sentidos contrarios.

Las transformaciones dinámicas definen una articulación y ejecutan su movimiento en el origen del sistema de referencia, esto pone de manifiesto la necesidad de posicionar un sistema de referencia en el extremo final de cada eslabón.

El sentido de movimiento sigue la norma de signos de los vectores unitarios para desplazamientos y la regla de la mano derecha para rotaciones.

6. DISEÑO DE LAS GUÍAS DE LABORATORIO DE ROBÓTICA

Partiendo del enfoque constructivista planteado en el anexo B, se diseñó un conjunto de guías de Laboratorio que aprovechan las ventajas de la representación visual de modelos Robóticos, para comprobar y entender los complejos conceptos matemáticos que se requieren para explicar la cinemática de los Robots.

La etapa de la presentación del tema se supone ha sido realizada por el profesor a través de clases tradicionales, o mediante exposiciones por parte de grupos de alumnos; de manera que a continuación se presentan los preceptos constructivistas y su aplicación en el diseño de experiencias propuestas para complementar el tema referente a cinemática de Robots manipuladores.

6.1 PARTIR DEL DESARROLLO DEL ALUMNO.

Las practicas de Laboratorio están sincronizadas con la clase, de manera que el tema de cada práctica corresponde a un tema previamente presentado y para el cual el estudiante ya ha tenido suficiente tiempo de de comprensión tanto individual como en grupo, incluidas las horas de consulta con el profesor.

6.2 ASEGURAR LA CONSTRUCCIÓN DE APRENDIZAJES SIGNIFICATIVOS.

Teniendo en cuenta que la práctica de laboratorio debe servir para complementar los conocimientos presentados durante el desarrollo de la clase, es importante diseñar actividades que permitan estudiar los conceptos más representativos y que tienen mayor relevancia para completar el proceso de construcción de conocimiento.

6.3 POSIBILITAR QUE LOS ALUMNOS REALICEN APRENDIZAJES SIGNIFICATIVOS POR SÍ SOLOS.

Aunque las guías de laboratorio vienen equipadas con un marco teórico que resume el tema de la clase presentando la información necesaria para el desarrollo de la práctica, existen algunos ítems que requieren una investigación por parte del alumno o la planeación de situaciones hipotéticas para resolver problemas determinados.

La etapa de revisión de esquemas, para saber como están construyendo el nuevo conocimiento los estudiantes, está contemplada en el desarrollo del pre-informe, en el cual el estudiante deberá resolver una tarea sencilla recurriendo a los conocimientos que pudo haber adquirido durante la clase y a través de la misma guía.

6.4 PROCURAR QUE LOS ALUMNOS MODIFIQUEN SUS ESQUEMAS DE CONOCIMIENTO.

Durante el desarrollo del pre-informe el estudiante trabaja con sus propios esquemas y realiza una construcción preliminar de la situación planteada en la guía. En los primeros minutos del laboratorio el estudiante tiene un espacio para realizar preguntas referentes a las dificultades que tuvo en la elaboración del pre-informe e igualmente pueda corregir los errores que cometió en la elaboración del mismo.

Igualmente a lo largo del laboratorio los estudiantes están enfrentando las situaciones planteadas en la guía usando las herramientas software y tienen al menos una persona capaz de solucionar las dificultades que se presenten en cualquier momento.

En el espacio de laboratorio se realiza la etapa más importante, pues es la oportunidad de experimentar con las herramientas software, comprobar la teoría presentada en clase y corregir o reconstruir los conceptos que en el momento del pre informe no estaban del todo claros, con la ventaja de realizar la actividad en grupos de trabajo dejando la oportunidad de discutir tanto entre los miembros del grupo como entre grupos vecinos.

6.5 ESTABLECER RELACIONES RICAS ENTRE EL NUEVO CONOCIMIENTO Y LOS ESQUEMAS DE CONOCIMIENTO YA EXISTENTES.

Luego de la práctica los estudiantes elaborarán un informe de laboratorio, en el cual deben presentar los resultados de los problemas planteados en la guía y una página con observaciones y conclusiones. Este espacio es el que permite la integración entre los conocimientos previos y las experiencias recientes, pues para redactar las observaciones y las conclusiones deben tener un pensamiento coherente y una comprensión del tema de la guía que les permita escribir con sentido.

7. CONCLUSIONES

- Como resultado de esta experiencia se tiene una herramienta computacional por medio de la cual se puede diseñar cualquier tipo de Robot manipulador, realizar toda la matemática que este involucra, crear un prototipo tridimensional virtual y programar secuencias de movimiento.
- Esta herramienta permite experimentar libremente con múltiples propuestas para el diseño de la arquitectura del Robot, de la celda de trabajo en la cual va a operar y la programación de las tareas así como la simulación de situaciones específicas durante la operación.
- Todo esto se realiza en el espacio virtual de un computador, de manera que no se requiere invertir dinero en materiales para fabricar prototipos, no se requiere un espacio de trabajo para pruebas y no hay pérdidas por accidentes de hardware o errores de programación.
- La herramienta Hemero de Matlab es un logro formidable mas no perfecto. Algunas de sus deficiencias pudieron corregirse, otras fueron estratégicamente eludidas. Sin embargo, no se descarta que durante el desarrollo de las prácticas planteadas, puedan aparecer otras cuya solución deberá ser encontrada sobre la marcha.
- El software de visualización, por su parte, exige habilidad y entrenamiento en construcción y concepción tridimensional. Deficiencias como el limitado menú de figuras y la necesidad de modificar continuamente los ejes de referencia, presentan obstáculos que requieren mucha atención y tiempo para superarlos.

- Cabe destacar, que la facilidad de resolver operaciones complejas y repetitivas a través de Matlab, reduce en gran medida el tiempo de programación y la posibilidad de cometer errores en este tipo de procesos y permite enfocar los recursos a etapas como diseño y programación de rutinas. La opción de resolver ecuaciones simbólicas tiene un gran valor didáctico en la medida en que permite observar comportamientos y realizar seguimientos de las variables en las complejas operaciones involucradas.
- El Programa RoboWorks 2.0 es una herramienta sencilla pero con una gran capacidad de adaptación, ya que sus resultados se limitan a la visualización de modelos y animación de los mismos, sin involucrarse con la teoría de Robótica, los algoritmos de programación, la solución de problemas, etc. Esto la convierte en una etapa de visualización que bien puede aplicarse a cualquier rama de la ingeniería que requiera mostrar objetos tridimensionales en movimiento.
- Las prácticas se diseñaron dividiendo el tema en etapas pequeñas con objetivos específicos sencillos, lo cual permite aclarar muchas dudas y dejar el espacio para que los estudiantes y usuarios detecten las deficiencias que tienen sobre los conceptos, el manejo de herramientas matemáticas y habilidades computacionales. El grado de complejidad se fue incrementando de manera gradual de forma que los conceptos fueran apareciendo de manera secuencial, hasta alcanzar el grado de complejidad involucrado en un modelo de seis grados de libertad.
- Para el diseño de este material se tuvo en cuenta el modelo pedagógico propuesto por la teoría constructivista la cual propone que el conocimiento se construya a través de la experiencia que el individuo tiene con el mundo y con los compañeros.

- Esta filosofía se aplica en la Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones donde casi todas las asignaturas que se ofrecen a los estudiantes tiene un componente práctico que se desarrolla en laboratorios específicamente adecuados para tal fin. Sin embargo es claro que todas las teorías planteadas sobre el proceso de aprendizaje son el resultado de investigaciones y por tal razón aplican para el momento en que fueron planteadas, y que en la medida en que la sociedad, los materiales didácticos, y los espacios de aprendizaje evolucionan, igualmente los métodos de enseñanza deben evolucionar, por lo que no debe considerarse como la única manera en que las personas aprenden, sino como una propuesta para solucionar algunas dificultades de enseñanza.
- Las guías que se diseñaron originalmente, fueron probadas por los estudiantes del grupo de Robótica, con gran aceptación y las críticas y sugerencias que de ellas surgieron sirvieron como aporte fundamental para la elaboración de las guías definitivas, aunque el espacio queda abierto para cualquier mejora.

8. RECOMENDACIONES

- En el presente proyecto se abordó el tema de la cinemática de Robots manipuladores; queda por realizar el modelado matemático de la cinemática de Robots móviles y todo lo referente a dinámica tanto de Robots manipuladores como de Robots móviles, para lo cual Hemero sólo ofrece herramientas en el espacio simulink de Matlab.
- Como ya se mencionó, RoboWorks tiene varias opciones de comunicación, por lo cual puede utilizarse como complemento en el desarrollo de proyectos que utilicen plataforma lenguaje C, LabView y comunicación de larga distancia mediante el protocolo TCP/IP.
- Presentar animaciones simultaneas de prototipos reales y su modelo en RoboWorks es una opción formidable que puede utilizarse para propósitos didácticos y de evaluación de modelos, porque permite observar el movimiento desde ángulos imposibles en el mundo físico y también realizar acercamientos para estudiar más detalladamente elementos específicos, durante la animación en tiempo real.
- La capacidad de modelado y animación de RoboWorks podría usarse también para presentar imágenes simuladas de algunos conceptos que resultan algo difíciles de entender en asignaturas como Sistemas de Control, Maquinas Eléctricas, Automatización de Procesos, Mecánica, Mecánica Analítica y en general todos los temas que involucren movimientos y relaciones tridimensionales.

9. REFERENCIAS BIBLIOGRAFICAS

- [1]. ANGULO USCATEGUI JOSÉ M^a "Robótica Práctica: Tecnología y Aplicaciones"
- [2]. ARMENTA JAVIER "Poder personal y psicoterapia Una visión narrativa centrada en la persona"
- [3]. BARRIENTOS ANTONIO, LUIS FELIPE PEÑIN, CARLOS BALAGUER, RAFAEL ARACIL "Fundamentos de Robótica"
- [4]. BONA B., INDRI M., SMALDONE N. "An Experimental Setup for Modelling, Simulation and Fast Prototyping of Mechanical Arms"
- [5]. CARRETERO MARIO "Constructivismo y Educación"
- [6]. CARRETERO MARIO "¿Qué es el Constructivismo?"
www.uls.edu.mx/~estrategias/constructivismo_educacion.doc Agosto 2005
- [7]. CORKE PETER I. "Robotics Toolbox (for use with MATLAB)". CSIRO Division of Manufacturing Technology, Preston Australia. 1994.
- [8]. CRAIG JOHN J. "Introduction to Robotics Mechanics and Control"
- [9]. CHENG FRANK S. "A Methodology for Developing Robotic Workcell Simulation Models" Proceedings of the 2000 Winter Simulation Conference.
- [10]. FOK, S. C., TAN F. L. "An Application of Computer Graphics For Teaching Robot Kinematics"
- [11]. FU K. S., GONZALEZ R. C., LEE C. S. G. "Robótica: Control, Detección, Visión e Inteligencia"

- [12]. GALLEGO RÓMULO "Discurso sobre el constructivismo"
- [13]. IBÁÑEZ-GUZMÁN J. "A Robotics And Automation Teaching Laboratory"
- [14]. INCE IIHAN, BRYANT KEITH, BROOKS THURSTON "Virtuality and Reality: A Video/Graphics Environment for Teleoperation"
- [15]. KARMY RODRIGO "Diferencias: Construccinismo y constructivismo"
- [16]. KROTKOV ERIC "Robotics Laboratory Exercises" IEEE Transactions on Education. February 1996.
- [17]. LEE D. M. A., ELMARAGHY W. H. "ROBOSIM: A CAD-based Off-line Programming and Analysis System for Robotic Manipulators" Computer-Aided Engineering Journal October 1990.
- [18]. LEGO "Invention Adventure Activity Book"
- [19]. NONAKA KENICHIRO "VRSC: Visual Robotic Simulation and Control with MaTX/RtMaTX Proceedings of the 1999 IEEE.
- [20]. OLLERO BATURONE ANIBAL."Robótica: Manipuladores y Robots móviles".
- [21]. PAOLINI CHRISTOPHER, VUSKOVIC MARKO "Integration of a Robotics Laboratory Using CORBA"
- [22]. ROGALLA O., POHL K., DILLMANN R. "A General Approach for Modelling Robots" Proceedings of the 2000 IEEE/RS International Conference on Intelligent Robots and Systems.
- [23]. VARGAS GERMÁN "¿Es posible una didáctica constructivista en el ámbito universitario?"

[24]. WANG JING "Methodology and Design Principles for a Generic Simulation Plattform for Distributed Robotic System Experimentation and Development"

ANEXO A GUÍAS PROPUESTAS PREVIAMENTE POR ESTUDIANTES DE ROBÓTICA

A continuación se presentan las guías que fueron diseñadas y propuestas por estudiantes de la asignatura Robótica en diferentes periodos académicos en que se dictó el curso, algunas de las cuales se utilizaron como material de trabajo para el desarrollo de este proyecto.

A.1. PRÁCTICAS USANDO LA VERSIÓN DEMO DEL PROGRAMA WORKSPACE

Propuesto por las estudiantes:
Maribel Anaya y Andrea Rosas

GUÍA No. 1 COMPROBACIÓN DE LA CINEMÁTICA DIRECTA UTILIZANDO UN ROBOT SCARA CINEMÁTICA DIRECTA

Objetivo general

Relacionar el movimiento del robot con el problema de resolución de la cinemática directa en un robot Scara.

Objetivos específicos

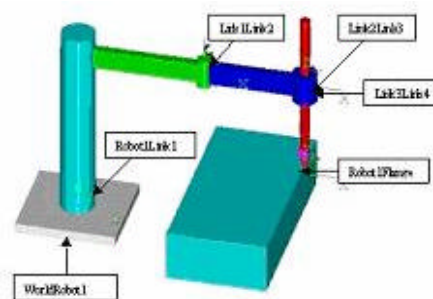
- Utilizar el programa Workspace para aplicar los conceptos de robótica.
- Implementar el desarrollo de una tarea por parte del robot.
- Visualizar la potencialidad de la robótica en procesos de producción.
- Obtener la matriz de Denavit Hartenberg asociada al robot (Configuración SCARA R2TR).

Procedimiento

1. Realizar los cálculos teóricos para determinar la matriz de Denavit Hartenberg (según la tabla 1).
2. Abrir workspace.
3. Ir a *Opciones* , *Visual Basic*.
4. Clic derecho sobre el nombre del proyecto.
5. *Importar archivo* (Nombre del robot).
6. Volver a Workspace.
7. Ejecutar la subrutina correspondiente a la creación de los elementos gráficos del robot (en la ventana de comandos teclear >>robot1)
8. Se selecciona el elemento base del futuro robot, se pide crear al robot

9. Para crear robot Ir a *Create, Robot*
10. Confirmar el número de juntas del robot.
11. Menú *robot, Edit Joints* (Posición y orientación de juntas según tabla 1)
12. Verificar la cinemática del robot, comparar los resultados teóricos y realizar los cambios correspondientes.
13. *Robot, Kinematics* (verificación de matriz de parámetros DH y configuración del robot Scara RRTR)

Fig. A1 Robot Scara



- La ubicación de cada uno de los sistemas de coordenados asociados a cada junta es:

Tabla A1. Configuración espacial del Robot

Número Junta	Juntas	X	Y	Z	Mín	Máx	Tipo
1	Robot1Link1	0	0	0	-150°	150°	R
2	Link1Link2	250	0	350	-150°	150°	R
3	Link2Link3	450	0	350	-100	80	T
4	Link3Link4	450	0	350	-150°	150°	R
5	WorldRobot1	0	0	0			
6	Robot1Flange	450	0	150			

GUÍA No. 2

COORDENADAS HOMOGÉNEAS Y MATRIZ DE TRANSFORMACIÓN

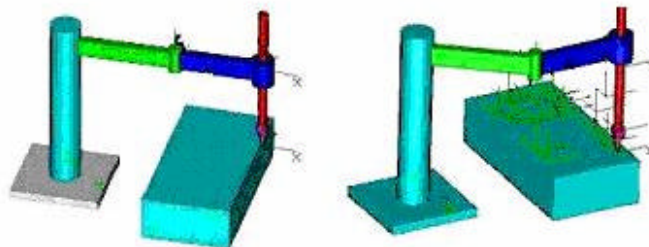
Objetivo General

Comprobar como la aplicación de las matrices de transformación relacionan el movimiento del Robot con su entorno dentro de su espacio de trabajo.

Objetivos Específicos

- Implementar el desarrollo de una tarea por parte del robot
- Relacionar el movimiento del efector final del Robot con los movimientos de giro y desplazamiento de cada una de sus articulaciones.
- Explorar las potencialidades de Workspace como software de simulación para robótica.

Fig. A2 Se observa la configuración inicial del Robot SACARA y la configuración final y las Secuencias del movimiento del efector final.



En este laboratorio se trabajará con los elementos del teach pendant: Con valor numérico de cada uno de los valores de las coordenadas de articulación, X, Y, Z posición cartesiana del efector final del robot, A, B, C: Orientación del efector final.

1. Para visualizar el movimiento por independiente de cada una de las articulaciones del robot se visualiza la consola *Teach Pendant: Menu View Pendant*.

2. Para relacionar el movimiento del efector final con el movimiento de cada una de las articulaciones se programa el robot para que realice una tarea específica, en este caso se especifica la secuencia de pasos para que el robot realice al trazo de las letras C y W.
 3. Desde el menú *Teach Pendant: Menu View Pendant* se editan cada uno de los pasos y se llaman en orden de ejecución para mayor sencillez
 4. Los pasos están referenciados al sistema de coordenadas de la base para facilitar el procedimiento y lograr la asimilación del comportamiento del Robot paso a paso.
- Debido a que la configuración del robot es SCARA se utilizará el robot para simular la escritura de dos letras sobre un bloque (Figura 2). Los puntos geométricos que deben ser programados son los siguientes:

Tabla A2. Puntos Geométricos para la C:

Gp	Tipo Mvto	X	Y	Z
0	Joint	450	0	150
1	Joint	300	50	150
2	Lineal	300	50	100
3	Circular	250	100	100
4	Circular	300	150	100
5	Lineal	350	150	100
6	Circular	400	100	100
7	Circular	350	50	100
8	Lineal	350	50	150
Via1	(Puntos 6 y 7)	385.4	64.6	100
Via2	(Puntos 3 y 4)	264.6	64.6	100

Tabla A3. Puntos geométricos para la W:

Gp	Tipo Mvto	X	Y	Z
9	Lineal	350	-50	150
10	Lineal	350	-50	100
11	Lineal	250	-75	100
12	Lineal	350	-100	100
13	Lineal	250	-125	100
14	Lineal	350	-150	100
15	Lineal	350	-150	150

Los puntos denotados como Via1 y Via2 son utilizados para poder realizar los movimientos de tipo circular.

Una vez en el menú se elige la opción de *nuevo GP* y se escriben cada uno de los pasos para al final crear el Path correspondiente.

Cuando todos los pasos hayan sido editados se configura el Path correspondiente a dicha secuencia de la siguiente forma:

1. Se crea un nuevo Path.
2. Se arrastra cada uno de los pasos hasta el icono del nuevo Path creado teniendo en cuenta el orden de ejecución, es decir, se agregan al Path en el orden en que deben ser ejecutados por el Robot para lograr que este cumpla con la tarea programada.
3. Finalmente se activa el Path en el nombre del ROBOT al dar clic derecho y seleccionar como el activo (*Set as active ...*) y marcar *Simulate*.
4. Para iniciar simulación *Robot, Play Simulation*.

A.2. PRÁCTICAS USANDO LA TOOLBOX HEMERO PARA MATLAB

Propuesto por los estudiantes:

Diego Tibaduiza y John Ruiz

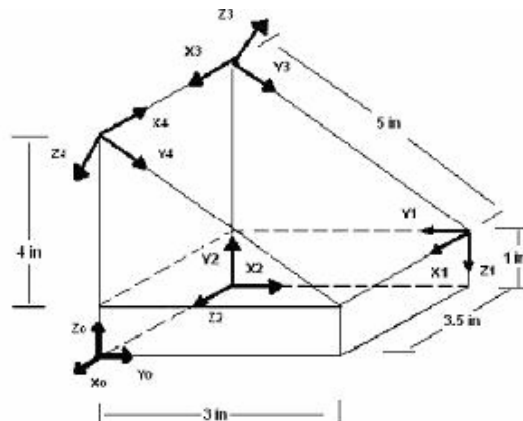
LABORATORIO DE ROBÓTICA

Practica de software 1

REPRESENTACIÓN DE LA ORIENTACIÓN Y LA POSICIÓN

1. Determinar la matriz de rotación para una rotación de un ángulo $\phi=30^\circ$ respecto al eje OX, seguida por una rotación de ángulo $\psi=90^\circ$ respecto al eje OW, seguida por una rotación de ángulo $\theta=45^\circ$ respecto al eje OY.
2. Determinar una matriz T que representa una rotación de ángulo $\alpha=60^\circ$ respecto al eje OZ, seguida por una traslación de 8 unidades de distancia a lo largo del eje OZ, seguida de una rotación de ángulo $\phi=30^\circ$ respecto al eje OV.
3. Para la figura que se muestra, encontrar las matrices de transformación homogénea 4×4 ${}^{i-1}A_i$ y 0A_i para $i = 1, 2, 3, 4$.

Fig A3. Sistemas en movimiento.



4. Se ha preparado una estación de trabajo de robot con una cámara de TV, tal como se muestra en la figura. La cámara puede ver el origen del sistema de coordenadas de la base donde se fija un robot de seis elementos, y también el

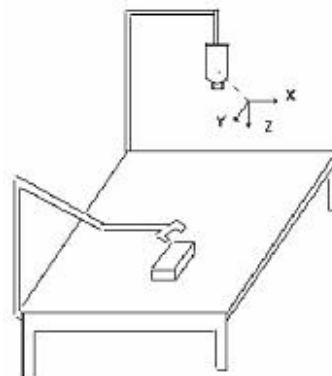
centro de un cubo que tiene que ser manipulado por el robot. Si se ha establecido un sistema de coordenadas local en el centro del cubo, entonces este objeto, tal como lo ve la cámara, se puede representar por una matriz de transformación homogénea T_1 .

También el origen del sistema de coordenadas de la base como lo ve la cámara se puede expresar por una matriz de transformación homogénea T_2 , donde:

$T_1 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ & & 1 & 0 & 0 & 10 \\ 0 & 0 & -1 & 9 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$T_2 = \begin{bmatrix} 1 & 0 & 0 & -10 \\ & & 0 & -1 & 0 & 20 \\ 0 & 0 & -1 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
--	--

- a) Desgraciadamente, después de que se ha colocado el equipo y se han tomado estos sistemas de coordenadas, alguien gira la cámara 90° respecto del eje z de la cámara. ¿Cuál es la posición/orientación de la cámara con respecto al sistema de coordenadas de la base del robot?
- b) Después de que haya calculado la respuesta a la pregunta a), la misma persona giro el objeto 90° respecto del eje X del objeto y lo trasladó cuatro unidades de distancia a lo largo del eje y girado. ¿Cuál es la posición/orientación del objeto con respecto al sistema de coordenadas de la base del robot? ¿Y con respecto al sistema de coordenadas de la cámara girada?

Fig A4. Sistema cámara-Robot y objeto.

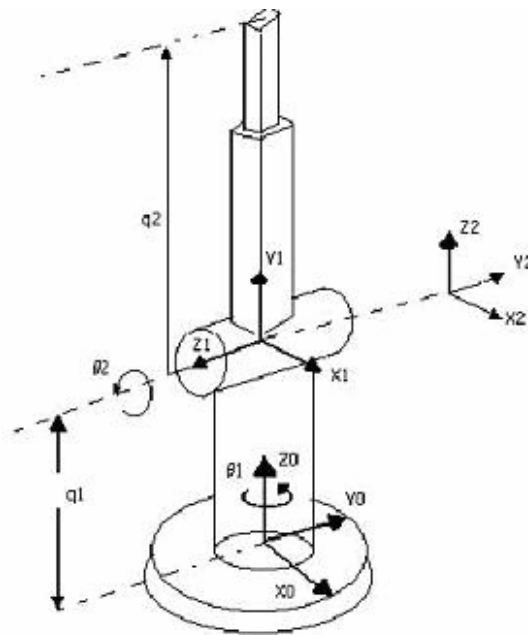


LABORATORIO DE ROBÓTICA
Practica de software 2

MODELOS CINEMÁTICOS DE ROBOTS

1. Asignar el sistema de referencia para la articulación de la muñeca
 2. Encontrar los parámetros de Denavit-Hartenberg para el robot mostrado en la figura.
 3. Hallar la matriz de transformación de cada articulación con respecto a la anterior.
 4. Obtenga el modelo directo del manipulador y compruebe el resultado teórico en Matlab con respecto a si se utiliza el comando fkine del mismo.
 5. Hallar el modelo inverso del manipulador de manera teórica y realizando un algoritmo que permita el uso del comando ikine en Matlab.
- Nota: entregar como preinforme escrito los puntos 1,2,3 y 4**

Fig A5. Robot .



ANEXO B GUÍAS DISEÑADAS PARA EL USO DE LA HERRAMIENTA ROBOTICA PROPUESTA CON MH Y RW.

A continuación se presentan las guías que fueron diseñadas con base en la herramienta propuestas para el uso de los paquetes **MH** y **RW**; las cuales están separadas en sesiones sincronizadas con las sesiones interactivas ya presentadas.

LABORATORIO DE ROBÓTICA

Prácticas de Software

Guía para el Desarrollo de las Prácticas

1. Las prácticas de laboratorio se realizarán en el horario determinado y en el aula del centro de cómputo que se asigne para ello.
2. La entrada al laboratorio será permitida dentro de los primeros diez (10) minutos después de la hora señalada.
3. Se formarán grupos de trabajo de máximo 3 estudiantes.
4. Cada estudiante debe haber elaborado su pre-informe a mano y lo entregará al profesor en el momento de la entrada al laboratorio. El pre-informe es requisito para ingresar al laboratorio.
5. Se dará un espacio inicial de tiempo para aclarar dudas sobre la guía y dificultades para la realización del pre-informe.
6. Un pre-informe escogido aleatoriamente será entregado a cada grupo como pre-informe de trabajo para el desarrollo de la práctica. Los errores que el grupo corrija en el pre-informe serán sumados positivamente a la calificación final de la práctica.
7. Se entregará un informe de laboratorio por grupo en la fecha de la siguiente práctica. Como anexo debe entregarse el pre-informe que se

utilizó en el desarrollo de la práctica con las correcciones que se realizaron.

8. Toda práctica que no pueda realizarse por el estudiante deberá ser justificada por el estudiante a través de los mecanismos que la Universidad Industrial de Santander tiene establecidos para ello y de esta forma acordaran con el profesor una fecha para su debida ejecución.

9. La calificación de cada práctica depende de:

- Pre-informe (individual).
- Informe de Laboratorio (grupal).
- Pre-informe de trabajo (grupal).
- Desempeño dentro del laboratorio (individual).

INSTRUCCIONES DE HEMERO

FUNCIÓN	PROPÓSITO
eul2tr	Generar a partir de los ángulos de Euler Z-Y-Z la transformación homogénea correspondiente.
frame	Representar gráficamente un cuadro de referencia con una posición y orientación dadas.
ishomog	Comprobar si el argumento que se le pasa es una transformación homogénea.
rotvec	Calcular la transformación de rotación en torno a un vector conservando la magnitud real del vector.
rotvec1	Calcular la transformación de rotación alrededor del vector unitario del vector determinado.
rotx, roty, rotz	Calcular las transformaciones de rotación en torno a los ejes X Y Z respectivamente.
rpy2tr	Obtener la transformación homogénea correspondiente a los ángulos RPY dados.
transl	Cálculos asociados a transformaciones de traslación.
tr2eul	Obtener los ángulos de Euler Z-Y-Z a partir de una matriz de transformación homogénea.
tr2rpy	Extraer de una transformación homogénea sus correspondientes ángulos RPY.
trinv	Calcular la transformación inversa de una <i>MTH</i> dada.
dh	Contener los parámetros de Denavit-Hartenberg del manipulador.
diff2tr	Convertir un vector de movimiento diferencial en la correspondiente transformación homogénea .
fkine	Calcular la cinemática directa de un robot manipulador.
ikine	Calcular la cinemática inversa de un manipulador.
jacob0	Calcular el jacobiano del manipulador expresado en el sistema de referencia base.
jacobn	Calcular el jacobiano expresado en el sistema de coordenadas del efector final
linktrans	Calcular las <i>MTH</i> de cada enlace del manipulador a partir de los parámetros de Denavit-Hartenberg.
numcols	Calcular el número de columnas de una matriz.
numrows	Calcular el número de filas de una matriz.
plotbot	Representar gráficamente el manipulador.
tr2diff	Convertir una transformación en un vector de movimiento diferencial y calcular la diferencia entre 2 transformaciones.
tr2jac	Calcular la matriz(que se podría llamar jacobiana) que permite pasar un vector de velocidades cartesianas de un cuadro de referencia a otro distinto.
velprop	Calcular el vector de velocidades en el espacio cartesiano expresado en el sistema de coordenadas del cuadro{n}.

LABORATORIO DE ROBÓTICA

Practica de software I

INTRODUCCION A ROBOWORKS 2.0

1. Introducción

La parte de visualización es particularmente importante para el entendimiento de la Robótica. Existen varios paquetes de software que ofrecen una visualización agradable de los modelos de Robots y su entorno en un ambiente tridimensional; uno de estos es el programa RoboWorks 2.0, el cual brinda una aceptable visualización y varias opciones de animación de los modelos construidos.

2. Requisitos

Se requiere que el estudiante tenga un conocimiento básico de Windows y habilidad para manejar menús y ventanas de visualización.

3. Objetivos

- Adquirir conocimiento y destreza en el manejo de menús del programa RoboWorks 2.0.
- Identificar las diferentes herramientas que tiene RoboWorks 2.0 para el manejo de modelos Robóticos.
- Examinar la capacidad que ofrece RoboWorks 2.0 para realizar animaciones de los modelos Robóticos.

4. Marco Teórico

4.1 Introducción a RoboWorks

RoboWorks es un paquete de software desarrollado por la compañía Newtonium bajo la dirección del Dr. Chetan Kapoor en Texas, USA.

Básicamente es un programa que ofrece la posibilidad de construir y animar Robots en un entorno tridimensional. Una de sus cualidades es la capacidad de comunicarse con otros programas y de recibir datos de ellos.

La interfase gráfica del RoboWorks se muestra en la figura 1 y consta de las siguientes partes:

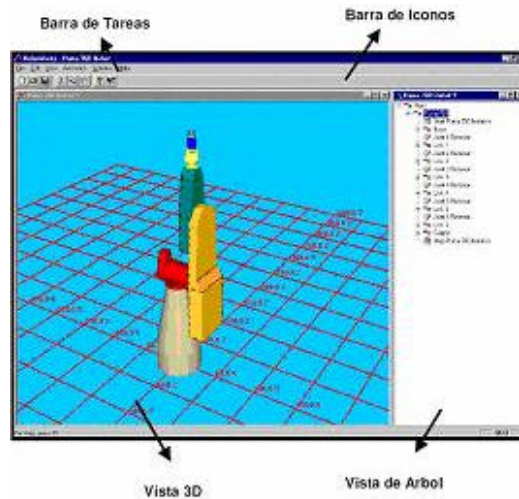
Vista en 3D: En esta vista se puede ver en 3 dimensiones el estado en que se encuentra un modelo tridimensional. Desde esta ventana no se puede modificar el modelo.

Vista de Árbol: En esta vista se crearán todos los modelos. Cada forma visible en la pantalla 3D se verá referenciada en esta vista; y sólo desde esta ventana se pueden modificar sus parámetros.

Barra de Tareas: Ofrece las clásicas tareas de todo programa basado en Windows: Archivo, Editar, Ver, Ventana y Ayuda, además de la opción propia del RoboWorks de Animación.

Barra de Íconos: En la barra de Íconos sólo se encuentran los íconos que pertenecen a las aplicaciones generales de Windows, estos son: Nuevo, Abrir, Guardar, Cortar, Copiar, Pegar, Acerca del RoboWorks y Ayuda.

Figura 1. Elementos de la visualización de RoboWorks 2.0.



Para realizar ajustes en la visualización se ofrecen las opciones del menú edit:

1. Background: Esta opción permite modificar el color de fondo.
2. Autorotation: Esta opción pone a girar el modelo continuamente.
3. Grid: Esta opción muestra una rejilla y por medio de esta, el sistema de referencia inicial.
4. Coordinate frames: Esta opción permite visualizar todos los orígenes creados por medio de transformaciones estáticas.
5. Antialias: Esta opción es útil para visualizar los sólidos con un renderizado de mayor calidad. El renderizado se refiere a la textura en que se puede visualizar un objeto. No se recomienda usar esta opción si su PC no tiene una tarjeta graficadora lo suficientemente buena.

6. Interactive mode: Esta opción presenta una ventana con cuatro posibilidades para modificar el punto de vista del observador. Estas opciones son: rotación translación, zoom y "picking". Esta última se utiliza para modificar las propiedades de los elementos del modelo por medio de un doble click sobre el elemento simulado.

7. Rendering: Permite escoger la visualización del Robot en tres modelos de textura. Las tres opciones son: Sombreado suave, sombreado plano y alambrado.

8. Projection: Esta opción sirve para visualizar el modelo en vista perspectiva o en forma ortográfica, en perspectiva se puede ver como las líneas se alejan del horizonte, en forma ortográfica se ven las medidas en tamaño real.

9. View: Permite visualizar el modelo desde las vistas comunes, arriba, abajo, al frente, atrás, etc. Y también permite ver el dibujo en su posición original.

10. No Clear: Al seleccionar esta opción, y cambiar el punto de vista, el punto de vista anterior permanece y se ven las dos posiciones simultáneamente.

El documento completo del cual se extrajo esta introducción junto a una versión *DEMO* de RoboWorks se encuentra en la página Newtonium.com

5. Procedimiento

1. Para ingresar a RoboWorks se da doble click en el icono de RoboWorks, posteriormente se cierran las ventanas emergentes y picando [no] a las opciones de guardar.

2. Para recuperar un ejemplo se da click izquierdo sobre el icono Abrir carpeta, a continuación seleccione alguno de los archivos identificado con el icono de RoboWorks; para este punto seleccione el archivo ejes.

Maximice esta imagen y observe la posición de los sistemas de coordenadas asociados al origen y a la única articulación. Para mejorar la visualización presione click izquierdo sobre el espacio alrededor del Robot y ajuste las opciones del menú edit.

3. Presione la tecla q y las teclas shift+q para mover la articulación del Robot. Puede ver que el eje z corresponde a la barra de color rojo, el eje y es azul y el x es verde. El sistema de coordenadas se mueve con el brazo, como debe ser.
4. Pique (click izquierdo de mouse) en la opción animation de la barra de tareas y seleccione from file; del menú emergente seleccione ejes. [Abrir]

En el tablero emergente.

Asigne un valor de doscientos (200) en la casilla time delay.

Presione la tecla *reproducción continua* que se encuentra entre el logotipo de pause y el logotipo de stop; esta selección permite una animación ininterrumpida del proceso de alineación de los dos sistemas de coordenadas.

Presione el botón con el logotipo play.

5. Detenga la animación en cualquier punto con el botón stop y presione el botón con el logotipo expulsión.
6. Presione la tecla q y las teclas shift + q; observe y concluya.
7. Visualice la ventana de árbol y ubíquela junto a la ventana 3D. Si no tiene esta ventana en su visualización, seleccione la opción view de la barra de tareas y escoja tree view en el menú emergente. Asegúrese que la ventana

de árbol está activa y pique la opción edit de la barra de tareas; seleccione highlight.

Desplace el puntero de mouse por la ventana de árbol, abra los diferentes grupos (tal y como se hace en el explorador de Windows); observe y concluya.

8. Observe los diferentes ejemplos que están archivados en la carpeta examples. Los archivos de animación tienen la extensión .dat. Para identificar cuales de estos ejemplos tienen animación, seleccione "todos los archivos" en la ventana de apertura.

9. Breve introducción al espacio de trabajo de Matlab.

- Inicie Matlab.
- En el prompt de Matlab digite guru1.
- Siga las instrucciones que le presenta el programa.

Explore y Experimente a gusto con el programa RoboWorks 2.0, observe y concluya.

LABORATORIO DE ROBÓTICA

Practica de software II

REPRESENTACIÓN DE LA ORIENTACIÓN Y LA POSICIÓN

1. Introducción

MATLAB es una poderosa herramienta para realizar operaciones de tipo matricial, por esta razón resulta muy efectiva para estudiar los procesos de Robótica que involucran la determinación de posición y orientación de sistemas de referencia. La Toolbox Hemero aporta un conjunto de instrucciones que reduce sustancialmente la cantidad de pasos que se deben indicar en el proceso de determinación de la posición y orientación de sistemas de referencia dejando más tiempo para la comprensión de lo que realmente está sucediendo.

2. Requisitos

Se requieren conocimientos en álgebra lineal, de MATLAB, de cinemática y capacidad de manejo abstracto del entorno tridimensional.

3. Objetivos

Adquirir conocimiento y habilidad en el manejo de los conceptos de posición y orientación.

Identificar los comandos necesarios para realizar operaciones de traslación y/o rotación de sistemas de referencia en el espacio tridimensional.

4. Marco Teórico

4.1. Matrices de rotación

Una matriz de rotación es una matriz 3×3 que representa la dirección de los ejes principales de un sistema de coordenadas rotado con respecto a un sistema de referencia fijo.

El espacio en el cual se va a mover el Robot es un volumen tridimensional donde la posición de un punto se determina mediante un vector de tres componentes en un sistema de referencia OXYZ.

OUVW es el marco de referencia solidario con el objeto al cual pertenece el punto. La posición del mismo objeto en el marco de referencia OUVW se obtiene multiplicando el vector original por una matriz que representa la rotación del marco de referencia OUVW respecto del marco de referencia estacionario OXYZ.

Las matrices de rotación básicas se pueden multiplicar entre sí para presentar una secuencia de rotaciones.

En una matriz de rotación T , cada columna representa los principales ejes del sistema de coordenadas rotante con respecto al sistema de referencia fijo.

La inversa de una matriz de rotación T^{-1} es equivalente a su transpuesta T^T y por lo tanto los vectores fila de la matriz T representan los ejes principales del sistema de referencia fijo con respecto al sistema de coordenadas rotado.

4.2. Matrices de transformación homogénea. (MTH)

La matriz de transformación homogénea (MTH) es una matriz 4x4 que representa la transformación de un vector en coordenadas homogéneas de un sistema de coordenadas a otro, indica la posición y la orientación de un sistema tridimensional móvil referenciado a un sistema tridimensional fijo.

La ecuación 1 muestra cómo una MTH está formada por cuatro submatrices de distinto tamaño: una submatriz $R_{3 \times 3}$ que representa la rotación; una submatriz $P_{3 \times 1}$ que corresponde al vector de traslación; una submatriz $F_{1 \times 3}$ que representa una transformación de perspectiva, y una submatriz $W_{1 \times 1}$ que corresponde a un escalado global.

$$T = \begin{vmatrix} R_{3 \times 3} & P_{3 \times 1} \\ F_{1 \times 3} & W_{1 \times 1} \end{vmatrix} = \begin{vmatrix} \text{Rotación} & \text{Traslación} \\ \text{Perspectiva} & \text{Escalado} \end{vmatrix} \quad [1]$$

En general se asumen las componentes de $F_{1 \times 3}$ como nulas y el valor de $W_{1 \times 1}$ la unidad.

Es posible multiplicar varias MTH para presentar una secuencia de posiciones, lo que se puede traducir en una trayectoria paso a paso. El resultado se llama matriz de transformación homogénea compuesta.

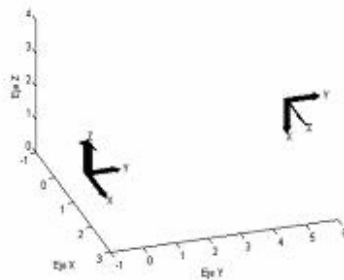
La figura 1 muestra un sistema de referencia OUVW que se encuentra en la posición (2,5,3) con respecto al sistema de referencia OXYZ S_0 y ha rotado 90° alrededor del eje OV; el cual se determina mediante la MTH S_1 .

$$S_0 = {}^0A_0 = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad [2]$$

$$S_1 = {}^0A_1 = \begin{vmatrix} 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & 5 \\ -1 & 0 & 0 & 3 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad [3]$$

Observe la figura y relacione las matrices de las ecuaciones 2 y 3 con la orientación de los vectores unitarios.

Fig 1. Sistemas coordenados.



4.3. Introducción a la Toolbox Hemero (MH)

Hemero (Herramienta Matlab-simulink para el Estudio de manipuladores y Robots móviles) es una Toolbox para Matlab que mediante la escritura de sólo un comando permite ejecutar toda la serie de operaciones de álgebra lineal requeridos para la manipulación de las matrices involucradas en el proceso.

Para conocer en **MH** el entorno tridimensional, se define el sistema de referencia mediante una matriz 4x4 llamada origen o sistema cero S0.

```
>>S0 = [1 0 0 0;0 1 0 0;0 0 1 0;0 0 0 1]
```

```
>>frame (S0, 'c', 1)
```

```
>>grid on
```

Estas instrucciones presentan los ejes ortonormales del origen. La letra c indica el color que se desea para las flechas, y se especifica con una letra:

'c' cyan

'm' magenta

'y' amarillo

'r' rojo

'g' verde

'b' azul

'w' blanco

'k' negro

El número 1 indica tamaño; determina el ancho de las flechas en la visualización.

Traslaciones

El desplazamiento de un sistema de coordenadas puede realizarse de dos formas. Cuando se conocen las coordenadas cartesianas del punto en el espacio tridimensional al cual se desea ubicar el nuevo sistema.

Si se desea desplazar el sistema S0 al punto (3, 5, 7)

```
>>S1 = transl (3,5,7) * S0
```

```
>>frame (S1,'g',1)
```

La otra forma es definir el vector posición que determina el punto al cual se desea ubicar el sistema y luego hacer el desplazamiento con base en ese vector.

```
>>V = [3;5;7]
```

```
>>S1 = transl (V) * S0
```

```
>>frame (S1,'g',1)
```

Rotaciones

Rotar el sistema de coordenadas significa cambiar la orientación de los ejes del mismo girándolo alrededor de un vector específico R.

```
>>R = [1;2;3]
>>S2 = rotvec1 (R, pi/2) * S0
>>frame (S2,'g',0.1)
```

La expresión entre paréntesis indica el vector alrededor del cual hay que rotar siguiendo la regla de la mano derecha, y el ángulo a rotar que se debe indicar en radianes. Si se desea rotar alrededor de uno de los vectores unitarios del sistema S_0 , entonces se usan las instrucciones:

```
>>Sa = rotx (pi/3) *S0
```

Para rotar $\pi/3$ radianes alrededor del eje x.

```
>>Sb = roty (pi/5)*S0
```

Para rotar $\pi/5$ radianes alrededor del eje y.

```
>>Sc = rotz(pi/6)*S0
```

Para rotar $\pi/6$ radianes alrededor del eje z.

Localización en el Espacio

Localizar un objeto en el espacio tridimensional involucra desplazamientos y rotaciones, lo que exige precisión en el orden en el cual se van a realizar los movimientos, porque estos son el resultado de productos matriciales consecutivos.

Si se desea rotar el sistema S_0 $\pi/7$ alrededor del eje y y luego desplazarlo al punto (2,5,7) del sistema:

```
>>St = roty (pi/7) * S0  
>>S4 = transl(2,5,7)*St  
>>frame (S4,'y',1)
```

Si se desea desplazar el sistema al punto (1,4,2) y rotarlo $\pi/7$ alrededor del eje y del propio sistema desplazado (eje v); se realiza todo el proceso en una sola instrucción:

```
>>S5 = transl (1,4,2)* roty (pi/7)*S0  
>>frame (S5,'k',1)
```

Es importante mencionar la función de la transformada inversa, ya que la *MTH* que expresa S_4 en S_0 es S_4 , pero la matriz que expresa S_0 en S_4 es precisamente S_4^{-1} y se va a llamar IS_4 .

```
>>IS4=trinv(S4)
```

Esto muestra que un sistema de coordenadas se puede expresar en términos de otro sistema dependiendo del marco de referencia que se esté manejando.

Para información más detallada de las funciones de cada instrucción puede digitar directamente en Matlab:

```
>>help instrucción
```

Para ver una guía sobre el uso de los comandos de la Toolbox Hemero puede digitar directamente en Matlab:

```
>>guru2
```

Para recordar algunos comandos básicos de Matlab consulte la guía de laboratorios de Sistemas de Control Clásico y Digital.

Tabla 1. Instrucciones MH requeridas para la traslación y orientación.

	Propósito
eul2tr	Generar a partir de los ángulos de Euler Z-Y-Z la transformación homogénea correspondiente
frame	Representar gráficamente un cuadro de referencia con una posición y orientación dadas
ishomog	Comprobar si el argumento que se le pasa es una transformación homogénea
rotvec	Calcular la transformación de rotación en torno a un vector
rotx, roty, rotz	Calcular las transformaciones de rotación en torno a los ejes \hat{X} , \hat{Y} y \hat{Z} respectivamente
rpy2tr	Obtener la transformación homogénea correspondiente a los ángulos RPY dados
transl	Cálculos asociados a transformaciones de traslación
tr2eul	Obtener los ángulos de Euler Z-Y-Z a partir de una matriz de transformación homogénea
tr2rpy	Extraer de una transformación homogénea sus correspondientes ángulos RPY
trinv	Calcular la transformación inversa de una dada

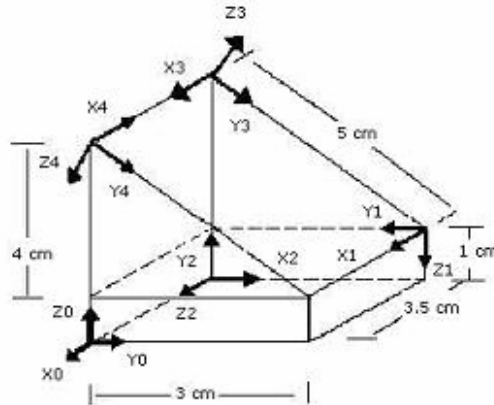
5. Procedimiento

1. Determine la **matriz de rotación** para una rotación de $f = 30^\circ$ respecto al eje OY, seguida por una rotación de $\alpha = 90^\circ$ respecto al eje OW, seguida por una rotación de $a = 45^\circ$ respecto al eje OZ.

2. Obtenga una MTH T que represente una rotación de ángulo a respecto al eje OZ, seguida por una traslación de b unidades de distancia a lo largo del eje OZ, seguida de una rotación de ángulo f respecto al eje OV. Respuesta en términos de a, b, f .

3. Para la figura 2, encuentre las MTH 4×4 ${}^{i-1}A_i$ y 0A_i para $i = 1, 2, 3, 4$.

Fig 2 Sistemas en movimiento.



Elabore un pre-informe a mano en el que incluya:

1. Introducción: En la que defina brevemente el significado y función de las matrices de rotación y de transformación homogénea. (ejemplos y dibujos)
2. Objetivos: Un párrafo explicando los objetivos de esta práctica con sus propias palabras.
3. Desarrollo: Describa los pasos a seguir para solucionar cada punto y realice claros dibujos referentes a cada situación de los elementos que pueda identificar, sin la ayuda del computador; obtenga las *MTH* con los respectivos sistemas de coordenadas solidarios. (Procure usar colores eje x verde, eje y azul y eje z rojo).

LABORATORIO DE ROBÓTICA

Práctica de software III

VARIACIÓN DEL SISTEMA DE REFERENCIA

1. Introducción

Las matrices de transformación homogénea (*MTH*) representan matemáticamente la posición y la orientación de un sistema móvil con base en un sistema de referencia fijo. Las instrucciones presentadas en la herramienta de MATLAB Hemero (**MH**) facilitan la determinación de posición y orientación de sistemas de referencia mediante operaciones de producto y transformada inversa.

2. Requisitos

Se requieren conocimientos en álgebra lineal, conocimiento y habilidad en el manejo de los conceptos de posición y orientación, comprensión de los comandos de MATLAB y de la herramienta Hemero para operaciones de rotación y/o traslación.

3. Objetivos

- Adquirir conocimiento y habilidad en el manejo de las *MTH*.
- Reconocer y entender el significado de la transformada inversa de una *MTH*.
- Identificar los comandos en **MH** necesarios para realizar operaciones de traslación, rotación y cambio de sistemas de referencia en el espacio tridimensional.

4. Marco Teórico

4.1 Operaciones con *MTH*

El sistema de referencia S1 está determinado con respecto al origen S0 mediante una *MTH* 0A_1 , igualmente un sistema de referencia S2 está determinado con respecto a S1 mediante la *MTH* 1A_2 .

Producto

El sistema de referencia S2 estará referenciado al sistema S0 mediante la *MTH* 0A_2 que se obtiene realizando el producto matricial mostrado en la ecuación 1.

$${}^0A_2 = {}^0A_1 * {}^1A_2 \quad [1]$$

De la misma manera se realizan los productos de las diferentes *MTH* consecutivas para obtener la cadena cinemática de los eslabones de un Robot.

Transformada inversa

Se sabe que la inversa de una matriz de rotación T^{-1} es equivalente a su transpuesta T^T y por lo tanto los vectores fila de la matriz T representan los ejes principales del sistema de referencia fijo con respecto al sistema de coordenadas móvil.

Las *MTH* están formadas por cuatro (4) submatrices dos (2) de las cuales deben permanecer constantes, la de perspectiva y la de escalado, por lo cual no se puede aplicar la transpuesta.

$$T = \begin{vmatrix} R_{3 \times 3} & P_{3 \times 1} \\ F_{1 \times 3} & W_{1 \times 1} \end{vmatrix} = \begin{vmatrix} \text{Rotación} & \text{Traslación} \\ \text{Perspectiva} & \text{Escalado} \end{vmatrix} = \begin{vmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad [2]$$

Para la *MTH* mostrada en la ecuación 2:

- El vector n representa las coordenadas del eje OU referenciado al sistema OXYZ.
- El vector s representa las coordenadas del eje OV referenciado al sistema OXYZ.
- El vector a representa las coordenadas del eje OW referenciado al sistema OXYZ.
- El vector p representa la posición del origen del sistema OUVW referenciado al sistema OXYZ.

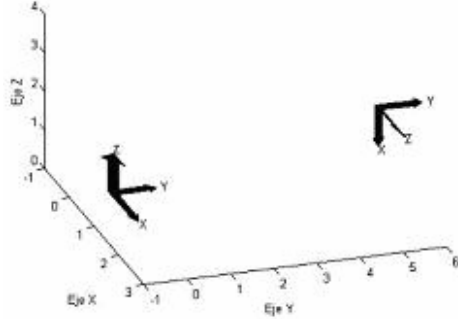
Para expresar el sistema OXYZ referenciado al sistema OUVW se aplica la técnica de la transformada inversa a la matriz T; cuya operación se explica en la ecuación 3.

$$T^{-1} = \begin{vmatrix} n_x & n_y & n_z & -n^T p \\ s_x & s_y & s_z & -s^T p \\ a_x & a_y & a_z & -a^T p \\ 0 & 0 & 0 & 1 \end{vmatrix} = \begin{vmatrix} R_{3 \times 3}^T & \begin{matrix} -n^T p \\ -s^T p \\ -a^T p \end{matrix} \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad [3]$$

Un sistema de referencia OUVW que se encuentra en la posición (2,5,3) con respecto al sistema de referencia OXYZ (representado por S0) y que ha sido rotado 90° alrededor del eje OV queda determinado mediante la *MTH* S1.

Es posible referenciar el sistema OXYZ al sistema OUVW, mediante una *MTH* S1 obtenida aplicando la técnica de la transformada inversa a la matriz S1.

Fig. 1. Sistemas y matrices.



$$S0=^0A_0 = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad [4]$$

$$S1=^0A_1 = \begin{vmatrix} 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & 5 \\ -1 & 0 & 0 & 3 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad [5]$$

Observando la figura 1 y las matrices respectivas representadas en las ecuaciones 4 y 5, realice las operaciones necesarias para obtener la *MTH* $S1=^1A_0$ de la ecuación 6 y que representa el sistema OXYZ referenciado al sistema OUVW.

$$S1=^1A_0 = \begin{vmatrix} 0 & 0 & -1 & 3 \\ 0 & 1 & 0 & -5 \\ 1 & 0 & 0 & -2 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad [6]$$

4.1. Toolbox Hemero (MH)

Producto de Matrices

En la **MH** se realizan los productos entre *MTH* de manera directa.

En el caso en que se conocen las MTH^0A_1 y 1A_2 que están representadas por S1 y S2 respectivamente, y se desea obtener la MTH^0A_2 que representa al sistema S2 referenciado al sistema S0 y va a ser llamada S20 se muestran las instrucciones necesarias y la ecuación 7 con la matriz resultante.

```
>>S0 = [1 0 0 0;0 1 0 0;0 0 1 0;0 0 0 1]
>>S1 = [0 0 1 2;0 1 0 5; -1 0 0 3;0 0 0 1]
>>S2 = [0 0 1 0;-1 0 0 3;0 -1 0 1;0 0 0 1]
>>S20 =S1*S2
>>frame (S0, 'k', 1)
>>frame (S1, 'y',1)
>>frame (S20, 'b',1)
>>grid on
```

$$S20 = \begin{vmatrix} 0 & -1 & 0 & 3 \\ -1 & 0 & 0 & 8 \\ 0 & 0 & -1 & 3 \\ 0 & 0 & 0 & 1 \end{vmatrix} [7]$$

Se pueden concatenar varias matrices en una sola operación, sólo se debe estar seguro del orden en el cual se ubican las matrices en la secuencia de productos ya que la multiplicación de matrices es una operación no conmutativa.

Transformada Inversa

La representación del sistema S_0 referenciado al sistema S_2 se denota 2A_0 y está representada por la matriz S_{02} de la ecuación 8. Ésta matriz se obtiene aplicándole la operación de transformada inversa a la $MTH {}^0A_2$ la cual está representada por la matriz S_{20} .

>S02 =trinv(S20)

$$S_{02} = \begin{bmatrix} 0 & -1 & 0 & 8 \\ -1 & 0 & 0 & 3 \\ 0 & 0 & -1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [8]$$

Para información más detallada de las funciones de cada instrucción puede digitar directamente en Matlab:

>>help *instrucción*

Para ver una guía de uso de los comandos de la **MH** puede digitar directamente en Matlab:

>>guru3

Para recordar algunos comandos básicos de Matlab consulte la guía de laboratorios de Sistemas de Control Clásico y Digital.

5. Procedimiento

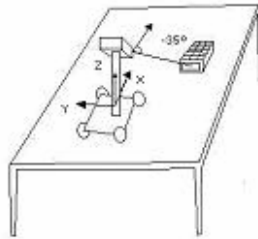
1. Sobre una mesa rueda libremente un robot móvil que tiene un sistema de coordenadas accesorio cuyo eje principal, eje x , apunta en la dirección *adelante*. Posee una cámara de TV ubicada en el extremo de una vara que la fija rígidamente a una altura sobre el plano de la mesa de 20 unidades de distancia. La cámara apunta hacia adelante (paralela al eje principal del móvil) y mantiene una inclinación permanente hacia el piso de -35 grados con respecto a la horizontal. Observe la figura 2. El móvil se desplaza por la superficie de la mesa, con base en un sistema de coordenadas tridimensionales cuyo origen está localizado en el centro del plano de la mesa, y se detiene en un punto tal que la base de la cámara queda ubicada en $(-5, 8)$ con una orientación de su eje principal formando un ángulo de 15° con el eje x .

La cámara observa un obstáculo cúbico fijo sobre la mesa y determina que el centro de éste con respecto a ella se encuentra determinado por la *MTH* $A1$.

$$A1 = \begin{vmatrix} 0.7917 & 0.2123 & -0.5729 & 12.6125 \\ -0.2590 & 0.9659 & 0 & 0.1888 \\ 0.5533 & 0.1484 & 0.8196 & -12.5355 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad A2 = \begin{vmatrix} -0.6567 & 0.4905 & -0.5729 & 15.1045 \\ -0.5985 & -0.8011 & 0 & -1.6118 \\ -0.4589 & 0.3428 & 0.8196 & -10.7938 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

- a) Determine la *MTH* del centro del cubo respecto al origen.
- b) Halle la *MTH* del origen respecto a la cámara.
- c) Calcule la longitud de la arista del cubo.
- d) Encuentre la posición de la cámara para que la *MTH* del centro del cubo respecto a la cámara corresponda a la matriz $A2$. (Opcional)

Fig. 2. Móvil, Cámara y cubo



2. Se ha preparado una estación de trabajo de robot con una cámara de TV, tal como se muestra en la figura. La cámara puede ver el origen del sistema de coordenadas de la base donde se fija un robot de seis elementos, y también el centro de un cubo que tiene que ser manipulado por el robot. Si se ha establecido un sistema de coordenadas local en el centro del cubo, entonces este objeto, tal como lo ve la cámara, se puede representar por una *MTH* T_1 .

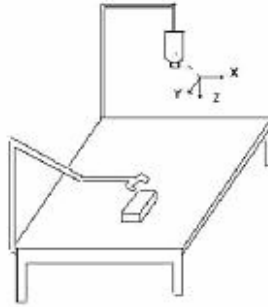
También el origen del sistema de coordenadas de la base como lo ve la cámara se puede expresar por una *MTH* T_2 , donde:

$$T_1 = \begin{vmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 10 \\ 0 & 0 & -1 & 9 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad T_2 = \begin{vmatrix} 1 & 0 & 0 & -10 \\ 0 & -1 & 0 & 20 \\ 0 & 0 & -1 & 10 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

- Después de que se ha colocado el equipo y se han tomado estos sistemas de coordenadas, alguien gira la cámara 90° respecto del eje z de la misma. ¿Cuál es la posición y orientación de la cámara con respecto al sistema de coordenadas de la base del robot?
- Luego de calcular la respuesta a la pregunta a), la misma persona giró el objeto 90° respecto del eje X del objeto y lo trasladó cuatro unidades de distancia a lo largo del eje y girado. ¿Cuál es la posición y orientación

del objeto con respecto al sistema de coordenadas de la base del robot?
¿Y con respecto al sistema de coordenadas de la cámara girada?

Fig. 3. Cámara, Robot y Objeto.



Elabore un pre-informe, a mano, en el que incluya.

1. Introducción: En la que explique brevemente el significado y función de las *MTH*. (ejemplos y dibujos)
2. Objetivos: Un párrafo explicando los objetivos de esta práctica con sus propias palabras.
3. Desarrollo: Descripción de los pasos a seguir para solucionar cada punto y realice claros dibujos referentes a cada situación de los elementos que pueda identificar, sin la ayuda del computador; obtenga las *MTH* con los respectivos sistemas de coordenadas solidarios. (Procure usar colores eje x verde, eje y azul y eje z rojo).

LABORATORIO DE ROBÓTICA

Practica de software IV

MODELOS CINEMÁTICOS DE ROBOTS

1. Introducción

Un Robot manipulador está constituido por varios elementos llamados eslabones, que están unidos entre sí mediante juntas las cuales permiten algún movimiento relativo llamadas articulaciones. El conjunto eslabón articulación otorga al Robot un grado de libertad y es una norma usada para describir a los manipuladores.

Para lograr una observación precisa del movimiento (orientación y traslación) de cada eslabón y del efector final se requiere establecer un conjunto de parámetros que describa la relación entre los eslabones asociando un sistema de coordenadas a cada grado de libertad y utilizar las propiedades de las matrices de transformación homogénea (*MTH*) que relacionan dichos sistemas de coordenadas.

2. Requisitos

Se requiere un total entendimiento de la interpretación, operaciones y propiedades de la *MTH*, habilidad con los comandos de MATLAB y la herramienta Hemero (**MH**) para realizar dichas operaciones, conocimientos de la representación Denavit-Hartenberg y el algoritmo de la misma.

3. Objetivos

- Identificar los tipos de articulaciones.

- Reconocer el concepto de variable articular.
- Familiarizarse con el algoritmo D-H y aprender a aplicarlo a cualquier Robot.
- Conocer el significado de los parámetros D-H e identificarlos fácilmente en cualquier Robot.
- Identificar los comandos en **MH** necesarios para configurar un Robot y determinar posiciones del mismo en el entorno tridimensional.

4. Marco Teórico

4.1. Articulaciones de los Robots

- Articulación rotacional.

En la cual los dos eslabones se encuentran atravesados por un eje imaginario que pasa justo por la unión y es alrededor de este eje que uno de los dos eslabones rota mientras el otro permanece fijo.

- Articulación prismática.

En la cual un eslabón yace sobre casi la totalidad de la longitud del otro o se encuentra adentro del otro y es la dirección de esta longitud la que determina el eje de articulación que permite que uno de los eslabones se deslice telescópicamente en su dirección mientras el otro eslabón permanece fijo.

4.2. La representación Denavit-Hartenberg

Es un método matricial para representar las relaciones translacional y rotacional entre elementos adyacentes de un Robot. En 1955 Denavit y Hartenberg propusieron éste método para establecer de manera sistemática un conjunto de coordenadas $\{S_i\}$ ligado a cada eslabón i de una cadena

articulada, pudiéndose determinar a continuación las ecuaciones cinemáticas de la cadena completa.

Para fines prácticos se va a utilizar la representación de Denavit-Hartenberg de Craig, que propone Ollero con el fin de introducir los datos en la herramienta Hemero.

4.3. Algoritmo Denavit-Hartenberg

A continuación se presentan los pasos a seguir para determinar la representación D-H de un Robot.

1. Establecer el eje de coordenadas de la base.

Se determina un sistema de coordenadas dextrógiro $S_0 = (x_0, y_0, z_0)$ en la base del robot, con el eje z_0 apuntando en la dirección del eje de la primera articulación.

2. Iniciar y repetir.

Para cada $i = 1, 2, \dots, n$; $n =$ número de grados de libertad del Robot, repetir los pasos 3 a 6.

3. Establecer los ejes coordenados de cada articulación.

Se hace coincidir el eje z_i con el eje de la articulación. En la dirección de desplazamiento para articulaciones prismáticas y en la dirección del eje de rotación para articulaciones rotacionales.

4. Establecer el origen del sistema de coordenadas i .

Se localiza el sistema de coordenadas i en la intersección de los ejes z_i y z_{i-1} o en la intersección de la normal común a los ejes z_i y z_{i-1} con el eje z_i .

5. Establecer el eje x_i .

Se determina el eje x_i en la dirección de la línea perpendicular común a los ejes z_i y z_{i-1} cuando estos son paralelos, por lo demás se sitúa apuntando a lo largo del eslabón i en dirección a la articulación $i+1$.

6. Establecer el eje y_i .

El eje y_i se ubica en la dirección que complete un sistema dextrógiro cumpliendo con la regla de la mano derecha con los vectores z_i y x_i .

7. Establecer el eje de coordenadas de la mano.

En la gran mayoría de los casos la articulación n -ésima es de tipo rotacional, por lo que se establece el eje z_n en la dirección del eje z_{n-1} , los otros ejes se sitúan de manera que completen un sistema dextrógiro.

8. Encontrar los parámetros D-H.

Los parámetros Denavit-Hartenberg mostrados en la tabla 1 definen las relaciones dimensionales entre los eslabones consecutivos y las variables articulares. Repetir los pasos de 9 a 13 para cada grado de libertad.

9. Encontrar a_{i-1} .

a_{i-1} es el ángulo entre el eje de la articulación $i-1$ y el eje de la articulación i ; medido como una rotación del eje $i-1$ alrededor de la perpendicular común hasta hacerlo coincidir con la dirección del eje i . a_{i-1} es el ángulo alrededor de x_{i-1} que debe girar este sistema para hacer coincidir los orígenes $(i-1)$ -ésimo e i -ésimo.

10. Encontrar a_{i-1} .

a_{i-1} es la distancia desde el eje de la articulación $i-1$ hasta el eje de la articulación i medida a lo largo de la línea perpendicular común a estos ejes. Cuando los ejes $i-1$ e i se cortan a_{i-1} vale cero. a_{i-1} es la distancia a lo largo del eje x_{i-1} que separa el origen $(i-1)$ -ésimo del origen i -ésimo.

11. Encontrar q_i .

q_i representa un giro alrededor del eje z_i hasta alinear el eje x_{i-1} con el eje x_i siguiendo la regla de la mano derecha pero también q_i es el ángulo de la variable articular para una articulación de rotación.

12. Encontrar d_i .

d_i representa la distancia desde el eje x_{i-1} hasta el eje x_i a lo largo del eje z_i , pero también d_i es la distancia de la variable articular para una articulación prismática.

13. El parámetro d_i .

En la **MH** se necesita el parámetro d_i para indicar el tipo de articulación. Se le asigna valor cero (0) para articulaciones rotacionales y valor uno (1) para las articulaciones prismáticas.

La representación D-H de un Robot permite conocer los movimientos de éste por las transformaciones sucesivas que relacionan el sistema de coordenadas i -ésimo con el sistema de coordenadas $(i-1)$ -ésimo realizando las operaciones matriciales de transformación homogénea básicas. Así se obtiene una matriz de transformación homogénea compuesta, ${}^iA_{i-1}$, llamada matriz de transformación D-H mostrada en la ecuación 1.

$${}^{i-1}A_i = T_{Z,d} T_{Z,q} T_{X,a} T_{X,\alpha} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos q_i & -\operatorname{sen} q_i & 0 & 0 \\ \operatorname{sen} q_i & \cos q_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$* \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos a_i & -\operatorname{sen} a_i & 0 \\ 0 & \operatorname{sen} a_i & \cos a_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos q_i & -\cos a_i \sin q_i & \sin a_i \sin q_i & a_i \cos q_i \\ \sin q_i & \cos a_i \cos q_i & -\sin a_i \cos q_i & a_i \sin q_i \\ 0 & \sin a_i & \cos a_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [1]$$

Tabla 1. Parámetros Denavit-Hartenberg (Según Craig).

ARTICULACIÓN i	a_{i-1}	a_{i-1}	q_i	d_i	
1	a_0 se asume cero	a_0 se asume cero	q_1 será cero si la articulación es prismática.	d_1 será cero si la articulación es de rotación.	
i	En general, a_{i-1} es el ángulo entre el eje Z_{i-1} y Z_i medido mediante la regla de la mano derecha con respecto a X_{i-1} .		Se toma a_{i-1} como la distancia que separa los ejes Z_{i-1} y Z_i medido a lo largo de X_{i-1} , si Z_{i-1} y Z_i se cortan, $a_{i-1} = 0$.	El ángulo q_i se mide mediante la regla de la mano derecha entre el eje X_{i-1} y el eje X_i con respecto al eje Z_i .	La distancia d_i se mide entre el eje X_{i-1} y el eje X_i a lo largo del eje Z_i .
La última articulación, n , si es de rotación, X_n está en línea con X_{n-1} cuando $q_n = 0$. Si es prismática, el origen de n está en la intersección de Z_n con X_{n-1} cuando $d_n = 0$.	a_{n-1}	a_{n-1}	q_n	d_n	

Esta matriz ${}^iA_{i-1}$ está determinada para un grado de libertad; para el caso de un Robot con n grados de libertad es necesario hallar las n matrices de transformación D-H y luego multiplicarlas consecutivamente como en la ecuación 2, para obtener 0T_n , la matriz de transformación homogénea del efector final referenciada al sistema de origen del Robot, que generalmente es el sistema de referencia de la base del mismo:

$${}^0A_1 * {}^1A_2 * {}^2A_3 * {}^3A_4 * \dots * {}^{n-1}A_n = {}^0T_n \quad [2]$$

4.4. Toolbox Hemero

En la **MH** se determina el modelo de un Robot guardando sus parámetros D-H en una matriz usualmente llamada dh.

La *MTH* asociada a cada uno de los enlaces se puede obtener y es de gran importancia, porque permite realizar un monitoreo de cada una de las articulaciones que conforman un Robot ya sea de manera individual, por grupos de interés o la cadena cinemática en su totalidad.

Las variables articulares toman el valor de los datos que se guardan en una matriz generalmente llamada q. Esta matriz puede ser una matriz fila para asignar un valor específico a cada variable articular, pero puede también ser una secuencia de posiciones de manera que cada fila de la matriz q corresponde a un fotograma que indica los valores que toman las variables articulares en cada paso.

En la visualización de los Robots **MH** grafica los eslabones como barras que unen los orígenes de los sistemas de coordenadas consecutivos mediante la instrucción plotbot.

```
>>plotbot(dh,q,'d,f,w,l,r')
```

En el paréntesis se especifica dh correspondiente a la matriz de los parámetros D-H, q es la matriz de los valores de las variables articulares y entre las comillas se designan los accesorios que se pueden escoger para mejorar la visualización.

- La opción d se utiliza para una representación en 2 dimensiones.
- La opción f dibuja los ejes de los sistemas de coordenadas asociados a cada articulación excepto el sistema del efector final.

- El sistema de coordenadas del efector final se visualiza mediante la opción w.
- La opción l se utiliza cuando no se desea que se borre el Robot en cada paso.
- Si se desea repetir una secuencia de movimientos la opción r ejecuta la secuencia de movimientos y la repite 50 veces.

Para el Robot de dos (2) grados de libertad mostrado en la figura 1 se encontró que sus parámetros D-H corresponden con los mostrados en la tabla 2.

Se ha asignado el valor cero (0) a las dos variables articulares aunque para **MH** este valor es irrelevante debido a que los valores serán tomados de la matriz q.

Fig. 1. Robot con dos grados de libertad.

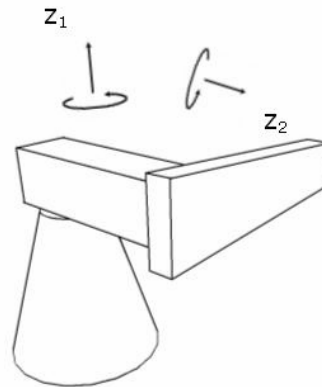


Tabla 2. Parámetros D-H del Robot de la figura 1.

ARTICULACIÓN i	a_{i-1}	a_{i-1}	θ_i	d_i	s_i
1	0	0	0	0.2	0
2	$-\pi/2$	0	0	0.15	0

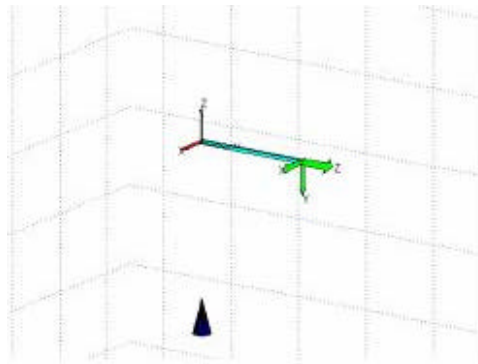
Para introducir los datos en **MH** se digita:

```
>>dh=[0 0 0 0.2 0;-pi/2 0 0 0.15 0]
```

```
>>q=[0 0]
```

```
>>plotbot(dh,q,'f,w')
```

Fig. 2. Modelo en **MH** del Robot de la figura 1.



La figura 2 muestra un acercamiento de la imagen obtenida luego de digitar los comandos anteriormente indicados donde el origen del Robot está representado por la base del cono en la parte inferior de la figura. El primer grado de libertad está representado por S1 y coincide con z_1 y el eslabón asociado corresponde a la distancia desde el origen hasta el sistema de coordenadas S1. El segundo grado de libertad está determinado por el sistema S2 y corresponde a z_2 y su eslabón asociado va desde el sistema de ejes de S1 hasta el sistema de ejes coordenados S2.

Como no se ha asignado un conjunto de parámetros para el extremo final del brazo, este no se muestra, ya que el último sistema de ejes coordenados que fue introducido es el de la posición del sistema S2.

Es posible ver una animación del movimiento de las articulaciones; para tal fin la herramienta necesita los valores de las variables articulares de cada paso; digite:

```
>>q=[0 0; pi/4 0; pi/2 0; (3*pi)/4 0; pi 0; pi pi/4; pi pi/2; pi (3*pi)/4; pi pi]
>>plotbot(dh,q,'f,w,r)
```

Verá cincuenta (50) veces pasar cada grado de libertad de la posición cero (0) a la posición pi.

De esta manera se puede posicionar el Robot en cualquier combinación de valores para las variables articulares y determinar secuencias especiales de movimiento y observar así su comportamiento y comprobar la asignación de los valores de la matriz dh.

Para ver una guía de uso de los comandos de la Toolbox Hemero puede digitar directamente en Matlab:

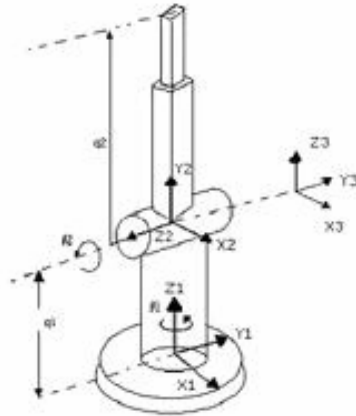
```
>>guru4
```

5. Procedimiento

Para el Robot que se muestra en la figura 3. elabore un programa en el editor de Matlab que le permita realizar los siguientes puntos.

1. Asigne a q1 el valor de 10 y a q2 un valor mínimo de 9 y un máximo de 15 y obtenga la matriz dh.
2. Muestre el modelo del Robot en **MH**.
3. Presente una animación en la que muestre el movimiento de cada una de las articulaciones y luego una combinación de todas.

Fig. 3. Robot con tres grados de libertad.



Elabore un pre-informe a mano en el que incluya:

4. Introducción: En la que explique brevemente los términos eslabón, articulación, grado de libertad y tipos de articulación. (ejemplos y dibujos)
5. Objetivos: Un párrafo explicando los objetivos de esta práctica con sus propias palabras.
6. Desarrollo: Solucione los puntos enunciados a continuación (sin computador) y realice claros dibujos, con los respectivos sistemas de coordenadas solidarios. (Procure usar colores eje x verde, eje y azul y eje z rojo).
 - a. Dibuje los sistemas de referencia para las articulaciones e identifique el tipo de articulación de cada una.
 - b. Determine un sistema de coordenadas para el extremo final del Robot.
 - c. Encuentre los parámetros de D-H para el Robot mostrado en la figura 3 y preséntelos en una tabla incluyendo el parámetro s_i .

LABORATORIO DE ROBÓTICA

Practica de software V

LA CADENA CINEMATICA

1. Introducción

La cadena cinemática es el resultado de articular los eslabones para formar el brazo. La ubicación del efector final respecto del sistema de referencia de la base se obtiene del producto de las matrices que representan el sistema de ejes coordinados solidarios a cada articulación.

A partir de la matriz D-H y de los valores que se le asignen a las variables articulares la **MH** genera las *MTH* que relacionan articulaciones adyacentes y realiza el producto de las mismas para obtener la ubicación del efector final.

2. Requisitos

Se requiere de un conocimiento completo de los tipos de articulaciones de Robots, habilidad en la aplicación del algoritmo D-H, capacidad para modelar diferentes configuraciones de Robots en **MH**,

3. Objetivos

- Obtener la *MTH* del efector final de cualquier Robot.
- Entender el significado de la *Matriz de Brazo*.
- Adquirir habilidad en la elaboración y animación de modelos de Robots en **MH**.
- Entender los conceptos de volumen de trabajo, destreza y restricción.

4. Marco Teórico

4.1. Definición de Robot

Según el Robot Institute of America, un Robot es un manipulador multifuncional reprogramable, diseñado para mover materiales, elementos, herramientas o dispositivos especiales mediante movimientos variables programables con el fin de desempeñar una variedad de tareas.

4.2. Arquitectura de los Robots

Las articulaciones empleadas en la construcción de Robots son únicamente de rotación y prismáticas, que otorgan al Robot un grado de libertad cada una, por lo que el número de grados de libertad coincide con el número de articulaciones.

Se pueden combinar estos tipos de articulaciones para lograr las diferentes configuraciones de Robots, las más frecuentes son:

- Cartesiana
- Cilíndrica
- Esférica o polar
- Angular
- Scara

4.3. Posición del efector final

La MTH que relaciona el sistema de coordenadas i -ésimo con el sistema de coordenadas $(i-1)$ -ésimo se denomina matriz de transformación homogénea compuesta. ${}^iA_{i-1}$, también llamada matriz D-H. Ecuación 1.

$${}^i A_{i-1} = \begin{bmatrix} \cos \mathbf{q}_i & -\cos \mathbf{a}_i \operatorname{sen} \mathbf{q}_i & \operatorname{sen} \mathbf{a}_i \operatorname{sen} \mathbf{q}_i & a_i \cos \mathbf{q}_i \\ \operatorname{sen} \mathbf{q}_i & \cos \mathbf{a}_i \cos \mathbf{q}_i & -\operatorname{sen} \mathbf{a}_i \cos \mathbf{q}_i & a_i \operatorname{sen} \mathbf{q}_i \\ 0 & \operatorname{sen} \mathbf{a}_i & \cos \mathbf{a}_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [1]$$

Esta matriz ${}^i A_{i-1}$ está determinada para un grado de libertad; para n grados de libertad es necesario multiplicar las n matrices D-H para obtener ${}^0 T_n$, que corresponde a la *MTH* del efector final referenciada al sistema de origen como se observa en la ecuación 2.

$${}^0 A_1 * {}^1 A_2 * {}^2 A_3 * {}^3 A_4 * \dots * {}^{n-1} A_n = {}^0 T_n \quad [2]$$

Para un Robot con seis grados de libertad la matriz ${}^0 T_6$ tiene la configuración que se describe en la ecuación 3.

$$T = \begin{bmatrix} x_6 & y_6 & z_6 & p_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^0 R_6 & {}^0 p_6 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} n & s & a & p \\ 0 & 0 & o & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad [3]$$

El significado de esta matriz, denominada "matriz del brazo" lo determina los diferentes vectores columna que la componen.

- \mathbf{n} = vector normal de la mano. Suponiendo una mano del tipo pinza con dedos paralelos, el vector \mathbf{n} apunta en dirección perpendicular a

los vectores **a** y **s** completando un sistema dextrógiro determinado por **sxa**.

- **s** = vector deslizamiento de la mano. Apunta en la dirección en la que se deslizan los dedos cuando se abre y se cierra la mano.
- **a** = vector de aproximación de la mano. Apunta en la dirección normal a la palma de la mano, es normal a la placa de montaje y paralelo a la longitud de los dedos.
- **p** = vector posición de la mano. Apunta desde el origen del sistema de coordenadas de la base hasta el origen del sistema de coordenadas de la mano, el cual se ubica en el punto donde hacen contacto los dedos al cerrar la mano.

4.4. Volumen de Trabajo

El espacio dentro del cual se mueve el efector final está determinado por la arquitectura del mismo; la cual depende del número de grados de libertad, el tamaño de los eslabones, el tipo de articulaciones y la forma como se combinan esas articulaciones de manera consecutiva, y en general se denomina Volumen de trabajo del Robot.

Este espacio se puede dividir en dos (2) espacios claramente diferenciables:

- Espacio alcanzable, o volumen que puede ser alcanzado por el efector final en por lo menos una (1) orientación.
- Espacio de manipulación con destreza, o volumen dentro del cual el efector final puede adoptar todas y cada una de las orientaciones posibles del espacio tridimensional para alcanzar su objetivo.

Para lograr un volumen de trabajo con destreza se requiere que el Robot cuente por lo menos con seis (6) grados de libertad, tres (3) grados de libertad para posicionar el efector final en un punto determinado del espacio y los otros tres (3) grados para permitirle la orientación adecuada. En algunas situaciones

se llega a posiciones degeneradas, que consiste en la pérdida de un grado de libertad porque quedan alineadas dos articulaciones de manera que cualquiera de las dos tiene el mismo efecto en la posición del efector final; en otros casos el volumen de trabajo está ocupado por obstáculos para los cuales se requiere introducir grados de libertad adicionales, que permiten obtener un volumen de trabajo con destreza aun con la presencia del obstáculo, en este caso se denominan grados de libertad redundantes. Algunas tareas requieren de menos grados de libertad, por lo tanto los Robots diseñados específicamente para tales tareas presentan restricciones de movimiento cuando se desean implementar para una función diferente.

4.5. Toolbox Hemero

Para obtener la matriz de brazo, T_n , la *MTH* que relaciona al efector final con el origen del sistema de coordenadas basta con completar la cadena cinemática de las matrices D-H, que son *MTH* que relacionan un grado de libertad con el anterior.

Esta Cadena cinemática puede calcularse mediante el comando `linktrans` y la multiplicación consecutiva de las *MTH* obtenidas. Representa la posición del efector final tal y como quedó ubicado por los valores de las variables articulares que fueron asignados respectivamente por t_i a cada articulación en la respectiva instrucción `linktrans` ejecutada.

Las *MTH* que están asociadas a cada eslabón se recuperan mediante la instrucción `linktrans`.

```
>>Ti = linktrans(dh(i,:),t_i)
```

El paréntesis muestra la matriz `dh`; el término `(i,:)` hace referencia al término i de la matriz `dh` y la expresión t_i define el valor asignado a la variable articular i .

Para obtener las *MTH* de los sistemas coordenados referenciados al sistema anterior digite:

```
>>T1 = linktrans(dh(1,:),0)
```

```
>>T2 = linktrans(dh(2,:),0)
```

```
>>T20 = T1*T2
```

T2 recupera el sistema S2 referenciado a S1; T20 lo referencia al origen.

```
>>Ti = linktrans(dh(:,i),ti)
```

```
>>Tn = T1*T2*...*TN
```

La visualización de **MH** presenta en color rojo, los ejes de coordenadas asociados a cada articulación y en color verde el sistema coordenado asociado al efector final, y es este eje de coordenadas el que determina los valores de la *MTH* obtenida al final de la cadena cinemática; las características del efector final, pinzas, soldador, pistola de pintura, remachador, etc., quedan a la imaginación del usuario, la **MH** se limita a presentar la ubicación del sistema de coordenadas correspondiente.

Para ver una guía de uso de los comandos de la Toolbox Hemero puede digitar directamente en Matlab:

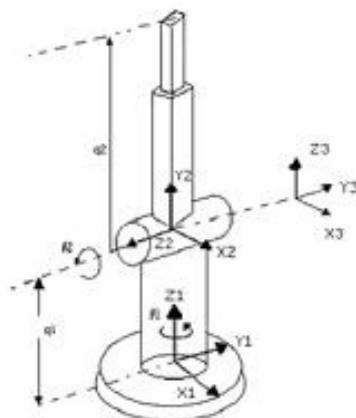
```
>>guru5
```

5. Procedimiento

Recupere el programa de la practica anterior y escriba nuevas líneas que resuelvan los problemas planteados a continuación.

1. Para el Robot de la figura 1 introduzca dos (2) nuevos grados de libertad con ejes de articulación perpendiculares entre sí; que correspondan al soporte rotatorio de una pinza en el extremo del Robot.
2. Obtenga las *MTH* de cada articulación referenciada a la articulación anterior y al origen.
3. Determine:
 - a. Una posición dentro del volumen de manipulación con destreza.
 - b. Una posición dentro del volumen alcanzable.
 - c. Una configuración degenerada.
 - d. Una restricción por dentro del volumen alcanzable.
 Para cada uno de los literales anteriores explique las propiedades de las configuraciones determinadas.
4. Obtenga las *MTH* del efector final referenciado al origen para las posiciones obtenidas en el punto anterior.

Fig. 3. Robot con tres grados de libertad.



Elabore un pre-informe a mano en el que incluya:

1. Introducción: En la que explique brevemente las configuraciones más frecuentes de Robots, los términos manipulación con destreza, volumen alcanzable, configuración degenerada y restricción. (ejemplos y dibujos)
2. Objetivos: Un párrafo explicando los objetivos de esta práctica con sus propias palabras.
3. Desarrollo: Imagine una posible solución al primero y segundo puntos enunciados en el procedimiento.
 - Realice claros dibujos, con los respectivos sistemas de coordenadas solidarios. (Procure usar colores eje x verde, eje y azul y eje z rojo).
 - Desarrolle el algoritmo D-H para esta posible configuración.

LABORATORIO DE ROBÓTICA

Practica de software VI

ANIMACION EN ROBOWORKS 2.0

1. Introducción

El programa RoboWorks presenta una composición sólida de los modelos que se diseñan en **MH**, además de permitir la reproducción de los movimientos que la **MH** genera; con la ventaja de lograr una óptima perspectiva.

Para lograr una mejor comprensión de las características cinemáticas de los modelos de Robots creados en RoboWorks, estos pueden ser animados mediante los archivos tipo *.dat; también se puede observar el estado de las variables articulares usando la ventana del monitor, de esta forma se puede conocer la relación entre las variables articulares y las posición del efector en particular y de todo el cuerpo del Robot en general, en el entorno tridimensional.

2. Requisitos

Habilidad en el manejo de los menús de RoboWorks. , habilidad en la animación de modelos mediante la **MH**.

3. Objetivos

- Adquirir habilidad en el manejo de las opciones de animación del programa RoboWorks.
- Reproducir las animaciones generadas en la **MH** en el correspondiente modelo de RoboWorks.

- Generar archivos tipo *.dat desde el block de notas para la animación en RoboWorks.
- Identificar las variables articulares en un modelo de RoboWorks.

4. Marco Teórico

4.1. Animación en RoboWorks

RoboWorks permite animar los modelos que han sido creados previamente mediante archivos tipo *.dat por alguno de los métodos explicados a continuación.

Cada modelo diseñado en RoboWorks tiene asignado un nombre a cada articulación, esto con el fin de identificar la variable articular y poder asignarle los valores deseados.

4.1.1. La opción monitor

En la barra de tareas, en la opción animation existe una opción llamada monitor, la cual despliega el conjunto de los nombres de las variables articulares, la tecla que comanda a cada articulación y el valor que tiene cada una en la posición actual del modelo, este monitor se está actualizando constantemente, de manera que cuando se mueve el modelo a través del teclado, los valores de las variables articulares indican los cambios introducidos.

4.1.2. Generación en la **MH**

MH genera los archivos tipo *.dat; se crean directamente en Matlab cuando se ejecuta alguno de los modelos que se han presentado como ejemplo para las diferentes funciones de la **MH** y quedan guardados en la carpeta examples de

RoboWorks, de manera que pueden ser llamados por para realizar la animación.

Este archivo contiene en su primer renglón los nombres de las variables articulares, cada nombre determina entonces una columna a la que serán asignados los valores que irá tomando la articulación respectiva a lo largo de toda la animación.

4.1.3. Generación en el block de notas

Los archivos tipo *.dat se pueden escribir directamente en el block de notas de Windows para luego ser llamados y ejecutados desde la opción Animation de RoboWorks o copiarlos en la carpeta examples de RoboWorks y luego ejecutarlos en la opción animation. Para editar estos archivos en el block de notas, es necesario conocer los nombres de las articulaciones y el efecto neto de los incrementos de las mismas en el movimiento de la articulación involucrada y la posición del efector final.

Conociendo los nombres de las variables articulares, la articulación y el tipo al que corresponde cada uno, se puede generar un archivo tipo *.dat para posicionar el modelo, en el caso de que sea un solo renglón de valores para las variables articulares o una animación cuando el archivo consta de varios renglones asignando los valores de las variables articulares para cada paso.

Una vez generado el archivo tipo *.dat en el block de notas, es posible animar el modelo correspondiente siguiendo los mismos pasos para la animación de modelos, sólo que en este caso debe buscarse en el block de notas, o puede haber sido guardado en la carpeta examples de RoboWorks para tener un acceso más directo a la animación.

5. Procedimiento

1. Animación de **MH** a RoboWorks.

En la **MH** dentro del guru6 se encuentran los siguientes subprogramas que modelan las siguientes arquitecturas de Robot del mismo nombre:

- scara
- cartesiano
- humano
- puma
- stanford

Ejecute cada uno de ellos en **MH** y luego despliegue el modelo del mismo nombre creado en RoboWorks para finalmente animarlo mediante el archivo tipo *.dat que fue creado en **MH** durante la ejecución de cada programa.

Ejemplo:

En el prompt de Matlab digite:

```
>>guru6
```

- Seleccione [1] para el Robot Scara.
- Lea cuidadosamente y siga las instrucciones que le presenta el programa.
- Luego de finalizada la ejecución de éste, entre a RoboWorks 2.0.
- Cierre las ventanas emergentes y pique [no] en las opciones de guardar.

- Pique en el logotipo de abrir carpeta de la barra de íconos o pique en file y luego open de la barra de tareas.
- Seleccione Scara.
- Experimente con el teclado para ver los movimientos y realice los ajustes de perspectiva, fondo, tamaño, etc.
- Pique Animation de la barra de tareas y seleccione from file.
- Seleccione el archivo scara.dat
- Realice los ajustes de time delay (200 es recomendable), reproducción continua, etc. Verá una animación de los diferentes grados de libertad que componen el Robot.

2. Escritura de archivos tipo *.dat.

Identifique los nombres de las variables articulares de los modelos presentados, genere un archivo tipo *.dat en el block de notas para posicionar el Robot en una configuración extrema, luego genere una animación que muestre una secuencia de posiciones extremas del modelo.

Ejemplo:

- Recupere el modelo scara en la ventana de RoboWorks.
- Pique en Animation en la barra de tareas.
- Seleccione Monitor.
- Tome nota de los nombres de las variables articulares.

- Pique en la ventana de vista 3D.
- Presione los números 1, 2 y 3 del teclado para observar el movimiento del modelo y los valores de las variables articulares.
- Determine el 'paso' de cada articulación y sus valores máximo y mínimo.
- Abra el block de notas en los accesorios de Windows.
- Escriba en el primer renglón los nombres de las variables articulares separados por un tab.

rot1 rot2 palo

- En el siguiente renglón escriba un valor máximo o mínimo para cada variable articular.
- Pique en "guardar como", asigne un nombre y guarde en la siguiente dirección.
- Archivos de programa/Newtonium/RoboWorks/examples.[guardar]
- Retorne a RoboWorks.
- Inicie la animación del archivo que acaba de guardar.

LABORATORIO DE ROBÓTICA

Practica de software VII

CINEMATICA DIRECTA

1. Introducción

El estudio cinemático de un Robot se centra en el análisis del movimiento de éste con respecto a un sistema de coordenadas de referencia fijo. En particular estudia la relación de los valores de las variables articulares con la posición y orientación del efector final del Robot. Aunque este problema puede ser tomado desde dos (2) puntos de vista, cinemática directa y cinemática inversa, en esta etapa se va a presentar el estudio de la cinemática directa de un Robot.

Ya se mostró una forma de completar la cadena cinemática usando la **MH**, sin embargo existe una manera más ágil de obtener la matriz de brazo en una sola instrucción, la cual sintetiza el problema de la cinemática directa, que depende de la matriz **dh** y de los valores articulares guardados en la matriz **q**.

2. Requisitos

Se requiere conocer las configuraciones de arquitectura básicas de los Robots, obtener y comprender la *matriz de brazo* de un Robot, habilidad en la construcción y animación de modelos en **MH**, comprender el concepto de volumen de trabajo con todo lo que ello implica

3. Objetivos

- Entender el problema de la cinemática directa de un Robot.

- Obtener la *MTH* del efector final de cualquier Robot como la solución al problema cinemático directo del mismo.
- Utilizar las instrucciones vistas en **MH** para conocer la posición de cualquier articulación de un Robot cuando se conoce la ubicación del efector final.
- Aplicar la cinemática directa en los modelos creados en RoboWorks desde la **MH**.

4. Marco Teórico

4.1. El Problema Cinemático Directo

La cinemática directa parte del conocimiento de los valores de las variables articulares y busca la posición y orientación del efector final que corresponde a esos valores. Para ello requiere un conocimiento preciso de la configuración física del Robot y de las relaciones de posición y orientación de los eslabones consecutivos, además del conocimiento de los posibles obstáculos dentro del volumen de trabajo.

El problema de la cinemática directa se puede resumir como la obtención de la matriz de brazo cuando se conoce el valor de todas las variables articulares para una configuración determinada; es resolver la ecuación 1 para obtener 0T_n , que corresponde a la *MTH* del efector final referenciada al sistema de origen.

$${}^0A_1 * {}^1A_2 * {}^2A_3 * {}^3A_4 * \dots * {}^{n-1}A_n = {}^0T_n [1]$$

4.2. Toolbox Hemero

Para obtener la matriz de brazo, T_n , la *MTH* que relaciona al efector final con el origen del sistema de coordenadas no es necesario obtener todas y cada una de las *MTH* que relacionan un grado de libertad con el anterior. La

Instrucción `fkine` realiza todo este proceso internamente y arroja como resultado la *MTH* del efector final tal y como quedó ubicado por los valores de las variables articulares que fueron asignados respectivamente por \ddot{q} a cada articulación en la matriz q ($1 \times n$; donde n es el número de grados de libertad).

```
>>Ti = fkine(dh,q)
```

La matriz T_i puede ser una posición expresada como una *MTH* cuando la matriz q tiene solo una fila, o una matriz tridimensional, $4 \times 4 \times m$, que representa una trayectoria de *MTH* en la que cada fila m es una *MTH* que representa un paso en la secuencia de posiciones de la trayectoria, cuando la matriz q esta conformada por varias filas cada una de las cuales representa un conjunto de valores para las variables articulares.

Es claro que el número de columnas de la matriz q debe coincidir con el número de variables articulares y los datos asignados en estas variables deben ser coherentes con el tipo de articulación al que serán asignados.

Para ver una guía de uso de los comandos de la Toolbox Hemero puede digitar directamente en Matlab:

```
>>guru7
```

5. Procedimiento

Recupere el programa de la practica anterior y escriba nuevas líneas que resuelvan los problemas planteados a continuación.

1. Introduzca los valores de las variables articulares que determinó para el pre-informe en una matriz q y visualice la posición mediante la instrucción `plotbot`; compare con su dibujo y concluya.

2. Obtenga la matriz de brazo en función de las variables articulares mediante las instrucciones:

```
>> q= [t1 t2 t3 t4 t5];  
>> T=simple(fkine(dh,q))
```

3. Obtenga las *MTH* del efector final referenciado al origen de la posición determinada para el pre-informe mediante *fkine* y mediante *linktrans*; compare y concluya.

4. Ejecute los modelos que se encuentran en el *guru5* de la **MH**:

- *scara*
- *cartesiano*
- *humano*
- *puma*
- *stanford*

5. Observe la animación de los respectivos modelos en *RoboWorks* mediante los archivos *Dnombre.dat* que fueron previamente generados durante la ejecución en **MH**.

Elabore un pre-informe a mano en el que incluya:

- **Introducción:** En la que explique claramente el problema de la cinemática directa.
- **Objetivos:** Un párrafo explicando los objetivos de esta práctica con sus propias palabras.
- **Desarrollo:** Recupere el modelo que obtuvo como resultado de agregar 2 grados de libertad en el Robot de la practica anterior.

- Realice claros dibujos, con los respectivos sistemas de coordenadas solidarios. (Procure usar colores eje x verde, eje y azul y eje z rojo).
- Determine un conjunto de valores para las variables articulares (diferentes de cero preferiblemente).
- Dibuje la posible posición del Robot con esos valores.

LABORATORIO DE ROBÓTICA

Practica de software VIII

CINEMATICA INVERSA

1. Introducción

El otro punto de vista del estudio cinemático de un Robot es la cinemática inversa, donde al Robot se le indica donde está el objetivo y él debe tomarlo con notable precisión, procesarlo o moverlo con notable precisión, para lo cual se debe encontrar los valores de las variables articulares que determinan la posición y orientación del efector final.

Se han planteado varios métodos para resolver este problema que en realidad consiste en un sistema de ecuaciones que además admite varias soluciones. La **MH** ofrece la solución de este problema mediante un método numérico iterativo; aunque otros métodos pueden ser implementados para agilizar la respuesta.

Además de recibir las soluciones por parte de la **MH**, o de cualquier otra herramienta matemática para resolver la cinemática inversa, RoboWorks también resuelve este problema de una manera práctica, ubicando mediante el teclado al efector final y tomando los valores del monitor. Un método empírico pero efectivo cuando no hay urgencia por una solución.

2. Requisitos

Se requiere entender el problema cinemático directo, conocer las configuraciones de arquitectura básicas de los Robots, obtener y comprender la *matriz de brazo* de un Robot, habilidad en la construcción y animación de

modelos en **MH**, comprender el concepto de volumen de trabajo con todo lo que ello implica, habilidad en la animación de modelos mediante RoboWorks.

3. Objetivos

- Entender el problema de la cinemática inversa de un Robot.
- Obtener el conjunto de valores para las variables articulares de cualquier Robot como la solución al problema cinemático inverso del mismo.
- Utilizar las instrucciones vistas en **MH** para conocer la posición de cualquier articulación de un Robot para una matriz de brazo específica.
- Aplicar la cinemática inversa en los modelos creados en RoboWorks desde la **MH**.

4. Marco Teórico

4.1. El Problema Cinemático Inverso

Consiste en resolver el sistema de ecuaciones que determina los valores que deben tomar las variables articulares para posicionar correctamente el efector final del brazo Robot con base en unas coordenadas de posición y orientación conocidas.

Este problema parte de la matriz 0T_n , una *MTH* que indica la posición y orientación del destino, las cuales están determinadas por los vectores **n**, **o**, **a** y **p**, y busca el valor para cada una de las variables articulares que posicionan el efector en las coordenadas dadas.

Existen varias técnicas para resolver estas ecuaciones, algunos son métodos netamente geométricos para Robots de pocos grados de libertad, pero con el incremento del número de articulaciones se aumenta la complejidad de la

solución, de manera que se hace necesario recurrir a métodos numéricos y técnicas computacionales iterativas.

4.2. Método geométrico

Este procedimiento es adecuado para Robots con pocos grados de libertad, o simplemente para solucionar la etapa de los tres primeros grados de libertad, que consiste en el posicionamiento del efector final sin tener en cuenta su orientación. El método consiste en encontrar un adecuado número de relaciones geométricas que involucran las coordenadas del efector final, las coordenadas articulares y las dimensiones físicas de los elementos.

4.3. Solución de la matriz de transformación homogénea

Como se parte de una *MTH*, entonces se tendrían dieciséis (16) incógnitas, pero la última fila de esta matriz es $[0 \ 0 \ 0 \ 1]$, por lo que se reduce el problema a doce (12) incógnitas, tres (3) de traslación, determinadas por la cuarta columna $[p_x \ p_y \ p_z]^T$ y nueve (9) de rotación, de las cuales solo tres (3) son independientes. Esto reduce el problema a seis (6) ecuaciones con seis (6) incógnitas no lineales e independientes.

4.4. Desacoplo cinemático

Una técnica muy útil es el desacoplo cinemático, que separa el proceso en dos etapas.

La primera etapa consiste en resolver el problema de posicionar el extremo del Robot, es decir lograr que el origen del sistema de coordenadas de la mano se localice en las coordenadas determinadas por el vector \mathbf{p} de la *MTH* que se va a resolver, "matriz de brazo", y se obtienen los valores de las primeras $n-3$ variables articulares q_1, q_2, \dots, q_{n-3} .

La segunda etapa se concentra en orientar el extremo del Robot basándose en los vectores \mathbf{n} , \mathbf{s} , \mathbf{a} y los resultados de la etapa anterior para calcular los valores de las tres últimas variables articulares q_{n-2} , q_{n-1} , q_n .

Para Robots con menos de seis (6) grados de libertad existen posiciones que sólo pueden ser alcanzadas con una orientación del efector final debido a restricciones de movimiento, es decir para algunas posiciones del efector final se presentan orientaciones prohibidas del mismo. Con seis (6) o más grados de libertad el efector final puede orientarse de cualquier forma y el volumen de trabajo, que queda casi idéntico al volumen alcanzable.

El problema de la cinemática inversa se puede resumir como la obtención del valor de las variables articulares cuando se conoce la matriz de brazo.

4.2. Toolbox Hemero

Para obtener los valores de las variables articulares, matriz q , la **MH** resuelve las ecuaciones que relacionan al efector final con el origen del sistema de coordenadas mediante la instrucción `ikine`, que realiza este proceso internamente utilizando un método numérico, y arroja como resultado la matriz q con los valores de las variables articulares que corresponden a la posición determinada por la *MTH* del efector final.

```
>> stol=1e-6; ilimit=1000  
>> q = ikine(dh,T,stol,ilimit)
```

Stol se refiere a la máxima diferencia que será admitida entre la *MTH* calculada y la *MTH* que fue ingresada en la matriz T .

Ilimit es el número máximo de iteraciones que se ejecutaran para calcular la solución.

Dependiendo de la característica de la matriz T, la respuesta que arroja ikine puede ser un vector fila, cuando la matriz T es una posición, o una matriz cuando T es una trayectoria de transformaciones.

Para Robots con menos de seis (6) grados de libertad, existen algunas restricciones las cuales generarían no convergencia por parte de ikine. Es necesario introducir una matriz M que actúa como una máscara, en la que designa con valor uno (1) las características móviles habilitadas por la arquitectura del Robot, y con valor cero (0) las restricciones del mismo. La matriz M presenta la siguiente interpretación: T para traslación a lo largo del eje determinado y R para Rotación alrededor del eje determinado)

$$M=[T_x T_y T_z R_x R_y R_z]$$

4.3. Animación En RoboWorks

Una vez **MH** ha encontrado el conjunto de valores de las variables articulares, se puede proceder a animar usando el archivo tipo *.dat generado.

Una manera directa de solucionar el problema cinemático inverso consiste en verificar con el monitor el nombre de cada articulación, la tecla que lo acciona, manipular el modelo usando el teclado para posicionar el efector final en la ubicación deseada y finalmente tomar nota del valor que corresponde a cada variable articular. Luego se puede generar la animación directamente en el block de notas, o se puede recurrir a **MH** para generar matrices con los valores que vayan desde cero hasta la posición hallada, construir la matriz de animación q y luego guardarla como archivo tipo *.dat mediante el comando creardat; para finalmente generar la animación en RoboWorks.

Para ver una guía de uso de los comandos de la Toolbox Hemero puede digitar directamente en Matlab:

>>guru8

5. Procedimiento

Recupere el programa de la practica anterior y escriba nuevas líneas que resuelvan los problemas planteados a continuación.

1. Introduzca la *MTH* del efector final del pre-informe y mediante el comando *ikine* obtenga la matriz *q* con los valores de las variables articulares, compare con los valores que determinó para el pre-informe.

2. Obtenga la matriz *q* con los valores de las variables articulares usando el método alternativo y compare con los valores para el punto anterior y el pre-informe.

3. Visualice la posición mediante la instrucción *plotbot*; de cada conjunto de valores de la matriz *q*, es decir un *figure* para los datos del pre-informe, otro para *ikine*; compare con su dibujo y concluya.

4. Ejecute los modelos que se encuentran en el gurut6 de la **MH**.

- *scara*
- *cartesiano*
- *humano*
- *puma*
- *stanford*

5. Observe la animación de los respectivos modelos en RoboWorks mediante los archivos *Inombre.dat* que fueron previamente generados durante la ejecución en **MH**; tal y como se hizo en la práctica anterior.

6. Observe el modelo DH Identifique los nombres de las variables articulares y genere una animación para que el modelo sujete la pieza amarilla con las pinzas.

Guárdela en el diskette de trabajo como anima.dat

Determine aproximadamente la *MTH* del efector final que utilizo para tomar la pieza, puede usar como guía los ejes coordenados asociados a cada articulación.

Elabore un pre-informe a mano en el que incluya:

1. Introducción: En la que explique claramente el problema de la cinemática inversa.
2. Objetivos: Un párrafo explicando los objetivos de esta práctica con sus propias palabras.
3. Desarrollo: Recupere el modelo que obtuvo como resultado de agregar 2 grados de libertad en el Robot de la practica anterior.
 - Recupere la matriz q con los valores de las variables que utilizo para el desarrollo de la práctica y la *MTH* que obtuvo para el efector final y realice claros dibujos, con los respectivos sistemas de coordenadas solidarios. (Procure usar colores eje x verde, eje y azul y eje z rojo).
 - Encuentre otras configuraciones (valores de las variables articulares) que permitan la misma ubicación del efector final.

LABORATORIO DE ROBÓTICA

Practica de software IX

MODELANDO EN ROBOWORKS 2.0

1. Introducción

RoboWorks es un programa de modelado tridimensional para realizar simulación de elementos mecánicos mediante animaciones; estas últimas ya se han manejado, por lo que ahora se va a mostrar la forma de construir los modelos.

RoboWorks maneja un entorno tridimensional gobernado por coordenadas cartesianas, en donde el plano horizontal esta formado por los ejes x y z, y la coordenada vertical es el eje y. Todos los objetos estarán referenciados a este sistema de coordenadas y sus posiciones y rotaciones deben ser especificados en el momento de la construcción.

2. Requisitos

Se requiere entender el problema cinemático directo e inverso, conocer las configuraciones de arquitectura básicas de los Robots, habilidad en la construcción y animación de modelos en **MH**, habilidad en la animación de modelos mediante RoboWorks.

3. Objetivos

- Adquirir conocimiento y habilidad en el manejo de programas tipo CAD.
- Identificar las figuras básicas para construir en RoboWorks.

- Obtener capacidad de descomponer un cuerpo en un conjunto de figuras básicas.
- Reconocer los tipos de articulaciones para implementarlas en los modelos generados en RoboWorks.
- Adquirir habilidad con los cambios de sistema de referencia para optimizar el modelado de Robots.
- Asignar variables articulares e identificar sus funciones.
- Adquirir la habilidad para mover los modelos construidos, ya sea desde el teclado o generando cualquier tipo de animación.

4. Marco Teórico

4.1. Ventana de vista de árbol

En la ventana de vista de árbol aparecerán todos los elementos de modelado a medida que sean creados.

Existen 3 tipos de elementos que podemos crear en esta ventana:

- Formas. Donde se encuentran las figuras básicas para realizar cualquier elemento tridimensional estas son: Cilindros, Conos, Discos, Esferas y Cubos.
- Transformaciones. Presenta todas las herramientas para generar movimiento en los diferentes eslabones de un Robot. Las opciones de movimiento son rotación y/o traslación y afecta a los miembros de la cadena cinemática que se encuentran entre transformation start y transformation stop.

Debe tenerse en cuenta que existen transformaciones estáticas y dinámicas. Estas últimas son traslación y rotación y generan estos mismos movimientos en los eslabones subsiguientes, de ser necesario las transformaciones estáticas permiten trasladar el origen del sistema de referencia para luego a partir de él construir los eslabones, cada uno de los cuales se ensambla utilizando las figuras geométricas básicas, las cuales deben conservar las posiciones relativas

durante el movimiento. Para lograr este efecto las transformaciones se incluyen en un bloque que inicia con la instrucción transformation start y se termina con transformation stop.

- Otros. Ofrece la opción de crear grupos y seleccionar materiales para los Sólidos.

4.1.1. Insert Child

Para insertar un elemento inmediatamente después de la raíz se debe ir al menú "edit" mientras la vista de árbol se encuentre activada y seleccionar "insert child". Después aparecerá una ventana que preguntará que es los que se desea insertar. Este elemento insertado está un subnivel jerárquico posterior al elemento que estaba resaltado cuando se dio el comando.

En esta ventana se encontrarán los tres tipos de elementos mencionados anteriormente: Formas, transformaciones y otros.

- Shape: (Formas) Permite seleccionar una figura tridimensional básica, una vez seleccionada esta figura el programa le presenta una ventana de opciones donde se asigna.
- Dimensions. Tamaño del objeto, coordenadas de la posición del centro del cuerpo, datos específicos referentes a la forma del cuerpo.
- Rendering. Permite cambiar el sombreado de la pieza, normalmente se encuentra con el sombreado de todo el modelo. Las opciones son: sombreado suave, sombreado plano y alambrado. También permite visualizarlo como una figura no sólida, y visualizarlo sombreado la cara delantera la de atrás o las dos.

- Material. Para definir un material únicamente para este modelo, o aplicarle el último material que se tiene predeterminado.

- Transformation: Existen dos tipos de transformaciones que utiliza el RoboWorks para otorgar animación y movimiento al modelo.

- Transformaciones estáticas. Estas transformaciones modifican la posición del punto de referencia de los elementos del dibujo, en otras palabras, permite mover el origen para hacer más sencilla la modelación y facilitar la animación. Tan pronto se selecciona translation o rotation, se abre una ventana con la opción name activa, en la cual se asigna el nombre de la transformación, el eje respecto del cual va a realizarse la transformación, initial value es el valor del cambio en sí.

- Translation sirve para desplazar el punto de referencia una distancia determinada, ya sea en el eje x, el eje y, o el eje z. Para desplazar el origen en una dirección que contenga más de un eje, tendrá que desplazarse en un eje primero y luego en el otro. Así cualquier objeto que se realice después de la transformación, se encontrará desplazado según lo especificado por la traslación. Esto simplifica enormemente el modelado pues así no es necesario recordar la posición de cada objeto con respecto al origen, y permite definir una posición con mayor sencillez.
- Rotation sirve para rotar un ángulo determinado alrededor de uno de los ejes del sistema de referencia que rige al cuerpo que se desea rotar. Esta rotación aplica a todos los cuerpos que se construyan a partir de ella.
- Transformation Start/Stop limita la transformación a los objetos contenidos entre estas dos instrucciones, actúa como un paréntesis separando el resto del modelo de los objetos que están contenidos. Esto

con el fin de realizar cambios muy precisos en segmentos específicos del modelo.

- Transformaciones dinámicas. Estas transformaciones no tienen un valor específico, su magnitud es una variable que depende de las indicaciones dadas permitiendo cambiar la posición de los objetos que domina esta transformación, esto permite lograr la animación de los modelos, ya sea por medio de otro programa, el teclado o un archivo de datos generado por el usuario.

Seleccionando control y luego dynamic se abre una ventana que pregunta por las especificaciones del control.

- Key selection asigna una tecla a la variable articular para activarlo mediante el teclado.
- Increment determina el paso de cambio por cada vez que se presiona la tecla, tanto en grados para rotaciones como en cm. para desplazamientos.
- Tag name etiqueta la variable articular y va a ser ese nombre el que leerá en la tabla del archivo tipo *.dat para asignar el valor a la articulación.
- Límits determina los valores máximo y mínimo para la articulación y se puede escoger que el programa informe de la ocurrencia de alguno de los dos durante la ejecución de una animación.

Las transformaciones dinámicas son las mismas que las estáticas, rotation y translation; con la única diferencia que su valor es variable, así las opciones "Transformation Start" y "Transformation Stop" no pueden ser usadas.

- Others: Define características de tipo grupal para separar diferentes segmentos del modelo y permitir un rápido y lógico acceso a los componentes que se están modelando.
- Group. Inicia una rama a partir de la cual se construyen las partes específicas de una sección del modelo y sólo requiere de un nombre, para luego usar esa ramificación como punto de partida en la construcción.
- Material. Aplica un color a todos los elementos que se van a construir a partir de ese punto, esto facilita limitar el modelo a grupos identificados por elementos del mismo color.

4.1.2. Insert After

Inserta inmediatamente después del elemento seleccionado (que se muestra resaltado con color azul) pero en la misma jerarquía o nivel de construcción.

4.1.3. Cut

Cortar, cuando se desea retirar el elemento seleccionado para luego ubicarlo en otra sección del árbol. El elemento desaparece de la ubicación actual y queda en memoria mientras se busca el nuevo sitio donde será insertado ya sea mediante paste o paste child.

4.1.4. Copy

Copia el elemento seleccionado en memoria para luego ser clonado mediante paste o paste child en una parte del árbol seleccionada, durante el proceso el elemento no desaparece de su posición original porque no va a ser removido.

4.1.5. Paste

Pega el elemento que se encuentra en memoria inmediatamente después del elemento que ha sido escogido como nueva locación del elemento cargado, pero lo pega en el mismo nivel jerárquico.

4.1.6. Paste Child

Pega el elemento residente memoria en un subnivel posterior al del elemento que se ha elegido como redireccionamiento.

5. Procedimiento

Recupere el programa escrito en la práctica anterior, con los dos (2) nuevos grados de libertad.

1. Grafique en la **MH** el modelo para la posición de cero y compárela con el dibujo del pre-informe.
2. Modele las partes restantes del Robot que se presenta como LAB9 en la carpeta examples de RoboWorks, tal y como lo hizo para el pre-informe.
3. Elabore animaciones sencillas que involucren todas las articulaciones en el modelo de la **MH** usando los comandos ikine y fkine.
4. Utilice los archivos tipo *.dat generados en el numeral anterior para animar el modelo de RoboWorks.

Elabore un pre-informe a mano en el que incluya:

- Introducción: En la que explique breve y claramente el programa RoboWorks.
- Objetivos: Un párrafo explicando los objetivos de esta práctica con sus propias palabras.
- Desarrollo: Recupere el modelo que obtuvo como resultado de agregar 2 grados de libertad en el Robot de la practica anterior.
- Recupere la matriz q con los valores de las variables que utilizó para el desarrollo de la práctica y la MTH que obtuvo para el efector final y realice claros dibujos, con los respectivos sistemas de coordenadas solidarios. (Procure usar colores eje x verde, eje y azul y eje z rojo).

ANEXO C. EL CONSTRUCTIVISMO, UN MODELO PEDAGÓGICO

C.1. EL CAMPO EDUCATIVO

La efectividad de un material didáctico reposa sobre el modelo pedagógico que se emplee para su diseño y el sistema educativo en el cual va a ser implementado. La visión que se tenga sobre el conocimiento determina la forma en que se espera enseñar dicho conocimiento. Si el conocimiento se asume como datos que se acumulan uno al lado del otro entonces se procede, en la enseñanza, a presentarle al estudiante datos para que el acumule. Si, por el contrario, y es nuestra propuesta, se asume que el conocimiento es una construcción humana, entonces se procede a hacer que el estudiante experimente por sí mismo esa construcción de modo que haya aprendizaje significativo.

En el ámbito universitario, particularmente en el área tecnológica se desaprovechan tanto la capacidad cognitiva que el estudiante ha venido desarrollando, como la cantidad de información que ha estado integrando a su estructura cognitiva. Asumimos, entonces una fundamentación constructivista en la enseñanza de la tecnología como la mejor fundamentación teórica posible, para asegurar un eficiente proceso educativo.

La metodología utilizada pretende que el estudiante esté en condición de completar la comprensión del conocimiento nuevo por su cuenta y prepararse para las evaluaciones simplemente con presentarle la parte teórica, y en algunas ocasiones la parte práctica, siguiendo un programa de clases pre-establecido. Asume, pues, que el estudiante acumule información que luego pueda utilizar.

Pero en realidad ese proceso que asume que el estudiante está en capacidad de entender completamente cualquier cantidad de información que se le presente por primera vez, y que todos los alumnos poseen los esquemas

necesarios para acoplar este nuevo conocimiento en su estructura cognitiva, resulta un fenómeno contraproducente porque los estudios de psicología cognitiva muestran que el tiempo en los procesos de aprendizaje resulta directamente proporcional a la calidad de comprensión sobre lo aprendido. Igualmente, la disposición previa, o lo que se ha denominado pre-conceptos, son condicionantes a la hora de construir nuevos conocimientos. De este modo, más que apropiarse de nuevos conocimientos lo que hay que entender es que el conocimiento se construye en un tiempo que puede extenderse o acortarse según las estructuras mentales previas que tenga el estudiante. Esto puede conducir a la aparición de una enorme brecha entre lo que los alumnos pueden, y tienen interés por aprender, y lo que les ofrece la institución.

La búsqueda de solución a los problemas mencionados es lo que suele subyacer a la utilización de conceptos y teorías psicológicas en los procesos educativos, es decir, los elementos que deben tenerse en cuenta a la hora de elaborar y concretar una serie de actividades y elementos que conciernen a las capacidades y disposiciones del individuo que aprende.

Para desarrollar una actividad didáctica que tenga en cuenta el modelo constructivista del proceso de aprendizaje, es necesario tener siempre en cuenta los siguientes preceptos[mario carretero].

1. Partir del desarrollo del alumno.
2. Asegurar la construcción de aprendizajes significativos.
3. Posibilitar que los alumnos realicen aprendizajes significativos por sí mismos.
4. Procurar que los alumnos modifiquen sus esquemas de conocimiento.
5. Establecer relaciones ricas entre el nuevo conocimiento y los esquemas de conocimiento ya existentes.

Estos postulados implican la búsqueda de un tipo de enseñanza, mediante actividades y decisiones educativas, que permita reconstruir el nuevo conocimiento a la vez que se busca aumentar la capacidad de solución de

problemas y mayor capacidad crítica, por lo tanto de saber aplicar el nuevo conocimiento y seguir el enriquecimiento del mismo.

Es necesario explicar en qué consiste el constructivismo, que surge de los estudios y aportes de autores como Piaget, Vygotsky, Ausubel y la actual psicología cognitiva.

C.2. ¿QUÉ ES EL CONSTRUCTIVISMO?

Puede tomarse al constructivismo como una tendencia que sostiene que el conocimiento no es una copia de la realidad adquirida mediante la impresión que del mundo se tiene a través de los sentidos y ciertas disposiciones internas determinadas por las condiciones ambientales que influyeron en el desarrollo individual, sino una construcción propia del ser humano, que se produce día a día como resultado de la interacción entre estos dos factores y que va enriqueciendo progresivamente el conocimiento mediante modelos explicativos cada vez más complejos y potentes. Esa construcción se realiza sobre esquemas previos que el individuo ya posee porque ha construido con anterioridad al nuevo momento pedagógico, es decir, con lo que ya construyó en su interacción con el medio que le rodea [mario carretero].

Esta construcción depende de:

- Representación que se tiene de la nueva información.
- Esquemas que se poseen para integrar el nuevo conocimiento.

Los procesos de construcción de conocimiento se sitúan como el eje del desarrollo de los procesos de aprendizaje y esto implica una búsqueda de estrategias que den lugar a ambientes de construcción de conocimiento y en consecuencia de aprendizaje, que permitan al estudiante experimentar dentro de una libertad controlada y asistida. Tal y como se ha dicho hasta ahora, se

aprende cuando se construye el conocimiento de algo por medio del estudio o de la experiencia.

El profesor estructura las experiencias para que los estudiantes construyan conocimiento. Tal construcción debe aceptar dos niveles diferenciados en la tradición epistemológica, el nivel psicológico y el nivel epistémico.

La construcción psicológica se realiza cuando el estudiante llega a comprender, valorar, interpretar y criticar el acumulado de conocimiento puesto a su disposición. Esto se logra cuando el aprendiz alcanza una relación efectiva entre el conocimiento previo y su propia interpretación del conocimiento nuevo que se le está presentando.

La construcción epistémica se realiza cuando el estudiante se enfrenta a problemas para los cuales no posee herramientas o no está del todo convencido de la solución, para lo cual él mismo concluye y produce nuevas comprensiones.

C.3. EL DESARROLLO DE LA INTELIGENCIA MEDIANTE ESQUEMAS Y ESTRUCTURAS

Un esquema es una representación de una situación concreta o de un concepto que permite enfrentar situaciones idénticas o parecidas en la realidad y los cuales van cambiando en la medida en que la inteligencia atraviesa fases cualitativamente distintas [mario].

Una estructura consiste en una serie de elementos cuyo efecto es diferente a la suma de los efectos de cada elemento cuando actúa por separado. Para el caso del desarrollo de la inteligencia, las estructuras son herramientas complejas que se van creando por y para la construcción de nuevos conocimientos y la experiencia.

Todas las funciones psicológicas superiores se adquieren, primero, en un contexto social y luego se internalizan como resultado del uso de un comportamiento cognitivo en dicho contexto. Se puede decir que la interiorización ocurre en la medida que un proceso interpersonal se convierte luego en un proceso intrapersonal. Esta internalización se puede aplicar igualmente a varios procesos:

1. Atención voluntaria
2. Memoria lógica
3. Formación de conceptos

El desarrollo mental se puede determinar una vez se reconocen los dos niveles que lo componen:

- El nivel real de desarrollo. Es la capacidad independiente de resolver problemas.
- El nivel de desarrollo potencial. Es la capacidad de resolver un problema bajo la guía o en colaboración con alguien más capaz.

Además, puede distinguirse, entre los dos, lo que se ha llamado la zona de desarrollo próximo, que es la distancia entre el nivel real de desarrollo y el nivel de desarrollo potencial. Esta zona es la que determina el proceso de aprendizaje constructivo pues es allí donde ocurre el proceso de internalización del conocimiento [.

Es claro, entonces, que lo que un individuo pueda aprender no depende únicamente de su actividad individual, ya que el carácter gregario favorece en gran medida el aprendizaje de conocimientos transmitidos o descubiertos en conjunto, lo que determina el ambiente activo de aprendizaje.

La importancia de la interacción social se refleja en el aprendizaje más eficaz del alumno cuando lo hace en un ambiente de colaboración e intercambio con los compañeros, mediante discusiones en grupo y el poder de la argumentación cuando se presenta discrepancia entre alumnos con distinto grado de conocimiento.

La actividad individual y el ambiente activo tiene un carácter complementario, y la naturaleza gregaria del hombre determina un tercer factor del constructivismo determinado por el producto social acumulado de las generaciones precedentes: la cultura, que es el medio que permite la interacción de las dos primeras para el desarrollo cognoscitivo.

C.4. EL ALUMNO DEBE ESTAR PREPARADO PARA APRENDER

Para enseñar, el conocimiento que se trasmite debe estar estructurado en sí mismo y respecto al conocimiento que ya posee el alumno.

Hay que ser conscientes que no todos los temas tienen el mismo nivel de dificultad de aprendizaje por parte del estudiante, lo que hace necesario desarrollar en el alumno esquemas que le sirvan de base para construir el nuevo conocimiento. La elaboración de tales esquemas depende de la selección del material y la dosificación del mismo la cual debe hacerse de forma tal mantenga una relación entre el conocimiento nuevo y el que ya posee el estudiante para construir un conocimiento estructurado [mario].

Aprender es comprender, ya que de esa forma el nuevo conocimiento queda integrado en la estructura de conocimientos. Pero explicar no necesariamente implica saturar al estudiante con una cantidad de material elaborado que castre su curiosidad nata y la tendencia humana de descubrir y concluir para luego poder predecir; por lo que se hace considerablemente importante conocer las representaciones que poseen los alumnos sobre lo que se les va a

enseñar y analizar el proceso de interacción entre lo que se les va a enseñar y lo que ya saben.

Uno de los más celebres estudiosos del proceso de aprendizaje es el psicólogo suizo Jean Piaget, cuyo trabajo se centró en postular una teoría sobre el aprendizaje como una construcción del conocimiento realizada por el individuo a través de diversos canales: la lectura, la escucha, la exploración y la experiencia con el medio ambiente.

Como resultado de su estudio Piaget definió tres mecanismos para el aprendizaje.

1. Asimilación: Consiste en adecuar una nueva experiencia en una estructura mental existente.
2. Acomodación: Consiste en revisar un esquema preexistente a la luz de una nueva experiencia.
3. Equilibrio: Consiste en buscar estabilidad cognoscitiva a través de la asimilación y la acomodación.

De otro lado, pero también con una posición netamente constructivista frente al aprendizaje, el filósofo y psicólogo ruso Lev Vygotsky enfocó su estudio desde un punto de vista social, en donde enfatiza la influencia de los contextos sociales y culturales en el conocimiento, atribuyendo un rol muy importante al maestro y presenta el desarrollo de las habilidades mentales de los estudiantes se presenta naturalmente a través de rutas de descubrimientos.

Los principales supuestos de Vigotsky son:

Construyendo significados:

- I. La comunidad tiene un rol central.
- II. El pueblo alrededor del estudiante afecta su percepción del mundo.

Instrumentos para el desarrollo cognoscitivo:

- I. El tipo y la calidad de estos instrumentos determina el patrón y la tasa de desarrollo.
- II. Los instrumentos deben incluir: personas importantes para el estudiante, la cultura y el lenguaje.

Tanto Piaget como Vigotsky convergen en la importancia que tiene el aula para el aprendizaje, y los resultados complementarios de sus investigaciones independientes presentan un interesante conjunto de principios que deben aplicarse en el aula.

Principios en el aula:

1. El profesor debe proveer un ambiente para la investigación y la experimentación espontánea. (Piaget)
2. El aprendizaje es un proceso activo en el cual se cometerán errores y las soluciones serán encontradas. (Piaget)
3. El aprendizaje es un proceso social que deberá suceder entre los grupos colaborativos con la interacción de posiciones contrarias que facilitan la comprobación de postulados en escenarios lo más naturales posible. (Piaget)
4. Dentro del ambiente de aprendizaje debe haber una persona que pueda orientar a los estudiantes cuando lleguen a un problema para el cual carecen de las capacidades y la experiencia necesarias. (Vigotsky)
5. El aprendizaje y el desarrollo son una actividad social y colaborativa que no puede ser enseñada a nadie, depende del estudiante construir su propia comprensión en su propia mente. (Vigotsky)
6. La zona de desarrollo próximo puede ser usada para diseñar situaciones específicas durante las cuales el estudiante podrá ser provisto del apoyo apropiado para el aprendizaje óptimo. (Vigotsky)

La comunidad tiene un rol fundamental en el aprendizaje, pues el individuo no es un organismo aislado sino que pertenece a una comunidad, en donde encuentra todos los instrumentos, incluso el lenguaje, que están relacionados con el desarrollo del entorno sociocultural. Este proceso cognoscitivo tiene un legado que trasciende generaciones y conserva una inercia cultural que no puede ser ignorada.

C.5. EL PROCESO DE CONSTRUCCIÓN INICIA CON LA PRESENTACIÓN DEL CONOCIMIENTO

Es evidente que la función del enseñante en primer lugar es la publicación oral de la información, la cual puede ser complementada mediante estrategias tales como cuadros sinópticos, mapas conceptuales, etc. y todos los estudiantes deben ser testigos de las diversas vertientes del conocimiento, estar al corriente de lo que ha acontecido, y luego de conocer la información, puede decidir su futuro intelectual e inclinarse por alguna especialidad que le resulte relevante. Este primer encuentro con el nuevo conocimiento permite al estudiante comparar los conocimientos que posee sobre el tema con los que se le están presentando, realizar la evolución de sus propios esquemas y prepararse para la construcción del conocimiento.

C.6. LA EXPERIENCIA COMPLETA EL PROCESO

El conocimiento resulta de un proceso de construcción por parte del sujeto, esta construcción depende de lo que se aprende y se entiende principalmente como resultado de la interacción con el entorno en el proceso de adquirir destrezas y conocimientos a través de la experiencia.

Toda experiencia es información que se detecta como una señal visual, auditiva, olfativa o somestésica que es registrada y procesada por el cerebro,

pero éste no es un mero recipiente donde se depositan las informaciones, sino una entidad que al ordenar y dar forma a estas señales, construye la experiencia y el conocimiento y finalmente da como resultado nuestra percepción de dicha experiencia.

El conocimiento se crea en el individuo, pero no es una descripción estática y fija de la realidad, sino que es un saber relativo y está basado en categorías convencionales y va siendo retocado de forma sucesiva para darle congruencia mientras se desplaza evolutivamente por el tiempo. El cuerpo junto con la capacidad de nuestro cerebro para evocar situaciones y vivencias que han tenido lugar en tiempos y lugares diferentes, son los pilares de la unidad mental.

La experiencia es el vínculo entre el mundo de lo dado y el mundo de lo posible, determina el lugar en donde se puede crear y se le puede abrir el mundo al estudiante, estimulando el desarrollo intelectual del sujeto como un proceso cognitivo de creación y descubrimiento de significados, gracias a la acción que el sujeto hace sobre el mundo y la que el mundo social hace sobre el sujeto; pues el trabajo experimental debe realizarse en grupos de estudiantes, con la correspondiente supervisión de una o unas personas capacitadas en la materia (acomodación y asimilación). Cabe aclarar que esta supervisión no puede ser de ninguna manera realizada por un dueño del conocimiento cuya opinión sea la última palabra, porque ningún cuerpo teórico abarca en sí mismo todos los puntos de vista que son pertinentes respecto a un conjunto de fenómenos concreto. Cada individuo realiza su propia construcción y para ello recibe influencia de las construcciones que tienen los demás, pues aun siendo irreductibles entre sí, son complementarias. Podría decirse que el individuo realiza su propia construcción partiendo de una construcción colectiva.

La comprensión de la realidad también se construye a partir de prácticas sociales y el conocimiento tiene una relación funcional con la realidad ya que

nos ofrece maneras viables de interactuar con el mundo externo. Bajo la idea construida de un 'yo' unificamos, seleccionadas experiencias internas asignándoles un sentido coherente para distinguirlas así del resto de la realidad. Pero es necesario compartir experiencias y ser testigo de construcciones realizadas por otros individuos, e igualmente comunicar nuestras propias percepciones de la realidad. La conversación permite ese intercambio ya que es una búsqueda y una exploración mutua, un intercambio de doble vía, un entrelazamiento de ideas donde los nuevos significados se desarrollan hacia la disolución de los problemas. Podría decirse que al hablar vamos creando la realidad con nuestros interlocutores.

C.7. LA EVALUACIÓN COMO REALIMENTACIÓN POSITIVA

La identificación de los errores nos informa cómo se está re-elaborando el nuevo conocimiento en la mente del alumno, de manera que en la evaluación, antes de resaltar los errores como una equivocación e incapacidad del alumno, deben servir para evaluar la calidad didáctica del proceso que se utilizó para presentar la información al alumno y explorar el camino que éste último siguió para llegar a una integración errada del nuevo conocimiento en su propia estructura cognitiva.

El aprendizaje es un proceso interno y depende de los esquemas y estructuras que se posee, los cuales capacitan al estudiante tanto para desarrollar nuevos esquemas, mejorar los ya existentes o desechar totalmente un esquema que se vuelve obsoleto.

Entre más altos son los niveles educativos, más adecuadas pueden ser las estrategias docentes basadas en la enseñanza receptivo-significativa. Pero se requiere también de una evaluación de estas estrategias, debido a que la utilización de las ideas constructivistas no es una aplicación dogmática de principios generales, sino la revisión sistemática de nuestras ideas a partir de

los datos y las teorías que nos proporcionen las investigaciones sobre un tema dado.

En otras palabras, la evaluación permite observar como el estudiante está realizando la construcción del nuevo conocimiento y en caso de que tal construcción esté desarrollando esquemas erróneos averiguar si el error es debido al estudiante o al diseño del ambiente de aprendizaje.

C.8. CONSTRUCTIVISMO EN LA UNIVERSIDAD

Los estudios e investigaciones sobre pedagogía, como el trabajo de Marta Vitalia, están mostrando la preponderancia que tiene el aprendizaje sobre la enseñanza, donde el aprendizaje se refiere a la actividad que realiza el estudiante para construir el conocimiento, y el profesor deja de ser un narrador de experiencias y esquemas ajenos y se convierte en un diseñador de ambientes de aprendizaje.

El ambiente de aprendizaje se refiere al conjunto de elementos, físicos (aulas, laboratorios, equipos, software, etc.) e institucionales (docentes, investigadores, directivos, etc.) que en conjunto permiten desarrollar experiencias, las cuales propician el proceso de construcción de conocimiento, que involucra creación de nuevos esquemas, evaluación de los ya existentes para reemplazarlos, mejorarlos y desecharlos con base en la nueva información que se está introduciendo.

Pero el ambiente universitario presenta dos territorios muy bien delimitados. En un lado se encuentra la universidad como una institución con un proyecto cultural dentro de cuyos objetivos está la formación de profesionales y del otro lado se encuentran los estudiantes que presentan, cada uno de ellos, un horizonte personal como aprendices. El puente entre estas dos posiciones es precisamente el ambiente de aprendizaje que diseña el profesor con base en el

conocimiento que se tiene de cada una de ellas, como un intérprete de la voluntad institucional y de las posibilidades de actuación de las personas dentro del proceso enseñanza-aprendizaje.

Dentro de los objetivos de la institución debe considerarse que las herramientas que se consiguen son para el uso y aprovechamiento máximo por parte de los estudiantes.