

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS  
SUPERIORES DE MONTERREY  
CAMPUS MONTERREY

PROGRAMA DE GRADUADOS DE LA DIVISION DE  
COMPUTACION, INFORMACION Y COMUNICACIONES



Planeación y Seguimiento de Trayectorias para un Robot  
Móvil Autónomo con Evasión de Obstáculos Dinámicos

TESIS

MAESTRO EN CIENCIAS  
EN AUTOMATIZACION

POR:  
SERGIO ARNULFO BAUTISTA CADENA

DICIEMBRE DE 1998

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS  
SUPERIORES DE MONTERREY

CAMPUS MONTERREY

DIVISIÓN DE COMPUTACIÓN, INFORMACIÓN Y  
COMUNICACIONES  
PROGRAMA DE GRADUADOS EN COMPUTACIÓN,  
INFORMACIÓN Y COMUNICACIONES



Planeación y Seguimiento de Trayectorias para un Robot  
Móvil Autónomo con Evasión de Obstáculos Dinámicos

Tesis

PRESENTADA COMO REQUISITO PARCIAL  
PARA OBTENER EL GRADO ACADÉMICO DE

MAESTRO EN CIENCIAS  
ESPECIALIDAD INGENIERÍA DE CONTROL

SERGIO ARNULFO BAUTISTA CADENA

DICIEMBRE DE 1998



**Planeación y Seguimiento de Trayectorias para un  
Robot Móvil Autónomo con Evasión de Obstáculos  
Dinámicos**

por

**Sergio Arnulfo Bautista Cadena**

**Tesis**

Presentada al Programa de Graduados en Computación, Información y

Comunicaciones

del

Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Monterrey

como requisito parcial para obtener el grado académico de

**Maestro en Ciencias**

en Automatización

**Instituto Tecnológico y de Estudios Superiores de Monterrey**

**Campus Monterrey**

Diciembre de 1998

© Sergio Arnulfo Bautista Cadena, 1998



Después de tres años tengo un nuevo pretexto para publicar lo que escribo. A veces, uno no siempre es uno. Uno es, algunas veces el que fué, otras veces el que es, y otras el que quiere ser. Hay unos que les gusta ser, lo que otros quieren que sea.

Hoy soy yo.

Pienso que la dedicatoria de este trabajo no es adecuada, porque esta tesis es solamente un requisito de trabajo. Tal vez lo adecuado sería reconocer a todos los que han sido parte de mi vida y que me han formado como persona. Yo asumo la responsabilidad de mis actos, pero ustedes tienen la culpa de lo que soy...bueno o malo.

A mi viejo, que nunca deja de estar allá atrás, quien me enseñó a no apoyarme en nadie, ni en él siquiera, pero que se hace el loco y le gana el amor de padre para ayudarme en cuanto me ve en problemas, este es un buen momento para decir que cada vez entiendo más su forma de ver la vida, tal vez porque aunque me gusta ser original, no puedo evitar a mis genes.

A mi madre, que me deja cada vez más sorprendido, es para mí increíble que una mujer pueda amar instintivamente a un hijo, sobre todo con la clase de hijo que tiene, si me hubiera graduado de primaria de nuevo, ella sería la más orgullosa, así de incondicional es el amor de una madre, ojalá todos lo entendiésemos.

A mi hermana, que escribe mucho mejor que yo y es mucho más humana. Me hubiera gustado ser el hermano menor y tener como ejemplo a ella, seguramente sería una mejor persona.

A mis abuelos y tíos que siempre se han preocupado por mí como si fuera uno de sus propios hijos, les he arrancado un poquito de su corazón y lo he traído siempre conmigo.

Estoy pensando en todos, mis amigos, mi familia entera, incluso mi pueblo, si hago algo bueno en esta vida, las circunstancias que me han rodeado están definidas por ellos. Así que dentro de algunos años espero atreverme a escribir de cada uno de ustedes, para darles el mérito.

Esta vez, no incluiré a alguna mujer en esto, no he vuelto a encontrar a alguien que me soporte lo suficiente, como para ser parte de mi vida. Ni modo, para la próxima...

A la Universidad de las Américas y todo lo que viví en esos años, que me han formado y me han dado una identidad.

Y sí, al mismo Tec de Monterrey, que aunque nunca estuvo en mis planes y todos los días me quejo, me dió mucho, me hizo crecer y me ofreció otra formación, pero ya estoy muy andado como para que me quieran educar.

Suficiente por ésta, creo que la próxima vez que escriba, va a tener que ser en una publicación que yo mismo patrocine porque con esta dedicatoria voy a provocar que nadie lea el trabajo técnico que contiene.



## Reconocimientos

Al Instituto Tecnológico y de Estudios Superiores de Monterrey, al Centro de Inteligencia Artificial, al Departamento de Control, que me ha apoyado en mi terquedad de terminar una maestría.

Al CONACYT que afortunadamente sigue tratando de promover la investigación científica en nuestro México.

A todos mis profesores, y en especial a los que he conocido más como personas, en el convivir diario del CIA: mi asesor el Dr. Horacio Martínez, mis sinodales y participantes del Proyecto de la Red de Investigación en Informática, el Dr. Rogelio Soto, el Dr. José de Jesús Rodríguez, el M.C. Jesús Santana, el Dr. José Luis Gordillo; y en general a todo el CIA que dirige el Dr. Cantú, que me permitió escapar de una deuda tremenda con el TEC, con un poco de trabajo; bien valió la pena.

Y finalmente a mis compañeros a los que les he robado muchos de sus conocimientos en el trabajo y experiencias diarias.

SERGIO ARNULFO BAUTISTA CADENA

*Instituto Tecnológico y de Estudios Superiores de Monterrey*  
*Diciembre 1998*





# Planeación y Seguimiento de Trayectorias para un Robot Móvil Autónomo con Evasión de Obstáculos Dinámicos

Sergio Arnulfo Bautista Cadena, M.C.  
Instituto Tecnológico y de Estudios Superiores de Monterrey, 1998

Asesor de la tesis: Dr. Horacio Martínez Alfaro

El trabajo de investigación presentado, describe la propuesta de desarrollo de una estructura de modelado, teleoperación, planeación de trayectorias y seguimiento para un robot móvil autónomo (*AGV*) en un ambiente virtual de manufactura. Ejecutable en un navegador de Internet, orientado su uso en laboratorios virtuales para tareas de educación y experimentación.

Esta investigación forma parte del Proyecto de laboratorios Virtuales y se complementa con otras estructuras de software y hardware para su inclusión en el esquema de la red de laboratorios virtuales.

Las principales partes de la investigación comprenden: la construcción de un ambiente virtual de manufactura en el cual se desarrollarán las actividades del *AGV*, generado mediante el lenguaje de programación *JAVA 3D*; la formalización matemática del modelado del vehículo autónomo, incluyendo el cálculo de la cinemática directa e inversa; la planeación de la trayectoria sobre la cual se desplazará el *AGV*, libre de colisiones con obstáculos fijos, mediante el algoritmo de recocido simulado; el seguimiento de la trayectoria planeada por parte del *AGV* basándose en la interfaz en lenguaje *C* del robot *Nomad 200* y su teleoperación también en lenguaje *C* para situaciones de movimientos simples que no requieran la planeación de una trayectoria.

La liberación y publicación de esta investigación como parte del proyecto de laboratorios virtuales, obliga a presentarse en un ambiente sencillo, fácil de utilizar y accesible para los usuarios externos que lo soliciten; es por esto que se implementa en una de las herramientas de programación orientada a objetos (OOP) más avanzado que existe actualmente: *JAVA*. El uso de esta herramienta de programación permitirá que el simulador-experimentador sea accesible en Internet vía un navegador como Netscape 4.04 que exprese su desempeño.

Las bases teóricas de cada uno de los elementos de la investigación están descritas en cada uno de los capítulos como condensados de los términos necesarios para comprender el desarrollo de cada una de las etapas; las referencias podrán ampliar los conocimientos que el usuario requiera.

Finalmente se presentan los resultados de cada una de las etapas del trabajo y de la integración de todos los elementos, analizando posteriormente el impacto que pueda tener como parte de un laboratorio virtual y evaluando sus alcances. La última etapa consistirá del trazado de la línea de investigación posterior para su evolución como parte de un proyecto de robótica virtual.

# Índice General

<b>Reconocimientos</b>	<b>vii</b>
<b>Resumen</b>	<b>ix</b>
<b>Índice de Tablas</b>	<b>xiv</b>
<b>Índice de Figuras</b>	<b>xv</b>
<b>Capítulo 1 Introducción</b>	<b>1</b>
1.1 Antecedentes de la investigación . . . . .	1
1.2 Definición, objetivo y propuesta de investigación . . . . .	2
1.2.1 Definición del problema . . . . .	2
1.2.2 Objetivo . . . . .	2
1.2.3 Propuesta . . . . .	2
1.3 Descripción del proceso de investigación . . . . .	3
1.3.1 Alcances . . . . .	3
1.3.2 Ambientes virtuales de manufactura . . . . .	3
1.3.3 Modelado, cinemática directa e inversa . . . . .	3
1.3.4 Planeación de trayectorias . . . . .	3
1.3.5 Seguimiento . . . . .	3
1.3.6 Resultados experimentales . . . . .	4
1.3.7 Conclusiones . . . . .	4
1.4 Producto de la investigación . . . . .	4
<b>Capítulo 2 AGVS en sistemas de manufactura</b>	<b>5</b>
2.1 AGVS . . . . .	5
2.1.1 Definición . . . . .	5
2.1.2 Aplicaciones . . . . .	6
2.1.3 Ruteo y control . . . . .	7
2.2 Manufactura en laboratorios virtuales . . . . .	8
2.2.1 Construcción de un laboratorio virtual . . . . .	8
2.2.2 Aplicación en manufactura de un laboratorio virtual . . . . .	10

2.3	Automatización de AGVS en sistemas virtuales de manufactura . . . . .	11
<b>Capítulo 3 Planeación de trayectorias</b>		<b>13</b>
3.1	Problema de planeación . . . . .	13
3.2	Curvas B-splines . . . . .	15
3.3	Recocido simulado . . . . .	16
3.3.1	Definición . . . . .	16
3.3.2	Proceso de recocido . . . . .	16
3.3.3	Algoritmo del recocido simulado . . . . .	17
3.3.4	Costos de la trayectoria . . . . .	19
3.4	Espacio C . . . . .	21
3.4.1	Trayectoria . . . . .	22
3.4.2	Obstáculos en la configuración de espacio . . . . .	23
3.4.3	Cálculo de la configuración de espacio . . . . .	23
<b>Capítulo 4 Robótica de un AGV</b>		<b>25</b>
4.1	Cinemática de vehículos autónomos . . . . .	25
4.1.1	Modelado cinemático . . . . .	25
4.1.2	Definiciones y condiciones . . . . .	26
4.1.3	Sistemas de coordenadas . . . . .	27
4.1.4	Matrices de transformación . . . . .	28
4.1.5	Modelo cinemático directo de posición para un WMR . . . . .	32
4.1.6	Modelo cinemático inverso de posición para un WMR . . . . .	33
4.2	Dinámica de vehículos autónomos . . . . .	33
4.3	Teleoperación y control de posición . . . . .	35
4.3.1	Teleoperación . . . . .	35
4.3.2	Control de posición . . . . .	35
<b>Capítulo 5 Seguimiento de trayectorias para un AGV</b>		<b>38</b>
5.1	Control de velocidad angular . . . . .	38
5.2	Control de velocidad lineal . . . . .	41
5.3	Evasión de obstáculos dinámicos . . . . .	43
<b>Capítulo 6 Simulación y resultados experimentales</b>		<b>46</b>
6.1	Ambientes virtuales . . . . .	46
6.2	Robótica de AGVS . . . . .	47
6.3	Planeación de trayectorias . . . . .	50
6.4	Seguimiento . . . . .	54
6.5	Integración del laboratorio virtual . . . . .	55
<b>Capítulo 7 Conclusión</b>		<b>62</b>



## Índice de Tablas

4.1	Sistemas de coordenadas . . . . .	29
4.2	Desplazamientos escalares, rotacionales y traslacionales. . . . .	30
4.3	Velocidades escalares, rotacionales y traslacionales. . . . .	30
4.4	Matrices de transformación. . . . .	31
5.1	Matriz de relación entre variables de entrada y salida . . . . .	41



## Índice de Figuras

2.1	Construcción del universo virtual para tareas de manufactura . . . . .	9
2.2	Vista superior del universo virtual . . . . .	10
3.1	Problema de planeación de trayectorias . . . . .	14
4.1	Sistemas de referencia de un robot móvil autónomo . . . . .	28
4.2	Diagrama de referencia del Nomad 200 . . . . .	33
4.3	Angulo de dirección de giro de las ruedas . . . . .	34
4.4	Modelo dinámico del Nomad 200 . . . . .	34
4.5	Control de posición - seguimiento puntual . . . . .	36
4.6	Estructura de seguimiento puntual en lazo abierto . . . . .	37
5.1	Esquema de control difuso . . . . .	39
5.2	Función de membresía para la señal $\Delta\theta$ de entrada . . . . .	40
5.3	Funcion de membresía para la señal $e\theta$ de entrada . . . . .	40
5.4	Función de membresía para la señal de salida $\omega$ . . . . .	41
5.5	Superficie generada por las funciones de membresía . . . . .	42
5.6	Tácticas de evasión de obstáculos dinámicos . . . . .	44
5.7	Estructura de planeación y seguimiento para un AGV con obstáculos dinámicos . . . . .	45
6.1	Universos virtuales de manufactura . . . . .	47
6.2	Cálculo de cinemática directa . . . . .	48
6.3	Cálculo de cinemática inversa . . . . .	49
6.4	Pantalla de teleoperación del AGV Nomad 200 . . . . .	51
6.5	Experimento de teleoperación . . . . .	52
6.6	Planeación de trayectorias . . . . .	53
6.7	Seguimiento de trayectorias . . . . .	54
6.8	Simulación de seguimiento . . . . .	56
6.9	Experimento de seguimiento . . . . .	57
6.10	Experimento de seguimiento: trazado de trayectoria . . . . .	58
6.11	Simulación de evasión . . . . .	59

6.12 Laboratorio virtual de robótica . . . . .	60
6.13 Ventana del Nomad 200 en el laboratorio virtual de robótica . . . . .	61

# Capítulo 1

## Introducción

### 1.1 Antecedentes de la investigación

Los robots han demostrado que su inclusión en los procesos productivos ha generado una revolución tecnológica al sustituir a operadores humanos, o inclusive realizando tareas que un humano no sería capaz de realizar en un sistema de manufactura flexible. Actualmente los robots móviles (*AGV*) incluidos en un sistema de manufactura llevan a cabo principalmente tareas de transporte entre etapas del proceso (celdas con manipuladores). Además, es también posible incluirlos en tareas de manipulación con la adecuada participación de un manipulador montado en el robot móvil. Para realizar dichas tareas de transporte, es necesario planear la trayectoria a realizar por el robot móvil y evadir los obstáculos que encuentre a su paso en el espacio de trabajo a lo largo de la línea de producción.

El seguimiento del robot móvil debe ser lo más aproximado a la trayectoria planeada, es por esto que se hace necesario una estructura de control inteligente, robusta, que controle sus movimientos y disminuya el error entre la trayectoria planeada y la trayectoria real descrita por el robot móvil.

El análisis de un esquema de manufactura es posible llevarlo a cabo estando directamente frente al sistema; sin embargo, existe la posibilidad de crear un ambiente de manufactura flexible virtual con los elementos necesarios. El análisis por tanto, puede estar al alcance de un usuario que puede determinar específicamente su espacio de trabajo, los elementos a utilizar, el producto y las etapas del proceso.

El propósito de este trabajo de investigación consiste en involucrar tareas de robótica y control para su conjunción en un espacio virtual en el que se simule una tarea de transporte en una celda de manufactura, en la que existan obstáculos móviles, y se alcance una posición y orientación final requerida al desplazar el robot móvil a lo largo de la trayectoria planeada.

## 1.2 Definición, objetivo y propuesta de investigación

### 1.2.1 Definición del problema

Se ha visto que en las plantas de manufactura existen problemas de modificación en el ruteo de trayectorias para los robots móviles cuando cambian las condiciones físicas de la planta.

La tarea a realizar por un *AGV* debe iniciar desde una posición inicial dada y alcanzar una posición final requerida dentro de el espacio físico de la planta. La trayectoria definida mediante la cual el robot móvil alcanza la posición final puede tener distintas propiedades de curvatura, sin embargo debe incluir en sus características la proximidad del robot con los obstáculos para evitar una colisión. Los objetos y el robot se representarán en un *espacio-C* y las trayectorias por curvas *B-spline*.

La planeación de la trayectoria será parcial y posible de modificar durante el desarrollo de su ejecución, esta información se presentará en un ambiente gráfico virtual tridimensional que muestre el espacio de trabajo, los obstáculos, la posición inicial, la posición final deseada, la trayectoria planeada, sus variaciones y el robot móvil.

### 1.2.2 Objetivo

El objetivo de esta investigación consiste en el desarrollo de un esquema de planeación de trayectorias y seguimiento inteligente, en un ambiente de trabajo virtual de manufactura, en el cual se determine la trayectoria óptima entre posición inicial y posición final para las circunstancias presentes de un robot móvil. Este espacio de trabajo virtual puede contener en su estructura obstáculos que se sumen al espacio inicial, e incluso que los obstáculos tengan dinámica propia.

Proporcionar una herramienta con tareas de robótica para la enseñanza y experimentación con robots móviles autónomos, en un espacio de simulación virtual tridimensional, orientado a realizar tareas de manufactura flexible.

### 1.2.3 Propuesta

La opción de evaluar un grupo de estrategias de robótica y control que culminen con un experimento de planeación y seguimiento de trayectorias para un robot móvil autónomo, orientado a tareas de manufactura flexible en ambientes virtuales.

## **1.3 Descripción del proceso de investigación**

### **1.3.1 Alcances**

La primera parte de la investigación consiste en la definición del universo virtual de manufactura, el cual se construirá en *JAVA 3D*; el siguiente capítulo comprende el modelado del robot móvil y la programación en *JAVA* de las rutinas de cinemática directa e inversa; posteriormente se resolverá el problema de planeación y seguimiento de trayectorias con evasión de obstáculos dinámicos para el *AGV* en ambientes virtuales de manufactura; de esta manera se integrarán todas las tareas al laboratorio virtual de robótica en su apartado de robots móviles.

### **1.3.2 Ambientes virtuales de manufactura**

En esta etapa se definen los conceptos para la generación de un laboratorio virtual que permita realizar pruebas y simulaciones de un caso particular de *AGV* que se desplace en un ambiente virtual de manufactura. Que se asemeje lo más posible a la realidad.

### **1.3.3 Modelado, cinemática directa e inversa**

Esta etapa describe los modelos cinemáticos directo e inverso del *AGV*, sus consideraciones teóricas, mediante las cuales se encontrarán las características y condiciones necesarias para el correcto control del *AGV*.

### **1.3.4 Planeación de trayectorias**

Se compone de tres partes principales (B-splines, recocido simulado y espacio-C) que llevan a cabo la planeación de la trayectoria por la cual se despazará el *AGV*, libre de colisiones con obstáculos fijos.

### **1.3.5 Seguimiento**

El seguimiento del cuerpo del robot deberá disminuir el error entre la trayectoria planeada y la trayectoria real, así como las situaciones extremas de emergencia al presentarse obstáculos dinámicos en la trayectoria planeada .

### **1.3.6 Resultados experimentales**

Se muestran los resultados obtenidos en cada una de las etapas. Debido a que esta investigación se complementa con otros elementos dentro del Proyecto de la Red de Investigación en Informática, algunos resultados serán mostrados numéricamente, pero la mayoría, serán mostrados gráficamente para una mejor comprensión y una presentación amigable al usuario. El movimiento del robot *Nomad 200* será presentado mediante secuencias de gráficas.

### **1.3.7 Conclusiones**

Finalmente, los resultados serán evaluados con respecto a sus propios índices y a las expectativas que se tienen para el proyecto. Se comentarán algunas de las experiencias en cada una de las etapas del desarrollo de esta herramienta. Finalmente, se harán reflexiones acerca del proceso de investigación, así como los siguientes pasos en el desarrollo de Laboratorios Virtuales.

## **1.4 Producto de la investigación**

La investigación debe generar resultados de simulación y pruebas con el robot móvil *Nomad 200*. El desarrollo de la técnica de seguimiento aplicada en base a la planeación de la trayectoria previamente realizada, debe describir el comportamiento de un robot móvil en un espacio de trabajo tridimensional virtual en el cual existan obstáculos móviles (que podrían ser otros robots móviles en un caso particular) y evadirlos para evitar colisiones.

Esta investigación tiene una profunda relación con el Proyecto de la Red de Investigación en Informática del Centro de Inteligencia Artificial del ITESM. Esta investigación está ligada como parte del Proyecto de laboratorios virtuales, que consiste en la simulación de ambientes virtuales de desarrollo experimental de robots móviles y manipuladores para procesos de manufactura flexible. La simulación de un robot móvil forma parte del proyecto, que tiene como finalidad ofrecer a un usuario, con conocimientos básicos de robótica, una opción de experimentación virtual que complete su experiencia en el área de robots móviles.

## Capítulo 2

# AGVS en sistemas de manufactura

## 2.1 AGVS

### 2.1.1 Definición

Una definición establecida para los “*Automated guided vehicles systems (AGVS)*” es la siguiente: *Vehículos energizados por baterías, con tracción, dirigidos automáticamente para seguir trayectorias definidas. Algunos son capaces de cargar y descargar automáticamente cargas unitarias. Usualmente, interactúan con otros sistemas automatizados para lograr los beneficios completos de la automatización integrada* (Groover, 1994).

El futuro de la automatización flexible en las plantas es prácticamente un hecho; las actuales fábricas CIM (Computer Integrated Manufacturing), son muy distintas a las que han quedado rezagadas en su desarrollo e implementación tecnológica. Las teorías emergentes están llevando a los productores a la modernización de sus procesos con base en la implementación de técnicas que recurren a la Inteligencia Artificial como solución a los problemas de calidad, tiempo, desempeño, planeación, situaciones no previstas, etc.

En un sistema de manufactura flexible, en cada celda de manufactura, se utilizan robots móviles y manipuladores robóticos, que realizan tareas de transporte, ensamble, construcción, corte, soldadura, etc.

Un *AGVS* es un sistema que maneja material y que es operado en forma independiente, es autopropulsado y su tarea consiste en seguir trayectorias en un espacio de trabajo determinado.

Estos vehículos son autónomos durante ciertos períodos de tiempo ya que la energía que consumen es generada por bancos de baterías que llevan consigo.

Por lo general, estos vehículos son guiados por cables en el piso, por pintura reflejante o incluso rieles. Obviamente, esta característica los hace dependientes de sensores específicos.

Existen distintos tipos de *AGVS*:



1. *Trenes sin conductor*
2. *Montacargas*
3. *Carga Unitaria*

Un sistema de vehículo autónomo guiado (*AGVS*) representa una forma versátil de mover materiales automáticamente. Un *AGVS* es adecuado en aplicaciones donde diferentes materiales deben ser movidos desde varios puntos de carga hacia puntos de descarga (Groover, 1994).

### 2.1.2 Aplicaciones

Los *AGVS* son utilizados en un creciente número y variedad de aplicaciones. Sus aplicaciones en la industria son diversas:

1. *Operaciones de trenes sin guías*
2. *Sistemas de acopio y distribución*
3. *Operaciones de ensamble en línea*
4. *Sistemas de manufactura flexible*
5. *Aplicaciones mixtas*

Estas operaciones incluyen el transporte de grandes cantidades de materiales entre distancias relativamente largas. Entre edificios de una planta de manufactura, por ejemplo. Estos trenes llevan entre 5 y 10 vagones regularmente.

Los montacargas son los elementos típicos a utilizar en estas aplicaciones. Las operaciones de distribución y llenado de “*stock*” incluyen el movimiento de materiales en cargas unitarias, inclusive movimientos individuales, desde o hacia localidades específicas. Este tipo de aplicaciones usualmente relacionan a los *AGVS* con sistemas que manipulan mercancía al “*stock*”, formando así un sistema AS/RS (*Automated storage/retrieval system*) en un centro de distribución.

Los *AGVS* entregan los elementos de entrada o unidades de carga desde el lugar de recibimiento hasta el AS/RS, el cual pone los elementos en el “*stock*” y entrega al *AGV* los materiales que saldrán al embarque, para que el *AGV* regrese y los ponga en el “*deck*” de carga. Cuando el índice de entradas y salidas están balanceadas, este modo de operación permite cargar en ambas direcciones a los *AGVS*, incrementando la eficiencia del sistema.

Los *AGVS* también están siendo usados en un creciente número de aplicaciones de ensamble en línea. En estas aplicaciones el índice de producción es relativamente bajo

y existen una variedad de modelos diferentes hechos en la línea de producción. Entre la estaciones de trabajo, los componentes son armados y ensamblados parcialmente sobre el *AGV* para ser terminadas las tareas en la estación a la cual se dirige. Las estaciones de trabajo son generalmente puestas en paralelo a la línea de producción para sumar flexibilidad a la línea. Unidades de carga y vehículos ligeros son el tipo de *AGVS* utilizados en las líneas de producción.

Otra aplicación que está creciendo en la tecnología de los *AGVS* es en sistemas de manufactura flexible (FMS). En esta aplicación los vehículos son utilizados como sistemas de manejo de materiales. Los vehículos realizan el trabajo del área donde se posicionan los elementos en superficies para las estaciones de trabajo individuales en el sistema. Los vehículos también se mueven entre las estaciones de trabajo del sistema de manufactura. En la celda de manufactura el trabajo es transferido de la plataforma del vehículo al área de trabajo de la estación para su procesamiento. Ya que la estación completó su trabajo, el vehículo toma nuevamente la pieza o el material y termina de trabajarlo mientras lo transporta al área de maquinado siguiente. Los sistemas *AGVS* proveen una sistema complementario de manejo de material versátil que vuelve eficiente la operación de un FMS (Flexible Manufacturing System).

Entre las aplicaciones no industriales de los sistemas de vehículos autónomos guiados se incluyen tareas como entrega de correo en edificios de oficinas y operaciones de manejo de material en hospitales. Los vehículos en los hospitales transportan comida, medicamentos, y material de laboratorio, así como otros materiales entre departamentos. Estas aplicaciones requieren de movimientos de vehículos en diferentes pisos, así que deben tener la capacidad de utilizar elevadores.

### 2.1.3 Ruteo y control

Existen funciones que deben ser desempeñadas para operar los sistemas de vehículos autónomos guiados de manera exitosa. Estas funciones son:

1. *Ruteo y guía*
2. *Control de tráfico y seguridad*
3. *Administración del sistema*

El sistema de guía, se refiere al método mediante el cual se planea una trayectoria que seguirá el *AGV* y el sistema de control del vehículo que intentará acercarse lo más posible a la trayectoria planeada. Existen dos métodos principales para definir trayectorias en el piso de una planta de manufactura que se mencionaron anteriormente, pintura y guías físicas.

El ruteo plantea el problema de seleccionar la trayectoria adecuada para el vehículo,

definiendo su movimiento desde un punto inicial hasta un punto final en el sistema. El propósito del control de tráfico para un *AGVS* es prevenir colisiones entre vehículos que estén desplazándose a lo largo de la misma ruta. Esto se logra mediante esquemas de control de bloqueo. Para poder llevar a cabo el bloqueo, es necesario que el *AGV* cuente con capacidad de sensado a bordo, o que se definan zonas bloqueadas. El equipo de sensado a bordo del *AGVS* incluye algún tipo de sensor que detecte la presencia de otros vehículos u objetos que se encuentren a lo largo de la trayectoria planeada. El concepto de zona bloqueada es simple. Para cada *AGVS*, se define su zona de trabajo separada, de manera que no interfieran unos con otros, no está permitido que uno de estos *AGV* se infiltre permanentemente en una zona que no le pertenece, así como restricciones en zonas de riesgo (límites de las zonas) en donde pueden acercarse unos a otros. El área de la zona debe contemplar el tamaño del vehículo, así como la zona de tolerancia o riesgo. Es necesario involucrar la seguridad de humanos que estén trabajando en zonas de trabajo de *AGVS*; por tanto, el sistema de detección ya sea por medio de sistemas ópticos, sonares o infrarojos debe poder detectar a un humano y tomar la consideración prudente de detenerse por completo o por lo menos disminuir su velocidad.

## **2.2 Manufactura en laboratorios virtuales**

### **2.2.1 Construcción de un laboratorio virtual**

Un laboratorio virtual es un concepto difícil de establecer. Su creación surge de la necesidad de una experimentación que aunque no sea real, ofrezca una sensación lo más cercana a la realidad de una planta de manufactura, sin efectuar pruebas con equipo utilizado en tiempo real en la planta productiva.

La utilidad de un laboratorio de este tipo concentra sus resultados en generar una experiencia al usuario tan real como si efectivamente lo estuviera llevando a cabo. Los usuarios que busquen una experiencia en el área de robótica móviles y que no tengan a su alcance la posibilidad de trabajar con un *AGV* real podrán experimentar en un laboratorio virtual con éste y otros elementos robotizados con la particularidad de hacerlo en una pantalla de computadora y con la posibilidad de realizar a distancia, una teleoperación del *AGV*, después de haber probado sus movimientos en el ambiente virtual.

El software que se ha desarrollado últimamente permite a los programadores algunas alternativas de reutilización y programación orientada a objetos (OOP), que hacen posible la generación de un ambiente virtual en el que se desarrollen robótica manipuladores y vehículos autónomos, formando así un laboratorio virtual en el que se tiene como propósito final llevar a cabo una tarea de manufactura totalmente automatizada.

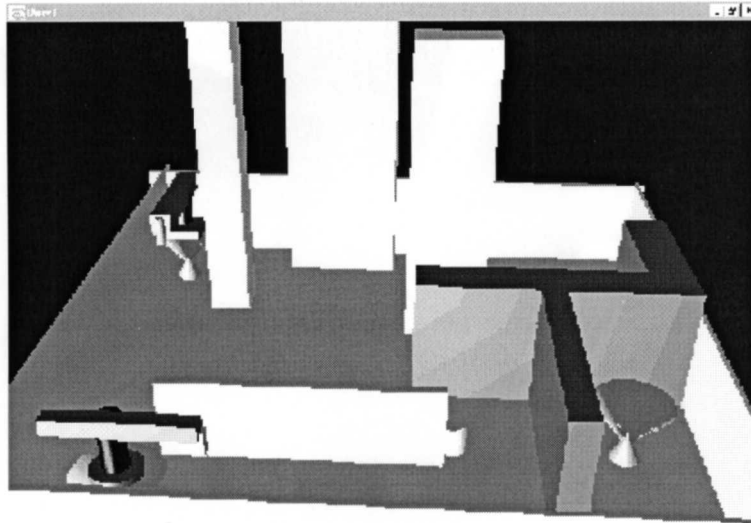


Figura 2.1: Construcción del universo virtual para tareas de manufactura

Si se cuenta con las máquinas adecuadas (hardware) y las interfaces de software que permitan la conexión entre equipo *AGV* y el laboratorio virtual, es posible pensar en el uso de un sistema de este tipo en algún proceso de manufactura industrial automatizado.

En un futuro, con el hardware, software, interfaces y medios de comunicación global adecuados, será posible desarrollar tareas de manufactura a distancia en una reconstrucción tridimensional virtual de un espacio real en una computadora, en la que concurren todos los elementos necesarios para realizar el proceso de manufactura, aunque en la realidad algunos de estos elementos se encuentren en lugares diferentes.

Para lograr esto, el laboratorio virtual, establece como principio de desarrollo, incluir cuatro elementos básicos:

1. *Software*
2. *Hardware*
3. *Interfaces*
4. *Medio de comunicación*

*Software.* Es necesario un lenguaje en el que sea posible programar de manera rápida, simple y abierta, tareas que puedan ser llamadas desde distintos programas para la realización de procedimientos básicos. El lenguaje debe tener la capacidad de ser transferible fácilmente y de ser ejecutable en un canal de comunicación establecido para

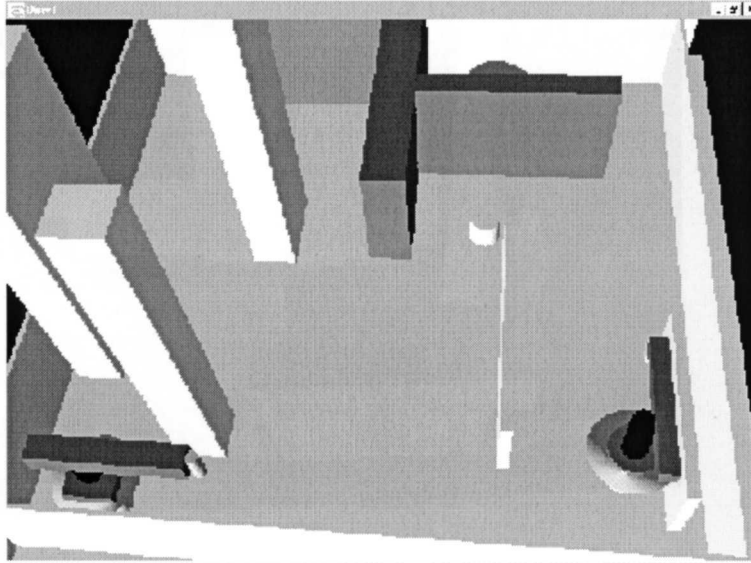


Figura 2.2: Vista superior del universo virtual

poder ser distribuible.

*Hardware.* Debe tener la particularidad de ser abierto en algunas de sus características para poder ser reconfigurable, conectable a un medio o vía de comunicación estándar, programable mediante un lenguaje como el descrito anteriormente y con interfaces que no requieran elementos específicos.

*Interfaces.* Deben ser un medio simple de acción entre el software y el hardware, transparente al usuario y suficientemente robustas.

*Medio de Comunicación.* Posiblemente éste es el elemento más complejo, porque debe integrar los elementos anteriores bajo una misma red de comunicación, que utilice el mismo código y que al usuario le presente un ambiente amigable y sencillo para poder realizar las tareas de manufactura virtual.

### 2.2.2 Aplicación en manufactura de un laboratorio virtual

Mientras el laboratorio virtual cumpla con los elementos requeridos, será posible entonces utilizarlo para el desarrollo de simulaciones de procesos de manufactura virtual. El propósito de desarrollar tareas de manufactura virtual es consecuencia de las nuevas tendencias en las empresas de manufactura, si la compañía manufacturera establece como estrategia competitiva la innovación de productos, es necesario que los cambios en el producto o los nuevos diseños antes de salir a los mercados deben ser probados. Sin embargo, realizar un prototipo y llevarlo a laboratorio de pruebas para identificar sus

repuestas en ambientes reales lleva un tiempo relativamente largo.

Si la estrategia competitiva establecida por la compañía ataca el problema de manufactura ágil, se enfrentará al problema de rediseñar algunos de sus procesos para hacerlos más eficientes y encontrar rutas alternativas de producción cuando encuentre problemas de paro en el equipo.

Así pues, tanto en la planeación de la producción como en el diseño y el desarrollo de nuevos productos, las empresas manufactureras buscan alternativas de simuladores que prueben sus conceptos nuevos o que optimicen sus tiempos de trabajo cuando sucedan eventos fuera de lo establecido.

Existen muchas herramientas comerciales, que se enfocan al diseño y construcción de elementos mediante CAD, posteriormente envían su información a un sistema CAM y se complementan para llevar a cabo un proceso CAE. *ProEngineer* y *CATIA* son algunas de estas herramientas de uso genérico en las empresas manufactureras.

La propuesta de un laboratorio virtual incluye algunas de las tareas que ofrecen estos softwares comerciales, pero bajo otras dimensiones y expectativas particularmente educativas, enfocadas a robóts manipuladores y *AGVS* que desempeñen procesos de manufactura simples, cooperativos, totalmente automatizados.

Suponiendo la situación en la que una empresa de manufactura utilice *AGVS* como medios de transporte entre celdas de manufactura para transportar partes de productos de ensamble o piezas en proceso de manufactura, es útil un espacio de pruebas virtual que permita experimentar situaciones difíciles que no estén contempladas en el desarrollo del proceso de manufactura. Si se experimentan en un ambiente virtual estas situaciones, es posible, programar las tareas de forma eficiente, óptima y con la seguridad requerida para el equipo real, automatizada completamente, y así verificar el desarrollo de la tarea encomendada al *AGV*. Todo esto desde una pantalla de computadora que tenga la capacidad de trabajar a distancia en un ambiente virtual reconstruido a partir de las características propias de la planta de manufactura sin existir en un espacio real.

## **2.3 Automatización de AGVS en sistemas virtuales de manufactura**

En un ambiente de manufactura totalmente estructurado, es posible ordenar a un robot móvil autónomo que realice y repita tareas de relativa complejidad.

En las secciones anteriores, se han determinado las características propias de los *AGVS*, de los Sistemas de Manufactura y de los Ambientes Virtuales; en esta última sección se concentra la información para poder establecer un esquema que proponga la forma de unir y complementar las partes.

Regularmente en las empresas de manufactura, los vehículos operan sobre rieles o guías para poder trasladar los componentes entre las celdas de manufactura o inclusive sobre la línea de producción. Esto ocasiona un continuo gasto en mantenimiento y un cuidado constante para mantener los rieles o guías totalmente libres de objetos que puedan alterar el desplazamiento del vehículo. Utilizar un *AGV*, en un ambiente de manufactura estructurado proporciona algunas ventajas y disminuye ciertos problemas. Un *AGV*, es un elemento que proporciona flexibilidad en la empresa manufacturera (Gasparetto et al., 1994), lo cual aumenta el valor agregado, implementando una estrategia competitiva de manufactura ágil.

Por ejemplo, si un obstáculo se presenta en la trayectoria de un vehículo guiado por rieles, el efecto dependerá de la capacidad del vehículo en la detección; y aún cuando pueda detenerse a tiempo, no podrá continuar su trayectoria mientras el obstáculo permanezca sobre su guía.

La modificación de trayectorias de desplazamiento para un vehículo autónomo es un problema de sustitución de nueva información en función de las condiciones que se presenten. Esto demuestra que mientras el vehículo guiado por rieles necesitará de ayuda física, el *AGV* necesitará de un cambio de trayectoria generado por software. Incluso, es posible que si el *AGV* detecta la presencia de un obstáculo en su trayectoria, automáticamente lo evada. Esto es posible debido a que no está restringido a seguir un riel y/o guía físico.

El comportamiento necesario en un ambiente estructurado de manufactura de un *AGV* es desempeñar tareas, cargar y transportar componentes del proceso de ensamble o transportar un manipulador entre las estaciones o sobre la línea de manufactura. Este *AGV* debe evitar colisiones con los obstáculos establecidos que se encuentren en la celda y con los obstáculos dinámicos no determinados que puedan presentarse sobre su trayectoria. Esto provocará un sistema altamente flexible que no tendrá necesidad de sustituir guías físicas si se presentan cambios en la configuración del ambiente físico. Los requerimientos serán: una base de datos que contenga la estructura nueva del ambiente y un tiempo de programación para trazar una nueva trayectoria de desplazamiento.

Ahora, incluyendo al ambiente virtual se genera un punto más de innovación tecnológica.

Si todas las consideraciones que se han descrito anteriormente con respecto al *AGV* se llevaban a cabo en una planta de una empresa de manufactura, es posible, mediante la construcción del espacio virtual que represente las características de la planta, tener el control de las tareas que llevará a cabo el *AGV* desde un monitor que permita al usuario asomarse a la planta sin estar físicamente ahí.

La planeación y seguimiento de una trayectoria contemplando la evasión de obstáculos dinámicos para un *AGV* en un ambiente virtual de manufactura propone al usuario la capacidad de tener el control sobre desplazamientos de un *AGV* que carga y transporta componentes es una planta desde su monitor, incluso sin estar físicamente en la planta.



## Capítulo 3

# Planeación de trayectorias

### 3.1 Problema de planeación

La planeación de trayectorias consiste en definir una curva suave que represente la trayectoria sobre la cual un *AGV* se desplazará. Esta función que define la forma de la trayectoria está formada por una sucesión de puntos; la trayectoria puede ser una función de segundo, tercer o cuarto orden, inclusive una serie, o cualquier tipo de representación matemática que determine los puntos en el espacio por los cuales se desplaza un objeto móvil.

El objetivo es una trayectoria, formada por una sucesión de puntos, sobre la cual viaja el robot móvil. La trayectoria será definida como una curva *B-spline*, construida mediante el método de optimización de recocido simulado; definida en un *espacio-C* en el cual existen obstáculos fijos.

La trayectoria deberá estar libre de colisiones con obstáculos. Si existiera un obstáculo no previsto en la trayectoria trazada, se recurrirá a un algoritmo de evasión de obstáculos dinámicos que se definirá en un capítulo más adelante.

El objetivo, por definición, del problema de planeación de trayectorias es mover el robot de una posición inicial dada a otra final dentro del espacio de trabajo  $\mathcal{W}$ , sin tomar en cuenta las propiedades dinámicas del robot, este problema se puede definir como (Latombe, 1991):

Sea  $\mathcal{A}$  un cuerpo rígido simple, *robot*, moviéndose en un espacio Euclidiano  $\mathcal{W}$ , llamado espacio de trabajo, representado como  $\mathcal{R}^N$ , con  $N = 2$  ó  $3$ .

Sean  $\mathcal{B}_1, \dots, \mathcal{B}_q$  cuerpos fijos rígidos distribuidos en  $\mathcal{W}$ . Donde los  $\mathcal{B}_i$  son llamados *obstáculos*. Se asume que tanto la geometría de  $\mathcal{A}$ ,  $\mathcal{B}_1, \dots, \mathcal{B}_q$  y la ubicación de los  $\mathcal{B}_i$  en  $\mathcal{W}$  son completamente conocidos. Se asume que no hay restricciones cinemáticas que limiten los movimientos de  $\mathcal{A}$ .

El problema se resume en: Dada una posición inicial y una orientación y una posición final y su respectiva orientación de  $\mathcal{A}$  en  $\mathcal{W}$ , se debe obtener una trayectoria  $\tau$  especificando una secuencia de posiciones y de orientaciones evitando contacto con las  $\mathcal{B}_i$ , comenzando con la posición y orientación inicial y terminando en la posición final y

su correspondiente orientación. Esta trayectoria encontrada debe cumplir por lo tanto con  $\mathcal{A} \cap \mathcal{B}_i = \phi$ .

Se asume que  $\mathcal{W}$  y los obstáculos  $\mathcal{B}_i$  son poliedros o polígonos fijos y  $\mathcal{A}$  es un poliedro o polígono móvil con orientación fija.

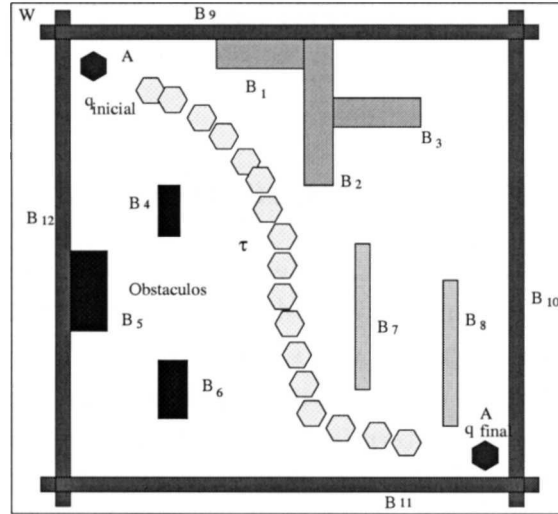


Figura 3.1: Problema de planeación de trayectorias

Como se describe en el párrafo anterior y la Figura 3.1, en el problema básico, se asume que el robot es el único objeto móvil en el espacio de trabajo y se ignoran las propiedades dinámicas del robot.

Es necesario mencionar la división del problema de obstáculos dinámicos para establecerlo como parte de la investigación desde este momento, aunque sea resuelto en un capítulo posterior. El problema de objetos móviles en un espacio de trabajo puede tomar dos vías principales:

1. Considerando el caso en que algunos o todos los obstáculos en el espacio de trabajo esten en movimiento. Este problema es conocido como problema dinámico de planeación de movimientos.
2. Considerado como una segunda extensión, es el caso en el que varios robots se mueven independientemente en un mismo espacio entre obstáculos estáticos. Este problema es conocido como problema de planeación de movimientos para múltiples robots.

## 3.2 Curvas B-splines

La generación de una curva *B-spline* tiene como objeto en este trabajo de investigación trazar una trayectoria libre de obstáculos en un espacio virtual tridimensional para el desplazamiento de un *AGV* a lo largo de ella.

Esta curva es generada en base a seis parámetros básicos:

- Orden de la curva:  $n$ .
- Numero de puntos de control:  $k$ .
- Numero de puntos de la curva:  $M$ .
- Vector de nodos:  $knot[ ]$ .
- Punto inicial:  $pini$ .
- Punto final:  $pin$ .

La curva de orden  $n$  es generada a partir de una línea recta trazada entre el punto inicial  $pini$  y el punto final  $pin$ , entre estos dos puntos se establecen  $k$  puntos de control con respecto a las distancias establecidas en el vector de nodos  $knot[ ]$ , los puntos de control serán perturbados posteriormente para generar un vector con el valor de las coordenadas de cada uno de los  $M$  puntos de la curva *B-spline*. La curva generada es del tipo uniforme, no periódica y no racional (Boor, 1977).

Una curva paramétrica *B-spline*,  $\mathbf{p}(u)$  de grado  $n$  está definida por  $(k + 1)$  puntos de control  $\mathbf{P}_i$ , el vector de nodos  $knot$  con los valores  $[x_0, x_1, \dots, x_i, \dots, x_m]$ , y la función:

$$\mathbf{p}(u) = \sum_{i=0}^n N_{i,k}(u) \mathbf{P}_i \quad (3.1)$$

donde las *Funciones Base Bernstein*  $N_{i,k}(u)$  se generan recursivamente por las funciones:

$$N_{i,0}(u) = \begin{cases} 1, & \text{Si } x_i \leq u < x_{i+1} \\ 0, & \text{Cualquier otro valor de } u \end{cases} \quad (3.2)$$

y

$$N_{i,k}(u) = \frac{(u - x_i)}{x_{i+k-1} - x_i} N_{i,k-1}(u) + \frac{(x_{i+k} - u)}{x_{i+k} - x_{i+1}} N_{i+1,k-1}(u) \quad (3.3)$$

El vector de nodos  $knot[ ]$  es el elemento que determina la variación de distribución de los puntos de control iniciales, los valores que le sean asignados influirán

directamente en las Ecuaciones 3.2 y 3.3, de las funciones base *Bernstein*. Estas funciones a su vez, al ser multiplicadas por cada uno de los puntos de control e incluidas en la sumatoria generarán los  $M$  puntos de la curva *B-spline* (Ulloa Perez, 1994). El algoritmo que se sigue en la generación de esta curva *B-Spline* está compuesto por tres métodos: inicializar puntos de control, funciones *Bernstein*, y evaluar puntos. Los datos de entrada a este algoritmo son los seis parámetros determinados en el párrafo anterior y los datos de salida se guardan en un vector llamado *muestra*  $[M]$ .

## 3.3 Recocido simulado

### 3.3.1 Definición

Recocido simulado es una técnica estocástica computacional derivada de los métodos estocásticos para encontrar soluciones cercanas al óptimo global en problemas de optimización combinatorios.

Por lo general, encontrar el valor óptimo global de una función objetivo con varios grados de libertad, sujeta a restricciones conflictivas, es un problema complejo; dado que la función objetivo tenderá a tener varios mínimos locales. Un procedimiento para resolver los problemas de optimización difíciles, es muestrear varios valores de la función objetivo de tal manera que se tenga una alta probabilidad de encontrar una solución cercana a la óptima.

### 3.3.2 Proceso de recocido

Un trozo de material a alta temperatura posee un estado de energía más elevado que otro trozo idéntico a baja temperatura. Si se quiere reducir la energía del material a su valor más bajo posible, es necesario disminuir la temperatura al cero absoluto, sin embargo, esta acción no asegura que el material se encuentra en su configuración de energía más baja. Se debe usar un proceso de recocido, consistente en elevar la temperatura del sistema y disminuirla gradualmente, esperando suficiente tiempo en cada etapa de decremento de temperatura para alcanzar el equilibrio térmico. Si el tiempo de espera en cada temperatura es suficientemente grande, especialmente cerca del punto de congelamiento, entonces la probabilidad de obtener una configuración de muy baja energía aumenta.

Para entender cómo el proceso de recocido ayuda al cristal a evitar un mínimo local, se menciona una discusión entre Hinton y Sejnowski: considérese un paisaje de energía simple. En el que una bola deslizante no tiene la suficiente energía inicial para rodar por una cuesta y descender por el otro lado para llegar al mínimo global. Si se *sacude* el

sistema completo, con esto se puede dar suficiente empuje a la bola para subir la cuesta. Con la fuerza que se emplea para sacudir al sistema más o menos en esa proporción será la energía que se dará a la bola para subir la pendiente. De acuerdo a esto, si se dá un sacudimiento vigoroso el empuje que se proporciona a la bola para salir del valle de mínima energía puede ocasionar que se regrese otra vez al lugar del mínimo local. Por el contrario, si se dá un sacudimiento generoso, entonces la bola sale del valle del mínimo local y no cuenta con la energía suficiente para regresar al lado del mínimo local. Sin embargo, aplicando un sacudimiento suave, se debe tomar un buen tiempo antes que se pueda aplicar a la bola el empuje exacto para que esté en el mínimo global en el primer intento.

De las situaciones anteriores se puede concluir que el recocido representa un compromiso entre los sacudimientos vigoroso y suave. A altas temperaturas, las energías térmicas grandes corresponden a los sacudimientos vigorosos mientras que las bajas temperaturas corresponden a sacudimientos suaves. Al recocer un objeto, se le eleva la temperatura y entonces se disminuye gradualmente hasta la temperatura ambiente, permitiendo al objeto buscar un equilibrio en cada paso a lo largo del procedimiento. La técnica de disminuir gradualmente es el mejor camino para asegurar que los mínimos locales puedan ser evitados sin tener que gastar una cantidad infinita de tiempo esperando por una transición de salida de un mínimo local.

### 3.3.3 Algoritmo del recocido simulado

El método se basa en el hecho de que cuando un sólido es calentado y después se le permite enfriar lentamente en forma controlable, la movilidad térmica de sus átomos se pierde y éstos se acomodan para alcanzar una estructura cristalina durante la solidificación. El estado cristalino es el estado libre de mínima energía del sistema. La clave del proceso es el enfriamiento lento, el cual permite un amplio margen de tiempo para la redistribución de los átomos, a la vez que van perdiendo movilidad.

El proceso de recocido simulado se utiliza para buscar en el espacio de soluciones de algún problema de optimización estocástica, una configuración  $z_i$  que minimice la función de costo,  $costo(x)$ , dada. Este mínimo valor se busca perturbando iterativamente configuraciones de acuerdo a algún mecanismo de generación, el cual determinará el tamaño de la perturbación. La aceptación de la configuración  $z_i$  perturbada se determina por la función de aceptación, la cual es una función del cambio en el costo,  $\delta costo$ , es igual a la diferencia en costo de la configuración perturbada y la configuración actual. La temperatura  $t$  se refiere al estado en el cual el algoritmo se está ejecutando actualmente. A medida que el algoritmo avanza, la temperatura es disminuida desde un valor alto dado hasta que el algoritmo termina en alguna temperatura cercana a cero. La función de aceptación más comúnmente usada es la de *Metrópolis*, según la cual una

configuración perturbada es aceptada si  $\delta costo \leq 0$ . En caso contrario,  $\delta costo > 0$ , entonces la probabilidad de que la configuración se acepte está dada por:

$$e^{\left(\frac{-\delta costo}{t}\right)} \quad (3.4)$$

Los pasos básicos del algoritmo de Recocido Simulado son:

1.  $x =$  solución inicial,  $t =$  temperatura inicial
2. Mientras no es satisfecho el criterio externo hacer lo siguiente:
  - (a) Mientras criterio del ciclo interno no se satisface:
    - i. Seleccionar un vecino  $\hat{x}$  de  $x$
    - ii.  $\delta costo \leftarrow costo(\hat{x}) - costo(x)$
    - iii. Si  $\delta costo \leq 0$ , entonces  $x \leftarrow \hat{x}$
    - iv. Si  $\delta costo > 0$ , entonces  $x \leftarrow \hat{x}$  con probabilidad  $Prob(\delta costo) = e^{\left(\frac{-\delta costo}{t}\right)}$
  - (b) Reducir la temperatura  $t$
3. Regresar  $x$

La configuración en este caso, es el vector  $O$  de los puntos de control. Considere  $\rho_{actual}$  como el tamaño de la máxima perturbación permitida para el ciclo actual del algoritmo, para reducir la longitud de la perturbación conforme disminuye la temperatura, se emplea:

$$\rho_{actual} = \rho_{inicial} \left( \frac{\log(t_{actual} - t_{final})}{\log(t_{inicial} - t_{final})} \right) \quad (3.5)$$

en la cual  $t_{inicial}$  es la perturbación máxima inicial y  $t_{actual}$ ,  $t_{inicial}$ ,  $t_{final}$  representan las temperaturas actual, inicial y final del sistema, respectivamente. El ciclo externo del algoritmo de recocido simulado se va reduciendo en porcentaje dado :

$$t_{i+1} = \sigma t_i \quad (3.6)$$

donde  $\sigma$  generalmente toma valores entre 0.80 y 0.90; sin embargo, dentro del ciclo interno, la temperatura se modifica proporcionalmente al valor óptimo actual de la función del costo; cuando un estado es aceptado en el ciclo interno  $i$ , la temperatura interna se modifica con:

$$t_i = \left( \frac{C_i}{C_{anterior}} \right) t_{anterior} \quad (3.7)$$

donde  $C_{anterior}$  y  $t_{anterior}$  son el costo y la temperatura del último estado aceptado respectivamente. Debido a las altas temperaturas, el equilibrio se alcanza más rápido, se introduce una fórmula para aumentar gradualmente el número de ciclos internos  $n_{interna}$  conforme la temperatura disminuye:

$$n_{interna} = n_{lib} \left[ 2 + 8 \left( 1 - \left( \frac{\log(t_{actual} - t_{final})}{\log(t_{inicial} - t_{final})} \right) \right) \right] \quad (3.8)$$

donde  $n_{lib}$  es el número de grados de libertad en el sistema, el cual se calcula multiplicando el número de puntos de control libres por la cantidad de números aleatorios que se generan para calcular la perturbación.

### 3.3.4 Costos de la trayectoria

Para obtener el costo total de la trayectoria, se muestrean varios puntos sobre la curva  $\mathbf{p}(u)$  en intervalos regulares y son sumadas la contribuciones de todos estos puntos por componente del costo. El costo total para cada punto de una configuración dada se define por:

$$C_{total} = C_{prox} + C_{long} + C_{dist} \quad (3.9)$$

donde :

$C_{prox}$  es el costo de la proximidad con los obstáculos,

$C_{long}$  representa el costo de exceso de longitud,

$C_{dist}$  es el costo de distribución paramétrica no uniforme

La función de costo depende del número de puntos muestreados  $M$ , el cual es un parámetro modificable y determina el grado de resolución de la curva resultante. Los componentes individuales de la función de costo son sumatorias de los puntos muestreados a intervalos paramétricos regulares de la siguiente manera:

$$u_0 = 0 \quad (3.10)$$

$$u_i = u_{i-1} + \frac{1}{M-1}, \quad i = 1, 2, \dots, M-1 \quad (3.11)$$

El componente  $C_{prox}$  requiere determinar si un punto se encuentra fuera o dentro de un obstáculo; si se encuentra fuera, es necesario calcular su distancia mínima. Se denotará como  $dist(p(x_i), \mathcal{B})$  a la función que calcula la distancia del punto  $p$  al



obstáculo  $\mathcal{B}$ , dando como resultado un valor negativo si el punto  $p$  se encuentra dentro del obstáculo  $\mathcal{B}$ . La función de costo de proximidad se puede definir como:

$$C_{prox} = \sum_{i=0}^{M-1} C_i \quad (3.12)$$

donde

$$C_i = \begin{cases} K_{prox1}, & \text{si } dist(p(x_i), \mathcal{B}) \leq 0 \\ K_{prox0}, & \text{si } 0 < dist(p(x_i), \mathcal{B}) \leq tolerancia \\ 0, & \text{si } tolerancia < dist(p(x_i), \mathcal{B}) \end{cases} \quad (3.13)$$

donde  $K_{prox1}$  y  $K_{prox0}$  son constantes, tomándose  $K_{prox1}$  arbitrariamente más grande que  $K_{prox0}$ ; *tolerancia* es la banda de tolerancia de un obstáculo dado y es un parámetro modificable. Esta restricción de proximidad permite, como soluciones posibles, muchas curvas con una longitud de arco demasiado grande, por lo tanto, se introduce el costo  $C_{long}$ , para minimizar la longitud del arco. Usando una aproximación cordal del total de longitud de arco  $long_a$ , se penaliza la curva tomando como base su desviación de la línea recta que conecta los puntos extremos:

$$C_{long} = K_{long} (long_a - | \mathbf{P}_f - \mathbf{P}_i |), \quad (3.14)$$

donde

$$long_a = \sum_{i=0}^{M-2} | \mathbf{p}(u_{i+1}) - \mathbf{p}(u_i) | \quad (3.15)$$

donde  $K_{long}$  es una constante positiva que penaliza el exceso de longitud de arco, y  $\mathbf{P}_f$  y  $\mathbf{P}_i$  son los puntos de control final e inicial, respectivamente. Mientras  $C_{prox}$  y  $C_{long}$  controlan el comportamiento global de la curva, el costo de la desigual distribución paramétrica,  $C_{dist}$ , se define de la siguiente manera:

$$C_{dist} = K_{dist} \sum_{i=0}^{M-1} \left| | \mathbf{p}(u_{i+1}) - \mathbf{p}(u_i) | - \frac{long_a}{M-1} \right|, \quad (3.16)$$

donde  $K_{dist}$  es otra constante positiva.

Las constantes  $K_{prox0}$ ,  $K_{prox1}$ ,  $K_{long}$ ,  $K_{dist}$  y  $K_{curv}$  determinan la importancia que se quiere dar a cada componente del costo total. Estos valores se determinan heurísticamente, experimentando con el programa de acuerdo a la calidad de la solución de

procesamiento. Considerando la dinámica del robot, los costos de proximidad deberán darse de tal forma que le ofrezcan seguridad física para pasar a una distancia adecuada de los obstáculos; en tanto que el costo de curvatura debiera permitir obtener una curva suave para que los movimientos del robot no contemplen cambios bruscos de curvatura.

### 3.4 Espacio C

La idea primordial de la configuración de espacio es representar al robot como un punto en un espacio apropiado y mapear los obstáculos en este espacio. Este mapeo transforma al problema de la planeación de movimiento del objeto dimensionado en un problema de planeación de movimiento de un punto. Esto hace que las restricciones del movimiento del robot sean más explícitas siendo más simple calcular la intersección de un punto con respecto a un conjunto de objetos que el cálculo de la intersección de objetos entre ellos mismos.

Sea el robot  $\mathcal{A}$  descrito por un subconjunto compacto de  $\mathcal{W} = \mathcal{R}_N$ ,  $N = 2$  ó  $3$  y los obstáculos  $\mathcal{B}_1, \dots, \mathcal{B}_q$  un subconjunto cerrado de  $\mathcal{W}$ . Sea, además,  $\mathcal{F}_\mathcal{A}$  y  $\mathcal{F}_\mathcal{W}$  marcos Cartesianos incrustados en  $\mathcal{A}$  y  $\mathcal{W}$  respectivamente. Se tiene que  $\mathcal{F}_\mathcal{A}$  es un marco móvil mientras que  $\mathcal{F}_\mathcal{W}$  es fijo. Por definición, dado que  $\mathcal{A}$  es un cuerpo rígido, cada punto  $a$  de  $\mathcal{A}$  posee una posición fija con respecto a  $\mathcal{F}_\mathcal{A}$ . Pero la posición de  $a$  en  $\mathcal{W}$  depende de la posición y orientación relativa de  $\mathcal{F}_\mathcal{A}$  con respecto a  $\mathcal{F}_\mathcal{W}$ . Dado que los obstáculos  $\mathcal{B}_i$  son cuerpos rígidos y fijos en  $\mathcal{W}$ , cada punto de los  $\mathcal{B}_i$  tiene una posición fija con respecto a  $\mathcal{F}_\mathcal{W}$ , para toda  $i \in [1, q]$ .

Una configuración de un objeto arbitrario es una especificación de la posición de cada punto de este objeto con respecto a un marco de referencia fijo (Latombe, 1991). Por lo tanto, una *configuración de  $\mathcal{A}$*  es una especificación de la posición  $\mathcal{T}$  y la orientación  $\Theta$  de  $\mathcal{F}_\mathcal{A}$  con respecto a  $\mathcal{F}_\mathcal{W}$ . La configuración de espacio de  $\mathcal{A}$  es el espacio  $\mathcal{C}$  de todas las configuraciones de  $\mathcal{A}$ . El subconjunto de  $\mathcal{W}$  ocupado por  $\mathcal{A}$  en la configuración  $\mathbf{q}$  se representa por  $\mathcal{A}(\mathbf{q})$ . De la misma forma, el punto  $a$  de  $\mathcal{A}$  en la configuración  $\mathbf{q}$  se representa por  $a(\mathbf{q})$  en  $\mathcal{W}$ . En este trabajo solo se tratará con orientación fija, por lo tanto la configuración de  $\mathcal{A}$  puede ser considerada como un punto  $x \in \mathcal{R}^k$ , este espacio de configuraciones  $k$ -dimensional es llamado *espacio  $\mathcal{C}_\mathcal{A}$* .

La configuración de un poliedro o polígono es un conjunto de parámetros independientes que caracterizan cada punto en el objeto, y se define con respecto a una configuración inicial. En esta configuración inicial, un vértice fijo del objeto coincide con el origen del eje global de coordenadas. Para un poliedro o polígono  $\mathcal{A}$ , este vértice es llamado *vértice de referencia* de  $\mathcal{A}$ , o  $vr_\mathcal{A}$ . El número de parámetros requeridos para especificar la configuración de un polihedro  $\mathcal{A}$  en un espacio  $k$ -dimensional, con relación a su configuración inicial, es  $d$ , donde

$$d = k + \binom{k}{2}, \quad (3.17)$$

es decir, son requeridos  $k$  parámetros para especificar la posición de  $vr_{\mathcal{A}}$  en el espacio  $\mathcal{R}^k$  y  $\binom{k}{2}$  se requieren para especificar la orientación de  $\mathcal{A}$ .

No todas las configuraciones posibles en  $espacioC_{\mathcal{A}}$  representan configuraciones válidas de  $\mathcal{A}$ ; las configuraciones de  $\mathcal{A}$  donde  $\mathcal{A} \cap \mathcal{B}_i \neq \emptyset$  no son válidas porque representan intersecciones. Estas configuraciones no válidas son el resultado de un mapeo de  $\mathcal{B}_i$  hacia  $espacioC_{\mathcal{A}}$ . Este mapeo explota dos propiedades fundamentales:

1. Su *rigidez*, la cual permite que las configuraciones sean caracterizadas por pocos parámetros.
2. Su *solidez*, que significa que un punto no puede estar dentro de más de un objeto.

### 3.4.1 Trayectoria

El concepto de continuidad es fundamental en la definición de una trayectoria. Para formalizar este concepto de continuidad, se debe especificar una función de distancia  $d : \mathcal{C} \times \mathcal{C} \rightarrow \mathbf{R}$ , donde la topología en  $\mathcal{C}$  será la topología métrica inducida por  $d$ . Para uniformizar criterios, la distancia entre dos configuraciones  $\mathbf{q}$  y  $\mathbf{q}'$  será decreciente y tenderá a cero cuando las regiones  $\mathcal{A}(\mathbf{q})$  y  $\mathcal{A}(\mathbf{q}')$  se aproximen una a otra y tiendan a coincidir. Una función de distancia que satisface esta condición es :

$$d(\mathbf{q}, \mathbf{q}') = \max_{a \in \mathcal{A}} \| a(\mathbf{q}) - a(\mathbf{q}') \| \quad (3.18)$$

donde  $\| x - x' \|$  representa la distancia Euclidiana entre dos puntos cualesquiera  $x$  y  $x'$  en  $\mathbf{R}^N$ .

La trayectoria de  $\mathcal{A}$  desde la configuración  $\mathbf{q}_{inicial}$  a la configuración  $\mathbf{q}_{final}$  es un mapa continuo:

$$\tau : [0, 1] \rightarrow \mathcal{C} \quad (3.19)$$

con  $\tau(0) = \mathbf{q}_{inicial}$  y  $\tau(1) = \mathbf{q}_{final}$ .

Donde  $\mathbf{q}_{inicial}$  y  $\mathbf{q}_{final}$  son las configuraciones *inicial* y *final* de la trayectoria, respectivamente. La continuidad  $\tau$  implica que:

para toda  $s_0 \in [0, 1]$ ,  $\lim_{s \rightarrow s_0} \max_{a \in \mathcal{A}} \| a(\tau(s)) - a(\tau(s_0)) \| = 0$ ,

donde  $s$  toma los valores entre  $[0, 1]$ . Se dice que  $\mathcal{A}$  es un *objeto de libre movimiento* cuando en ausencia de obstáculos, cualquier trayectoria definida como anteriormente se indicó, es factible.

### 3.4.2 Obstáculos en la configuración de espacio

Cada obstáculo  $\mathcal{B}_i$ , de  $i=1$  a  $q$ , en el espacio de trabajo  $\mathcal{W}$  mapeado de  $\mathcal{C}$  a una región:

$$\mathcal{CB}_i = \{ \mathbf{q} \in \mathcal{C} \mid \mathcal{A}(\mathbf{q}) \cap \mathcal{B}_i \neq \phi \} \quad (3.20)$$

donde  $\mathcal{CB}$  es definido como *obstáculo-C*. La unión de todos los *obstáculos-C*:

$$\cup_{i=1}^q \mathcal{CB}_i \quad (3.21)$$

el cual es llamada la *región del Obstáculo-C* y el conjunto:

$$\mathcal{C}_{libre} = \mathcal{C} \mid \cup_{i=1}^q \mathcal{CB}_i = \{ \mathbf{q} \in \mathcal{C} \mid \mathcal{A}(\mathbf{q}) \cap (\cup_{i=1}^q \mathcal{B}_i) = \phi \} \quad (3.22)$$

es llamado *el espacio libre*. Cualquier configuración en  $\mathcal{C}_{libre}$  es llamada *configuración libre* (Latombe, 1991).

Una trayectoria libre entre dos configuraciones  $\mathbf{q}_{inicial}$  y  $\mathbf{q}_{final}$  es un mapa continuo :

$$\tau : [0, 1] \rightarrow \mathcal{C}_{libre} \text{ con } \tau(0) = \mathbf{q}_{inicial} \text{ y } \tau(1) = \mathbf{q}_{final} \quad (3.23)$$

Dos configuraciones pertenecen a la misma componente conectada de  $\mathcal{C}_{libre}$  si y solo si están conectadas por una trayectoria libre.

### 3.4.3 Cálculo de la configuración de espacio

Se caracterizará  $\mathcal{CB}_i$  en términos de sumas de conjuntos para el cálculo de los obstáculos de *espacioC*. La *suma*, *resta* y *negación* de conjuntos se definen sobre conjuntos de puntos en  $\mathcal{R}^N$  de la siguiente manera:

$$\mathcal{A} \oplus \mathcal{B} = \{a + b \mid a \in \mathcal{A}, b \in \mathcal{B}\} \quad (3.24)$$

$$\mathcal{A} \ominus \mathcal{B} = \{a - b \mid a \in \mathcal{A}, b \in \mathcal{B}\} \quad (3.25)$$

$$\ominus \mathcal{A} = \{-a \mid a \in \mathcal{A}\} \quad (3.26)$$

Si un conjunto  $\mathcal{A}$  consiste de un solo punto,  $a$ , entonces  $a \oplus \mathcal{B} = \{a\} \oplus \mathcal{B}$ . Además,  $\mathcal{A} \ominus \mathcal{B} = \mathcal{A} \oplus (\ominus \mathcal{B})$ . Nótese que  $\mathcal{A} \oplus \mathcal{A} \neq \{2a \mid a \in \mathcal{A}\}$ , y  $\mathcal{A} \ominus \mathcal{A} \neq \phi$ .

Cuando  $\mathcal{A}$  consiste en un simple punto, no tiene caso hablar de orientación. Entonces, al igual que  $\mathcal{W}$ , el espacio de configuración de  $\mathcal{A}$  es una copia de  $\mathcal{R}^N$  y por lo tanto es un espacio euclidiano. Los *obstáculos-C* son idénticos a los obstáculos en el espacio de trabajo.

Cuando  $\mathcal{A}$  es un disco o cuando  $\mathcal{A}$  es un objeto adimensionado, permite trasladarlo libremente sin rotación, la configuración espacio  $\mathcal{C}$  es también  $\mathcal{R}^N$ . En estos casos los *obstáculos-C* se obtienen “creciendo” los obstáculos por la forma de  $\mathcal{A}$ .

## Capítulo 4

# Robótica de un AGV

## 4.1 Cinemática de vehículos autónomos

### 4.1.1 Modelado cinemático

Los robóts móviles con ruedas son más eficientes en su uso de energía que los que tienen extremidades o los que se arrastran sobre superficies difíciles. Los vehículos de transporte con ruedas que siguen trayectorias marcadas, ya sea por una cinta que provoque refracción; pintura, o por cables guías, se encuentran actualmente en aplicaciones industriales.

Las aplicaciones de los *AGVS* también se desarrollan en exploraciones submarinas, espaciales, manejo de materiales nucleares y explosivos, seguridad, maquinaria agrícola, militar, educacional, movilidad para discapacitados y robóts personales.

El control de robóts móviles requiere una metodología para el modelado, análisis y diseño que ponga en paralelo o imite a las metodologías utilizadas en manipuladores estacionarios.

El objetivo en este apartado consiste en el modelado cinemático de un *WMR* (*Wheeled Mobile Robot*).

Cinemática es el estudio de la geometría del movimiento. El análisis cinemático está basado en la asignación de ejes coordenados entre el robot y su ambiente de trabajo, ver Figura 2.1, y la aplicación de matrices de transformación entre los respectivos sistemas coordenados.

Una metodología establecida para el modelado de la cinemática de robots manipuladores estacionarios empieza con la aplicación de la convención *Denavit-Hartenberg* (Denavit y Hartenberg, 1955) para asignar el eje coordenado a cada una de las uniones del robot. Sucesivamente los ejes coordenados del robot son asociados a los sistemas de coordenadas mediante transformaciones homogéneas de las matrices **A**. Las matrices **A** son especificadas completamente por cuatro parámetros característicos (dos desplazamientos y dos rotaciones) entre sistemas de coordenadas consecutivos. Cada matriz **A** describe tanto la forma como el tamaño de la *extremidad* del robot, y la traslación (para una unión prismática) o rotación (para una unión rotacional) de la unión asociada.

En el caso de los *WMR* se asignan los ejes coordenados a las extremidades y a las ruedas del *WMR*, a los cuales se les aplica la convención *Sheth-Uicker* (Muir, 1994) para definir las matrices transformacionales. Además, la convención *Sheth-Uicker* separa las constantes de forma y tamaño de los parámetros variables de las uniones de las ruedas, y simplifica la formulación de las matrices. La convención *Sheth-Uicker* permite modelar la relación entre el par máximo de cada una de las ruedas del *WMR* y el piso.

La posición y orientación en coordenadas base del efector final de un manipulador estacionario es encontrado multiplicando en cascada las matrices **A** de transformación desde la base hasta el efector final. Las transformaciones de velocidad y aceleración son encontradas diferenciando las transformaciones de coordenadas. La velocidad de las uniones individuales están relacionadas con las velocidades del efector final mediante la matriz Jacobiana del manipulador en la solución directa.

En los robots móviles, la matriz Jacobiana inversa es aplicada en la solución inversa, para calcular las velocidades de las uniones de las variables de cada rueda del *WMR* a las velocidades del cuerpo del robot. Debido a que los *WMR*'s pueden ser cadenas múltiples, las soluciones directa e inversa pueden obtenerse simultáneamente resolviendo las ecuaciones cinemáticas de movimiento en todas las ruedas.

#### 4.1.2 Definiciones y condiciones

Un *WMR*, es un robot capaz de trasladarse sobre una superficie a través de la actuación de ruedas montadas en la base del cuerpo del robot y en contacto con la superficie. Una rueda ensamblada es un elemento que provee o permite movimiento relativo entre la superficie y el equipo que este montado sobre ella, entre los cuales se intenta tener un solo punto de contacto.

Se asume en el diseño que:

- *EL WMR no contiene partes flexibles.*
- *Existen cero o una extremidad por rueda.*
- *Todos los ejes de las extremidades son perpendiculares a la superficie.*

Se asume en la operación que:

- *El WMR se mueve sobre una superficie plana.*
- *La fricción traslacional y el punto de contacto con la superficie es lo suficientemente grande para que no ocurra un deslizamiento.*
- *La fricción rotacional en el punto de contacto entre la rueda y la superficie es lo suficientemente pequeña para que ocurra un deslizamiento rotacional.*

La asignación del sistema de coordenadas es el primer paso en el modelado cinemático de un manipulador estacionario. Haciendo esta analogía, se aplica primeramente la convención *Sheth-Uicker* (Sheth y Uicker, 1971) para asignar los sistemas de coordenadas y modelar cada rueda como una superficie planar en el punto de contacto. Esta convención permite eliminar ambigüedades en las matrices de transformación de coordenadas. La convención *Sheth-Uicker* es ideal para modelar ruedas esféricas; las velocidades angulares de las ruedas son convertidas directamente en velocidades transformacionales a lo largo de la superficie. Los movimientos de ruedas planares están restringidos a solo incluir dos grados de libertad.

Se asignan sistemas de coordenadas en ambos extremos de los elementos del *WMR*. Los elementos de la cadena de un *WMR* son el piso, el cuerpo del robot, y las extremidades. Las uniones tienen un par de revolución en cada eje de la extremidad, un par planar para modelar cada rueda y un par planar para modelar el cuerpo del robot.

Los sistemas coordinados instantáneos coincidentes se consideran como el alcance de un sistema coordinado a otro, en un punto común, en un momento exacto. Si se considera a un sistema coordinado de un robot  $\mathbf{R}$  que se mueve en tres dimensiones:  $X$ ,  $Y$ ,  $\theta$ , además existe un sistema coordinado  $\bar{\mathbf{R}}$  y un sistema coordinado  $\mathbf{F}$ , ambos estacionarios; entonces en el momento en que  $\mathbf{R}$  partiendo de  $\mathbf{F}$ , alcance a  $\bar{\mathbf{R}}$ , existirá un sistema coordinado coincidente instantáneo y  $\mathbf{F}$  permanecerá como un sistema de coordenadas de referencia convencional estático.

### 4.1.3 Sistemas de coordenadas

El sistema de coordenadas del piso  $\mathbf{F}$  es estacionario en relación a la superficie de traslado, y sirve como coordenadas de referencia para los movimientos del robot. El sistema de coordenadas del robot  $\mathbf{R}$  es asignado al cuerpo del robot, así, la posición del desplazamiento del *WMR* está directamente relacionado con la distancia al sistema coordinado del piso  $\mathbf{F}$ . El sistema de coordenadas de la rodilla  $\mathbf{H}_i$  es asignado al punto en el que el cuerpo del robot es intersectado con el eje de *steering* de la rueda  $i$ . El sistema coordinado de la extremidad  $\mathbf{S}_i$ , es asignado en el mismo punto a lo largo de eje de la rueda  $i$ , pero fijo en relación a la extremidad. Se asigna el sistema de coordenadas  $\mathbf{C}_i$  al punto de contacto entre cada rueda y el piso.

Aún cuando la estructura anterior es adecuada, existe la flexibilidad de asignar los sistemas coordinados en las posiciones y orientaciones que convengan a la estructura del modelo cinemático.



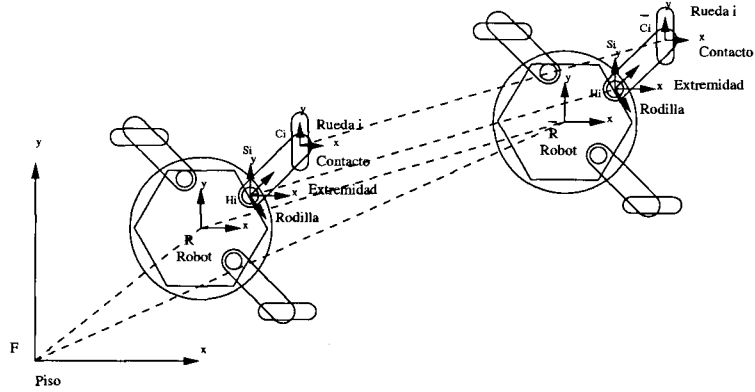


Figura 4.1: Sistemas de referencia de un robot móvil autónomo

#### 4.1.4 Matrices de transformación

Las matrices de transformación homogénea expresan las posiciones relativas y las orientaciones de los sistemas de coordenadas. Las matrices de transformación homogénea  ${}^A\Pi_B$  transforma las coordenadas del punto  $B_r$  en el marco de referencia coordenado  $B$  a sus correspondientes coordenadas en el marco de referencia  $A$ :

$${}^A r = {}^A\Pi_B B_r \quad (4.1)$$

La notación utilizada es la siguiente:

- Las cantidades escalares son denotadas por minúsculas.
- Los vectores son denotados por *letras minúsculas bold*.
- Las matrices son denotadas por *letras mayúsculas bold*.
- Los superíndices denotan referencia al sistema de coordenadas.
- Los subíndices son utilizadas para denotar sistemas de coordenadas o componentes de un vector o matriz.

Los vectores que denotan puntos en el espacio, están formados por tres coordenadas Cartesianas y un factor de escala como cuarto elemento:

$${}^A r = \begin{pmatrix} A r_x \\ A r_y \\ A r_z \\ 1 \end{pmatrix} \quad (4.2)$$

Tabla 4.1: Sistemas de coordenadas

<b>F</b> ( <i>Floor</i> ) <i>Piso</i>	Sistema de coordenadas estacionario con el eje z ortogonal a la superficie de traslado.
<b>R</b> <i>Robot.</i>	Sistema coordinado que se mueve con el cuerpo del <i>WMR</i> , con el eje z ortogonal a la superficie de traslado.
<b>H<sub>i</sub></b> ( <i>Hip</i> ) <i>Rodilla (De i=1...N).</i>	Sistema coordinado que se mueve con el cuerpo del <i>WMR</i> , con su eje z coincidente con el eje de la unión <i>i</i> si existe; coincidente con el sistema coordinado de punto de contacto <b>C<sub>i</sub></b> si no existe una unión con la extremidad.
<b>S<sub>i</sub></b> ( <i>Steering</i> ) <i>Extremidad (De i=1...N).</i>	Sistema coordinado que se mueve con la extremidad <i>i</i> , con su eje z coincidente con el eje z de <b>H<sub>i</sub></b> , y su origen coincidente con el origen de <b>H<sub>i</sub></b> .
<b>C<sub>i</sub></b> ( <i>Contact</i> ) <i>Punto de contacto (De i=1...N).</i>	Sistema coordinado que se mueve con la extremidad <i>i</i> , con su origen en el punto de contacto entre la rueda y la superficie; el eje y es paralelo a la rueda y el plano x-y es tangente a la superficie.

Siempre se usa un factor de escala unitario.

Las matrices de transformación contienen una matriz rotacional de  $(3 \times 3)$  con tres componentes  $(\mathbf{n}, \mathbf{o}, \mathbf{a})$  y el vector traslacional  $\mathbf{p}$ .

$${}^A\Pi_B = \begin{pmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.3)$$

Los tres componentes  $\mathbf{n}$ ,  $\mathbf{o}$  y  $\mathbf{a}$  de la matriz rotacional expresan la orientación de los ejes  $x$ ,  $y$ ,  $z$  respectivamente, del sistema de coordenadas  $B$  con respecto al sistema de coordenadas  $A$  y son ortonormales.

Los tres componentes  $p_x$ ,  $p_y$ ,  $p_z$  del vector traslacional  $\mathbf{p}$  expresan el desplazamiento

del origen del sistema de coordenadas  $B$  con respecto al sistema de coordenadas  $A$  a lo largo de los ejes  $x$ ,  $y$ ,  $z$  del sistema de coordenadas  $A$ .

En general, cualquier par de sistemas coordenados  $A$  y  $B$  en el modelo del  $WMR$  están localizados en coordenadas  $x$ ,  $y$ ,  $z$  distintas de cero relacionadas. La matriz de transformación debe entonces contener las traslaciones  ${}^A d_{Bx}$ ,  ${}^A d_{By}$ ,  ${}^A d_{Bz}$ . Se asignan todos los sistemas coordenados con el eje  $z$  perpendiculares a la superficie de traslado, para que las rotaciones entre sistemas de coordenadas sean con respecto al eje  $z$ .

Para desplazamientos rotacionales y traslacionales iguales a cero, la matriz de transformación de coordenadas se reduce a la matriz de identidad.

Tabla 4.2: Desplazamientos escalares, rotacionales y traslacionales.

${}^A \theta_B$	El desplazamiento rotacional alrededor del eje $z$ del sistema coordenado $A$ entre el eje $x$ del sistema coordenado $A$ y el eje $x$ del sistema coordenado $B$ .
${}^A d_{Bj}$	(Para $j \in [x, y, z]$ ): El desplazamiento traslacional a lo largo del eje $j$ del sistema de coordenadas $A$ entre el origen del sistema de coordenadas $A$ y el origen del sistema de coordenadas $B$ .

Tabla 4.3: Velocidades escalares, rotacionales y traslacionales.

${}^A \omega_B$	La velocidad rotacional ${}^A \dot{\theta}_B$ alrededor del eje $z$ del sistema coordenado $A$ entre el eje $x$ del sistema coordenado $A$ y el eje $x$ del sistema coordenado $B$ .
${}^A v_{Bj}$	(Para $j \in [x, y, z]$ ): La velocidad traslacional ${}^A \dot{d}_{Bj}$ a lo largo del eje $j$ del sistema de coordenadas $A$ entre el origen del sistema de coordenadas $A$ y el origen del sistema de coordenadas $B$ . Ya que todo movimiento es en el plano $x$ - $y$ , la componente $z$ de ${}^A \dot{d}_{Bj}$ de la velocidad traslacional es cero.

Tabla 4.4: Matrices de transformación.

Transformación Floor-Robot	${}^F\mathbf{T}_{\bar{R}}$
Transformación $\bar{R}$ Robot-Robot	$\bar{R}\Phi_R$
Transformación Robot-Hip	${}^R\mathbf{T}_{H_i}$
Transformación Hip-Steering	${}^{H_i}\Phi_{S_i}$
Transformación Steering-Contact	${}^{S_i}\mathbf{T}_{C_i}$
Transformación $\bar{C}$ ontact-Contact	$\bar{C}_i\Phi_{C_i}$
Transformación Floor- $\bar{C}$ ontact	${}^F\mathbf{T}_{\bar{C}_i}$

$${}^F\mathbf{T}_{\bar{R}} = \begin{pmatrix} \cos^F\theta_{\bar{R}} & -\sin^F\theta_{\bar{R}} & 0 & {}^F d_{\bar{R}x} \\ \sin^F\theta_{\bar{R}} & \cos^F\theta_{\bar{R}} & 0 & {}^F d_{\bar{R}y} \\ 0 & 0 & 1 & {}^F d_{\bar{R}z} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.4)$$

$$\bar{R}\Phi_R = \begin{pmatrix} \cos^{\bar{R}}\theta_R & -\sin^{\bar{R}}\theta_R & 0 & \bar{R}d_{Rx} \\ \sin^{\bar{R}}\theta_R & \cos^{\bar{R}}\theta_R & 0 & \bar{R}d_{Ry} \\ 0 & 0 & 1 & \bar{R}d_{Rz} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.5)$$

$${}^R\mathbf{T}_{H_i} = \begin{pmatrix} \cos^R\theta_{H_i} & -\sin^R\theta_{H_i} & 0 & {}^R d_{H_i,x} \\ \sin^R\theta_{H_i} & \cos^R\theta_{H_i} & 0 & {}^R d_{H_i,y} \\ 0 & 0 & 1 & {}^R d_{H_i,z} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.6)$$

$${}^{H_i}\Phi_{S_i} = \begin{pmatrix} \cos^{H_i}\theta_{S_i} & -\sin^{H_i}\theta_{S_i} & 0 & 0 \\ \sin^{H_i}\theta_{S_i} & \cos^{H_i}\theta_{S_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.7)$$

$${}^{S_i}\mathbf{T}_{C_i} = \begin{pmatrix} \cos^{S_i}\theta_{C_i} & -\sin^{S_i}\theta_{C_i} & 0 & {}^{S_i} d_{C_i,x} \\ \sin^{S_i}\theta_{C_i} & \cos^{S_i}\theta_{C_i} & 0 & {}^{S_i} d_{C_i,y} \\ 0 & 0 & 1 & {}^{S_i} d_{C_i,z} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.8)$$

$$\bar{C}_i\Phi_{C_i} = \begin{pmatrix} \cos^{\bar{C}_i}\Phi_{C_i} & -\sin^{\bar{C}_i}\Phi_{C_i} & 0 & \bar{C}_i d_{C_i,x} \\ \sin^{\bar{C}_i}\Phi_{C_i} & \cos^{\bar{C}_i}\Phi_{C_i} & 0 & \bar{C}_i d_{C_i,y} \\ 0 & 0 & 1 & \bar{C}_i d_{C_i,z} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.9)$$

$${}^F\mathbf{T}_{\bar{C}_i} = \begin{pmatrix} \cos^F\theta_{\bar{C}_i} & -\sin^F\theta_{\bar{C}_i} & 0 & {}^F d_{\bar{C}_i} x \\ \sin^F\theta_{\bar{C}_i} & \cos^F\theta_{\bar{C}_i} & 0 & {}^F d_{\bar{C}_i} y \\ 0 & 0 & 1 & {}^F d_{\bar{C}_i} z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.10)$$

#### 4.1.5 Modelo cinemático directo de posición para un WMR

Las estructuras descritas anteriormente tiene la finalidad de inducir un procedimiento para el modelado de las referencias de posición de un *WMR* (Muir, 1994).

En este apartado se describe el desarrollo para encontrar la posición del robot al definir el ángulo al que debe girar el motor del *AGV* para llegar a un punto con referencia al eje de coordenadas establecido (Kelly, 1996).

Primeramente es necesario modelar, definir y caracterizar al *AGV* con el que se va a trabajar, es por esto que se sugiere un procedimiento para modelar al *AGV* en cuanto a sus características físicas, marcos de referencia asignados y posiciones alcanzadas. De esta manera, se encontrará un punto  $x$ - $y$  con referencia a un eje de referencia establecido que indique la distancia y posición del robot.

Los pasos recomendados para llevar a cabo un modelado cinemático de un *AGV* son los siguientes:

1. *Realizar un diagrama del WMR.* Muestre la posición de las ruedas y de las extremidades. El diagrama no necesita ser a escala. Una vista superior y lateral son suficientes, ver Figura 4.2.
2. *Asignar los sistemas coordenados.* Sistemas coordenados del robot, de la rodilla, de la extremidad, del punto de contacto y del piso, de acuerdo a las convenciones ya mencionadas en la Tabla 4.1.
3. *Desarrollar las matrices de transformación de coordenadas.* La matriz *Robot-Hip*, *Hip-Steering*, *Steering-Contact* de acuerdo a la Tabla 4.4.
4. *Formular las ecuaciones de movimiento de posiciones.* Las posiciones relativas y las orientaciones de los dos sistemas de coordenadas estan determinadas por la multiplicación en cascada de las matrices, de la Ec. 4.11.

$${}^F\mathbf{T}_{\bar{C}} = {}^F\mathbf{T}_{\bar{R}} \cdot {}^{\bar{R}}\Phi_{\bar{R}} \cdot {}^R\mathbf{T}_{H_i} \cdot {}^{H_i}\Phi_{S_i} \cdot {}^{S_i}\mathbf{T}_{C_i} \cdot {}^{\bar{C}_i}\Phi_{C_i} \cdot {}^F\mathbf{T}_{\bar{C}_i} \quad (4.11)$$

$${}^F\mathbf{T}_{\bar{C}} = {}^F\mathbf{T}_{\bar{R}} \cdot {}^{\bar{R}}\Phi_{\bar{R}} \cdot I \cdot I \cdot I \cdot {}^{\bar{C}_i}\Phi_{C_i} \cdot {}^F\mathbf{T}_{\bar{C}_i} \quad (4.12)$$

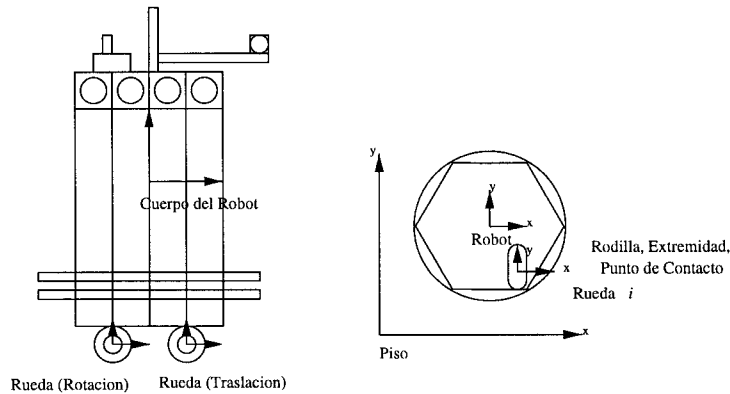


Figura 4.2: Diagrama de referencia del Nomad 200

#### 4.1.6 Modelo cinemático inverso de posición para un WMR

La secuencia en la enseñanza de la robótica básica inicia en la cinemática directa, donde al robot se le entrega un ángulo y se calcula la posición  $x$ - $y$  relativa a un marco de referencia establecido.

Posteriormente se introduce a la cinemática inversa donde la tarea consiste en el cálculo de los ángulos necesarios para alcanzar una posición requerida en términos de coordenadas referidas al mismo marco de referencia.

La tarea de cinemática inversa puede ser calculada de distintas maneras: mediante una aproximación geométrica, ángulos de Euler (Spong y Vidyasagar, 1989), o un cálculo de matrices inversas (Goldenberg y Benhabib, 1985).

Debido a las características del *AGV*, en el cual los motores están desacoplados, esto es, no guardan una relación directa en la que los movimientos del motor de movimiento angular (dirección), afecten al motor que determina los movimientos lineales (traslación), se utiliza la aproximación geométrica para calcular el ángulo y desplazamientos necesarios para alcanzar el punto requerido.

Para esto es necesario mostrar la referencia gráfica en el plano y la manera de calcular el ángulo, ver Figura 4.3.

$$\theta = \Pi + \text{Atan}(\mathbf{p}_x, \mathbf{p}_y) \quad (4.13)$$

$$\mathbf{d} = \sqrt{(d_x)^2 + (d_y)^2 + (d_z)^2} \quad (4.14)$$

## 4.2 Dinámica de vehículos autónomos

El modelo cinemático parece ser adecuado y permite tener un estimado de la posición del vehículo. Esto es debido al hecho de que la velocidad a la cual el vehículo se mueve

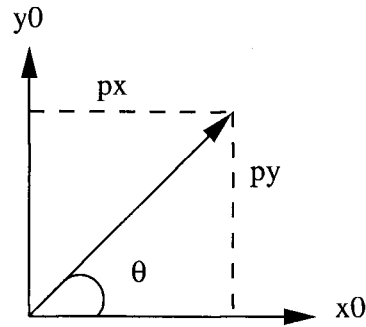


Figura 4.3: Angulo de dirección de giro de las ruedas

es lenta y se hacen a un lado los efectos dinámicos que no introducen errores significativos al sistema.

Sin embargo, un modelo que se aproxime más a las características reales del vehículo, y que toma en cuenta los efectos importantes asociados al movimiento del vehículo, como la carga del motor, debe considerar también los parámetros dinámicos del vehículo.

Así que se describe un modelo dinámico del vehículo y consecuentemente una forma dinámica de estimación de la posición, basándose en las fuerzas que actúan sobre las ruedas del vehículo (Gasparetto et al., 1994), ver Figura 4.4.

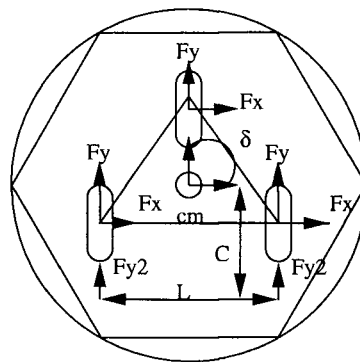


Figura 4.4: Modelo dinámico del Nomad 200

$$\tau = \mathbf{I}\ddot{\theta} \quad (4.15)$$

$$\ddot{\theta} = (L/2(\mathbf{F}_{x2}) - C(\mathbf{F}_{y2}) + (\mathbf{F}_{y1}\cos\delta + \mathbf{F}_{x1}\sin\delta)(\mathbf{B} - \mathbf{C}))/\mathbf{I} \quad (4.16)$$

## 4.3 Teleoperación y control de posición

### 4.3.1 Teleoperación

La teleoperación de robóts surge como una necesidad de realizar movimientos y tareas en ambientes no adecuados para un humano, en los cuales no se pueden realizar las tareas de forma adecuada, debido a las condiciones ambientales presentes.

La teleoperación puede ser entendida como una rama de la robótica actual que consiste en uno o más robóts esclavos dirigidos por un operador maestro humano. Las aplicaciones de la teleoperación se realizan en exploración espacial, manejo de desechos nucleares o tóxicos y exploración submarina, donde todos estos ambientes son peligrosos y difíciles para el humano; así, es preferible tener al operador remoto en un lugar seguro (Joseph, 1985).

En este trabajo de investigación se ofrece al usuario una estrategia sencilla de teleoperación del *AGV Nomad 200*, para que el usuario remoto pueda teledirigir al *AGV* en el ambiente virtual construido. Los movimientos están definidos en cuanto a su longitud (cortos -25 pulg.- y largos -100 pulg.-), y los giros que puede realizar el *AGV* son de  $\pm 90^\circ$ .

La teleoperación se lleva a cabo mediante un grupo de programas ejecutables realizados en *C*, pero llamados desde *JAVA*. Esto se debe a que el *Nomad 200* realiza sus tareas mediante funciones propias generadas en lenguaje *C*. *JAVA* aumenta el potencial de estas rutinas debido a que permite su ejecución desde una página HTML en Internet, ocasionando así un fácil acceso a pruebas de teleoperación de un *AGV* desde lugares remotos, vía un espacio virtual en Internet.

Esta parte de teleoperación, es una forma de ejemplificar el cálculo de cinemática directa en el desplazamiento real de un *AGV*, esto permite al usuario entender el proceso de cálculo de cinemática directa como una herramienta útil de operación del robot y no sólo como la multiplicación de matrices que generen un punto en el espacio pero que no muestren tangiblemente un desplazamiento del robot con las especificaciones de ángulos y distancias.

Las opciones presentadas por el proceso de teleoperación son por el momento relativamente restringidas, pero establecen la base de comprensión del proceso en el laboratorio virtual de robótica. Las opciones presentadas desde la página en Internet permiten mover al robot una distancia específica y con ángulos rectos, estableciendo una referencia mediante un botón que establece al robot en el espacio.

### 4.3.2 Control de posición

A partir de los puntos  $x-y$  generados mediante el proceso de planeación de trayectorias descrito en el capítulo tres, se realiza una estrategia de seguimiento puntual sobre la



trayectoria. Los puntos generados que describen la curva *B-spline* son una sucesión de coordenadas  $x-y$  que construyen una curva de tercer o cuarto orden relativamente suave. El *AGV* debe desplazarse entre estos puntos, calculando los ángulos a los que tiene que girar para poder llegar al punto inmediato siguiente. El desplazamiento entre puntos se calcula mediante las Ec. 4.13 y 4.14, al igual que el ángulo entre ellos (Spong y Vidyasagar, 1989).

La página principal en Internet que presenta esta investigación contiene en su ventana de teleoperación un botón que construye un archivo de datos, transmitido desde el programa en *JAVA*. Le ordena al robot *Nomad 200* mediante el programa en *C*, que realice el seguimiento puntual de la trayectoria generada por la etapa de planeación.

Manejando un análisis formal, este seguimiento puntual es prácticamente un control

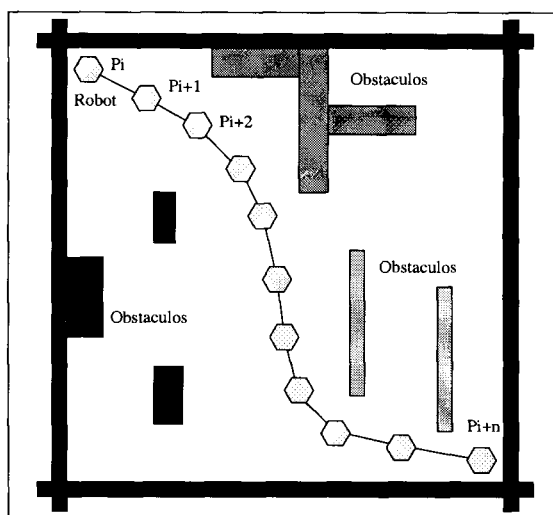


Figura 4.5: Control de posición - seguimiento puntual

de posición, o una sucesión de cálculos de cinemática inversa para el *AGV*, donde se encuentra el ángulo necesario para alcanzar el punto inmediato siguiente de la trayectoria. Se desplaza una distancia calculada, frena y busca el siguiente ángulo. El propósito de incluir este seguimiento puntual es debido a la complementación de este trabajo con la investigación en control de motores dentro del Proyecto de la Red de Investigación en Informática (Cantu de la Garza, 1998) y la demostración del cálculo de cinemática inversa como una experiencia básica en el estudio de la robótica.

Se tiene un módulo de control de motor sincrónico que simula ser utilizados por el *AGV's*, el control - ya sea adaptable, difuso o clásico - recibe como entradas valores de ángulos, y suaviza el arranque en cada cambio de dirección del motor. Es por esto que los motores reciben como entrada un vector de ángulos, mediante los cuales el *AGV* se desplazará a lo largo de la trayectoria y el cambio de referencia será recibido

por el controlador constantemente tratando de suavizar el cambio en los escalones de referencia.

Si bien este proceso de seguimiento puntual no es óptimo, puede ofrecer al usuario una idea de control de posición consecutivo por parte de los motores con efectos sobre un *AGV*, de cualquier manera, el control de velocidad sobre el robot móvil, en el cual se desarrolla una trayectoria real suave sin frenados entre punto y punto a lo largo de la trayectoria será descrita en el siguiente capítulo.

El *AGV* debe trasladarse a lo largo de una trayectoria suave, planeada mediante un proceso fuera de línea en el que se encuentre una trayectoria libre de obstáculos, optimizada en longitud de arco y en proximidad con los obstáculos. Esta trayectoria descrita por las coordenadas  $x-y$  debe ser ejecutada por el *Nomad 200*, desplazándose lo más cerca posible a los parámetros de entrada.

Si el sistema se establece hasta este punto, lo que se presenta es un sistema en lazo abierto en el cual al *AGV* se le ordena viajar sobre una trayectoria planeada sin generar información de posición relativa a su parámetro de entrada. El diagrama presentaría las características mostradas en la Figura 4.6.

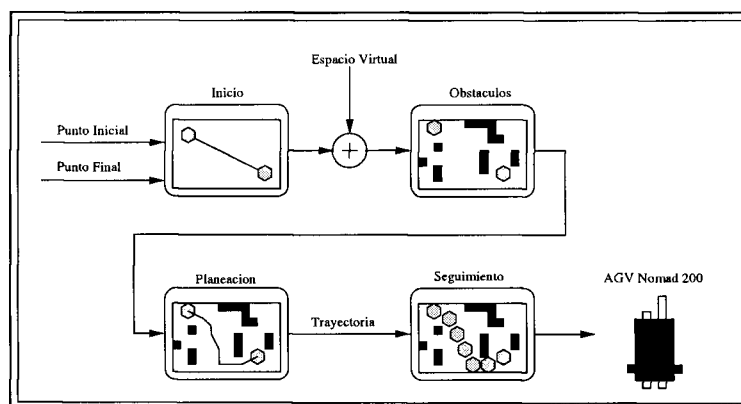


Figura 4.6: Estructura de seguimiento puntual en lazo abierto

## Capítulo 5

# Seguimiento de trayectorias para un AGV

El seguimiento de trayectorias para un *AGV* requiere la programación de:

- *Un Control de velocidad angular.* Mediante este controlador se le ordena al *Nomad 200* la velocidad de ángulo de giro que tiene que modificar para poder encontrar el punto siguiente dentro de la trayectoria, sin detenerse a virar su ángulo de dirección (Gomez Garcia, 1997).
- *Un Control de velocidad lineal.* Mediante este controlador se le ordena al *Nomad 200* la velocidad con la que se desplaza a lo largo de la trayectoria.
- *Una interfaz de retroalimentación de posición.* Que indique su ubicación actual con respecto a la trayectoria trazada.
- *Una interfaz de retroalimentación de obstáculos dinámicos.* Que permita estar en constante monitoreo de obstáculos dinámicos no previstos durante el proceso de planeación.

## 5.1 Control de velocidad angular

La trayectoria planeada para el desplazamiento del *AGV* se compone de un número de puntos determinados, espaciados paramétricamente entre el punto inicial y el punto final. El cálculo de los ángulos correspondientes a las trayectorias entre los puntos fue descrito en el capítulo anterior mediante la cinemática inversa; sin embargo, ahora es necesario identificar la velocidad del movimiento angular entre los puntos para que el robot pueda desplazarse suavemente sobre la trayectoria, girando progresivamente en términos de los ángulos entre los puntos intermedios alcanzados.

El medio mediante el cual se recibe información de retroalimentación al lazo de control debe proveer de información relativa a la posición que guarda el robot en relación con la trayectoria planeada. El instrumento disponible para el sensado de ésta variable son los encoders acoplados a las flechas de los motores de viraje y traslación. Estos instrumentos generan una señal de posición en  $x$  y posición en  $y$  relativas al punto de

inicio, de manera que retroalimentarán la posición real y podrá ser comparada con las coordenadas establecidas por la planeación.

Es necesaria una estrategia de control que regule la velocidad de este movimiento angular con referencia a la trayectoria planeada, esto es, se necesita un controlador que regule la velocidad del movimiento angular y corrija los errores en el desplazamiento leídos por los encoders acoplados a las ruedas del robot móvil.

Una estrategia de seguimiento difuso parece cumplir con los requerimientos que el

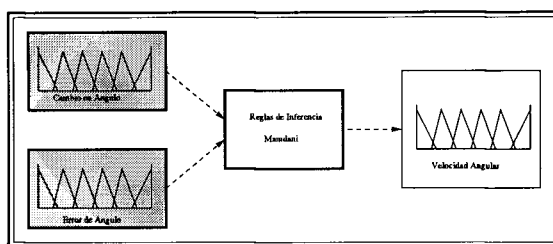


Figura 5.1: Esquema de control difuso

problema de control busca. Si se analiza la idea del control de velocidad para un motor sincrónico de corriente directa como el que tiene el *AGV*, se advierte que los cambios en velocidad angular son sutiles entre los puntos de la curva. Difícilmente se tendrá una trayectoria de desplazamiento para el robot que tenga cambios de dirección extremos, además la señal de velocidad de salida no es un valor exacto, sino un valor con límites ligeramente extendidos que permitan al robot desplazarse suavemente entre puntos con cierta libertad. La solución al problema de seguimiento debe disminuir el error a cero entre la curva planeada y la descrita por el robot, la diferencia entre velocidades angulares menores a cinco pulgadas por segundo no es notoria, y cumple con la disminución del error entre la trayectoria planeada y la real.

Debido a que se contempla la posibilidad de encontrarse con obstáculos dinámicos, es necesario que el controlador a implementar soporte situaciones de perturbaciones extremas, y disminuya a cero el error en la trayectoria. Las perturbaciones no están caracterizadas en magnitud, por tanto, se necesita robustez en el controlador.

Es por estas razones que se determina como una opción factible un controlador difuso para el seguimiento de la trayectoria planeada para el *Nomad 200*.

La opción propuesta por (Gomez Garcia, 1997) es utilizada para este fin. Un controlador difuso que emplea como variables de entrada los cambios de pendientes en los diferentes puntos de la trayectoria y el error originado al desplazarse el robot, y como variable de salida una señal de velocidad angular para el motor que regula la dirección o viraje del robot.

Para cada una de las variables, tanto de entrada como de salida se construyeron siete

conjuntos difusos, definidos para caracterizar el tamaño de la variable como: NG (Negativo Grande), NM (Negativo Mediano), NP (Negativo Pequeño), CERO (Cero), PP (Positivo Pequeño), PM (Positivo Mediano), PG (Positivo Grande).

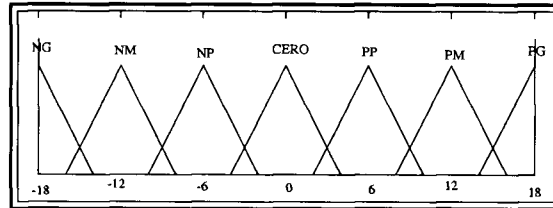


Figura 5.2: Función de membresía para la señal  $\Delta\theta$  de entrada

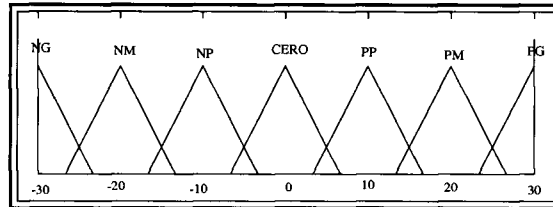


Figura 5.3: Funcion de membresía para la señal  $e\theta$  de entrada

Ya que los conjuntos estuvieron definidos, se construyen cuarenta y nueve reglas de inferencia difusas que determinan una salida en función del tamaño de las variables de entrada. Este gran número de reglas de inferencia permiten cubrir todas las opciones de ocurrencia generadas por las variables de entrada y se le asigna una señal a la variable de salida que corresponda a la decisión tomada ante tal situación.

Las funciones de membresía de las señales de entrada y salida son de forma triangular, las características de cada una fueron determinadas en función del conocimiento del diseñador que ha tenido contacto con la planta (*Nomad 200* (Nomad 200 Hardware and Language Reference Manual, 1997)), respetando así la esencia del control difuso: *el control difuso está basado en el conocimiento de un experto en el manejo y caracterización de las condiciones de la planta* (Kortenkamp et al., 1998).

La relación que guardan las reglas de inferencia de las funciones de membresía estan expuestas en la Tabla 5.1.

Aún cuando la esencia del controlador difuso es la propuesta por Gómez (Gomez Garcia, 1997), es necesario adecuar algunos parámetros, condiciones e incluso relaciones en las

Tabla 5.1: Matriz de relación entre variables de entrada y salida

		$e_\theta$						
		NG	NM	NP	CERO	PP	PM	PG
$\Delta\theta$	NG	NG	NM	NP	CERO	PP	PM	PG
	NM	NM	NM	NP	CERO	PP	PM	PM
	NP	NM	NM	NP	CERO	PP	PM	PM
	CERO	NM	NP	NP	CERO	PP	PM	PM
	PP	NM	NP	NP	CERO	PP	PM	PM
	PM	NM	NP	NP	CERO	PP	PM	PM
	PG	NM	NP	NP	CERO	PP	PM	PG

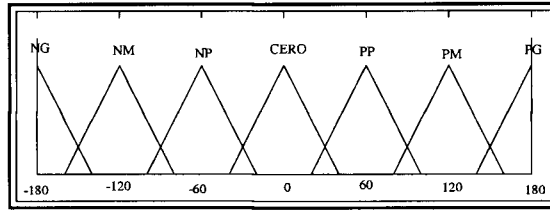


Figura 5.4: Función de membresía para la señal de salida  $\omega$

reglas de inferencia para poder llevar a cabo el control difuso desde la plataforma en *JAVA*, sin acceder a un programa en *C* que se tenga que ejecutar por separado, y permita al usuario llevar a cabo el control difuso de velocidad desde el navegador en Internet.

## 5.2 Control de velocidad lineal

Así como fue necesario el controlador difuso para determinar el ángulo de dirección o giro del robot, también es necesario encontrar la velocidad de traslación y la distancia durante la cual se mantendrá la misma dirección. Para esto es necesario establecer como señal de retroalimentación las señales de los sonares que tiene el *Nomad 200*. De ésta manera los sonares indicarán la distancia relativa a la cual se encuentra algún objeto, y tomará una decisión de disminuir o aumentar la velocidad en función de la distancia a los obstáculos. Esto será más útil en cuanto se contemple la situación de evadir obstáculos que no esten previstos dentro de la planeación fuera de línea de la trayectoria de desplazamiento, ésta situación se analiza en la siguiente sección.

Por el momento, se determina una señal de retroalimentación como entrada que indica la

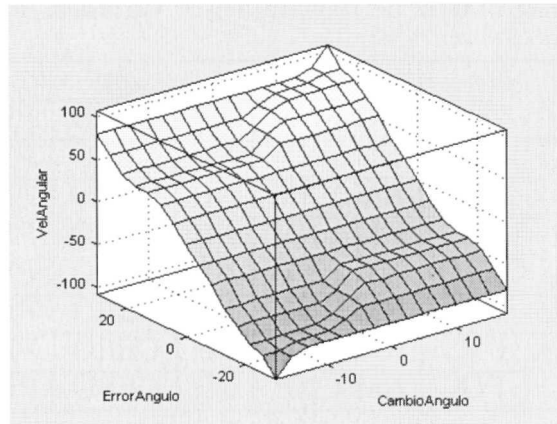


Figura 5.5: Superficie generada por las funciones de membresía

proximidad del robot con los obstáculos, esta señal es generada por los sonares. La señal es recibida y caracterizada de acuerdo a la longitud de proximidad con los obstáculos, se establece en un rango y se determina una señal de salida correspondiente a la situación de proximidad. Una estrategia de este tipo es muy parecida a la programación de ganancias en la que se establece una relación en la cual, a un grupo de señales de entrada les corresponde una señal de salida.

El rango de señal de los sonares se encuentra entre 17 y 255 pulgadas. Es por esto que se establece que, si los sonares envían una lectura de proximidad de objetos en los rangos establecidos, se establece una velocidad de navegación que disminuye inversamente proporcional a la proximidad con el obstáculo. Los rangos y las señales correspondientes de salida son:

- Para una lectura de sonares entre 60 y 255 pulg., la velocidad es de 110 pulg./s. Esto indica que los obstáculos están a una distancia considerable en la cual no ofrecen un peligro inminente para el robot.
- Para una lectura de sonares entre 50 y 59 pulg., la velocidad es de 100 pulg./s. Este es el momento de alerta ya que es la primera señal en la que se detecta una proximidad con un obstáculo que ofrece la posibilidad de una colisión.
- Para una lectura de sonares entre 40 y 49 pulg., la velocidad es de 90 pulg./s. Para una situación en la que se detecte un incremento en la proximidad con el obstáculo se determina una segunda disminución en la velocidad, ya que puede ser una situación de peligro o una situación prevista en la planeación en la que se desplaza el robot muy cerca del obstáculo pero con la seguridad de no colisionarse con éste.

- Para una lectura de sonares entre 30 y 39 pulg., la velocidad es de 80 pulg./s. Al igual que en la situación anterior, se contempla la posibilidad de ser parte de la trayectoria establecida, aún con una proximidad con algún obstáculo muy cercana. Sin embargo, la disminución en la velocidad se establece debido a la consideración de que obstáculo detectado sea un objeto móvil desplazándose en el universo con una trayectoria cercana al *AGV* y con posibilidades de colisionarse.
- Para una lectura de sonares entre 20 y 29 pulg., la velocidad es de 70 pulg./s. Esta es una situación difícil en la que se disminuye la velocidad con el objeto de no realizar un frenado abrupto, sino suave, pero ya se contempla la posibilidad de una colisión con un obstáculo desconocido, no previsto en la planeación.
- Para una lectura de sonares entre 0 y 19 pulg., la velocidad es de 0 pulg./s. El frenado es el principio de una acción evasiva que el robot debe realizar ya que la proximidad con un obstáculo móvil es crucial y debe responder a tal situación mediante una maniobra evasiva al encontrarse en una situación de choque inmediato.

Estas relaciones establecidas por las reglas de inferencia difusas sobre la velocidad lineal del motor que produce la traslación en el *Nomad 200*, conducen a la siguiente sección de esta investigación en donde se establecieron los algoritmos de la estrategia de evasión cuando se suscite una situación de emergencia para evadir un obstáculo dinámico en el universo.

### 5.3 Evasión de obstáculos dinámicos

La situación en la que se encuentra el *AGV* es emergente, si consideramos la secuencia de sucesos en las secciones anteriores. El *AGV* se encuentra detenido y con un obstáculo en movimiento aproximándose hacia él. Para evitar una colisión es necesario interrumpir el seguimiento de la trayectoria, realizar una maniobra evasiva y tomar nuevamente el seguimiento sobre la trayectoria planeada.

Suponiendo una situación en la que se presente un obstáculo sobre la trayectoria planeada para el *AGV*, éste no tendrá, hasta el momento, manera de “saber” que existe un obstáculo que no le permitirá realizar su trayectoria, ya que la planeación fue realizada fuera de línea. Los sonares con los que cuenta el *Nomad 200* (*Nomad 200 Hardware and Language Reference Manual*, 1997) ofrecen la posibilidad de detección de un obstáculo que se encuentre total o parcialmente sobre la trayectoria.

Para un evento de este tipo existen un sinnúmero de soluciones. Partiendo de dos situaciones básicas de caracterización del obstáculo detectado, se toman tres decisiones: frenar al *AGV* por completo, acelerar su desplazamiento o disminuir su velocidad, ver



Figura 5.6.

La situación en la que se encuentra el *AGV* es emergente. Si se contempla la primera

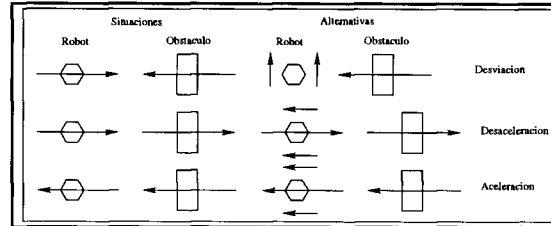


Figura 5.6: Tácticas de evasión de obstáculos dinámicos

opción, el *AGV* se encuentra detenido y con un obstáculo fijo frente a él. La solución es rodearlo y retomar la trayectoria con un error grande que será corregido mediante el controlador difuso. Puede suceder también que el obstáculo esté en movimiento aproximándose hacia él. Para evitar una colisión es necesario interrumpir el seguimiento de la trayectoria, realizar una maniobra evasiva y tomar nuevamente el seguimiento sobre la trayectoria planeada.

La opción de acelerar, será ejecutada en situaciones de aproximación de un obstáculo dinámico por los flancos, en la cual se desarrollará más velocidad para evitar la colisión y que el obstáculo dinámico pase sin causar daños; o por detrás, donde se suscite una parcial persecución del obstáculo dinámico, en ésta situación también se ejecutará una aceleración para evitar ser alcanzado, y se regresará nuevamente a velocidad cruceo cuando no sea detectado el obstáculo por los sonares.

Por último, si se suscita una situación de persecución del robot a un obstáculo que se este desplazando delante de él, se deberá disminuir la velocidad con respecto a la distancia con el objeto móvil, con el fin de evitar la colisión.

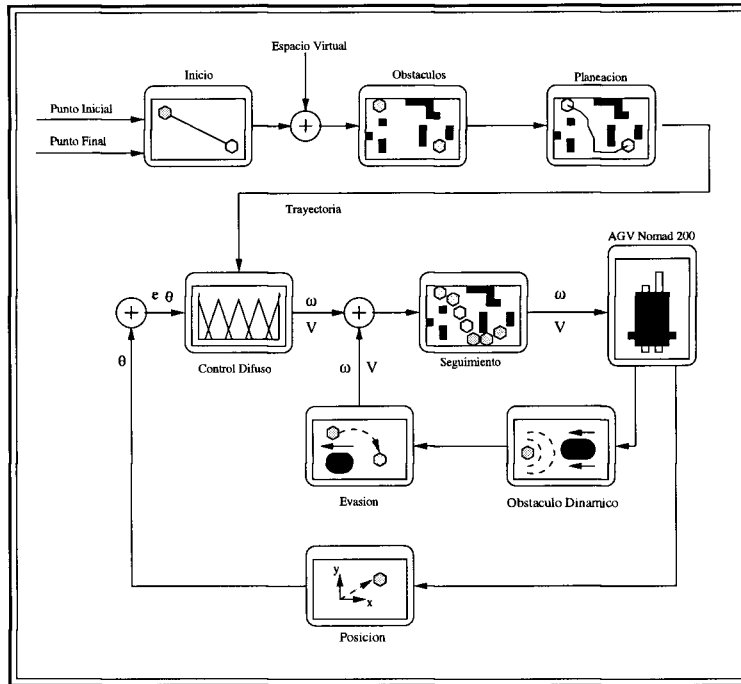


Figura 5.7: Estructura de planeación y seguimiento para un AGV con obstáculos dinámicos

## Capítulo 6

# Simulación y resultados experimentales

### 6.1 Ambientes virtuales

El propósito de trabajar con un lenguaje orientado a objetos como *JAVA*, es desarrollar interfaces gráficas, amigables para el usuario. Si además este trabajo está enfocado a la enseñanza de conceptos de robótica y al trabajo en ambientes virtuales de manufactura, lo propio es presentar una interfaz que permita al usuario una ejecución sencilla de las tareas. La interfaz presenta las opciones pertinentes para que el usuario modifique los valores de entrada para su procesamiento y observe los resultados generados ante distintas circunstancias.

Ante esta situación, el primer resultado que se presenta es el establecimiento de un universo virtual en el cual se llevan a cabo las tareas de robótica. El universo se presenta en tres dimensiones y se utilizará una vista superior en dos dimensiones para tener una perspectiva de la trayectoria que desarrollará el *AGV*.

Se presentan dos universos virtuales, uno de los cuales el usuario podrá elegir como espacio virtual para sus tareas de robótica.

Los universos están formados por:

- *Paredes*: Que acotan el espacio virtual tridimensional.
- *Superficie*: Que establece el plano sobre el que están los objetos.
- *Obstáculos*: Que se encuentran dentro del espacio de trabajo.
- *Manipuladores*: Que conviven en este espacio de manufactura con los *AGVS*.
- *AGV* : Que es el objeto que se desplaza sobre la superficie, entre los obstáculos sin colisionarse con alguno de ellos, sin salirse del espacio acotado por las paredes y se traslada entre estaciones cercanas a los manipuladores.

El objetivo de este programa es que el usuario pueda escoger un universo adecuado a las tareas que tiene planeado realizar. Sería recomendable, para realizar la teleoperación

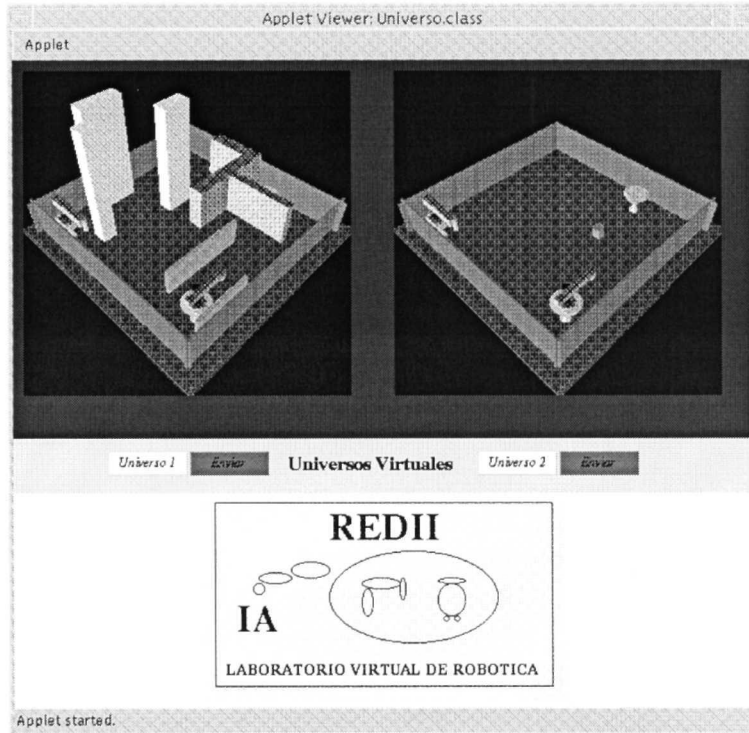


Figura 6.1: Universos virtuales de manufactura

del *AGV*, un universo que no contenga muchos obstáculos. Así, el usuario puede aprender a mover al *AGV* sin ocuparse de evadir obstáculos.

Los universos que se presentan en estas gráficas fueron generados mediante *JAVA 3D*. Se crearon como muestra para el desarrollo de universos virtuales de manufactura, dando un contexto adecuado a las tareas de robótica y control, ver Figura 6.1. De esta manera, se realizan cálculos, de posición (cinemática directa), ángulo (cinemática inversa), coordenadas (planeación de trayectorias) y velocidades (seguimiento), dentro de un ambiente que establece al usuario en su espacio y genera una realidad virtual dentro de la cual el usuario estará experimentando.

## 6.2 Robótica de AGVS

Aún cuando ésta sección no presenta un gráfico como resultado, sino cálculos numéricos, es necesaria incluirla como un formalismo en el aprendizaje de robótica de *AGVS*.

Es necesario recordar que antes de llevar a cabo los cálculos de cinemática directa e inversa, se debe realizar el modelado del robot. Esto es una práctica común en el

análisis de manipuladores robóticos, y debido a las muy variadas formas de los *AGVS* es difícil establecer modelos básicos, sin embargo, para llevar a cabo un correcto análisis y estudio de robótica de *AGVS* es necesario llevarlo a cabo.

El cálculo de cinemática directa se lleva a cabo mediante la multiplicación de las



Figura 6.2: Cálculo de cinemática directa

matrices de transformación. Los datos entregados a éstas matrices son propios del *AGV Nomad 200*. Los valores de los ángulos que la interfaz requiere indican la rotación de los ejes de las llantas. El programa de cinemática directa encontrará las coordenadas  $x$  y  $y$  a las que llegará el *AGV*, ver Figura 6.2.

El cálculo de éstas matrices es un formalismo matemático, sin embargo, al pasar por este proceso, el usuario que tiene conocimientos básicos de robótica tendrá una mejor visualización y aplicación de lo que puede ser una abstracción matemática.

La cinemática inversa en los robots móviles presenta ciertas ventajas en algunos casos como el del *AGV Nomad 200*. El motor que dirige a las ruedas está desacoplado del que las traslada. Esto es, el motor que indica a las ruedas el ángulo al que deben girar para alcanzar un punto, es distinto al que les proporciona la fuerza para el desplazamiento, lo cual permite hacer un análisis simple para encontrar el ángulo necesario para que

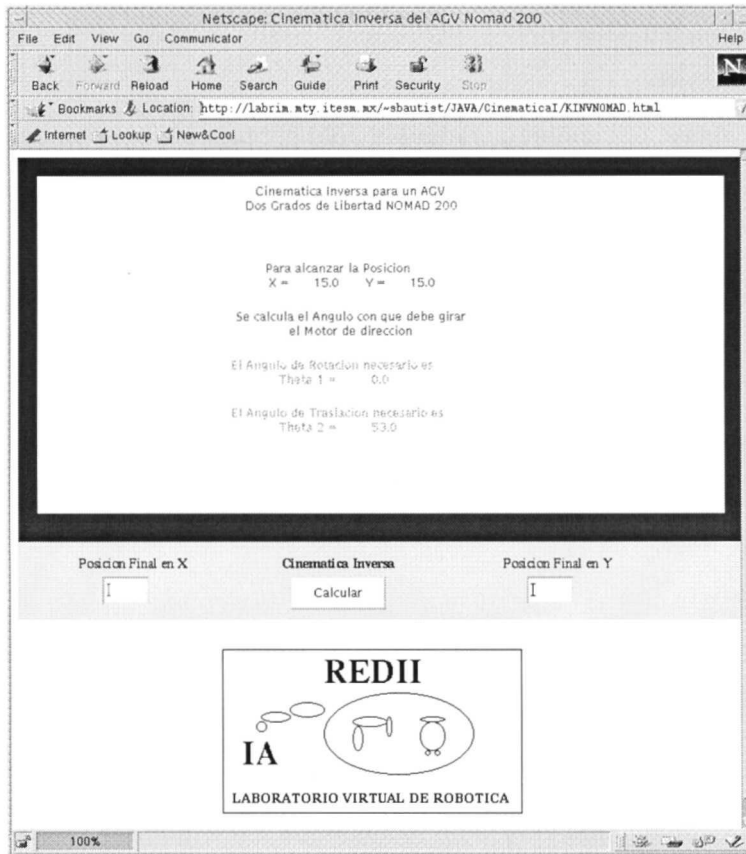


Figura 6.3: Cálculo de cinemática inversa

mediante el motor de giro de las ruedas el robot alcance las coordenadas deseadas, ver Figura 6.3.

La teleoperación del *AGV* es la primera experiencia que el usuario tendrá desde un navegador de Internet con el equipo moviéndose en tiempo real. La interfaz presenta 10 botones que realizan tareas específicas, ver Figura 6.4:

- *Forw-L*: Mueve al *Nomad 200* hacia adelante una distancia de 100 pulgadas. Adecuado para espacios libres de obstáculos donde se pueden tener recorridos largos.
- *Forw-S*: Mueve al *Nomad 200* hacia adelante una distancia de 25 pulgadas. Adecuado para espacios cercanos a obstáculos donde se debe tener precaución para evitar una colisión.
- *Back-L*: Mueve al *Nomad 200* hacia atrás una distancia de 100 pulgadas. Adecuado para espacios libres de obstáculos donde se pueden tener recorridos largos.

- *Back-S*: Mueve al *Nomad 200* hacia atrás una distancia de 25 pulgadas. Adecuado para espacios cercanos a obstáculos donde se debe tener precaución para evitar una colisión.
- *Rigth-L*: Mueve al *Nomad 200* hacia la derecha una distancia de 100 pulgadas. Adecuado para espacios libres de obstáculos donde se pueden tener recorridos largos.
- *Rigth-S*: Mueve al *Nomad 200* hacia la derecha una distancia de 25 pulgadas. Adecuado para espacios cercanos a obstáculos donde se debe tener precaución para evitar una colisión.
- *Left-L*: Mueve al *Nomad 200* hacia la izquierda una distancia de 100 pulgadas. Adecuado para espacios libres de obstáculos donde se pueden tener recorridos largos.
- *Left-S*: Mueve al *Nomad 200* hacia la izquierda una distancia de 25 pulgadas. Adecuado para espacios cercanos a obstáculos donde se debe tener precaución para evitar una colisión.
- *Ref. Cero*: Pone la dirección de las llantas del *Nomad 200* en su referencia *home*, se debe dar este comando entre cada uno de los pasos para evitar una pérdida de orientación por parte del usuario.
- *Trayect*: Realiza una secuencia puntual de una trayectoria.

La Figura 6.5, muestra una secuencia de eventos realizada experimentalmente con el teleoperador. En este experimento, se manejó al *Nomad 200*, desde la página en Internet del teleoperador. Se llevó al robot por un pasillo angosto en el cual los movimientos debían ser en un principio de longitud larga y al llegar a una esquina se debían utilizar los botones de movimientos cortos. La trayectoria y los pasos del movimiento son mostrados en la Figura 6.5. Con este experimento se lleva a cabo el primer contacto de teleoperación del robot desde Internet.

### 6.3 Planeación de trayectorias

La planeación de la trayectoria a seguir por el *AGV* fue generada por el algoritmo de recocido simulado. Al igual que en la sección anterior, una interfaz gráfica fue generada en *JAVA* y presentada de manera simple al usuario para que éste pueda seleccionar el punto inicial y el punto final de la trayectoria a trazar.

La interfaz en *JAVA* genera una gráfica de dos dimensiones que presenta una vista superior de alguno de los universos seleccionado en la sección anterior, ver Figura 6.6.

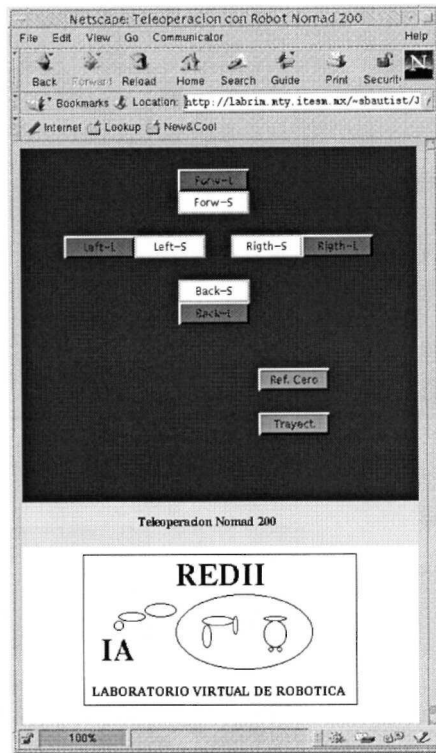
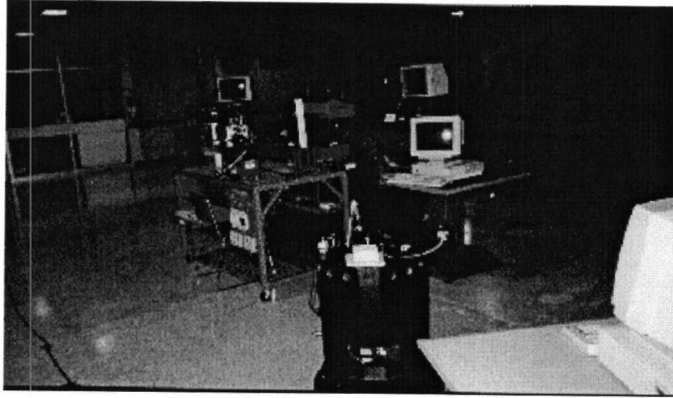


Figura 6.4: Pantalla de teleoperación del AGV Nomad 200

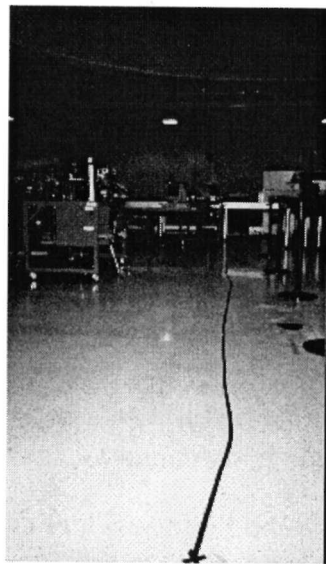
El proceso para generar la trayectoria está indicado por los botones de la interfaz, de manera que el usuario puede ir siguiendo la siguiente secuencia lógica:

- *Definir*: Define el espacio bidimensional en el cual se va a trazar la trayectoria. Después de haber presionado el botón con el indicador del *mouse*, aparecerá la vista superior del universo en el cual se encuentran los obstáculos. Es necesario presionar el botón de *Regresar* para que aparezca nuevamente la gráfica del espacio de dos dimensiones.
- *Ptos. Iniciales*: Al presionar este botón aparecerán deslizadores que permiten al usuario modificar las coordenadas  $x-y$  de los puntos inicial y final. Es necesario presionar el botón de *Regresar* para que aparezca nuevamente la gráfica del espacio de dos dimensiones con dos pequeños cuadros negros correspondientes a las posiciones de los puntos inicial y final, y una línea recta que los une, la cual describe el vector inicial de puntos de control de la curva *B-spline*.
- *Planeación*: Este botón presenta en pantalla una ventana en la que se muestran dos deslizadores correspondientes a la temperatura inicial y final del recocido simulado. Debido a que esta técnica es lenta, los valores de temperatura pueden

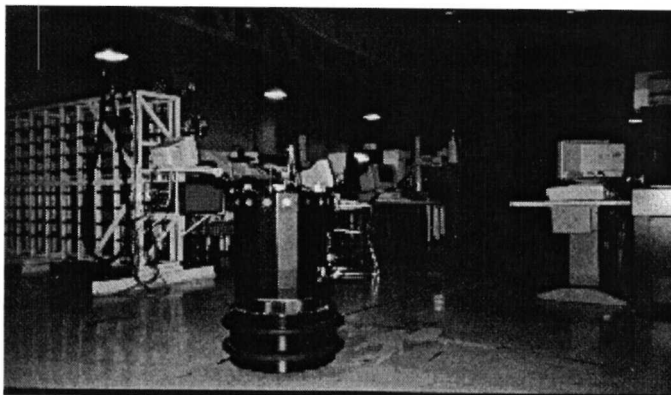




(a) Primera



(b) Segunda



(c) Tercera

Figura 6.5: Experimento de teleoperación  
52

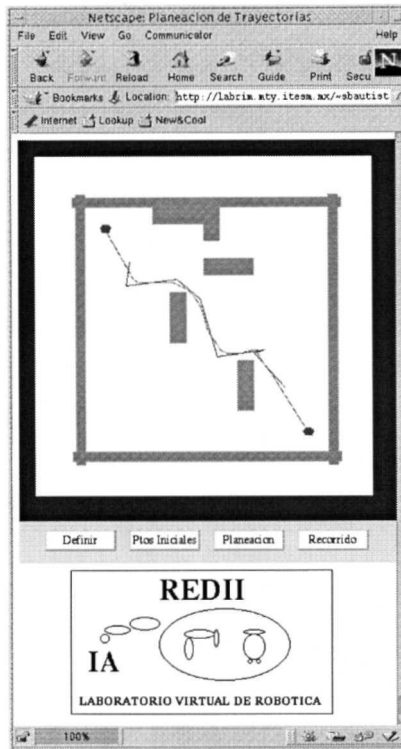


Figura 6.6: Planeación de trayectorias

ser modificados para agilizar el algoritmo, pero esto provoca una alteración en el resultado óptimo de la trayectoria. El proceso del algoritmo como se menciona anteriormente es lento, el botón de *Regresar* ejecutará el procesamiento del algoritmo y regresará a la ventana principal la trayectoria resultante.

- *Recorrido*: Muestra la sucesión de posiciones que alcanza la figura hexagonal del robot a lo largo de la trayectoria encontrada. Si el robot está libre de colisiones con los obstáculos, se ha encontrado una trayectoria adecuada para realizar un seguimiento en el universo virtual.

El objetivo en ésta sección es claro: encontrar una trayectoria adecuada en el universo, para que el *AGV* pueda desplazarse a lo largo de ésta. El algoritmo de recocido simulado, optimiza trayectorias a partir de una línea recta trazada entre punto inicial y final. Esta línea recta es perturbada en cada uno de sus puntos de control para modificarla, ocasionando que se adopte una curva con un menor costo. El costo de la curva está en función de la proximidad con los obstáculos, la longitud del arco y la distribución de los puntos, como ya se estableció en el capítulo tres.

## 6.4 Seguimiento

La integración de las secciones y tareas anteriores es realizada en esta interfaz gráfica. Contando con los elementos dispuestos como el universo virtual, el modelado del robot y el cálculo de cinemática directa e inversa, la teleoperación y la planeación de la trayectoria, se puede llevar a cabo una tarea en la que se integren los elementos anteriores, en la cual un *AGV* real como el *Nomad 200* será el elemento de acción de ésta tarea de seguimiento.

La interfaz gráfica presenta en la ventana correspondiente un universo tridimensional, una vista superior de este universo y un panel de botones. Los botones ejecutan tareas que han sido explicadas como pasos o experimentos anteriores, ver Figura 6.7.

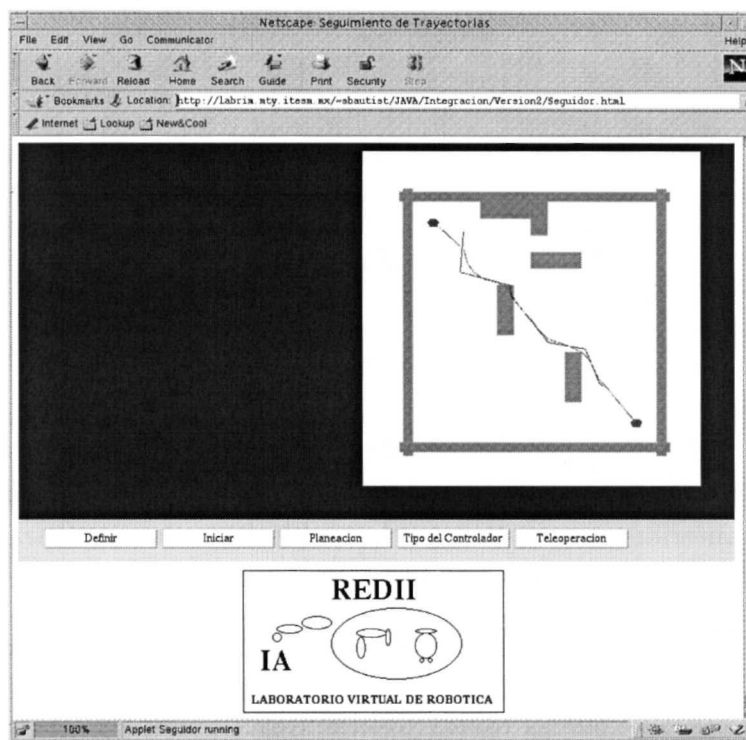


Figura 6.7: Seguimiento de trayectorias

- *Definir*: Define el espacio bidimensional en el cual se va a trazar la trayectoria. Es necesario tomar en cuenta que aquí se llevará a cabo un experimento en tiempo real, esto es, el *AGV* se desplazará en un espacio real tridimensional, por tanto el universo debe ser construido con el mayor parecido posible con la realidad.
- *Inicio*: Si se han tomado las precauciones de establecer un espacio o universo virtual parecido al real, en el cual se llevará a cabo el movimiento del robot, entonces

también es necesario incluir cierta dosis de reserva en cuanto al posicionamiento adecuado de los puntos inicial y final. La distancia que se determine entre estos puntos es proporcional a la distancia que recorrerá el *AGV* en el espacio real.

- *Planeación*: El encontrar una trayectoria óptima, libre de colisiones con los obstáculos es una tarea que el usuario ejecutará al usar el planeador. La estructura de seguimiento tiene su propio planeador de trayectorias, funciona de la misma forma que el programa planeador, solo que en esta sección los datos referentes a la curva trazada serán enviados al *AGV* para que se desplace en referencia a esos parámetros. Por tanto, se deben hacer las pruebas necesarias con el planeador hasta encontrar una trayectoria libre de colisiones para ser enviada al robot.
- *Recorrido*: El botón que ejecuta ésta acción permitirá observar un hexágono que representa al robot en cada uno de los puntos que describen la curva *B-spline* de la trayectoria, esto permitirá observar si el *AGV* efectivamente está libre de colisiones con los obstáculos del universo. Si es así, la trayectoria es adecuada para la ejecución del *Nomad 200*.
- *Seguimiento*: Cuando la trayectoria es generada por el planeador, el siguiente botón del panel es el de seguimiento, realmente es una ventana de teleoperación como la descrita en la sección anterior en la que el botón de *Trayect* enviará los datos generados en el planeador y provocará la ejecución del programa de seguimiento para el robot. Es necesario mencionar que el *AGV Nomad 200* tiene una interfaz propia en lenguaje *C*. En este programa se establece un algoritmo de control difuso mencionado en el capítulo correspondiente al seguimiento y la evasión de obstáculos dinámicos que puedan presentarse en el transcurso del recorrido. La interfaz gráfica en *JAVA* permite la ejecución de movimientos del *AGV* desde el navegador de Internet, lo cual hace amigable, sencillo y disponible para cualquier usuario que cuente con éste recurso, el manejo del robot.

La integración de las tareas anteriores construye un experimento que utiliza las herramientas básicas del modelado de *AGV's*, de la planeación de trayectorias y del control; este experimento involucra a las tareas y experimentos básicos en una herramienta que ejecutará el objetivo perseguido por el presente trabajo de investigación: la planeación y seguimiento de trayectorias para un *AGV* evadiendo obstáculos dinámicos. La interfaz de este experimento, las imágenes del robot y las trayectorias descritas se presentan como evidencia del experimento, en las Figuras 6.9 y 6.10.

## 6.5 Integración del laboratorio virtual

El trabajo que realizan los programas descritos intentan llevar a un usuario con intereses en robótica de vehículos autónomos, a un objetivo final de aprendizaje que lleve

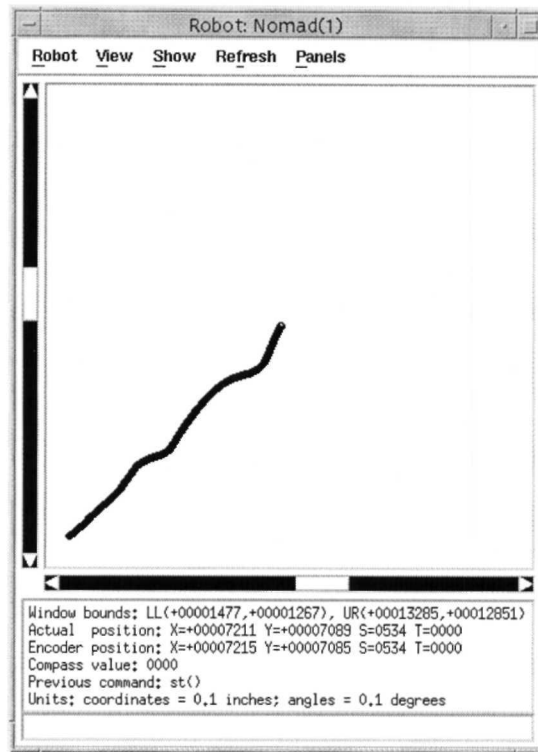


Figura 6.8: Simulación de seguimiento

una secuencia lógica.

Estableciendo primeramente un espacio de manufactura en donde es necesario un vehículo móvil para transportar material entre celdas de trabajo. Este espacio de manufactura es la descripción de la circunstancia del trabajo del *AGV*, le otorga al usuario una perspectiva de actuación para el robot, del tipo de trabajo que se realiza en una celda de manufactura y la utilidad de un robot móvil para éstas tareas.

Si bien el modelado y cálculos de cinemática directa e inversa no son determinantes y deben ser transparentes al usuario que no tenga conocimientos suficientes en el área de robótica, es un formalismo que debe cumplirse en el estudio de la robótica ya sea de manipuladores o de *AGV*'s. Esta tarea le da al usuario una caracterización del elemento que va a controlar, de sus características físicas estática y dinámicas.

Los primeros experimentos de teleoperación desde una interfaz en Internet ocasionarán que el usuario tenga un contacto con el *AGV* y entienda como el modelado y el cálculo de cinemática directa e inversa le dan una idea clara de los movimientos que realiza.

La planeación de trayectorias es una estructura que entregará curvas optimizadas, pero que el usuario tendrá que decidir cual utilizar en términos de los puntos inicial y final. La trayectoria tendrá que estar libre de obstáculos para ser enviada al *AGV*.



(a) Primera



(b) Segunda

Figura 6.9: Experimento de seguimiento



(a) Primera



(b) Segunda



(c) Tercera



(d) Cuarta

Figura 6.10: Experimento de seguimiento: trazado de trayectoria

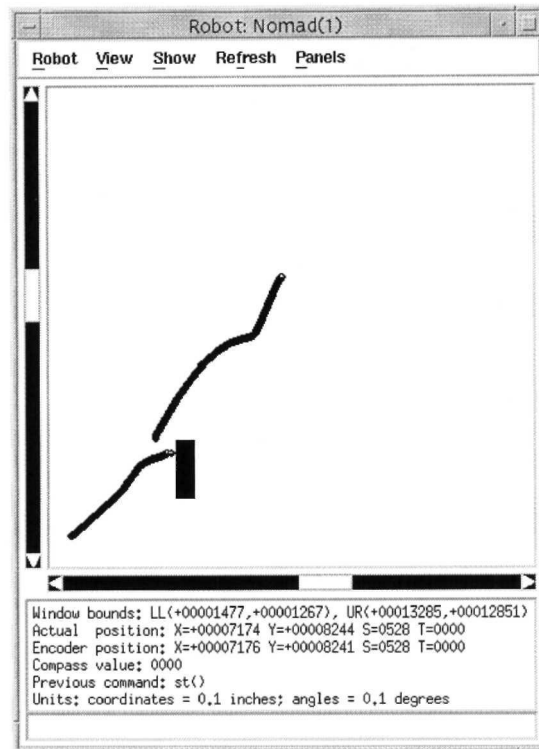


Figura 6.11: Simulación de evasión

Finalmente el seguimiento de la trayectoria y la evasión de obstáculos dinámicos serán los elementos de control sobre el *AGV* en tiempo real.

El laboratorio virtual de robótica, ver Figura 6.12 es un proyecto en desarrollo, por el momento se cuentan con los módulos descritos que caracterizan al *Nomad 200*, ver Figura 6.13. En otra sección del laboratorio se lleva a cabo una descripción de controladores para un manipulador y para un motor síncrono (Cantu de la Garza, 1998). Las ventanas que presentan a manipuladores robóticos tendrán capacidades similares a la ventana del *Nomad*. Los universos serán construidos mediante una interfaz gráfica que permita al usuario poner y quitar obstáculos mediante el indicador del *mouse*, para así poder modificar los universos de manera sencilla. Las trayectorias también serán trazadas en el universo virtual y los movimientos del robot se llevarán a cabo con una animación dentro del universo virtual, conjuntando así un grupo de experiencias en la enseñanza de la robótica.





Figura 6.12: Laboratorio virtual de robótica



Figura 6.13: Ventana del Nomad 200 en el laboratorio virtual de robótica

## Capítulo 7

### Conclusión

El objetivo de la investigación es la planeación y seguimiento de trayectorias para el *AGV* ante obstáculos dinámicos; para lograr esto, es necesario llevar a cabo una sucesión de tareas de robótica y control que al integrarse logren el objetivo planteado.

Cada una de las tareas establece sus parámetros de entrada y salida generando una experiencia al usuario en cada una de las tareas que ocasionan un conocimiento que se ocupará en el desarrollo del experimento final.

Debido a que ésta investigación es parte del Proyecto de de laboratorios virtuales de la Red de Investigación en Informática, toda la secuencia de tareas se establece como un elemento en el aprendizaje de robótica, específicamente en el apartado de vehículos autónomos.

Actualmente es posible ejecutar en Internet, mediante un navegador, las tareas descritas en este reporte.

El ambiente virtual de manufactura se construyó como una necesidad de establecer un espacio de trabajo en el cual el *AGV* desarrolle sus tareas. Se pensó en un ambiente de manufactura porque sería una aplicación real de un *AGV*. Una planta de manufactura es el espacio en el cual los vehículos móviles tienen su mayor aplicación, realizando tareas de transporte de materiales. Los espacios construidos son una aproximación de la realidad, factible de ser modificado. Para las tareas de robótica y control era necesario establecer un marco de referencia en el cual se desarrollarían dichas tareas, así que la creación de éstos universos le dan una idea al usuario de los espacios que tiene para moverse y las longitudes y cotas físicas que tiene como límite .

El modelado del *AGV* se ha dicho que es un mero formalismo, sin embargo, es necesario establecer un esquema matemático descriptivo de las características físicas del robot y así poder dar una idea del esquema matemático que acompaña a la robótica. Para el caso propio del *AGV Nomad 200* se indican cuales fueron las matrices que se tomaron y las consideraciones necesarias hechas para poder establecer el modelo y el cálculo de cinemática directa. Después de haber hecho el modelado y el cálculo de la cinemática directa del *AGV*, se procede al cálculo de la cinemática inversa, ésta tendrá una relación directa con el experimento de seguimiento, ya que es la fase que calcula el ángulo y la distancia con la que se mueve el *AGV*. Las interfaces gráficas que realizan los calcu-

los respectivos necesitan: los ángulos con los que se moveran las ruedas (dirección y traslación) para el problema de cinemática directa y las coordenadas del punto  $x-y$  que se quiere alcanzar (cinemática inversa). Los resultados numéricos son mostrados en pantalla, el usuario tendrá que interpretarlos en relación a las coordenadas requeridas y a los ángulos dados.

En la planeación de trayectorias, el método de recocido simulado como optimizador de una trayectoria recta inicial ha presentado buenos resultados (Martinez Alfaro, 1993), (Ulloa Perez, 1994), (Gomez Garcia, 1997) en anteriores investigaciones. Los cambios hechos en este proyecto consistieron en el parafraseo entre lenguajes de programación  $C$  (Deitel y P.J., 1995) a  $JAVA$  (Deitel y P.J., 1998). Aún cuando los lenguajes en su sintáxis son parecidos,  $JAVA$  por ser un lenguaje orientado a objetos no contiene estructuras como los apuntadores en  $C$ . Las gráficas bidimensionales que se presentan en el proceso de planeación son generadas mediante la secuencia de las tareas de definición del universo, determinación de puntos iniciales, planeación mediante el algoritmo de recocido simulado y el recorrido del robot sobre la curva generada. La secuencia es sencilla para el usuario y le permite modificar algunos parámetros (coordenadas  $x-y$  punto inicial y final, temperatura inicial y final de recocido).

El seguimiento y la evasión de obstáculos dinámicos merecen una distinción debido, principalmente al lenguaje en que fueron programados:  $C$ . Las etapas anteriores fueron hechas en  $JAVA$  (Sun  $JAVA$  3D API Reference Manual, 1997), sin embargo, el  $AGV$  *Nomad 200* tiene como plataforma de desarrollo lenguaje  $C$ , el algoritmo de control difuso utilizado está planteado en una investigación anterior (Gomez Garcia, 1997). El algoritmo de control difuso disminuye el error entre la trayectoria planeada y el recorrido real, sin embargo, cuando la trayectoria tiene curvas muy pronunciadas el  $AGV$  no las puede realizar y falla. Es por esto que también se deben tomar en cuenta en la planeación el costo de la derivada de la curva propuesta en (Martinez Alfaro, 1993). La evasión de obstáculos dinámicos es una situación emergente, que se utilizará en bajo condiciones de peligro de colisión con obstáculos no definidos en la planeación. El *Nomad 200* se detiene cuando detecta mediante los sonares la presencia de algun obstáculo dentro de su rango de sensado, toma alguna de las maniobras de evasión y regresa al recorrido de la trayectoria. La trayectoria recorrida por el  $AGV$  en tiempo real es mostrada en las Figuras 6.8, 6.10.

En términos globales, los resultados de este trabajo de investigación son adecuados a las condiciones del desarrollo del trabajo. La disposición de los programas de las tareas de robótica y control, así como la página principal del laboratorio virtual de robótica están probados en los navegadores de Internet Netscape y HotJava. Es probable que algunas de estas tareas sufrirán una evolución y otras se verán seriamente modificadas; la base del proyecto está construida, los cambios se harán dependiendo de los requerimientos y pruebas hechas a los programas, pero el establecimiento de una secuencia de tareas que funcionen para la enseñanza a usuarios con conocimientos básicos de

robótica permitirá una experimentación de la teoría de vehículos autónomos (*AGVS*) y la experimentación en tiempo real con un robot móvil como el *Nomad 200*, desde una plataforma común como lo es Internet.

## Bibliografía

- Boor, C. D. (1977). A practical guide to splines. *Journal of Mathematics*.
- Cantu de la Garza, A. (1998). Control y simulación de motores y manipuladores. Tesis de maestría, Instituto Tecnológico y de Estudios Superiores de Monterrey.
- Deitel, H., y P.J., D. (1995). *C how to program* (2nd. ed.). New York, NY: Prentice Hall.
- Deitel, H., y P.J., D. (1998). *Java how to program* (1st. ed.). Englewood Cliffs, NJ: Prentice Hall.
- Denavit, J., y Hartenberg, R. (1955). A kinematic notation for lower-pair mechanisms based on matrices. *Journal of Applied Mechanics*.
- Gasparetto, A., Rossi, A., y Robb, I. (1994). Control system design and dynamic simulation of an autonomous vehicle for factory automation. En *International Symposium on Measurement and Control in Robotics*.
- Goldenberg, A., y Benhabib, B. (1985). Robotics. *Journal of Robotics*.
- Gomez Garcia, S. (1997). Planeacion y seguimiento de trayectorias para un robot movil. Tesis de maestría, Instituto Tecnológico y de Estudios Superiores de Monterrey.
- Groover, M. P. (1994). *Automation, production systems, and computer-integrated manufacturing* (2nd. ed.). Englewood Cliffs, NJ: Prentice Hall.
- Joseph, Y. (1985). Robotics. *Journal of Robotics*.
- Kelly, A. (1996). *Essential kinematics for autonomous vehicles* (Tech. Rep.). Pittsburgh, PA: The Robotics Institute, Carnegie Mellon University.
- Kortenkamp, D., Bonasso, P., y Robin, M. (1998). *Artificial intelligence and mobile robots* (1st. ed.). Cambridge, MA: AAAI Press/The MIT Press.
- Latombe, J.-C. (1991). *Robot motion planning* (1st. ed.). Norwell, MA: Kluwer Academic Publishers.
- Martinez Alfaro, H. (1993). *Collision-free path planning for robots using b-splines and simulated annealing*. PhD thesis, Iowa State University.
- Muir, P. F. (1994). *Kinematic modeling of wheeled mobile robots* (Tech. Rep.). Pittsburgh, PA: The Robotics Institute, Carnegie Mellon University.

- (1997). *Nomad 200 hardware and language reference manual*. Nomadic Technologies, Inc., Mountain View, CA.
- Sheth, P., y Uicker, J. (1971). A generalized symbolic notation for mechanisms. *Journal of Engineering for Industry, Series B*.
- Spong, M. W., y Vidyasagar, M. (1989). *Robot dynamics and control* (1st. ed.). Englewood Cliffs, NJ: John Wiley and Sons.
- (1997). *Sun java 3d api reference manual*. SUN Microsystems, Inc., Mountain View, CA.
- Ulloa Perez, A. (1994). Planeacion de trayectorias en espacio-c usando b-splines y recorrido simulado. Tesis de maestría, Instituto Tecnológico y de Estudios Superiores de Monterrey.



