



# INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

CAMPUS ESTADO DE MEXICO  
DIVISION DE GRADUADOS E INVESTIGACION  
DIRECCION DE MAESTRIAS EN INGENIERIA

**BIBLIOTECA**



## GENERACION DE TRAYECTORIAS COMPLEJAS PARA ROBOTS INDUSTRIALES USANDO B-SPLINES RACIONALES

TESIS QUE PARA OPTAR POR EL GRADO DE  
MAESTRO EN SISTEMAS DE MANUFACTURA

P R E S E N T A

**BERNARDO ISIDRO DE LA MERCED SANCHEZ**

ASESOR: DR. PEDRO LUIS GRASA SOLER

COMITE DE TESIS: DR. CUAUHEMOC CARBAJAL

DR. PEDRO LUIS GRASA SOLER

M. EN C. AGUSTIN VASQUEZ ARVALLO

JURADO: DR. CUAUHEMOC CARBAJAL

PRESIDENTE

DR. PEDRO LUIS GRASA SOLER

SECRETARIO

M. EN C. JUAN CARLOS PEDROZA

VOCAL

ATIZAPAN DE ZARAGOZA, MEXICO, MAYO DE 1995

TESIS  
TS  
191.8  
.M4  
1995

15 DIC 1997 ITESM-CEM

2000  
MAY 2000

07 AGO 1998

18 DIC 1996

18 DIC 1996

15 DIC 1998

29 MAY 1996

ITESM-CEM

63889

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY  
CAMPUS ESTADO DE MÉXICO

DIVISIÓN DE GRADUADOS E INVESTIGACIÓN      DIRECCIÓN DE MAESTRÍAS EN INGENIERÍA

2. SOLICITUD DE REGISTRO DE TESIS

**Dr. EMIL LIEBERMANN G.**  
Director de Maestrías en Ingeniería  
ITESM-CEM

De acuerdo con los Arts. 5, inc. c; 8, inc. b y 9 del Reglamento de Tesis para las maestrías en ingeniería del ITESM-CEM, solicito que se registre el tema de la tesis que desarrollaré para optar el grado de Maestro. Adjunto a esta solicitud la propuesta de tesis correspondiente.

Autor (Nombre, matr., dir, tel. ): Bernardo de la Merced Sánchez 310038. Sector 10 A depto. 201 Infonavit Norte, C. Izcalli C.P. 54720 Tel. 8733059 Especialidad: Sistemas de Manufactura  
Título propuesto: Generación de trayectorias complejas para robots industriales usando B-Splines racionales.

Asesor (Nombre, puesto, dirección, teléfono) Dr. Pedro Luis Grasa Soler, Director del Centro de Tecnología y Productividad.

Miembros Comité (Nombres, dir. tel.) Dr. Cuauhtémoc Carbajal  
M. en C. Agustín Vásquez Arvallo

Semestre en que se inscribirá en la materia Tesis: \_\_\_\_\_


Atizapán de Zaragoza, \_\_\_\_\_

Atentamente,



\_\_\_\_\_  
**Bernardo I. de la Merced Sánchez**

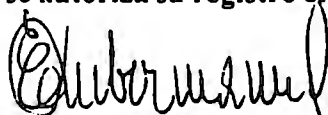
Apruebo la propuesta de tesis adjunta que fue elaborada con mi asesoramiento y recomiendo su registro,



\_\_\_\_\_  
**Dr. Pedro Grasa Soler**  
Asesor

Se acepta la propuesta de tesis y se autoriza su registro el día \_\_\_\_\_

22 agosto 1994



\_\_\_\_\_  
**Dr. Emil Liebermann G.**  
Director de Maestrías en Ingeniería

## **Agradecimientos.**

Estos últimos dos años han sido muy importantes en mi vida y en mi formación profesional. Quiero expresar mi más sincero agradecimiento a todas las personas que hicieron esto posible, especialmente por el apoyo y la ayuda brindada para la satisfactoria culminación de la maestría; por ello dedico este trabajo:

A mis padres, hermanos, y especialmente a mis sobrinas Deyanira y Stephanie; deseándoles toda la felicidad y éxito del mundo.

A Guadalupe Vázquez por todo lo que representa para mí.

A mis profesores de la maestría por su invaluable enseñanza.

A mis compañeros de grupo por la ayuda y experiencias que me han regalado.

A mi asesor Dr. Pedro Grasa Soler y a mi "coasesor" M. en C. Agustín Vázquez Arvalo, por el enorme apoyo que me han dado antes y durante el transcurso de la maestría.

Finalmente quiero dedicar esta tesis a la memoria del Ing. Rubén Arbona, por haberme brindado la oportunidad de empezar a trabajar en el área de la robótica.

Bernardo de la Merced  
Mayo de 1995

---

---

# INDICE

---

---

## CAPÍTULO 1. INTRODUCCIÓN

1.1 Robótica industrial . . . . .	1
1.1.1 Aplicaciones de la robótica industrial . . . . .	2
1.1.2 Tecnología del robot . . . . .	3
1.2 El problema fundamental de la robótica . . . . .	5
1.3 Antecedentes del estudio de trayectorias . . . . .	7
1.4 Importancia de la planeación de trayectorias . . . . .	8
1.5 Planteamiento del problema . . . . .	9
1.6 Objetivos de la tesis . . . . .	10

## CAPÍTULO 2. FUNDAMENTOS DE TRAYECTORIAS PARA ROBOTS INDUSTRIALES

2.1 Caracterización de tareas . . . . .	11
2.1.1 Herramientas para la caracterización de tareas . . . . .	12
2.1.2 Metodología . . . . .	14
2.2 El problema de planeación de rutas óptimas . . . . .	15
2.3 El problema de la cinemática inversa . . . . .	18
2.4 Restricciones para trayectorias . . . . .	20
2.5 Planeación de rutas usando el modelo dinámico . . . . .	20
2.6 Planeación de rutas libres de colisión . . . . .	21
2.7 Resumen del problema de planeación de trayectorias . . . . .	21

## CAPÍTULO 3. ANÁLISIS GEOMÉTRICO Y MATEMÁTICO DE TRAYECTORIAS

3.1 Características de las curvas planas y espaciales . . . . .	23
3.1.1 Construcción de curvas paramétricas . . . . .	23
3.1.1.1 Curvas paramétricas planas . . . . .	24
3.1.1.2. Curvas paramétricas espaciales . . . . .	24
3.1.2 Curvas diferenciales parametrizadas . . . . .	25
3.1.3 Curvas parametrizadas por longitud de arco . . . . .	25
3.1.4 Teorema fundamental de existencia y unicidad de curvas en el espacio . . . . .	25
3.2 Métodos para la generación de curvas complejas . . . . .	26
3.2.1 Métodos generales . . . . .	26
3.2.1.1 Polinomio de grado $n$ . . . . .	26
3.2.1.2 Splines . . . . .	27

3.2.2 Métodos locales . . . . .	27
3.2.2.1 Aproximación con segmentos de línea recta . . . . .	27
3.2.2.2 Aproximación con arcos de círculo . . . . .	28
3.2.2.3 Aproximación con splines . . . . .	29
3.2.2.4 B-Splines racionales . . . . .	29
3.3 Selección del método de generación de curvas complejas . . . . .	30
3.4 El polinomio de aproximación de Bernstein . . . . .	31
3.4.1 Aproximación polinomial . . . . .	31
3.4.2 Teorema de aproximación de Weierstrass . . . . .	31
3.4.3 Definición del polinomio de aproximación de Bernstein . . . . .	32
3.4.4 Propiedades matemáticas . . . . .	33
3.5 El método de Bernstein-Bézier para trazar splines . . . . .	33
3.5.1 Algoritmo para la construcción de una curva de Bézier . . . . .	33
3.5.2 Propiedades de las curvas de Bézier . . . . .	35
3.6 Curvas racionales de Bézier (B-splines racionales) . . . . .	36
3.6.1 La forma general de los B-splines racionales . . . . .	36
3.6.2 Propiedades de los B-splines racionales . . . . .	37
3.6.3 Los parámetros de control de la forma geométrica de la curva . . . . .	38
3.6.3.1 Implicaciones de la variación de los pesos sobre las funciones base . . . . .	38
3.6.3.2 Significado geométrico de los parámetros de forma . . . . .	39
3.7 Implementación del método de los B-splines racionales . . . . .	42
3.7.1 Funciones base de Bernstein . . . . .	43
3.7.2 Polinomios de Bernstein-Bézier . . . . .	45
3.7.3 B-splines no racionales . . . . .	46
3.7.4 Factores de forma (pesos) . . . . .	47
3.7.5 B-splines racionales . . . . .	48
3.7.6 B-splines racionales para posicionamiento y orientación . . . . .	52

**CAPÍTULO 4. GENERACIÓN DE TRAYECTORIAS**

4.1 Introducción . . . . .	53
4.2 Espacio de articulación . . . . .	54
4.2.1 Interpolación polinomial . . . . .	56
4.2.1.1 Ligaduras para la planificación de trayectorias de articulación interpoladas . . . . .	57
4.2.1.2 Algoritmo . . . . .	58
4.3 Espacio cartesiano . . . . .	59
4.3.1 Movimiento cartesiano en línea recta . . . . .	62
4.3.2 Problemas geométricos con rutas cartesianas . . . . .	63
4.3.2.1 Puntos intermedios inalcanzables . . . . .	63
4.3.2.2 Altas velocidades de articulación cerca de singularidades . . . . .	64
4.3.2.3 Puntos inicial y final alcanzables en diferentes soluciones . . . . .	64
4.4 Aproximación cartesiana en espacio de articulación . . . . .	65
4.5 Topología del espacio de configuración . . . . .	66
4.6 Generación de trayectorias usando B-splines racionales . . . . .	69
4.6.1 Metodología para la planeación de la trayectoria . . . . .	69

**CAPÍTULO 5. SISTEMA GENERADOR DE TRAYECTORIAS COMPLEJAS**

5.1 Descripción del problema .....	76
5.2 Planeación de ruta .....	78
5.3 Generación de trayectoria .....	79
5.4 Consideraciones cinemáticas y dinámicas .....	80

**CAPÍTULO 6. CONCLUSIONES**

6.1 Conclusiones .....	83
------------------------	----

**APÉNDICE**

A1 Corridas del programa generador de trayectorias .....	86
Ejemplo 1 .....	86
Ejemplo 2 .....	94

<b>BIBLIOGRAFÍA Y REFERENCIAS</b> .....	102
-----------------------------------------	-----



# CAPÍTULO 1

## INTRODUCCIÓN

### 1.1 ROBÓTICA INDUSTRIAL

En los últimos años, la exigencia de aumentar la productividad y mejorar la calidad, conjuntamente con el aumento en la diversidad de los productos fabricados en las mismas instalaciones, hace inadecuada la automatización rígida de las primeras décadas del siglo XX, ver Figura 1.1. La robótica y los *Sistemas de Manufactura Flexible (FMS)* nacieron de la necesidad no de la fantasía. Un amplio estudio sobre la historia y los fundamentos de la robótica es presentado en PROMAH[1].

AUTOMATIZACION	◦ FIJA. (Volúmenes de producción altos)	
	◦ FLEXIBLE	- Por lotes. (Vol. medios)
		- Por piezas. (Vol. bajas)
		ROBOTICA

Figura 1.1 Técnicas de automatización.

La robótica hace referencia a una automatización importante de numerosos sectores de la actividad humana en los cuales se estimaba imprescindible, hasta hace poco tiempo, la presencia del hombre.

Aunque el concepto actual de robot es muy ambiguo, hay dos características específicas para distinguir a un robot:

- a) **Versatilidad.** Potencialidad estructural de ejecutar diferentes tareas y/o realizar una misma tarea en formas diversas.
- b) **Autoadaptabilidad** Posibilidad de un robot para alcanzar su objetivo por sí mismo, a pesar de las perturbaciones imprevistas del entorno a lo largo de la ejecución de la tarea.

### 1.1.1 Aplicaciones de la robótica industrial

Según la Robotic Industries Asociation (RIA), anteriormente el Robotic Institute of America (RIA):

*"Un robot industrial es un manipulador multifuncional reprogramable diseñado para desplazar materiales, piezas, herramientas o dispositivos especiales mediante movimientos programados variables para la ejecución de tareas diversas".*

Las tareas desempeñadas por un robot han aumentando en complejidad en forma rápida. Las diferencias entre las tareas que puede realizar sin dificultad y aquellas que (posiblemente) no podrá realizar jamás son muy sutiles, ver Figura 1.2. Las razones que hacen que una tarea sea difícil para un robot son:

- a) Que involucre decisiones que cambian continuamente en tiempo real.
- b) Que el criterio de decisión no pueda reducirse a un conjunto de ecuaciones por el gran número de variables involucradas en el modelo.

Cosas que pueden hacer los robots de hoy	Cosas que la próxima generación de robots serán capaces de hacer	Cosas que un robot muy sofisticado del futuro puede ser capaz de hacer	Cosas que ningún robot será capaz de hacer (probablemente)
<ul style="list-style-type: none"> <li>- Tocar el piano</li> <li>- Cargar / descargar máquinas de CNC</li> <li>- Cargar / descargar máquinas de fundición en moldes permanentes, de moldeo, etc.</li> <li>- Pintar con spray en una línea de montaje</li> <li>- Cortar materiales con rayo láser</li> <li>- Hacer moldes</li> <li>- Desbarbar piezas vaciadas en arena</li> <li>- Manipular herramientas</li> <li>- Ensamblar piezas eléctricas y mecánicas simples (pequeños motores, radios, grabadoras de cinta, juguetes, etc.)</li> </ul>	<ul style="list-style-type: none"> <li>- Cargar / descargar una máquina de corte o soplado de vidrio</li> <li>- Ensamblar piezas grandes y/o complejas (Televisores, hornos de microondas, refrigeradores, automóviles, etc.)</li> <li>- Operar máquinas de carpintería</li> <li>- Caminar sobre dos patas</li> <li>- Afeitarse</li> <li>- Limpiar ventanas</li> <li>- Limpieza de paredes con chorros de arena o agua</li> <li>- Limpiar cascos de barcos</li> </ul>	<ul style="list-style-type: none"> <li>- Poner una mesa</li> <li>- Limpiar una mesa</li> <li>- Hacer juegos de mano con una bola</li> <li>- Cargar una lavaplatos</li> <li>- Soldar una pieza agrietada</li> <li>- Tender una cama</li> <li>- Localizar y reparar fugas dentro de un tanque y tuberías</li> <li>- Hacer el diseño de un bordado</li> <li>- Hacer un nudo</li> <li>- Engrasar una máquina</li> <li>- Afinar un coche</li> <li>- Cargar, operar y descargar una máquina de coser</li> <li>- Construir una barda de ladrillo</li> <li>- Cambiar un neumático</li> <li>- Operar un tractor o cosechadora en un campo plano</li> <li>- Poner gasolina</li> <li>- Recoger fruta</li> <li>- Realizar un salto mortal</li> </ul>	<ul style="list-style-type: none"> <li>- Cortar un diamante y pulir piedras preciosas</li> <li>- Pelar una uva</li> <li>- Reparar una silla o un plato roto</li> <li>- Remendar ropa</li> <li>- Jugar tenis o ping-pong a nivel de campeón</li> <li>- Saltar con garrocha</li> <li>- Bailar un ballet</li> <li>- Montar en bicicleta en tráfico</li> <li>- Arboricultura</li> <li>- Ensamblar el esqueleto de un dinosaurio</li> <li>- Cortar el pelo elegantemente</li> <li>- Realizar obras de arte</li> <li>- Eliminar el apéndice en una operación quirúrgica</li> <li>- Tocar el violín</li> <li>- Tallar madera o mármol</li> <li>- Limpiar obras de arte</li> <li>- Cargar a un niño</li> <li>- Cortar y trinchar carne</li> <li>- Fijar una fractura múltiple</li> </ul>

Figura 1.2 Tareas presentes y futuras de los robots Groover[2].

La robótica industrial es uno de los grandes campos de aplicación de los robots. Las principales ventajas en las aplicaciones de robótica industrial están relacionadas con el aumento de productividad y mejora sustancial de la calidad de los productos; estas ventajas son:

- a) Aumento en la velocidad de los procesos,
- b) Elevado tiempo de funcionamiento sin fallas (trabajo uniforme e ininterrumpido),

- c) El diseño modular reduce tiempo de mantenimiento,
- d) Optimización en el empleo del equipo y/o la maquinaria atendida por el robot,
- e) Flexibilidad en la producción de series cortas y medianas,

Una breve descripción de las aplicaciones de los robots en la industria se presenta en la Figura 1.3, la información fue tomada de Groover[2] y Ferráte[3].

APLICACIONES DE LOS ROBOTS INDUSTRIALES	Manipulación	Transferencia de Material	- Operaciones de transporte (asir y situar)	- Piezas inmóviles - Piezas en transportador - Reconocimiento de formas
			- Paletización y operaciones relacionadas	- Paletización - Empaque individual - Embalaje
		Carga y descarga de máquinas	- Carga (alimentación)	- Forjadoras - Cortadoras - Troqueladoras
			- Descarga	- Piezas inyectadas - Vidrio soplado - Piezas extruidas
			- Carga / Descarga	- Tornos - Fresas - Taladros - Centros de maquinado - Carrusel de herramienta
	Procesamiento	Soldadura	- Por puntos - De arco - Continua	
		Acabado superficial	- Pintura con spray - Metalizado	
		Maquinado	- Taladrado - Desbastado y desbarbado - Pulido - Corte (láser, chorro de agua o arena, etc.)	
	Montaje	Coincidencia de piezas	- Montajes árbol/agujero - Aplamamiento	
		Unión de piezas	- Tornillo / tuerca - Pasadores y seguros - Soldadura fina (estaño, cobre, ultrasonido, etc.) - Encolado (pegado) - Coaldo	
	Inspección	Medición	- Visual - Táctil	
		Comparación	- Otros sentidos	

Figura 1.3 Aplicaciones de la robótica industrial.

### 1.1.2 Tecnología del robot

De acuerdo con PROMAHI[1], un robot en funcionamiento puede representarse por cuatro subsistemas funcionales como se indica en la Figura 1.4.

1. Brazo mecánico (robot físico).
2. Entorno. Universo en que esta inmerso el robot físico.
3. Cerebro del robot. Dispositivo de tratamiento de información.
4. Tareas a realizar. Trabajos encomendados a la primera unidad.

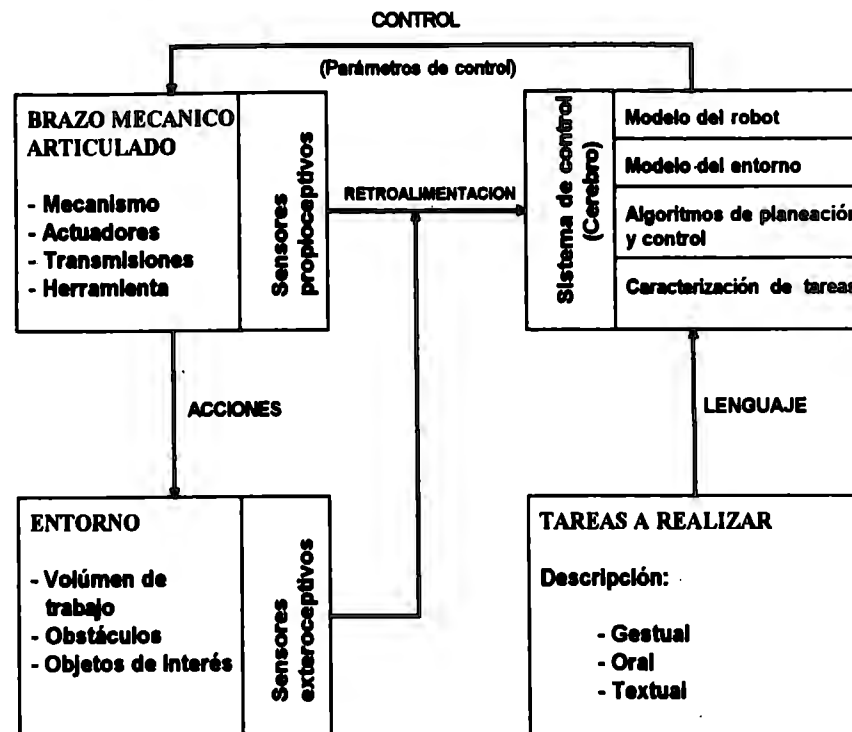


Figura 1.4 Modelo general de un robot.

La mayoría de los robots industriales actuales son menos sofisticados, la relación con el entorno es de sólo una dirección, carecen de retroalimentación exteroceptiva (captación de información externa al robot físico). En su instalación y operación se previene al máximo las perturbaciones del entorno para sortearlas con programación. Tienen sistemas de control sencillos y son relativamente económicos; pero están limitados a la realización de tareas repetitivas.

Los robots industriales están diseñados para ejecutar diversas tareas. El trabajo se efectúa haciendo que el robot desplace su cuerpo, brazo y muñeca en su espacio de trabajo. Los movimientos del robot pueden dividirse en dos categorías generales: los *movimientos de cuerpo y brazo* permiten al robot *desplazar* el efector final a una posición deseada dentro de su volumen de trabajo; los *movimientos de la muñeca* permiten *orientar* al efector final en el espacio para el desempeño de la tarea.

Los movimientos del robot se realizan por medio de articulaciones accionadas por motores. Cada tipo de articulación posee sus propias formas características para las superficies pareadas, y cada una permite un tipo de movimiento relativo entre elementos contiguos. En el diseño de robots generalmente se utilizan sólo dos tipos de articulaciones: *prismáticas* (lineales) y *de revoluta* (giratorias).

Los elementos de conexión entre articulaciones se denominan eslabones y corresponden al cuerpo físico del robot. Desde el punto de vista de la Teoría de Máquinas y Mecanismos (TMM), se llama Grados de Libertad (*gdl*) de un mecanismo, al número de parámetros necesarios para determinar su posición.

En robótica, el número de grados de libertad depende de la tarea que el robot debe realizar. No todas las tareas requieren de 6 gdl., por ejemplo:

- Posicionamiento de un punto en el espacio: 3 gdl (soldadura por puntos ).
- Posicionamiento de una recta en el espacio: 5 gdl (taladrado, soldadura con arco, etc.).

Los movimientos de una herramienta o dispositivo de aprehensión no se incluyen en los *gdl* del robot.

## 1.2 EL PROBLEMA FUNDAMENTAL DE LA ROBÓTICA

El fin último de un robot es la ejecución de una tarea en forma rápida y eficiente. Alcanzar este fin es precisamente lo que se denomina el "*problema fundamental de la robótica*". Se puede representar la complejidad del problema con la Figura 1.5 .

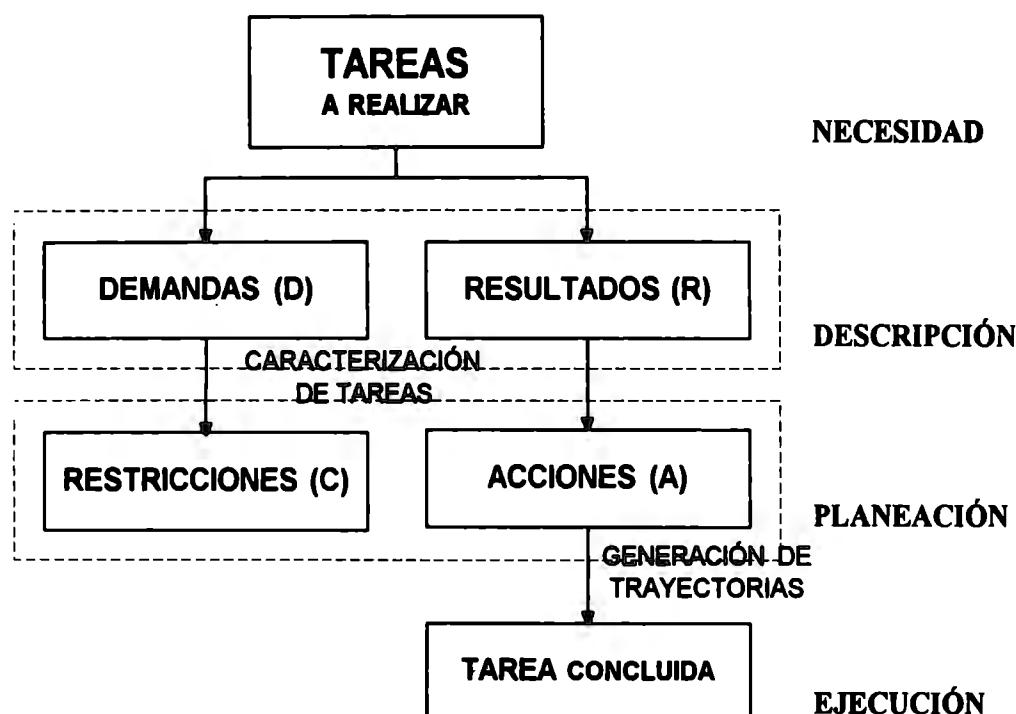


Figura 1.5 Descripción del problema fundamental de la robótica.

Las tareas a realizar por un robot plantean la necesidad de obtener resultados específicos con procesos específicos. Se necesita hacer una descripción de los resultados (R) que se deben obtener y las demandas y requerimientos (D) que se deben satisfacer para completar su trabajo satisfactoriamente. El conjunto de demandas se plantean a lo largo de la ruta espacial por la cual debe desplazar su órgano terminal, y en los lugares donde debe efectuar alguna operación (*puntos de trabajo*). La siguiente etapa es traducir las descripciones cualitativas en parámetros cuantitativos que representen las restricciones (R) para el movimiento del robot.

La información sobre las restricciones para la tarea, junto con la del modelo matemático del robot se utiliza para generar un conjunto de *parámetros de control* que describan en todo momento el estado de su *órgano terminal* (posición, orientación, velocidad, aceleración, fuerza aplicada, etc.), durante el tiempo en que se ejecuta la tarea.

Los parámetros de control del *órgano terminal* se introducen al *modelo cinemático inverso* para obtener un nuevo conjunto de parámetros (*parámetros de actuador*), que se utilizan para accionar a los motores que mueven al mecanismo y lo llevan a realizar su tarea.

De manera general, se puede describir el problema de la robótica en tres etapas, ver Figura 1.6. Tales etapas son:

1. Caracterización de tareas (task planning) Necesidad --> Descripción
2. Planeación de la trayectoria (path planning) Planeación
3. Seguimiento de la trayectoria (path tracking) Ejecución

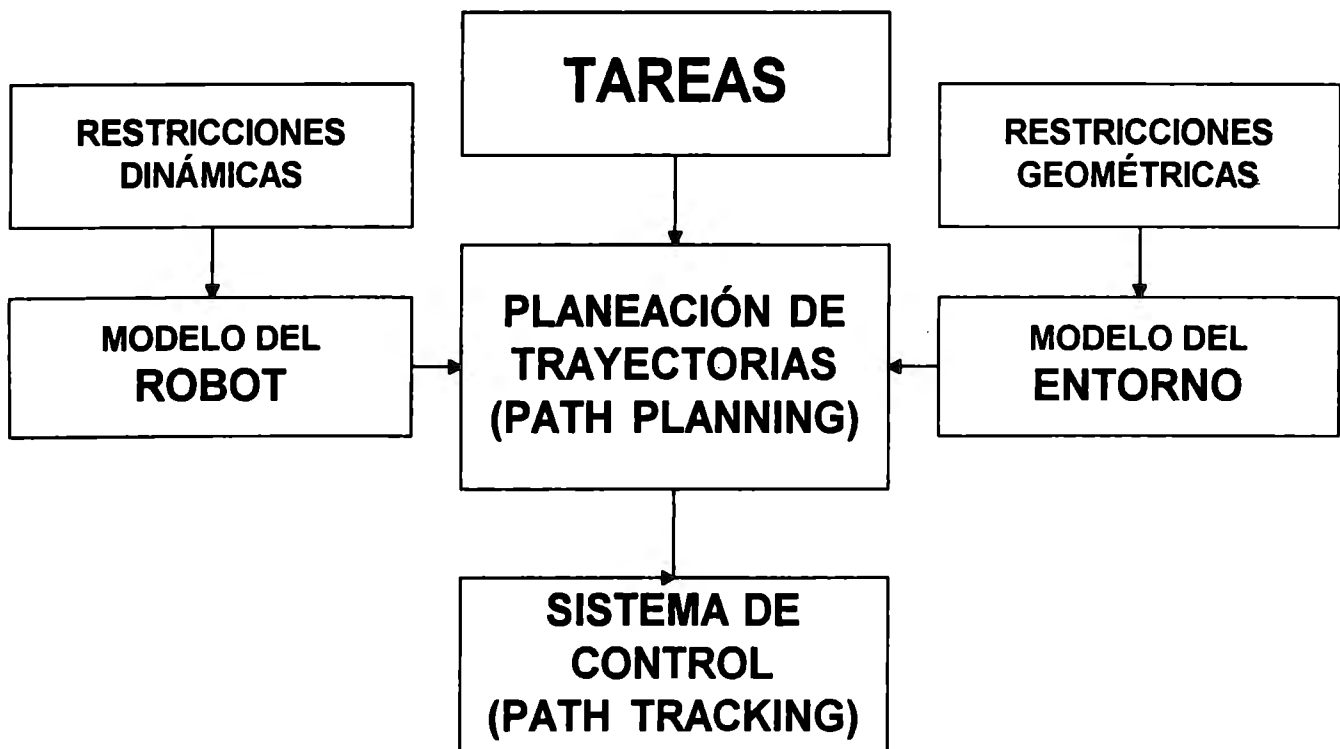


Figura 1.6 Etapas del problema fundamental de la robótica.

## 1.3 ANTECEDENTES DEL ESTUDIO DE TRAYECTORIAS

---

La *planeación de trayectorias* se refiere a la obtención de la ecuación vectorial que representa la ruta espacial a seguir por el órgano terminal del robot, en conjunto con los parámetros que gobiernan el movimiento y la dinámica del manipulador a lo largo de esa ruta. Una trayectoria se puede definir por tres tipos parámetros:

a) Geométricos.

- Ecuación vectorial de la ruta espacial de desplazamiento del órgano terminal.
- Puntos de trabajo, de control y/o de paso.
- Repetibilidad.

b) Cinemáticos.

- Velocidad lineal y angular del órgano terminal en cada punto o intervalo de la ruta.
- Exactitud de posicionamiento del órgano terminal.

c) Dinámicos.

- Aceleración del órgano terminal a lo largo de la ruta.
- Fuerzas y torques en el órgano terminal a lo largo de la ruta.

Existen algunos métodos que dan solución al problema geométrico de la planeación de trayectorias para tareas específicas. Esos métodos generan rutas para trayectorias de los tipos siguientes:

- Punto a punto
- Rectas
- Curvas indefinidas
- Curvas definidas (control de velocidad, aceleración y fuerza)

Para generar tales rutas, se utilizan técnicas de interpolación lineal, circular, y polinomial. Entre las últimas destaca el uso de splines segundo a cuarto orden, y de splines de Bézier (*B-splines*) cúbicos. Los trabajos sobre aplicación de splines en planeación de trayectorias para robots son escasos:

- Curvas fractales en tareas de pulido con robots (Y. Misugaki, M. Sakamoto; IEEE-1991).
- B-splines cúbicos en control de trayectorias de robots (P. Andre, M.C. Haddad; IEEE- 1992).

A la fecha, no hay información sobre el tratamiento del problema con *B-splines racionales*, que son B-splines expresados en forma de cociente y en términos de algunos parámetros de control para la forma de las curvas, ver Capítulo 3.

Son pocos los métodos que consideran los problemas de restricciones dinámicas y singularidades en el espacio de trabajo durante el proceso de obtención de trayectorias complejas. Por ejemplo:

- Minimum time trajectory planner for the discrete dynamic robot model. H.H. Tan. R. B.Potts. IEEE-1988.
- Planning collision-free paths in cartesian space. Tak-Lai, Daryl Luk. Robotics Today-1992.

## 1.4 IMPORTANCIA DE LA PLANEACIÓN DE TRAYECTORIAS

Un robot industrial que realiza tareas de ensamble, manipulación o procesamiento debe mover su órgano terminal desde una posición inicial a una final en el espacio tridimensional satisfaciendo las restricciones impuestas por la estructura tecnológica, espacio de trabajo y calidad de la producción. Por lo tanto, controlar estos movimientos tiene una importancia fundamental, sobre todo si se introduce una estrategia de optimización en la ejecución de las tareas.

Los requerimientos de optimización en muchas situaciones con restricciones ha llevado a revisar los modelos de sistemas articulados para considerar la dinámica completa de los robots. Es necesario optimizar el tiempo de aplicación de fuerzas/torques por parte de los actuadores de estos sistemas, para que puedan moverse a velocidades que no excedan sus propias restricciones físicas. En estas situaciones no es realista despreciar las fuerzas inerciales y centrífugas, así como la fricción viscosa.

Es claro que la calidad en la ejecución de la tarea esta estrechamente relacionada con la calidad de la trayectoria que el robot sigue para desarrollarla. En [4] se identifican diversos criterios para describir la calidad de los esquemas de planeación de trayectorias:

1. La trayectoria debe ser eficiente, tanto en su etapa de planeación (path planning) como en la de ejecución (path tracking).
2. La trayectoria debe estar completamente definida para todos los puntos de la ruta del brazo robot.
3. La posición, velocidad y aceleración deben ser funciones suaves y continuas en el tiempo.
4. Garantía de que la posición, velocidad o aceleración no excedan los límites físicos del sistema.
5. La razón de cambio de aceleración respecto al tiempo o *jerk* también debe ser continua.

Si el sistema de planeación entrega trayectorias que respeten estos criterios, la calidad de la tarea que se ejecute estará garantizada

Antes, se mencionó que existen soluciones particulares al problema de planeación de trayectorias para tareas específicas. Cuando se implementan éstas soluciones en sistemas robóticos los dotan de una alta confiabilidad, pero reducen su flexibilidad. Por tal causa, los sistemas de robots comerciales adolecen de una falta de instrucciones para definir trayectorias de mayor complejidad que interpolaciones cúbicas o circulares inclusive, ver Figura 1.7..

	Soluciones	Tareas ejecutables
Se tiene	Particulares	Específicas
Se requiere	Generales	Diversas / complejas

Figura 1.7 Tareas versus soluciones.

Este hecho conduce a la concepción de un sistema experto de planeación de trayectorias para usarse en cualquier sistema robótico y dotarle de una verdadera multifuncionalidad en la ejecución de tareas. En la Figura 1.8 se muestra un modelo del problema de planeación de trayectorias que toma en cuenta todos los criterios antes mencionados.



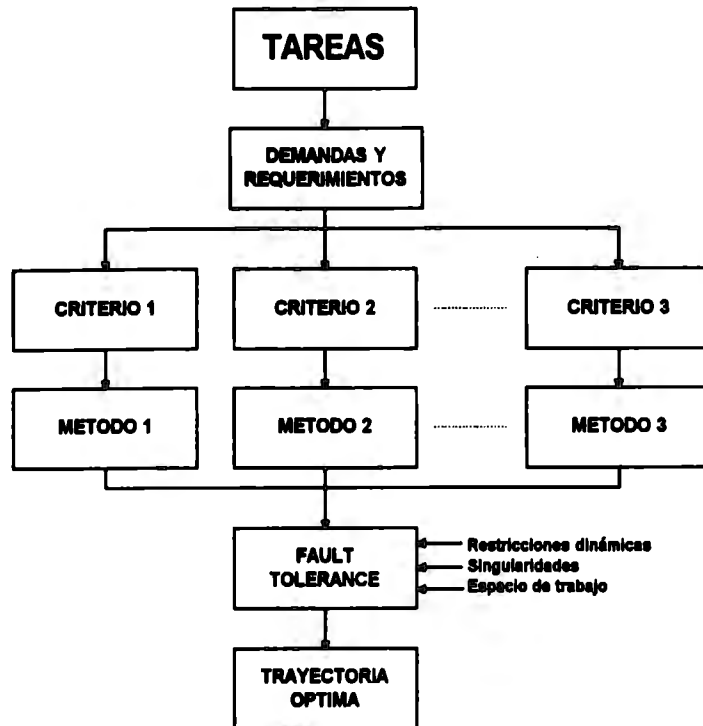


Figura 1.8. Modelo de Sistema de Planeación de Trayectorias.

Para alcanzar la meta del sistema experto, se necesita implementar en computadora métodos que den solución a las dos primeras partes del *problema fundamental de la robótica* (ver Sección 1.2), a saber: la caracterización de tareas (task planning), y la generación de trayectorias (path planning). Tales métodos deben ser eficientes, robustos y completos, para que las trayectorias arrojadas satisfagan las demandas y requerimientos de las tareas para las cuales se desarrollen.

## 1.5 PLANTEAMIENTO DEL PROBLEMA

La investigación bibliográfica realizada mostró que todavía no existe un método completo para la planeación de trayectorias complejas suaves, sujetas a restricciones dinámicas. La complejidad del problema puede resumirse en las siguientes líneas:

- El criterio para cumplir con los requerimientos de máxima suavidad exige curvas o rutas continuas con un control total de las propiedades geométricas de las mismas.
- Se necesita generar curvas complejas con la mayor fidelidad (máxima exactitud) y con el menor número de puntos de paso (ligaduras) posibles.
- El método debe trabajar en tiempo real de ser posible, tal restricción es crítica cuando se tiene un número muy grande de puntos por los que debe pasar la curva de la trayectoria; sin embargo, la naturaleza de los problema que requieren de trayectorias complejas permiten hacer la generación fuera de línea (off line).

## 1.6 OBJETIVOS DE LA TESIS

---

- a) Establecer una metodología para caracterizar tareas de robot y extraer criterios para la elección de un método de planeación de trayectorias.
- b) Establecer un método de generación de curvas suaves y complejas para trayectorias de robot.
- c) Implementar un modelo computacional del método de generación de curvas suaves y complejas.
- d) Integrar las restricciones dinámicas del robot al método de planeación de trayectorias suaves, como parte de los criterios de Fault Tolerance.
- e) Generar un modelo computacional para el método de planeación de trayectorias suaves.

# CAPÍTULO 2

---

---

## FUNDAMENTOS DE TRAYECTORIAS PARA ROBOTS INDUSTRIALES

---

---

### 2.1 CARACTERIZACIÓN DE TAREAS

---

Se dice que un proceso está "*robotizado*" cuando los seres humanos que lo realizaban han sido sustituidos por uno o varios robots. Dado este hecho, se plantean numerosos problemas técnicos a superar antes de instalar con éxito un robot industrial.

Cada tarea en la industria tiene sus requerimientos de tiempo y calidad. Las trayectorias están relacionadas con la naturaleza de la tarea a realizar. Cada tarea plantea demandas y requerimientos diferentes que debe satisfacer el robot al moverse. La *caracterización de tareas* permite reconocer tales demandas por medio de los *criterios de selección*, y traducirlas a un conjunto de parámetros que puedan utilizarse en la planeación y control del trabajo del robot. Así, requerimientos como:

- *Mínimo tiempo* para trasladar el órgano terminal de un punto de trabajo a otro.
- *Máxima suavidad* de la curva espacial que representa la ruta a seguir por el órgano terminal; esto es, ausencia de discontinuidades e indefiniciones en la curva y sus derivadas.

se traducen en parámetros tales como:

- Exactitud en el posicionamiento,
- Velocidad (media, máxima, etc.) del órgano terminal,
- Aceleración del mismo,
- Carga útil.

Los parámetros anteriores representan las características de la trayectoria óptima (para la tarea especificada), que satisface los requerimientos y demandas obtenidas en la caracterización de tareas. La obtención de estos parámetros es el principal objetivo de la planeación de trayectorias.

## 2 FUNDAMENTOS DE TRAYECTORIAS PARA ROBOTS INDUSTRIALES

Definir una técnica completa de *caracterización de tareas* esta más allá del alcance de esta tesis debido a la complejidad involucrada en tal labor. Sin embargo, uno de los objetivos es establecer las bases para el desarrollo futuro de tal metodología.

### 2.1.1 Herramientas para la caracterización de tareas

El objetivo es encontrar una herramienta o conjunto de herramientas que permitan traducir las demandas y requerimientos cualitativos de cada tarea en la información cuantitativa necesaria para resolver el *problema fundamental de la robótica*, ver Figura 2.1.

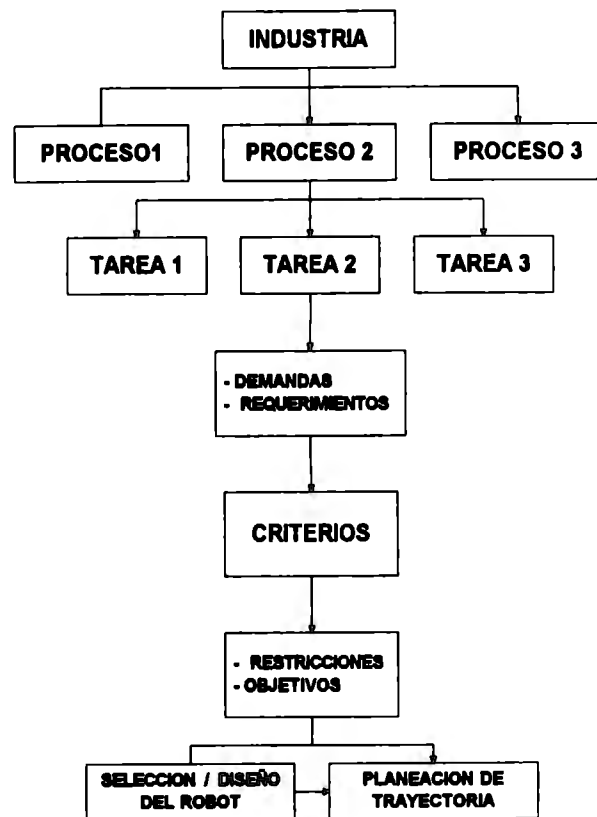


Figura 2.1 Proceso de caracterización de tareas.

Se pueden identificar tres necesidades básicas involucradas en el proceso de caracterizar tareas para robots, y estas tienen que ver con la transición entre etapas. Por lo tanto, el problema ahora se refiere a encontrar una herramienta para cada una de las transiciones, ver Figura 2.2.

Etapa inicial	Etapa final	Herramienta(s)
Aplicaciones	Tareas	Tablas aplicaciones de robots industriales
Tareas	Demandas/Requerimientos	Análisis de tiempos y movimientos
Demandas/Requerimientos	Objetivos/Restricciones	Análisis de micromovimientos Redes de Petri Funciones de calidad (QFD)

Figura 2.2 Etapas y herramientas

## 2 FUNDAMENTOS DE TRAYECTORIAS PARA ROBOTS INDUSTRIALES

Es claro que la etapa médular del problema es la formulación de los *criterios* de transformación. Afortunadamente existe una herramienta del control total de calidad (Total Quality Control o TQC) que permite hacer un correlación entre etapas y rubros de información: Las Funciones de calidad (Quality Functions Deployment o QFD).

Las QFD son las herramientas que ayudan a traducir los requerimientos de los consumidores en requerimientos básicos apropiados para cada una de las etapas de desarrollo del producto, Hartley [5]. Las QFD utilizan la técnica de enlace de matrices para asegurar que la voz del consumidor sera apropiadamente capturada y usada para definir, diseñar y manufacturar un producto.

Una primera matriz se construye relacionando los "requerimientos del consumidor" con los "requerimientos de diseño". Una vez terminada, el eje de los "requerimientos de diseño" es enlazado a otro eje etiquetado como "diseño de partes". Los requerimientos de diseño son entonces usados para diseñar apropiadamente cada una de las partes del producto final.

De esta manera, las matrices se van desplegando a través de todas las funciones de la empresa como fichas de dominó (ver Figura 2.3), hasta cerrar el ciclo con una matriz cuyo eje horizontal sea etiquetado como "requerimientos del consumidor". Por supuesto la información al final del "Dominó" de matrices QFD debe ser la misma que con la que se inicio el proceso.

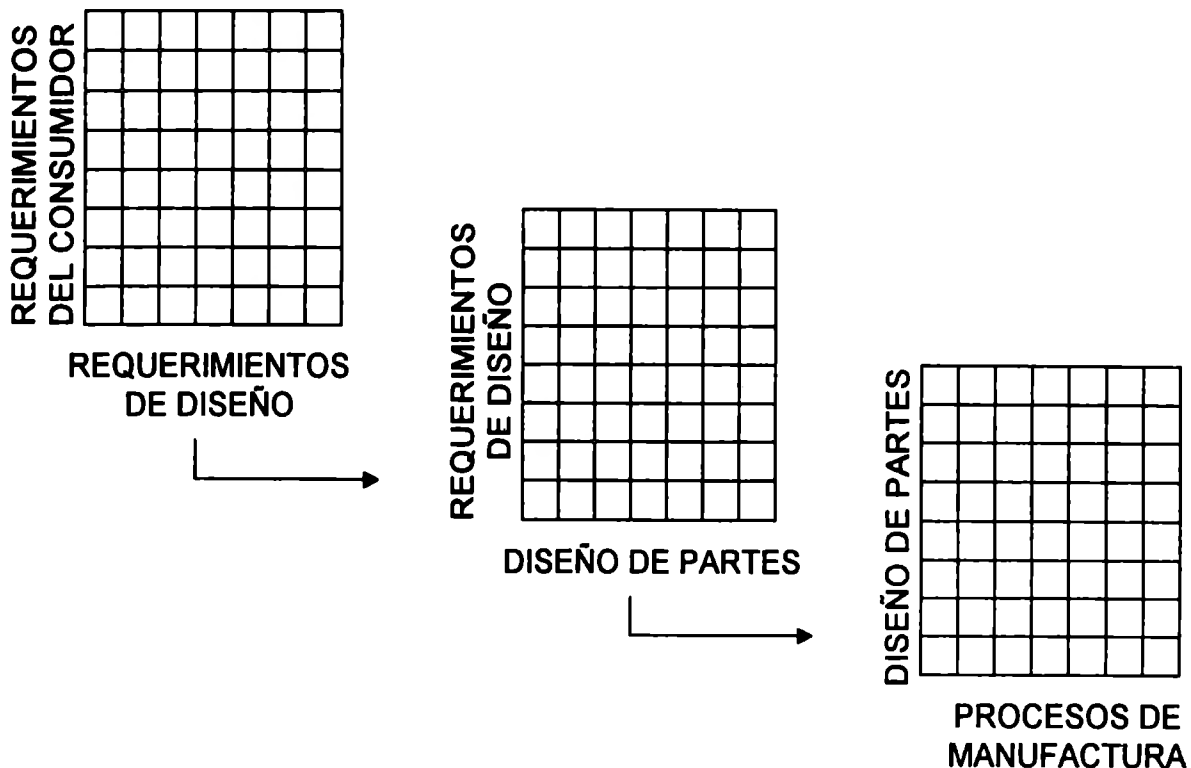


Figura 2.3 "Dominó" de matrices QFD.

2.1.2 Metodología

La caracterización de tareas para robots puede dividirse en dos partes de la manera siguiente:

1. Desarrollo de herramientas.
  - a) Generación de los *perfiles de desempeño* para los modelos comerciales de robots (y para los nuevos diseños), utilizando las técnicas de estudio del *performance*.
  - b) Formulación de las *redes de Petri* para correlacionar la información los estudios de tiempos y movimientos de las aplicaciones de robots industriales.
  - c) Definición de las matrices *QFD de criterios* con la información de las redes de Petri.
  
2. Caracterización de tareas específicas.
  - a) Identificar el tipo de industria y la tarea a realizar por un robot (sector industrial y aplicación), ver Figura 2.4.
  - b) Seleccionar los *perfiles de desempeño* más adecuados para establecer las demandas y requerimientos que la tarea plantea. Una primera aproximación de los perfiles se muestra en las Figuras 2.5 y 2.6.
  - c) Con la información anterior se recurre a las matrices *QFD de criterios* para obtener la información necesaria para la selección/diseño del robot más adecuado, y para la generación de la trayectoria para la tarea planteada; específicamente las restricciones y los objetivos, ver Figura 2.7. Es preciso señalar que los objetivos se refieren a expresiones de acontecimientos cuantificables que se necesita sucedan, y que para ser efectivos, estos deben ser precisos, cuantificables, cronometrables, aceptables y alcanzables.

TIPO DE APLICACIÓN →  SECTOR INDUSTRIAL ↓	Soldadura	Trat. superficiales	Maquinado	Ensamble	Alimentación	Transporte	Paletización	Embalaje	Inspección	Pintura
Agroalimentario		X			X	X	X	X		
Material agrícola	X	X	X							
Químico y plásticos				X	X	X	X	X		
Textil y moda		X					X	X		
Madera y muebles		X	X	X	X	X	X			X
Mat. de construcción y vidrio		X			X	X	X			
Obras públicas	X	X	X		X					
Mecánica general	X	X	X		X	X			X	
Mantenimiento					X	X	X	X		
Metalúrgico	X		X	X	X	X	X			
Automotriz	X	X	X	X	X				X	X
Construcción eléctrica		X	X	X	X	X		X	X	
Electrodomésticos		X	X	X					X	X
Electrónica				X	X				X	

Figura 2.4 Principales aplicaciones de los robots por sectores industriales, tomada de Ferráte[3].

## 2 FUNDAMENTOS DE TRAYECTORIAS PARA ROBOTS INDUSTRIALES

PROBLEMAS TÉCNICOS PLANTEADOS DEBIDOS A		CURTIDOS	CAUCHO	PLÁSTICO	MAT. P/CONSTRUC. Y VIDRIO	TEXTIL Y MODA	MADERAMUEBLES	METALÚRGIA	SOLDADURA	ENSAMBLE	SECTORES INDUSTRIALES  TEMAS EN LOS QUE DEBEN EFECTUARSE TRABAJOS DE INVESTIGACIÓN Y DESARROLLO
MAQUINARIA NO ADECUADA	Máquina-herramienta	●	●	●	●	○		○	○	●	Cinemática y control del RI, efector final, control de máquina-herramienta
	Sistema de transporte	●	●		●		○	●	○	○	Sistemas de transporte, sensores, dispositivos específicos
	Almacén y plataformas	●		○	●			●		○	
	Tecnología anticuada		○		●	○	○	●			Tecnologías nuevas o más avanzadas
ELEMENTOS A MANEJAR	Gran variedad	●	●	●	●	○		●		●	Pinzas, dispositivos específicos y sensores
	Frágiles o deformables	●	●	●	●		○	○		○	
	Carga elevada		○				○	○			Cinemática, control, elementos motores
	Grandes dimensiones		○	○		○	○			●	Cinemática, pinza
	Perfil variable	●	●	○	○	●	●	●	●	●	●
EL PROPIO PROCESO	Tiempo de ciclo corto	●	●	○	●	●		●		●	Cinemática y elementos motores para mejorar velocidad y aceleración
	Dificultad de inspección	●		●	●	○	●	●		●	Sensores, control, autodetección de averías
	Serie pequeñas	●			●	●		●		●	Pinza, programación, dispositivos específicos
	Variable de ciclo a ciclo							●	●	●	Sensores, control, software
CONDICIONES DE TRABAJO	Espacio limitado				●	●		●		●	Cinemática, tipo de instalación
	Entorno perjudicial para el RI			●	○			●			Protección del robot y elemento terminal
	Grandes recorridos		○		○	○	○	○			Cinemática
	Movimientos complejos	○	○		○	○	○	●	●	●	Cinemática y control

Figura 2.5 Problemas técnicos en las aplicaciones de robots, tomada de PROMAHI[1].

Nomenclatura: RI Robot Industrial.

● Problemas que afectan a todo un sector en general.

○ Casos particulares en algunas empresas del sector.

### 2.2 EL PROBLEMA DE PLANEACIÓN DE RUTAS ÓPTIMAS

La *planeación de trayectorias* convierte una descripción de un movimiento deseado a una *trayectoria de robot*. Una trayectoria es definida como una secuencia temporal de configuraciones intermedias entre una posición inicial y una final, junto con las velocidades y aceleraciones intermedias correspondientes Cela-Hamam[6].

El movimiento de un brazo robot industrial con muchas articulaciones es impulsado por actuadores individuales en cada una de ellas. Es común especificar el movimiento deseado del efector final como una curva en el espacio tridimensional, llamada la *ruta (path)* de la trayectoria, ver Figura 2.8. Si se requiere que el robot recorra la ruta prescrita, el sistema de control debe seguir las trayectorias de las articulaciones individuales para efectuar con exactitud el movimiento de un punto a otro a lo largo de la ruta.

## 2 FUNDAMENTOS DE TRAYECTORIAS PARA ROBOTS INDUSTRIALES

Aplicación	Característica técnica típica necesitada				
	Anatomía	No. de gdl	Sistema de impulsión	Sistema de control	Programación
Transferencia de material		3 a 5	Neumático o Hidráulico	Secuencia limitada o punto a punto	Manual Gestual
Carga de máquinas	Polar Cilíndrico Brazo articulado	4 a 5	Eléctrico o Hidráulico	Secuencia limitada o punto a punto	Textual Gestual
Soldadura por puntos	Polar Brazo articulado	5 a 6	Hidráulico o Eléctrico	Punto a punto	Gestual
Soldadura por arco	Polar Cartesiana Brazo articulado	5 a 6	Hidráulico o Eléctrico	Trayectoria continua	Manual Gestual
Recubrimiento al spray	Brazo articulado	≥ 6	Hidráulico	Tray. continua	Gestual
Ensamble	Brazo articulado Cartesiana SCARA	3 a 6	Eléctrico	Punto a punto ó Trayectoria continua	Textual Gestual

Figura 2.6 Características técnicas de un robot en función de su aplicación, tomada de PROMAHI[1].

Clasificación según aplicaciones	Exactitud			Velocidad									Secuencia del programa			
				Escala			Valores			Trayectoria						
Tipo y denominación del proceso	Baja $\epsilon \geq 1$ mm	Media 0.1-1 mm	Alta $\epsilon \leq 0.1$ mm	Una	Varias	Continua	Baja $\leq 0.1$ m/s	Media 0.1-1 m/s	Alta $\geq 1$ m/s	Punto a punto	Recta	Curva definida	Curva indefinida	Continua	Discontinua	Condiccionada
Manipulación																
Moldeo por inyección	X					X		X					X			X
Forja	X				X			X					X			X
Troquelado	X				X			X	X						X	
Maquinado		X		X	X			X		X					X	
Paletizado	X				X			X		X					X	
Proceso																
Moldeado		X				X		X				X		X		
Desbarbado		X				X		X				X		X		
Pulido		X			X			X				X	X	X		
Granallado	X			X				X				X	X	X		
Limpiar fundición	X			X				X				X	X			X
Limpiar a vapor	X			X				X				X	X			X
Montaje			X		X			X	X			X	X		X	
Cosido			X			X		X				X	X		X	
Soldadura por puntos	X				X			X		X		X	X		X	
Soldadura por arco	X					X	X	X				X	X		X	
Soldadura blanda		X						X				X	X		X	
Pintura	X					X		X				X	X		X	
Temple	X				X		X	X		X		X	X		X	X
Medida																
Palpado			X			X		X	X			X		X	X	
Radiación (X, luz, etc.)	X			X	X		X	X		X		X		X	X	

Figura 2.7 Especificaciones de los robots industriales en función de las aplicaciones, tomada de PROMAHI[1].



## 2 FUNDAMENTOS DE TRAYECTORIAS PARA ROBOTS INDUSTRIALES

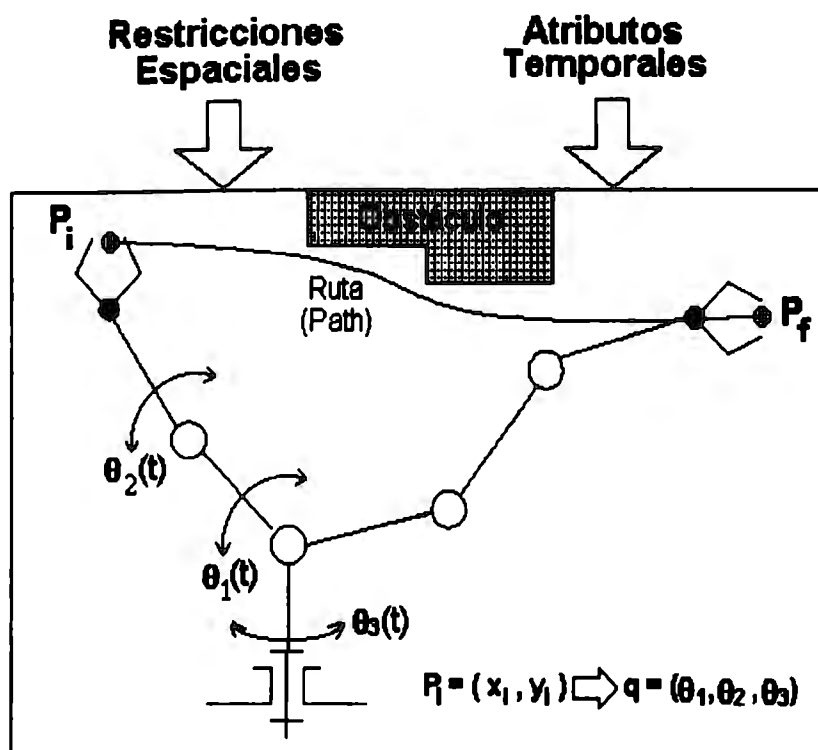


Figura 2.8 El problema de la generación de trayectorias.

La salida de un *sistema generador de trayectorias* es una secuencia de configuraciones del brazo que forman la entrada al *sistema de control* (feedback control system) del robot. Dado que las trayectorias se generan en computadoras digitales, los puntos son calculados a una determinada tasa o frecuencia. Típicamente la *servo-tasa de muestreo* (path update rate) debe estar entre 20 y 200 Hz para un *seguimiento* (tracking) suave y predecible.

El problema de planeación de rutas óptimas (Optimal Path Planning Problem - OP PP) para robots manipuladores es enunciado por Mayorga-Wong[7] como:

*"Dadas una posición inicial, una final y una trayectoria factible especificada en el espacio operacional, encontrar la ruta y la historia temporal correspondiente en el espacio de configuración; y las funciones de control de mínimo tiempo o energía para satisfacer los requerimientos del robot"*

El OP PP y el problema de la cinemática inversa asociado puede estudiarse desde dos enfoques: métodos *locales* y métodos *globales*. Los métodos globales tienen dos ventajas principales:

- En la mayoría de los casos, las restricciones impuestas para algunas variables son suficientes para obtener trayectorias libres de singularidades.
- Puede considerarse la dinámica del robot.

Sin embargo, estos métodos están limitados a la programación fuera de línea y no son fáciles de implementar en ambientes con retroalimentación exteroceptiva.

## 2 FUNDAMENTOS DE TRAYECTORIAS PARA ROBOTS INDUSTRIALES

Asumiendo que la ruta a seguir por el brazo es especificada por un número de puntos a lo largo de la trayectoria deseada, y que deben tomarse en cuenta las restricciones físicas del mismo; el cálculo del movimiento propio de un robot manipulador es resuelto en dos etapas:

### 1. Planeación de la trayectoria/ruta.

Involucra básicamente los aspectos geométricos y cinemáticos del movimiento. Esta etapa incluye:

- a) Interface con el usuario      o      ¿Cómo describir una trayectoria espacial?.
- b) Representación interna      o      ¿Cómo la manejará la computadora?.
- c) Generación de la trayectoria.

Esta parte se divide en los subproblemas de la tabla siguiente:

No.	PARTE	PARÁMETROS		
		G	C	D
I	Planeación de la ruta (Path planning)	✓		
II	Historia temporal (posición, velocidad, aceleración, jerk)	✓	✓	✓
III	Fault tolerance		✓	✓
G = GEOMÉTRICOS		C = CINEMÁTICOS		D = DINÁMICOS

### 2. Seguimiento de la trayectoria/ruta.

Etapas que involucra la dinámica del robot

La descripción detallada del proceso se realizará en el Capítulo 4.

## 2.3 EL PROBLEMA DE LA CINEMÁTICA INVERSA

La robótica de manipuladores es usualmente servocontrolada en el espacio de de articulación, mientras que las tareas del robot son descritas en el espacio cartesiano, ver Figura 2.7.

<b>Variables de manipulación: Descripción de la tarea → Coordenadas de trabajo</b>			
$x_j = x_j(t) ; j = 1, 2, 3, \dots, m$			
<b>Variables de articulación: Descripción del movimiento → Coordenadas de control</b>			
$\theta_i = \theta_i(t) ; i = 1, 2, 3, \dots, n$			
<b>Espacios euclidianos:</b>	$R^m$	m-dimensional	→ No. de dimensiones espaciales
	$R^n$	n-dimensional	→ No. de articulaciones del robot
Si $m < n$ → Manipuladores redundantes			
Si $m = n$ → Manipuladores no redundantes			

Figura 2.9 Espacio cartesiano y espacio de articulación.

## 2 FUNDAMENTOS DE TRAYECTORIAS PARA ROBOTS INDUSTRIALES

La *Cinemática Directa* consiste en determinar la posición y la orientación de la herramienta respecto de un sistema de referencia conveniente (generalmente la estación de trabajo), dando las coordenadas (ángulos) de articulación del manipulador, Craig[9]:

$$\mathbf{X}(t) = \mathbf{f}(\boldsymbol{\theta}(t))$$

donde

$$\begin{aligned}\mathbf{X} &\equiv \mathbf{X}(t) = [x_1, x_2, \dots, x_m]^T \in \mathbb{R}^m \\ \boldsymbol{\theta} &\equiv \boldsymbol{\theta}(t) = [\theta_1, \theta_2, \dots, \theta_n]^T \in \mathbb{R}^n\end{aligned}$$

La *Cinemática inversa* consiste en determinar el conjunto de ángulos de las articulaciones del robot, dando la posición y orientación de la herramienta:

$$\boldsymbol{\theta}(t) = \mathbf{f}(\mathbf{X}(t))$$

Para resolver el problema se recurre a una formulación alterna de la ecuación anterior, haciendo uso del *Jacobiano* del manipulador:

$$\dot{\mathbf{x}}(t) = \mathbf{J}(\boldsymbol{\theta}(t)) \dot{\boldsymbol{\theta}}(t)$$

$$\mathbf{J}(\boldsymbol{\theta}) \equiv \mathbf{J}(\boldsymbol{\theta}(t)) = \frac{\partial \mathbf{f}(\boldsymbol{\theta}(t))}{\partial \boldsymbol{\theta}(t)}$$

En los manipuladores redundantes donde  $m < n$ ,  $\mathbf{J}(\boldsymbol{\theta}(t))$  no es una matriz cuadrada, y por lo tanto pueden existir un número infinito de velocidades de articulación  $\dot{\boldsymbol{\theta}}(t)$  que satisfagan la última ecuación para cada vector velocidad  $\dot{\mathbf{x}}(t)$  en cualquier  $t \in [t_0, t_f]$ . En general, la solución proporcionada por la última ecuación, no garantiza la generación de rutas de articulación libres de singularidades en el intervalo de tiempo de la tarea. La cinemática inversa para manipuladores no redundantes es muy complicada.

Para que un robot alcance un punto con una orientación deseada, la solución de la cinemática inversa es esencial, Wu[10]. La cinemática inversa es un problema complicado debido a tres razones principales:

1. Involucra la resolución de un sistema de  $n$  ecuaciones trascendentales.
2. La existencia de soluciones involucra la forma y dimensión del espacio de trabajo.
3. La cinemática inversa para un punto particular dentro del espacio de trabajo puede generar más de una solución.

En la Sección 5.6 se trata con algunos métodos de solución del problema de la cinemática inversa.

### 2.4 RESTRICCIONES PARA TRAYECTORIAS

---

De acuerdo con Çela-Hamam[6], las restricciones que afectan el movimiento de un robot se clasifican en dos tipos:

1. Restricciones inherentes.
  - a) Restricciones de torque o actuador.
  - b) Restricciones de posición.
    - Singularidades.
    - Alcance máximo.
    - Repetibilidad.
  - c) Restricciones de velocidad.
  - d) Restricciones de aceleración.
  - e) Restricciones de jerk.
2. Restricciones del espacio de trabajo.
  - a) Volúmen de trabajo efectivo.
  - b) Obstáculos.

Las *singularidades* son configuraciones de articulaciones particulares en donde las velocidades angulares de articulaciones vecinas son muy elevadas, para producir pequeños cambios en la posición del efector final. La configuración del espacio de trabajo se construye con las posiciones relativas entre el robot y los obstáculos como función del tiempo. Si las funciones de transformación (mapeo) para los obstáculos y singularidades son predefinidas, la configuración del espacio de trabajo depende sólo de las variables de control del robot.

La formulación del OPPP puede darse para el caso donde sólo se consideren las *restricciones inherentes*, y también para el caso donde se adicionan las *restricciones del espacio de trabajo*. Esta distinción hace posible dividir al OPPP en dos casos:

- a) OPPP sin obstáculos,
- b) OPPP con obstáculos (trayectorias libres de colisiones).

### 2.5 PLANEACIÓN DE RUTAS USANDO EL MODELO DINÁMICO

---

Usualmente cuando se calcula la *historia temporal* al estar generando la trayectoria, se usan valores máximos o promedios para la aceleración de cada articulación (grado de libertad). Para evitar sobrepasar las capacidades físicas del sistema, se debe tener cuidado cuando se eligen aceleraciones máximas, por lo que tal selección debe ser muy conservativa. Hacer esto significa simplificar en exceso el comportamiento real de cualquier robot; por lo cual es natural que con esta metodología nunca se aproveche al máximo la capacidad de movimiento de los manipuladores.

Craig [9] plantea que los métodos actuales deben asegurar que la cantidad de aceleración que el manipulador es capaz de alcanzar en cualquier instante sea una función de la dinámica del brazo y de los límites de sus actuadores (curvas de comportamiento velocidad - torque).

**BIBLIOTECA**



### 2.6 PLANEACIÓN DE RUTAS LIBRES DE COLISIÓN

---

En la Sección 2.4 se planteó que la planeación de trayectorias libres de colisión es el caso general del OPPP. La mayor parte de las tareas actuales realizadas por robots se caracterizan por la dificultad con que se generan las trayectorias sorteando los obstáculos fijos dentro del volumen de trabajo. La dificultad estriba en que el sistema de control no involucra modelo alguno del entorno que interactúe con el modelo del robot mismo. Esto provoca una carencia total de procedimientos e instrucciones adecuados para tratar con obstáculos en las trayectorias definidas. el problema se acentúa cuando los obstáculos se están moviendo (bandas transportadoras, vehículos de transferencia, etc.). Otro robot puede estar trabajando en la misma área y por lo tanto cada brazo debe ser considerado como un obstáculo móvil por el otro.

El sistema debe tener modelos del manipulador, del área de trabajo y de todos los obstáculos potenciales en tal área. Como se mencionó antes, si esto se cumple, entonces la configuración del espacio de trabajo depende sólo de las variables de control del robot.

Los sistemas de planeación de trayectorias libres de obstáculos no están aún disponibles comercialmente. De acuerdo con Craig[9], las investigaciones para dar solución al problema están concentradas en dos áreas principales:

- a) Formación de una conexión de gráficos o "células" para representar el espacio libre del robot, y búsqueda de un "gráfico de conectividad" para una secuencia de células adyacentes, o "canal". La primera célula contiene la configuración inicial y la última la configuración final. El resultado es una ruta libre de colisión. Un método en extremo interesante es el presentado en Maciejewsky-Fox[11].
- b) Creación de campos potenciales alrededor de los obstáculos para evitarlos, mientras se traza la ruta hacia un polo atractivo artificial en el punto final.

### 2.7 RESUMEN DEL PROBLEMA DE PLANEACIÓN DE TRAYECTORIAS

---

Para concluir el planteamiento del problema de planeación/generación de trayectorias, se presenta un resumen gráfico en la Figura 2.10. En el cuadro se pueden identificar las estrategias que se siguen para generar trayectorias ya sea en espacio de articulación o en espacio cartesiano. La información se tomó de PROMAHI[1], Groover[2], Craig[9], Fu[12], Angulo[13], Brady[14] y Canny[15].

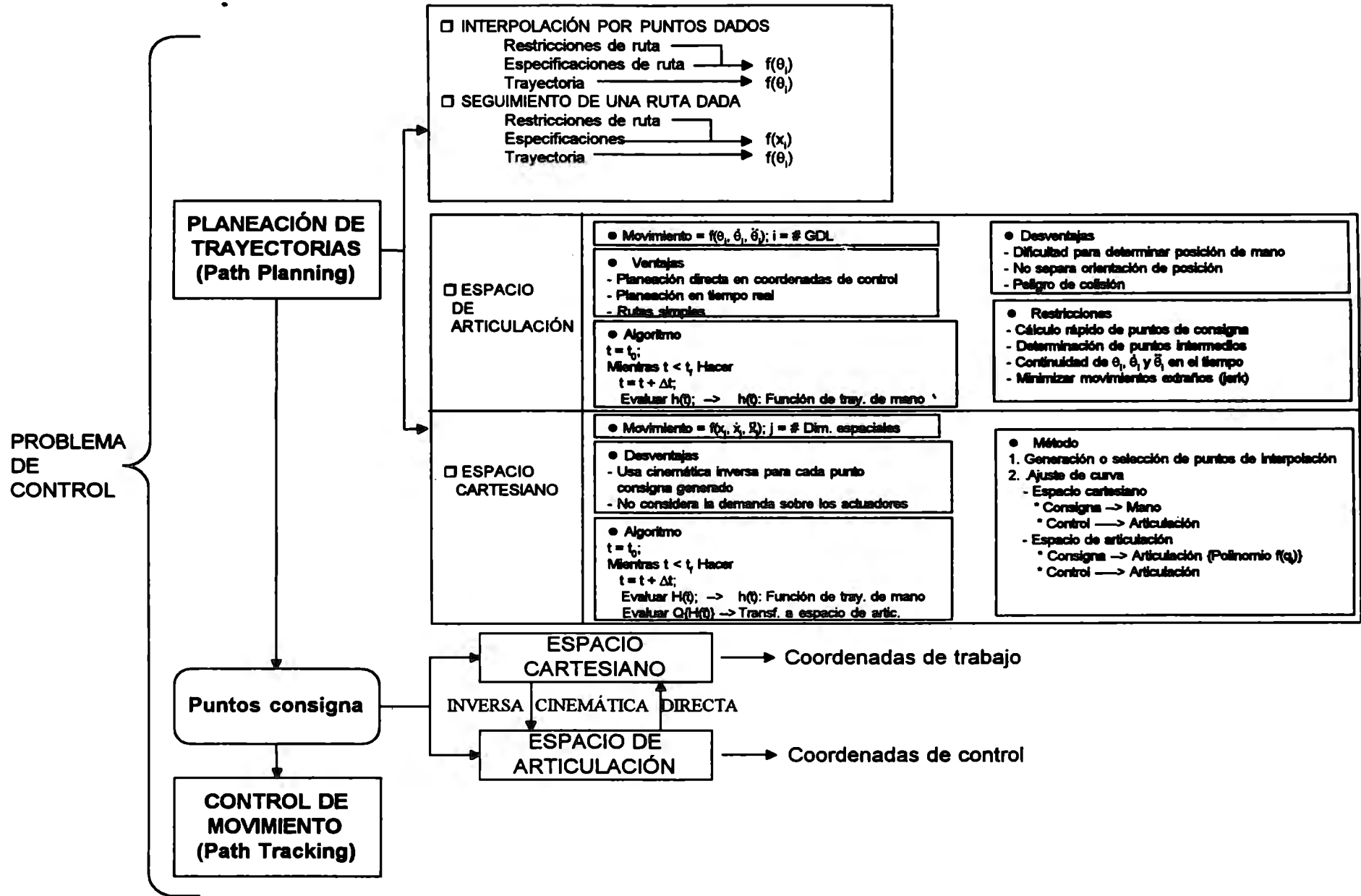


Figura 2.10 Resumen del problema de planeación de trayectorias.

# CAPÍTULO 3

---

## ANÁLISIS GEOMÉTRICO Y MATEMÁTICO DE TRAYECTORIAS

---

### 3.1 CARACTERÍSTICAS DE LAS CURVAS PLANAS Y ESPACIALES

---

Una función valuada en vector  $t \rightarrow f(t)$  es una regla que asocia un vector  $f(t)$  a cada número real  $t$  en un intervalo definido; y dará a cada punto de un lugar geométrico (curva o superficie) en un espacio  $R^n$ , un vector de posición.

Hill[16] y Vásquez[17] plantean que existen dos configuraciones de curvas usadas ampliamente en los Sistemas de Asistencia Computarizados (SAC's):

#### 1. Curvas Analíticas.

Son curvas descritas por una simple función. Las curvas analíticas más utilizadas son las de segundo grado (cónicas). Tales curvas son conjuntos cuyas coordenadas cartesianas satisfacen una ecuación cuadrática. Las cúbicas (curvas analíticas de tercer grado) también encuentran una aplicación importante.

#### 2. Curvas Paramétricas.

Una *curva paramétrica* es aquella cuyas ecuaciones de definición están dadas en términos de una sola variable simple e independiente denominada *variable paramétrica* o *parámetro*.

#### 3.1.1 Construcción de curvas Paramétricas

Cada punto en una curva tridimensional se puede expresar mediante sus componentes como:

$$\begin{array}{ll} \mathbf{x} = \mathbf{x}(s) & \\ \mathbf{y} = \mathbf{y}(s) & \text{Ecuaciones paramétricas} \\ \mathbf{z} = \mathbf{z}(s) & \\ \mathbf{x}, \mathbf{y}, \mathbf{z}: & \text{Variables dependientes} \\ \mathbf{s}: & \text{Variable paramétrica} \end{array}$$

### 3 ANÁLISIS GEOMÉTRICO Y MATEMÁTICO DE TRAYECTORIAS

Se puede enunciar algunas reglas para la construcción o elección de las funciones matemáticas de una curva paramétrica:

- i) Definir cada componente por una sola variable paramétrica.
- ii) Cada punto en la curva corresponde a un único valor de la variable paramétrica.
- iii) Cada valor de la variable paramétrica debe corresponder a un punto único en la curva.

#### 3.1.1.1 Curvas paramétricas planas

Para definir curvas paramétricas planas, generalmente se usan como funciones paramétricas polinomios de segundo grado.

$$x(s) = a_x s^2 + b_x s + c_x$$

$$y(s) = a_y s^2 + b_y s + c_y$$

$$z(s) = a_z s^2 + b_z s + c_z$$

---

$$\vec{P}(s) = \vec{a} s^2 + \vec{b} s + \vec{c} \quad , s \in [0,1]$$

La anterior expresión es la FORMA ALGEBRAICA de la ecuación de una curva plana en el espacio tridimensional. Conociendo las coordenadas de tres puntos se obtienen los valores de  $s$  para cada uno de ellos y entonces se pueden determinar los valores de los vectores constantes  $\vec{a}$ ,  $\vec{b}$ , y  $\vec{c}$  esto nos lleva a la FORMA GEOMÉTRICA de una curva plana:

$$\vec{P}(s) = (2s^2 - 3s + 1)\vec{P}_0 + (-4s^2 + 4s)\vec{P}_1 + (2s^2 - s)\vec{P}_2 \quad , s \in [0,1]$$

#### 3.1.1.2. Curvas paramétricas espaciales

Las funciones paramétricas para curvas espaciales, generalmente, son polinomios de tercer grado.

$$\vec{P}(s) = \vec{a} s^3 + \vec{b} s^2 + \vec{c} s + \vec{d} \quad , s \in [0,1]$$

En esta FORMA ALGEBRAICA se pueden identificar 12 incógnitas: 3 coordenadas por vector y 4 vectores constantes. Son necesarios 4 puntos para resolver el sistema y llegar a la FORMA GEOMÉTRICA de las curvas espaciales.



$$\begin{aligned} \vec{P}(s) = & (-4 s^3 + 9s^2 - 5.5 s + 1) \vec{P}_1 \\ & + (13.5 s^3 - 22.5 s^2 + 9 s) \vec{P}_2 \\ & + (-13.5 s^3 + 185 s^2 - 4.5 s) \vec{P}_3 \\ & + (4.5 s^3 - 4.5 s^2 + s) \vec{P}_4 \quad , s \in [0,1] \end{aligned}$$

### 3.1.2 Curvas diferenciales parametrizadas

Uno de los requisitos fundamentales que deben cumplir las curvas para trayectorias de robots (y otras aplicaciones), es que sean continuas en todo el intervalo de definición, y además, que por lo menos sus dos primeras derivadas también lo sean. Para cumplir con tal restricción las *curvas paramétricas* deben ser *diferenciales*, es decir, que las funciones que las definen tengan derivadas de todos los órdenes de todos los puntos dentro de su intervalo.

Una *curva diferencial parametrizada regular* es aquella cuya primera derivada es diferente de cero para todo punto del intervalo.

### 3.1.3 Curvas parametrizadas por longitud de arco

Definiendo con  $f(t)$  a una curva regular parametrizada por  $t \in [a,b]$ ; la longitud de arco de la curva desde el punto  $t_0$ , es por definición:

$$S(t) = \int_{t_0}^t |f'(t)| dt$$

Así, para valores dados de  $t$  se definen valores para  $s(t)$ . Si además, el vector tangente a la curva es unitario en todo el intervalo acotado por  $t$ , es decir,  $|f'(t)| = 1$ ; entonces se puede utilizar la longitud de arco  $s$  también como parámetro alternativo a  $t$ .

### 3.1.4 Teorema fundamental de existencia y unicidad de curvas en el espacio

Sean  $k(s)$  y  $\tau(s)$  dos funciones continuas arbitrarias en el intervalo  $a \leq s \leq b$ . Entonces, existe una y sólo una curva espacial  $f(s)$  cuya curvatura es  $k(s)$ , su torsión es  $\tau(s)$  y tiene a  $s$  como parámetro natural a todo lo largo de ella.

La curvatura se define como:

$$k(s) = |f''(s)|$$

Para determinar el valor de la torsión en un punto determinado por  $s$ , se debe considerar tres vectores unitarios asociados a ese punto

$\hat{t}(s)$ : Vector tangente a la curva en el punto  $s$ .

$\hat{n}(s)$ : Vector normal a la curva en el punto  $s$ .

$\hat{b}(s)$ : Vector binormal a la curva en el punto  $s$ .

Y sus relaciones que son:

$$\begin{aligned} t'(s) &= f'(s) \quad , \quad t'(s) = K(s) n(s) \\ b'(s) &= t(s) n(s) \quad , \quad b'(s) = J(s) n(s) \end{aligned}$$

## 3.2 MÉTODOS PARA LA GENERACION DE CURVAS COMPLEJAS

---

Para describir matemáticamente formas geométricas complejas y arbitrarias se utilizan métodos basados en la técnica de aproximación polinomial. Estos métodos son de dos clases distintas:

### 1. Aproximación a una función discreta.

Asociar un polinomio de aproximación a un conjunto de puntos que representan el comportamiento de una función  $y = f(x)$ , que no se conoce en forma explícita.

### 2. Aproximación a una función continua y conocida.

Asociar un polinomio a una función  $y = f(x)$  conocida por una función más simple  $y = P_n(x)$ .

### 3.2.1 Métodos generales

Los métodos generales son aquellos que definen una sola función que toca o se aproxima a todos los  $n+1$  puntos del conjunto dado. Cuando se modifica la posición de un punto, es necesario recalcular toda la curva.

#### 3.2.1.1 Polinomio de grado $n$

La curva generada es una curva analítica o paramétrica descrita por un polinomio de grado  $n$ , que necesita  $n+1$  puntos para definirla completamente. En las Secciones 3.1.1 y 3.1.2 se trataron estos polinomios. Su principal característica es que pasan exactamente por los puntos dados.

En las secciones mencionadas se puede observar que la obtención del polinomio es resultado de la resolución de un sistema de  $n+1$  ecuaciones con  $n+1$  incógnitas, problema que se complica para más de 5 puntos dados.

#### 3.2.1.2 Splines

Para trazar manualmente una curva suave a través de un conjunto de puntos, se utiliza un dispositivo de madera, plástico o metal llamado SPLINE (curvígrafo). Los splines modernos son dispositivos deformables plásticamente, es decir, conservan la forma que se les da aún después de dejar de manipularlos. Con estos splines es posible hacer pasar a una curva por todos los puntos definidos en el papel.

Este concepto físico para el trazo de curvas complejas ha dado lugar al concepto matemático de los splines, es decir, a la teoría de aproximación polinomial para el ajuste de curvas a un conjunto predefinido de puntos. El grado del polinomio algebraico que representa el lugar geométrico depende del número de puntos que definen el polígono de definición.

Para generar rutas con requerimientos de máxima suavidad, el uso de B-splines proporciona diversas ventajas sobre las otras técnicas:

- a) Una curva de grado  $n$ -ésimo pasa por los puntos inicial y final del conjunto dado y se aproxima a los intermedios con desviación mínima.
- b) Permiten controlar las propiedades geométricas de la curva.
- c) La función que define la curva es independiente del sistema coordenado de referencia.
- d) Tiene un efecto de suavidad mayor.

En las Secciones 3.4 y posteriores se tratarán a fondo los B-splines, tanto en forma no racional como en forma racional.

#### 3.2.2 Métodos locales

Los métodos locales para generar curvas complejas, son aquellos que utilizan formas geométricas simples y bien estudiadas para aproximar subconjuntos pequeños de puntos. Posteriormente se unen todos los "pedazos" de curva para generar las formas complejas.

Estos métodos eliminan la necesidad de recalcular toda la curva cuando cambia la posición de un punto, pues la distorsión sólo se presenta en el "pedazo" donde se sitúa dicho punto.

Las desventajas se refieren a la "baja" calidad de las formas generadas, y a la incertidumbre sobre las condiciones de continuidad de la curva y de sus derivadas en los puntos de unión. En ocasiones es necesario cambiar la posición del punto para satisfacer tan importantes requisitos.

##### 3.2.2.1 Aproximación con segmentos de línea recta

Es la técnica más sencilla para el trazo de curvas complejas. Sin embargo sus desventajas son demasiadas. Primero, necesita una gran cantidad de puntos para obtener curvas con formas suaves. Segundo, aunque la función "a pedazos" parezca suave, nunca será continua (ni tampoco sus derivadas). Existen grandes problemas en los vértices de este polígono gigante.

3.2.2.2 Aproximación con arcos de círculo

Hay muchos tipos de curvas que no tienen descripción matemática, por lo que se tiene que recurrir a técnicas de ajuste para representarlas. Para generar curvas en las máquinas de control numérico actuales (NC o CNC), sólo se puede recurrir a funciones de interpolación circular. Tales máquinas no están equipadas con funciones de interpolación cúbica o de grados superiores.

Análogamente al trazo manual de curvas complejas con compás, el método desarrollado por Moreton-Parkinson[18] de la Universidad de Liverpool, se basa en el enlace de puntos mediante la unión de arcos de círculo de radios diferentes. La información necesaria para la generación de las curvas se establece en dos niveles:

1. Curva total: Conjunto de puntos *ligadura*.
2. *Paneles* individuales (sección entre dos puntos *ligadura*):
  - a) Coordenadas de inicio del arco,
  - b) Coordenadas finales del arco,
  - c) Coordenadas del centro del centro del arco o su radio,
  - d) Dirección de trazo (sentido horario o antihorario).

El uso de un arco singular sencillo sobre cada *panel* es insuficiente para dar una curva suave y continua. Pero si se usan dos arcos de círculo en cada *panel*, entonces se puede lograr tales características en casi toda la curva, ver Figura 3.1; sin embargo se pueden presentar los mismos problemas que el método anterior en los puntos de unión.

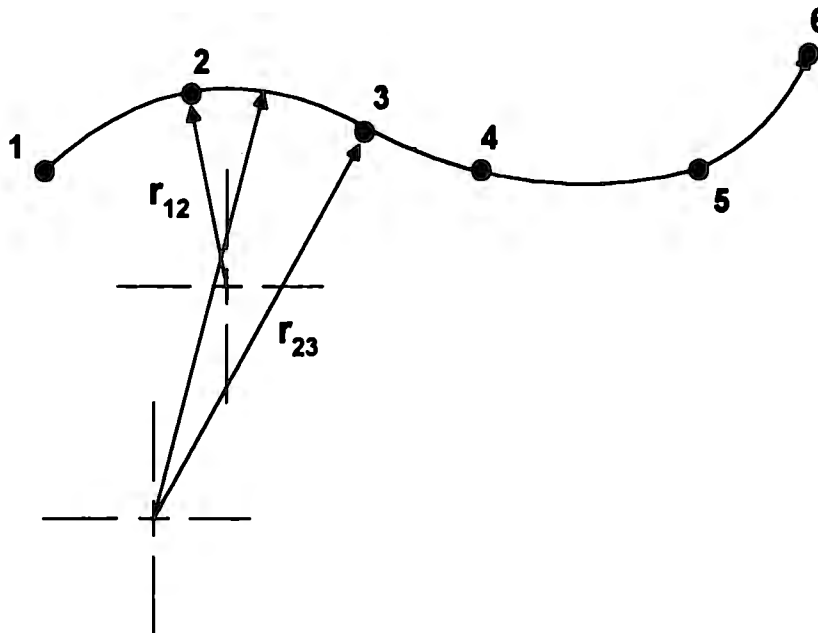


Figura 3.1 Interpolación con la técnica de biarco.

### 3.2.2.3 Aproximación con splines

A diferencia de los polinomios analíticos/paramétricos, los splines no pasan por todos los puntos del conjunto, sino que se aproximan a ellos. La generación de los splines se hace utilizando algoritmos recursivos cuya complejidad no va más allá de los medios computacionales para su implementación. El error intrínseco depende tanto del grado del spline (número de puntos del conjunto), como de la técnica utilizada.

Vásquez[17] hace un estudio de los principales métodos para trazar splines (comenzando con el *Método de Ferguson*), y comprueba como los splines generados con el *Método de Bezier-Bernstein* son los que más se aproximan al polígono formado por los puntos predefinidos.

Usando B-splines no racionales, la aproximación depende sólo de las funciones base del polinomio de Bernstein. Por ello, cuando el error intrínseco es mayor que la tolerancia preespecificada, la única manera de mejorar la forma y aumentar la exactitud es cambiar la posición de algunos puntos, ver Figura 3.2. Vásquez[17] y Farin[19] desarrollan algoritmos para generar curvas complejas usando este método.

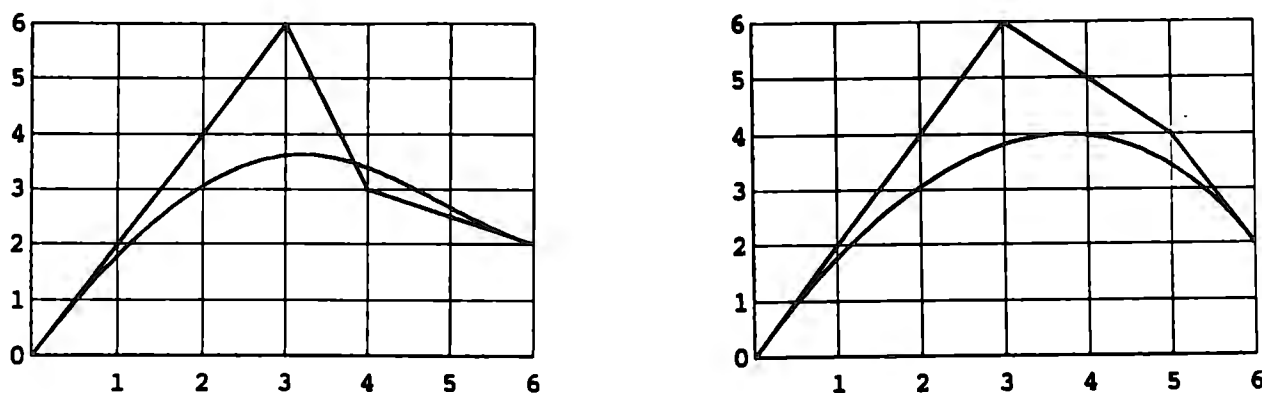


Figura 3.2 Cambiando la forma de un B-Spline no racional.

### 3.2.2.4 B-Splines racionales

Los B-Splines racionales son splines basados en el método de Bernstein-Bezier. A diferencia de los splines antes mencionados la ecuación que los define esta dada en forma de cociente, e involucra un conjunto de  $n+1$  parámetros de forma (el mismo número que la cantidad de puntos).

Es posible mejorar la forma y la exactitud del spline modificando los valores de los parámetros de forma tanto como se requiera.

La complejidad del método radica en que los parámetros son dependientes de manera no lineal, por lo que es difícil encontrar expresiones explícitas para cada uno de ellos. Entonces se recurre a técnicas eurísticas de cálculo. Vásquez[17] y Chou-Piegl[20] desarrollan algoritmos que unen B-splines racionales cúbicos para crear curvas largas y complejas en 2 o 3 dimensiones.

### 3.3 Selección del método de generación de curvas complejas

Para seleccionar una técnica de trazo de curvas complejas, se recurre a la evaluación que hace Tehrani[21] de diversos métodos, en la definición de contornos de imágenes médicas. Su objetivo es definir los contornos a partir de pocos puntos frontera, y minimizar la discontinuidad de la segunda derivada.

Los resultados obtenidos de una prueba con 100 puntos frontera, se muestran en la Figura 3.2. Los criterios de evaluación son numéricos, se basan en la comparación entre los métodos usados y una cuidadosa interpolación manual. Estos criterios se dividen en dos tipos principales:

1. Medidas de la suavidad de la curva.
  - Energía máxima
  - Energía promedio
2. Medidas de la exactitud de forma de la curva.
  - Error promedio de forma
  - Error máximo de forma
  - Area XORed: Porcentaje de error de área entre las fronteras definidas manualmente y los métodos de interpolación utilizados.

MÉTODO	Longitud de la curva	Energía Máxima	Energía Promedio	Error <sub>max</sub> de forma	Error <sub>prom</sub> de forma	AreaXOR <sub>ed</sub> [ % ]	Desventajas
Manual	1252.76	0.29	0.032	0.00	0.000	0.00	
Lineal	1134.29	1.36	0.039	2.91	0.384	24.32	- No es suave - No es exacta
(Spline) cónico	1638.62	9.72	0.154	45.51	6.631	20.55	- Resultados pobres cuando hay inflexiones
B-spline	1216.81	0.21	0.032	4.90	0.256	7.58	- No pasa por los puntos borde
Spline cúbico (primer orden)	1236.60	3.30	0.044	4.48	0.296	5.58	- Problemas con arcos muy agudos
Spline cúbico (optimizado)	1245.01	0.32	0.029	3.66	0.232	2.30	- No garantiza la continuidad de las derivadas de orden superior

Figura 3.2 Evaluación de los métodos para trazar curvas complejas.

Los resultados de la evaluación indican que los splines cúbicos optimizados son una buena alternativa para el trazo eficiente de curvas complejas. Su única pero gran desventaja es la poca confiabilidad que tienen para asegurar la continuidad de las derivadas de orden superior, condición que es crítica para la aplicación a la generación de trayectorias.

El otro método con buenas propiedades es precisamente el de B-plines. La desventaja que tienen es que no pasan por los puntos intermedios, sino que tan sólo se aproximan a ellos. Sin embargo, como se vera en las secciones siguientes, los B-splines racionales proporcionan una forma de minimizar los errores de forma de la curva tanto como se requiera.

### 3.4 El polinomio de aproximación de Bernstein

---

#### 3.4.1 Aproximación polinomial

Vásquez[17] realiza un estudio detallado sobre el polinomio de Bernstein. A continuación se incluye un resumen de su tratamiento.

Un método de aproximación polinomial puede definirse como:

$$f(x) = P_n(x) + E(x)$$

$f(x)$ : Función real conocida (analíticamente o por medio de una tabla).

$P_n(x)$ : Polinomio de aproximación.

$E(x)$ : Función de error.

Un polinomio de aproximación tiene la forma siguiente:

$$P_n(x) = A_0\beta_0(x) + A_1\beta_1(x) + \dots + A_n\beta_n(x)$$

$A_i$ : Parámetros cuyo valor debe determinarse.

$\beta_i(x)$ : Funciones base del conjunto de polinomios de grado  $n$ .

Además existen los requerimientos de continuidad para todas las derivadas del polinomio:

$$\begin{aligned} P(x_0) &= f(x_0) \\ P'(x_0) &= f'(x_0) \\ &\vdots \\ P^{(n)}(x_0) &= f^{(n)}(x_0) \end{aligned}$$

Si tales condiciones se cumplen, entonces el polinomio de aproximación converge a la función y sus derivadas.

#### 3.4.2 Teorema de aproximación de Weierstrass

Los polinomios algebraicos, son funciones muy conocidas y útiles, para realizar aproximaciones a funciones continuas. el Teorema de Weierstrass nos permite formular esta idea con una expresión matemática formal.

Sea  $f(x)$  cualquier función continua en  $[a,b]$  entonces para cualquier  $\epsilon > 0$  existe un número entero  $n = n(\epsilon)$  y un polinomio  $P_n(x)$  de grado  $n$  tal que

$$|f(x) - P_n(x)| < \epsilon \quad \forall x \in [a,b] \quad , n = n(\epsilon)$$

por lo tanto si  $n \rightarrow \infty$  entonces  $\epsilon \rightarrow 0$ .

### 3.4.3 Definición del polinomio de aproximación de Bernstein

El polinomio de aproximación de Bernstein es una función valuada en escalar.

$$B_n[f(s)] = \sum_{j=0}^n f\left(\frac{j}{n}\right) R_n^j(s) \quad , s \in [0,1]$$

$B_n[f(s)]$ : Polinomio de aproximación de Bernstein.

$f(n)$  : Función a aproximar

$n$  : Grado del polinomio.

$j/n$  : Puntos finales de los  $n$  intervalos en que se divide la función

$f(j/n)$ : Coeficientes de las funciones base. Función evaluada en los puntos  $j/n$

$R_n^j(s)$ : Funciones base del polinomio

Las funciones base se definen como:

$$R_n^j(P) = \binom{n}{j} P^j q^{n-j} \quad , \quad \binom{n}{j} = \frac{n!}{(n-j)! j!} \quad , \quad q = 1 - s \quad , \quad P = s$$

Sus características se resumen a continuación:

- a) Toda las funciones que forman la familia son continuas en  $[0,1]$ .

$$R_n^j(s) \in [0,1]$$

- b) El máximo de cada función elemento de la familia  $R_n^j(s)$  ocurre en  $s = j/n$ .

- c)  $R_n^j$  y  $(R_n^{j'})'$  son simétricas si y sólo si

$$\frac{j + j'}{n} = 1$$

- d) Propiedad de partición de la unidad:

$$\sum_{j=0}^n R_n^j(s) = 1 \quad \forall s \in [0,1]$$

- e) El polinomio de aproximación de Bernstein pasa por unos extremos de la función  $f(x)$ , y a los demás sólo se acerca.



### 3.4.4 Propiedades matemáticas

Para la demostración de estas propiedades, referirse a Vásquez[17].

- a)  $B_n [f(s)]$  es tangente en los puntos finales a la línea recta que une un punto extremo con el punto vecino interior.
- b)  $B_n [f(s)]$  imita a la función a la cual se aproxima (y a todas sus derivadas ).
- c)  $B_n [f(s)]$  es tan suave como  $f(s)$  en términos de número de oscilaciones respecto a cero (Teorema de Schoenberg).
- d) El efecto de suavidad de  $B_n [f(s)]$  no la tiene ningún otro método de aproximación polinomial (Teorema de Popiviciu).

## 3.5 EL MÉTODO DE BERNSTEIN-BÉZIER PARA TRAZAR SPLINES

---

La construcción geométrica de Bezier asocia a un conjunto de puntos (elementos de un espacio euclidiano  $E^*$ ), un polígono. El polígono se construye enlazando mediante líneas rectas puntos sucesivos del conjunto; y es a este polígono al que se asocia la curva de Bézier mediante una interesante técnica geométrica.

### 3.5.1 Algoritmo para la construcción de una curva de Bézier

1. Definir un conjunto de puntos en el espacio euclidiano  $E^*$ :  $(P_1, P_2, \dots, P_n)$ .
2. Enlazar los puntos con líneas rectas para formar el polígono de definición.
3. Parametrizar cada recta con  $s$ , donde  $0 \leq s \leq 1$ .
4. Especificar un  $\Delta s$  para "barrer" todo el intervalo  $[0,1]$ .
5. Desde  $s = 0$  y para  $s = s + \Delta s$  hacer
  - a) Localizar todos los puntos que corresponden a la distancia  $s$ , desde el punto  $P_j$  hasta el  $P_{j+1}$  sobre la línea correspondiente  $(P_0^{(j)}, P_1^{(j)}, \dots, P_n^{(j)})$ .
  - b) Trazar un nuevo polígono entre los nuevos puntos.
  - c) Repetir (a) y (b) para el valor actual de  $s$  hasta que sólo exista un punto:  $P_0^{(n)}(s)$ , punto de la curva de Bézier para el valor actual de  $s$ .
6. Repetir (5) hasta que  $s = 1$ .
7. Trazar el B-spline enlazando todos los puntos  $P_0^{(n)}(s)$ .

En la Figura 3.3 se muestra la aplicación del algoritmo para el trazo de un B-spline 2D.

Partiendo del algoritmo anterior Vásquez[17] plantea una técnica vectorial para formular la "*Ecuación algebraica del polinomio de Bézier*".

Si se tiene  $n+1$  puntos iniciales, la localización de cada punto se puede expresar como:

$$P_j^{(n)} = P_j^{(n+1)} + s [ P_{j+1}^{(n-1)} - P_j^{(n-1)} ]$$

### 3 ANÁLISIS GEOMÉTRICO Y MATEMÁTICO DE TRAYECTORIAS

Denotado la expresión anterior en función de los puntos del polígono de definición representa la "Ecuación algebraica del polinomio de Bézier".

$$\vec{P}_0^{(n)} = \sum_{j=0}^N s^j (1-s)^{n-j} \binom{n}{j} \vec{P}_j$$

Comparando esta nueva expresión con la función polinomio de Bernstein, Vázquez[17] concluye que la "Ecuación algebraica de la construcción geométrica de Bézier" es exactamente la "Función del polinomio de Bernstein valuada en vector", es decir:

$$\vec{B}_n[f(s)] = \sum_{j=0}^n \vec{P}_j R_n^j(s)$$

$B_n[P(s)]$  : Conjunto de vectores posición de los elementos de la curva de Bézier

$P_j$  : Vectores posición de los puntos que definen el polígono de definición

$R_n^j(s)$  : Funciones base del polinomio de Bernstein o funciones de Bézier

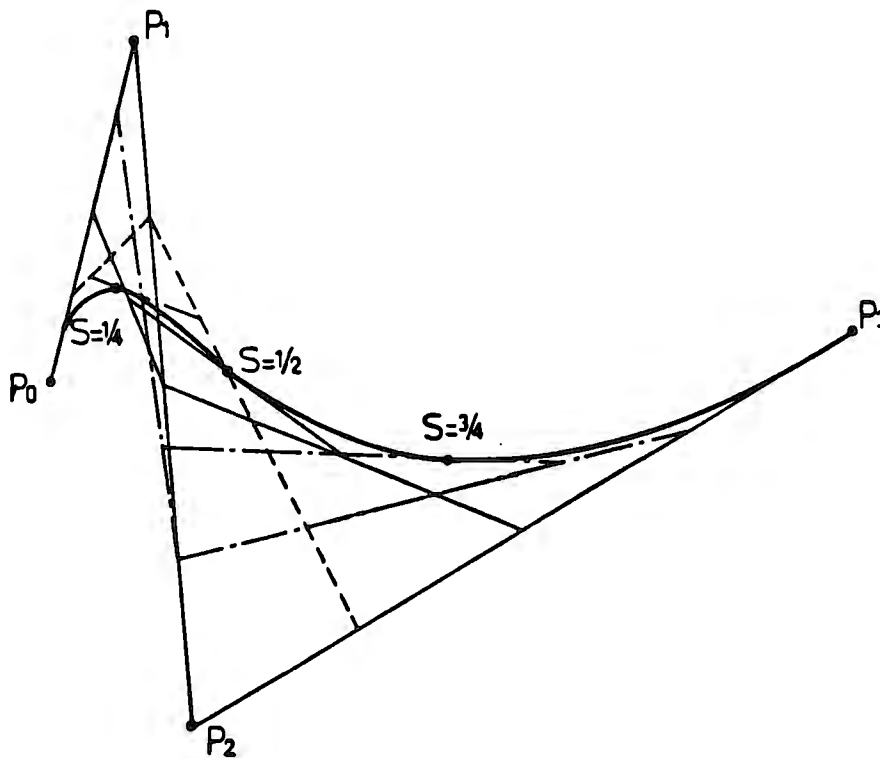


Figura 3.3 Construcción de curvas de Bézier.

### 3.5.2 Propiedades de las curvas de Bézier

Al tener como ecuación algebraica al polinomio de Bernstein, la construcción geométrica de Bézier asume todas las propiedades observadas para este polinomio.

- El grado de la función de Bézier, depende del número de puntos que definen el polígono de definición. Si  $n$  es el grado del polinomio de Bézier, entonces se necesitan  $n+1$  para generar el polinomio.
- La curva de  $n$ -ésimo grado expresada por los vectores  $B_n[P(s)]$  pasa por los puntos inicial y final del conjunto, y se aproxima a los intermedios.
- La curva es tangente en los puntos inicial y final a las líneas que los unen con los puntos vecinos interiores correspondientes.
- El B-spline permanecerá en el interior de la pared que forma el polígono de control.
- El B-spline se aproxima al polígono de definición cuando  $n$  aumenta.
- la función valuada con el vector  $B_n[P(s)]$  es invariable bajo transformaciones del sistema cordenado de referencia.
- La curva de grado  $n$ -ésimo expresada por los vectores de posición  $B_n[P(s)]$ , tiene un efecto de suavidad que ningún otro método de aproximación polinomial.
- Al ser una ecuación vectorial, se puede expresar en  $k$  ecuaciones para las  $k$  componentes de un espacio euclidiano  $E^k$ .
- La única manera de cambiar la forma de B-spline, es cambiando la posición de algunos de los puntos del polígono de definición.

La Figura 3.4 ilustra algunas de las propiedades con el trazo de diversos B-splines.

Como se mencionó en la Sección 3.2, se puede utilizar la expresión general de las curvas de Bézier (B-splines) para trazar curvas de  $n$ -ésimo grado con  $n+1$  puntos; o trazar  $m$  B-splines cónicos o cúbicos (tomando 3 o 4 puntos respectivamente) para generar la curva compleja.

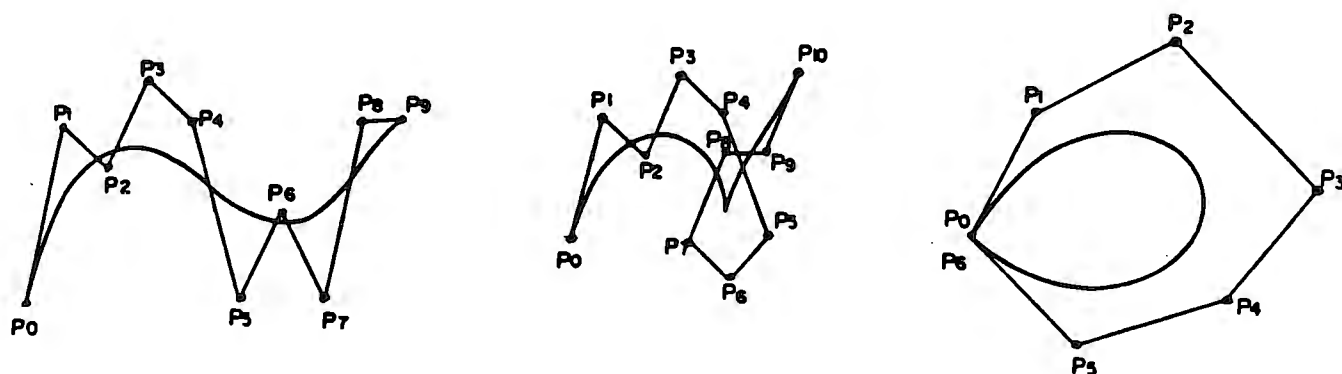


Figura 3.4 Algunos B-splines planos asociados a sus respectivos polígonos.

### 3.6 CURVAS RACIONALES DE BEZIER (B-SPLINES RACIONALES)

La principal desventaja de trabajar con el método general para el trazo de B-splines, es que el gran número de puntos para generarlos conducen a la obtención de polinomios de grado elevado, junto con elementos cuyo significado es difícil de interpretar. Tales elementos pueden producir problemas sobre el control de propiedades de la curva al crear puntos de inflexión y torceduras en el lugar geométrico que se define.

Existe una forma alterna para escribir el polinomio de Bernstein-Bézier. Cuando esta ecuación se escribe en coordenadas homogéneas, se permite mayor libertad para manipular la forma geométrica de la curva. De esta manera se pueden corregir los problemas causados por el alto grado del polinomio.

Esto se logra incluyendo en la ecuación algunos *parámetros de control de la forma de la curva*, denominados *pesos* ( $W_i$ ).

#### 3.6.1 La forma general de los B-splines racionales

Una curva de Bézier en forma racional (B-spline racional), se define por la ecuación algebraica:

$$\vec{r}(s) = \frac{\sum_{j=0}^n W_j \vec{r}_j R_n^j(s)}{\sum_{j=0}^n W_j R_n^j(s)}, \quad s \in [0,1]$$

$\vec{r}_j(s)$ : Vectores posición que localizan a los puntos  $P_j$  en un espacio Euclidiano  $E^*$ . Estos puntos forman un polígono de control al que se asocia una curva de Bézier.

$W_j$ : Parámetro de control de la forma de la curva asociados a los puntos  $P_j$ .

$R_n^j(s)$ : Funciones de Bézier.

Para estudiar las propiedades de los B-splines racionales, se puede reescribir la ecuación anterior en forma integral.

$$\vec{r}(s) = \sum_{j=0}^n \vec{r}_j B_j^n(s)$$

$$B_j^n(s) = \frac{W_j R_j^n(s)}{\sum_{k=0}^n W_k R_k^n(s)}$$

### 3.6.2 Propiedades de los B-splines racionales

- a) El polinomio de Bézier en forma racional es un caso particular de la forma racional.  
 Cuando  $W_j = W_k = 1 \quad \forall j, k = 0, 1, 2, \dots, n$

$$\sum_{k=0}^n W_k R_k^n(s) = \sum_{k=0}^n R_k^n(s) = 1$$

$$B_j^n(s) = R_j^n(s)$$

$$\therefore \vec{r} = \sum_{j=0}^n \vec{r}_j R_j^n(s) = \vec{B}[\vec{r}(s)]$$

- b) La relación del B-spline racional con su polígono de definición es independiente del sistema de referencia utilizado.

$$\sum_{j=0}^n B_j^n(s) = 1$$

- c) La curva de Bézier asociada al polígono de definición permanece por entero dentro de él.

$$\text{Si } W_j \geq 0 \quad \forall j = 0, 1, 2, \dots, n \rightarrow B_j^n(t) \geq 0$$

- d) La curva de Bézier pasa por los puntos inicial y final del polígono.

$$\vec{r}(0) = \vec{r}_0$$

$$\vec{r}(1) = \vec{r}_n$$

- e) El B-spline racional es tangente en los puntos inicial y final al polígono de definición.

$$\dot{\vec{r}}(0) = n \frac{W_1}{W_0} [\vec{r}_1 - \vec{r}_0]$$

$$\dot{\vec{r}}(1) = n \frac{W_{n-1}}{W_n} [\vec{r}_n - \vec{r}_{n-1}]$$

### 3.6.3 Los parámetros de control de la forma geométrica de la curva

La forma geométrica de un B-spline racional depende de:

- a) La forma del polígono de definición, es decir, de la posición de los puntos que definen sus vértices.
- b) Los valores de los parámetros de forma  $W_j$  (pesos).

Asignando diferentes valores a los pesos se puede cambiar la forma de la curva, haciendo que se aproxime o aleje a los puntos vértices del polígono. El problema es entonces, determinar los valores adecuados para los pesos que nos generen una curva con la forma deseada.

#### 3.6.3.1 Implicaciones de la variación de los pesos sobre las funciones base

- a) Para  $B_j^n(s) = 0$  cuando  $W_j = 0$ ,  $j = 0, 1, 2, \dots, n$ ; la curva no se define, es un punto.
- b) Si  $W_0 = W_n \neq 0$  y  $W_j = 0 \forall j = 1, 2, 3, \dots, n-1$ ; el gráfico representa una recta entre los puntos definidos por:

$$r_0 \text{ y } r_n$$

- c) Si  $W_j = 1 \forall j = 0, 1, 2, \dots, n \rightarrow B_j^n(s) = B_j^n(t)$ ; la curva esta dada en forma no racional.
- d) La curva se acerca al punto vértice del polígono, es decir:

$$\vec{r}(s) = \sum_{j=0}^n r_j B_j^n(s)$$

Si y sólo si:

$$\lim_{W_j \rightarrow +\infty} B_j^n(s) = 1, \quad s \in [0,1]$$

- e) La curva se aleja de los puntos vecinos a  $P_j$  (efecto del punto vecino), si y sólo si:

$$\lim_{W_j \rightarrow +\infty} B_j^n(s) = 0, \quad k \neq j, \quad s \in [0,1]$$

3.6.3.2 Significado geométrico de los parámetros de forma

De las consideraciones antes descritas sobre la variación de los pesos, se puede establecer que el problema de evaluarlos no es un asunto sencillo. El valor de cada parámetro  $W_j$  asociado a su correspondiente punto vértice  $P_j$  es función de los puntos vecinos.

$$W_j = \frac{\frac{1 - U}{U}}{\frac{1 - V}{V}}$$

$$U = \frac{B_j^n(s)}{\sum_{\substack{k=0 \\ k \neq j}}^n W_k B_k^n(s) + B_j^n(s)}$$

$$V = \frac{W_j B_j^n(s)}{\sum_{k=0}^n W_k B_k^n(s)}$$

Vásquez[17] establece un método de evaluación mediante una analogía geométrica. Sin embargo la velocidad de convergencia es muy lenta y poco equitativa (algunos puntos convergen más rápido que otros). Farin[19] hace un estudio a fondo sobre los B-splines racionales cónicos, en los que existen 3 pesos y sólo uno de ellos es desconocido (siempre se cumple que los pesos  $W_0 = W_n = 1$ ). Chou-Piegl[20] presenta un método de trazo usando B-splines racionales cúbicos, cuyos valores de los pesos (2 por cada conjunto de tres puntos) pueden obtenerse siempre por simple manipulación algebraica.

A falta de una técnica directa de evaluación. La analogía geométrica resulta la alternativa más viable para resolver el problema, aunque no es óptima en modo alguno.

Para compensar las desventajas del método de Vásquez[17], se se presenta un método derivado alternativo.

Se inicia el estudio suponiendo la existencia de un polígono con  $n$  puntos vértices  $(P_0, P_1, \dots, P_n)$  y  $n$  pesos fijos  $(W_0, W_1, \dots, W_n)$ , ver Figura 3.5, de tal manera que se cumple la condición:

$$W_0, W_n > 0 \text{ y } W_k \geq 0 \text{ para } k = 1, 2, 3, \dots, n-1.$$

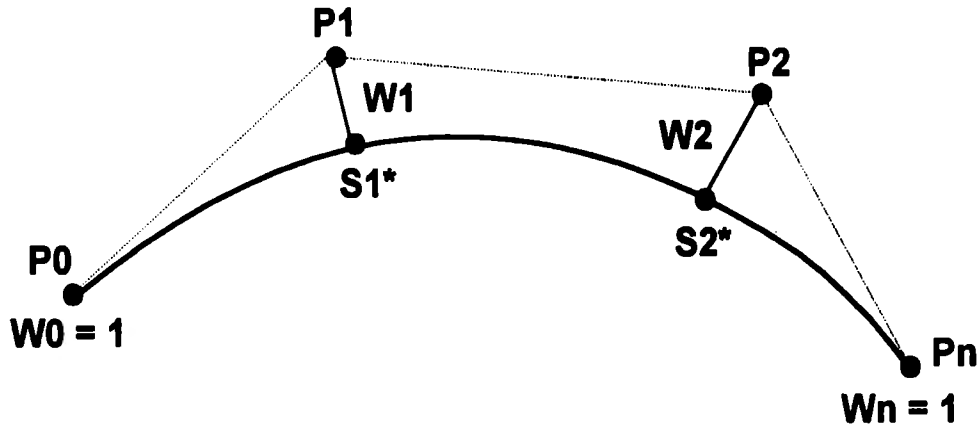


Figura 3.5 Parámetros de forma del B-spline racional

De las consideraciones de la Sección 3.6.3.1 se desprende de los pesos  $W_0 = W_n = 1$ , al corresponder a los puntos externos del B-spline. Experimentalmente se encontró que variando su valor, el B-spline se distorsiona en su vecindad.

También se establece lo siguiente:

- a) Cuando  $W_j \rightarrow \infty \quad \forall j = 1, 2, \dots, n-1$ ; la longitud del B-spline racional tiende a igualarse con la longitud de polígono de definición.
- b) Los puntos  $s_j^* \quad \forall j = 1, 2, \dots, n-1$ ; se localizan sobre el B-spline no racional cuando  $W_j = 1 \quad \forall j = 1, 2, \dots, n-1$ .
- c) Cuando  $W_j \rightarrow \infty \quad \forall j = 1, 2, \dots, n-1$ ; Los puntos  $s_j^* \rightarrow P_j$ .

Se propone un parámetro auxiliar  $V$  (*factor de acercamiento*), para obtener una relación de evaluación para los pesos  $W_j$ . De la Figura 3.4 se desprende que:

$$\text{Cuando } [s_j \rightarrow P_j] \rightarrow [V \rightarrow 0] \rightarrow [W_j \rightarrow \infty]$$

$$\text{Cuando } [s_j \rightarrow s_j^*] \rightarrow [V \rightarrow 1] \rightarrow [W_j \rightarrow 1]$$

Por lo tanto:

$$W_j = \frac{1 - V}{V} + 1$$



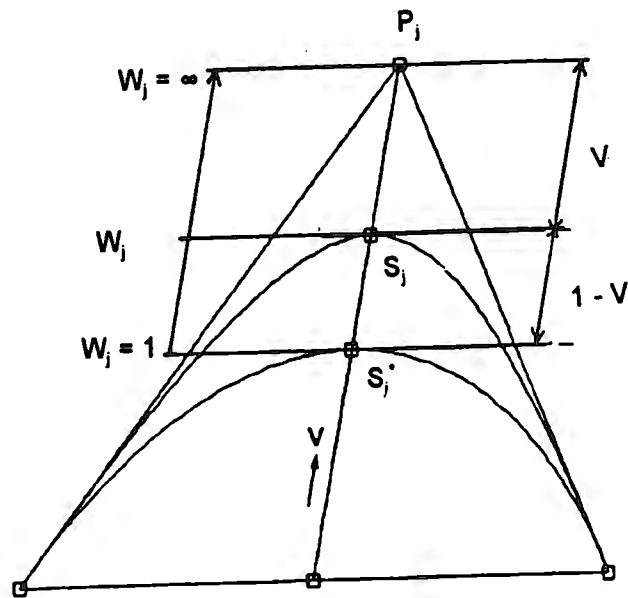


Figura 3.6 Interpretación geométrica de los parámetros de forma.

El método para evaluar los pesos puede resumirse como sigue:

1. Obtener la longitud del polígono de definición considerando que corresponde a un B-spline cuyos  $W_j$  tienden todos a infinito.
2. Parametrizar el polígono de definición respecto de  $s$  y encontrar los valores de  $s$  a que corresponden los puntos vértices intermedios  $P_j$ .
3. Encontrar los puntos  $s_j^*$  sobre el B-spline no racional (todos los  $W_j$  son igual a uno), usando los valores de  $s$  obtenidos en el inciso anterior.
4. Determinar las distancias entre cada par de puntos  $(P_j, s_j^*)$  y encontrar la mínima del conjunto.
5. Proponer un valor para el factor de acercamiento  $V$ . Por ejemplo,  $V = 0.25$  si se necesita que la separación entre cada par de puntos  $(P_j, s_j^*)$  se reduzca a la cuarta parte.
6. Dividir todo el conjunto de distancias entre la distancia mínima y multiplicar los resultados por la *relación de evaluación*. El resultado será un vector de pesos  $W$  relacionado con el vector de puntos  $P$ .

Es necesario mencionar que éste método al ser una simplificación del problema de evaluación de pesos original, debe aplicarse con reservas. El principal inconveniente es el *efecto del punto vecino*, que tiende a alejar la curva de los puntos vecinos cuando se incrementa el valor de un peso buscando acercarla a un punto específico. A pesar de este hecho, la técnica funciona bastante bien como se mostrará en la siguiente sección.

### 3.7 IMPLEMENTACIÓN DEL MÉTODO DE LOS B-SPLINES RACIONALES

---

La implementación del método se hizo en Mathematica 2.0 versión para Microsoft Windows, y se corrió en una computadora personal con microprocesador Intel 80486. Las características de la máquina son las requeridas por el paquete para un funcionamiento adecuado: 8 Mb de memoria RAM, 20 Mb libres en el disco duro y Windows versión 3.1 instalado.

El paquete puede generar generar B-splines racionales y no racionales tanto literal como numéricamente; al igual que trazar las curvas correspondientes en dos o tres dimensiones. El resultado último del paquete es la generación de un B-spline racional para posicionamiento y orientación de una herramienta de robot en el espacio. Esto se logra concatenando dos splines *cuasiparalelos*, con el mismo número de puntos, donde la línea que une puntos correspondientes representa la orientación requerida, ver Figura 3.5. En el capítulo siguiente se tratarán a fondo estos puntos.

A continuación se presentan algunos ejemplos de como trabaja el paquete desde sus funciones más elementales, hasta llegar a la generación del B-spline para una trayectoria de robot.

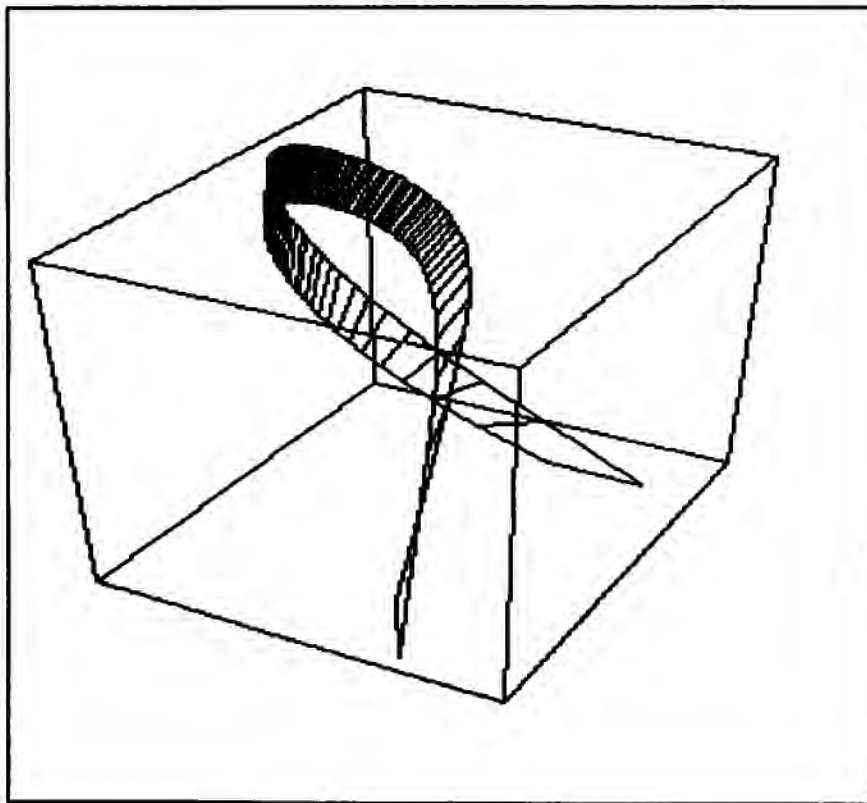


Figura 3.7 B-spline racional para posicionamiento y orientación.

### 3 ANÁLISIS GEOMÉTRICO Y MATEMÁTICO DE TRAYECTORIAS

#### 3.7.1 Funciones base de Bernstein

##### 3.7.1.1 Elemento de una familia

$R_{nj}[n,j]$

n: Grado del polinomio de Bernstein (grado de la familia de funciones base).

j: Número del elemento ( $0 < j < n$ ).

$R_{nj}[1,0]$

$$1 - s$$

$R_{nj}[3,2]$

$$3 (1 - s) s^2$$

$R_{nj}[10,5]$

$$252 (1 - s)^5 s^5$$

##### 3.7.1.2 Elemento de una familia evaluado en s

$NR_{nj}[n,j,s]$

n: Grado del polinomio de Bernstein (grado de la familia de funciones base).

j: Número del elemento ( $0 < j < n$ ).

s: Variable paramétrica ( $0 < s < 1$ ).

$$NR_{nj}[1,0,.2] = 0.8$$

$$NR_{nj}[2,1,.25] = 0.375$$

$$NR_{nj}[3,2,.33] = 0.218889$$

$$NR_{nj}[10,5,1] = 0$$

##### 3.7.1.3 Familia de curvas

$R_n[n]$

n: Grado del polinomio de Bernstein (grado de la familia de funciones base).

$R_n[1]$

$$\{1 - s, s\}$$

$R_n[3]$

$$\{(1 - s)^3, 3 (1 - s)^2 s, 3 (1 - s) s^2, s^3\}$$

$R_n[10]$

$$\{(1 - s)^{10}, 10 (1 - s)^9 s, 45 (1 - s)^8 s^2, 120 (1 - s)^7 s^3, 210 (1 - s)^6 s^4, 252 (1 - s)^5 s^5, 210 (1 - s)^4 s^6, 120 (1 - s)^3 s^7, 45 (1 - s)^2 s^8, 10 (1 - s) s^9, s^{10}\}$$

### 3 ANÁLISIS GEOMÉTRICO Y MATEMÁTICO DE TRAYECTORIAS

#### 3.7.1.4 Familia de curvas evaluada en $s$

**NRn[n,s\_]**

n: Grado del polinomio de Bernstein (grado de la familia de funciones base).

s: Variable paramétrica ( $0 < s < 1$ ).

**NRn[1,.2]**

{0.8, 0.2}

**NRn[2,.5]**

{0.25, 0.5, 0.25}

**NRn[4,.75]**

{0.00390625, 0.046875, 0.210937, 0.421875, 0.316406}

#### 3.7.1.5 Elemento de una familia evaluado en [0,1]

**ValueRnj[n,j,ds\_]**

n: Grado del polinomio de Bernstein (grado de la familia de funciones base).

j: Número del elemento ( $0 < j < n$ ).

ds: Incremento de la variable paramétrica para "barrer" el intervalo [0, 1].

**ValueRnj[1,1,.2]**

{0, 0.2, 0.4, 0.6, 0.8, 1}

**ValueRnj[2,1,.25]**

{0, 0.375, 0.5, 0.375, 0.}

**ValueRnj[5,3,.2]**

{0, 0.0512, 0.2304, 0.3456, 0.2048, 0.}

#### 3.7.1.6 Familia de curvas evaluada en [0,1]

**ValueRn[n,ds\_]**

n: Grado del polinomio de Bernstein (grado de la familia de funciones base).

ds: Incremento de la variable paramétrica para "barrer" el intervalo [0, 1].

**ValueRn[1,.2]**

{{1, 0.8, 0.6, 0.4, 0.2, 0.}, {0, 0.2, 0.4, 0.6, 0.8, 1}}

**ValueRn[2,.2]**

{{1, 0.64, 0.36, 0.16, 0.04, 0.}, {0, 0.32, 0.48, 0.48, 0.32, 0.},  
{0, 0.04, 0.16, 0.36, 0.64, 1}}

### 3 ANÁLISIS GEOMÉTRICO Y MATEMÁTICO DE TRAYECTORIAS

ValueRn[3,.1]

$$\{ \{1, 0.729, 0.512, 0.343, 0.216, 0.125, 0.064, 0.027, 0.008, 0.001, 1.36846 \cdot 10^{-48}\},$$

$$\{0, 0.243, 0.384, 0.441, 0.432, 0.375, 0.288, 0.189, 0.096, 0.027, 3.69779 \cdot 10^{-32}\},$$

$$\{0, 0.027, 0.096, 0.189, 0.288, 0.375, 0.432, 0.441, 0.384, 0.243, 3.33067 \cdot 10^{-16}\},$$

$$\{0, 0.001, 0.008, 0.027, 0.064, 0.125, 0.216, 0.343, 0.512, 0.729, 1\} \}$$

#### 3.7.2 Polinomios de Bernstein-Bézier

##### 3.7.2.1 Polinomio de grado n

$B_n[n\_]$

n: Grado del polinomio de Bernstein (grado de la familia de funciones base).

$B_n[1]$

$$(1 - s) P_j[0] + s P_j[1]$$

$B_n[3]$

$$(1 - s)^3 P_j[0] + 3 (1 - s)^2 s P_j[1] + 3 (1 - s) s^2 P_j[2] + s^3 P_j[3]$$

$B_n[5]$

$$(1 - s)^5 P_j[0] + 5 (1 - s)^4 s P_j[1] + 10 (1 - s)^3 s^2 P_j[2] +$$

$$10 (1 - s)^2 s^3 P_j[3] + 5 (1 - s) s^4 P_j[4] + s^5 P_j[5]$$

##### 3.7.2.2 Polinomio de grado n evaluado en s

$NB_n[P, s\_]$

n: Grado del polinomio de Bernstein (grado de la familia de funciones base).

P: Conjunto de vectores que localizan los vértices del polígono de definición.

$P = \{ \{0,1\}, \{2,2\}, \{3,6\}, \{4,3\}, \{6,2\} \};$

$NB_n[P, 0]$

$\{0, 1\}$

$NB_n[P, .25]$

$\{1.6875, 2.57422\}$

$NB_n[P, .75]$

$\{4.3125, 3.26172\}$

### 3 ANÁLISIS GEOMÉTRICO Y MATEMÁTICO DE TRAYECTORIAS

#### 3.7.2.3 Puntos del B-spline no racional

BezierPoints[P, ds]

P: Conjunto de vectores que localizan los vértices del polígono de definición.

ds: Incremento de la variable paramétrica para "barrer" el intervalo [0, 1].

P={{0,1},{2,2},{3,6},{4,3},{6,2}};

BezierPoints[P,.25]

{0, 1}, {1.6875, 2.57422}, {3., 3.6875}, {4.3125, 3.26172},  
{6., 2.}}

P={{5,1,0},{3,6,3},{1,4,5},{3,3,3},{5,4,1}};

BezierPoints[P,.1]

{5, 1, 0}, {4.2152, 2.6113, 1.1287}, {3.5152, 3.5648, 2.0752},  
{2.9672, 4.0273, 2.7927}, {2.6192, 4.1488, 3.2512},  
{2.5, 4.0625, 3.4375}, {2.6192, 3.8848, 3.3552},  
{2.9672, 3.7153, 3.0247}, {3.5152, 3.6368, 2.4832},  
{4.2152, 3.7153, 1.7847}, {5., 4., 1.}}

#### 3.7.3 B-splines no racionales

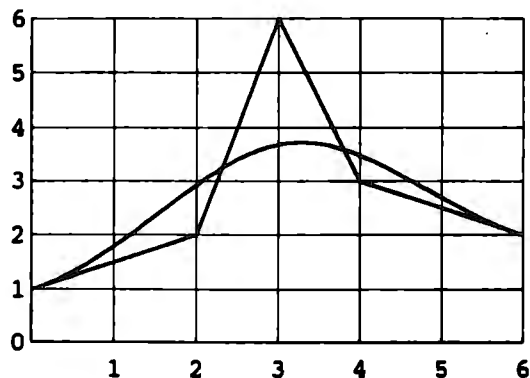
PlotBezier[P,ds\_]:

P: Conjunto de vectores que localizan los vértices del polígono de definición.

ds: Incremento de la variable paramétrica para "barrer" el intervalo [0, 1].

P={{0,1},{2,2},{3,6},{4,3},{6,2}};

PlotBezier[P,.01]



3.7.4 Factores de forma (pesos)

3.7.4.1 Valor de s correspondiente al punto P<sub>k</sub> del polígono de definición

ValueS[P<sub>,n,k,Lpd</sub>]

P: Conjunto de vectores que localizan los vértices del polígono de definición.

n: Grado del polinomio de Bernstein (grado de la familia de funciones base).

k: Número del punto a evaluar.

Lpd: Longitud del polígono de definición.

P={{0,1},{2,2},{3,6},{4,3},{6,2}};

N[ValueS[P,5,2,11]] = 0.203279

N[ValueS[P,5,3,11]] = 0.578107

N[ValueS[P,5,4,11]] = 0.865586

3.7.4.2 Distancia del punto P<sub>k</sub> al punto S<sub>k</sub>\*

DistPtoS[P<sub>,n,k</sub>]

P: Conjunto de vectores que localizan los vértices del polígono de definición.

n: Grado del polinomio de Bernstein (grado de la familia de funciones base).

k: Número del punto a evaluar.

N[DistPtoS[P,5,2]] = 0.687284

N[DistPtoS[P,5,3]] = 2.28302

N[DistPtoS[P,5,4]] = 0.668632

3.7.4.3 Vector de distancias de los puntos P<sub>k</sub> a los puntos S<sub>k</sub>\*

VectDist[P<sub>,n</sub>]

P: Conjunto de vectores que localizan los vértices del polígono de definición.

n: Grado del polinomio de Bernstein (grado de la familia de funciones base).

N[VectDist[P,5]]

{0, 0.687284, 2.28302, 0.668632, 0}

3.7.4.4 Vector de factores de forma (psos)

VecW[P<sub>,V</sub>]

P: Conjunto de vectores que localizan los vértices del polígono de definición.

V: Factor de accrcamiento.

N[VecW[P,.5]]

{1., 2., 6.64359, 1.94572, 1.}

N[VecW[P,.1]]

{1., 10., 33.2179, 9.72861, 1.}

3.7.5 B-splines racionales

3.7.5.1 Polinomio racional de grado n

RBn[n\_]

n: Grado del polinomio de Bernstein (grado de la familia de funciones base).

RBn[0]

Pj[0]

RBn[1]

$$\frac{(1 - s) Pj[0] Wj[0] + s Pj[1] Wj[1]}{(1 - s) Wj[0] + s Wj[1]}$$

RBn[3]

$$\frac{((1 - s)^3 Pj[0] Wj[0] + 3 (1 - s)^2 s Pj[1] Wj[1] + 3 (1 - s) s^2 Pj[2] Wj[2] + s^3 Pj[3] Wj[3])}{((1 - s)^3 Wj[0] + 3 (1 - s)^2 s Wj[1] + 3 (1 - s) s^2 Wj[2] + s^3 Wj[3])}$$

3.7.5.2 Puntos del B-spline racional

RBezierPoints[P\_,W\_,ds\_]

P: Conjunto de vectores que localizan los vértices del polígono de definición.

W: Vector de pesos.

ds: Resolución del B-spline racional.

P={{0,1},{2,2},{3,6},{4,3},{6,2}}

W=N[VecW[P,.05]]

{1., 20., 66.4359, 19.4572, 1.}

RBezierPoints[P,W,.05]

{0, 1.}, {1.86199, 2.54191}, {2.21018, 3.25974}, {2.40318, 3.75545}, {2.53807, 4.1  
 {2.64267, 4.39198}, {2.72913, 4.59885}, {2.80409, 4.75443}, {2.87172, 4.8683},  
 {2.93495, 4.94654}, {2.99611, 4.99276}, {3.05726, 5.00863}, {3.12046, 4.99405},  
 {3.18804, 4.94711}, {3.26301, 4.86372}, {3.34959, 4.73678}, {3.4546, 4.5546},  
 {3.59053, 4.2976}, {3.786, 3.93015}, {4.14044, 3.37039}, {6., 2.}



3.7.5.3 B-spline racional

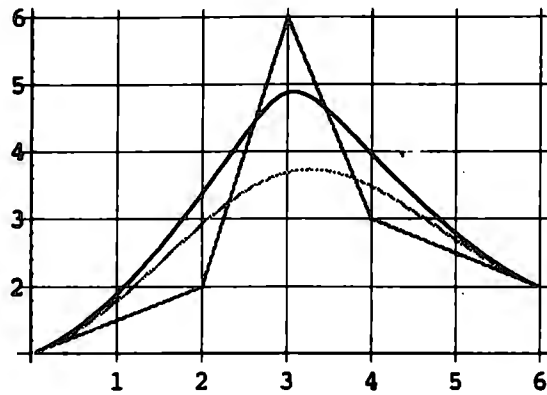
PlotRBezier[P\_,ds\_,V\_]

P: Conjunto de vectores que localizan los vértices del polígono de definición.

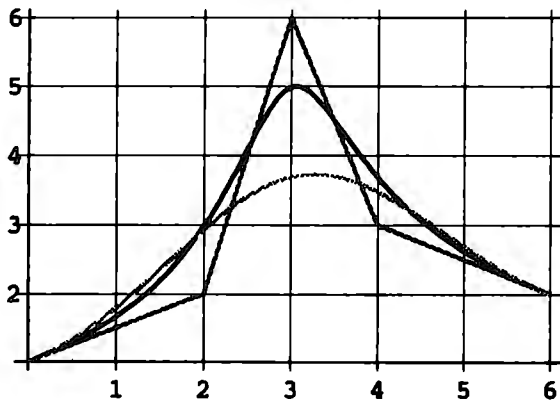
ds: Resolución del B-spline racional.

V: Factor de acercamiento.

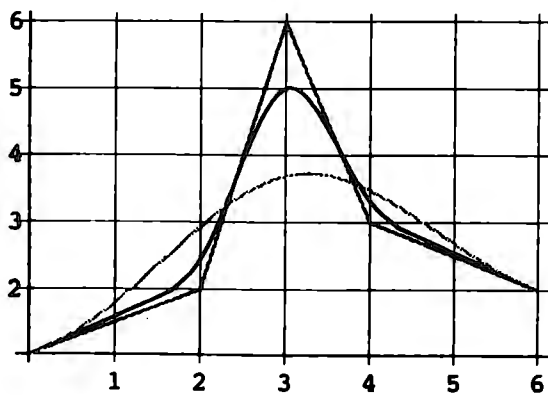
PlotRBezier[P,01,.5]



PlotRBezier[P,01,1]



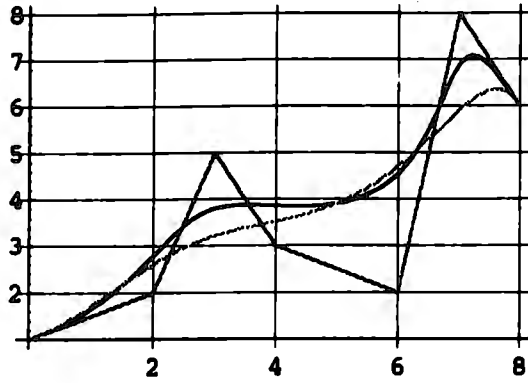
PlotRBezier[P,01,01]



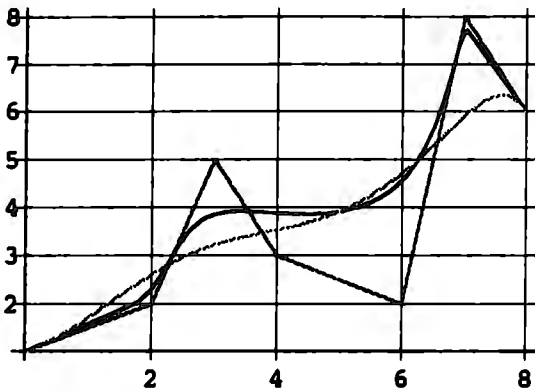
### 3 ANÁLISIS GEOMÉTRICO Y MATEMÁTICO DE TRAYECTORIAS

$P = \{(0,1), (2,2), (3,5), (4,3), (6,2), (7,8), (8,6)\}$ ;

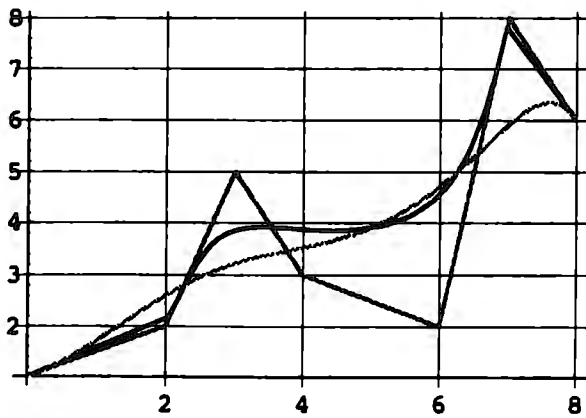
PlotRBezier[P,.01,.2]



PlotRBezier[P,.01,.01]



PlotRBezier[P,.01,.001]



### 3 ANÁLISIS GEOMÉTRICO Y MATEMÁTICO DE TRAYECTORIAS

#### 3.7.5.4 B-spline racional tridimensional

`PlotRBezier3D[P_,ds_,V_]`

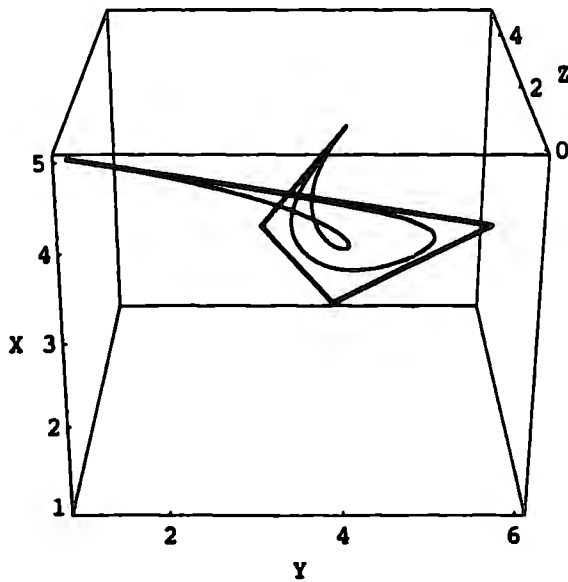
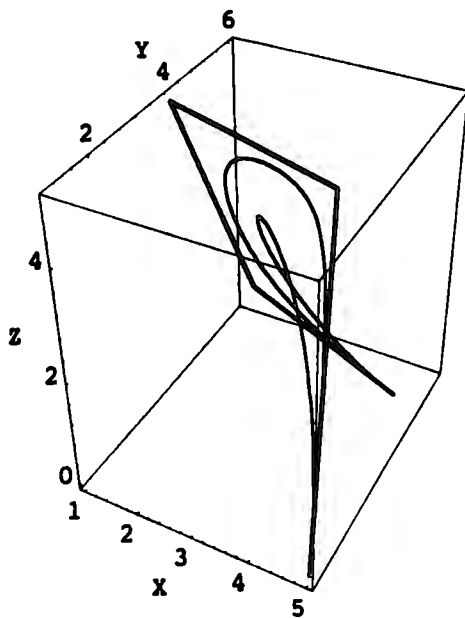
P: Conjunto de vectores que localizan los vértices del polígono de definición.

ds: Resolución del B-spline racional.

V: Factor de acercamiento.

`P={{5,1,0},{3,6,3},{1,4,5},{3,3,3},{5,4,1}};`

`PlotRBezier3D[P,.01,.05];`



### 3 ANÁLISIS GEOMÉTRICO Y MATEMÁTICO DE TRAYECTORIAS

#### 3.7.6 B-splines racionales para posicionamiento y orientación

PlotRBezier3DAII[P,Dir\_ds,V\_]

P: Conjunto de vectores que localizan los vértices del polígono de definición.

Dir: Conjunto de puntos para definir los vectores de orientación.

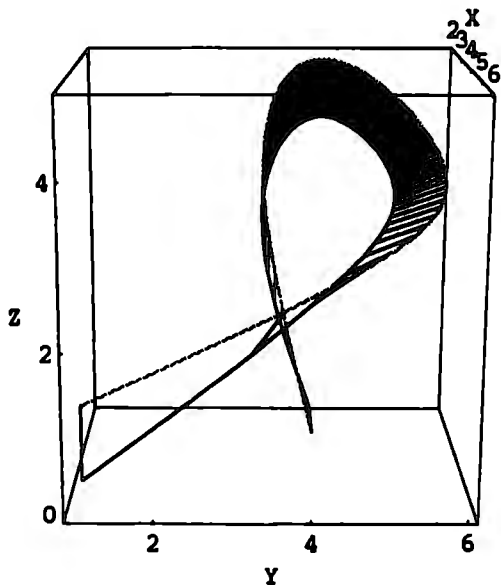
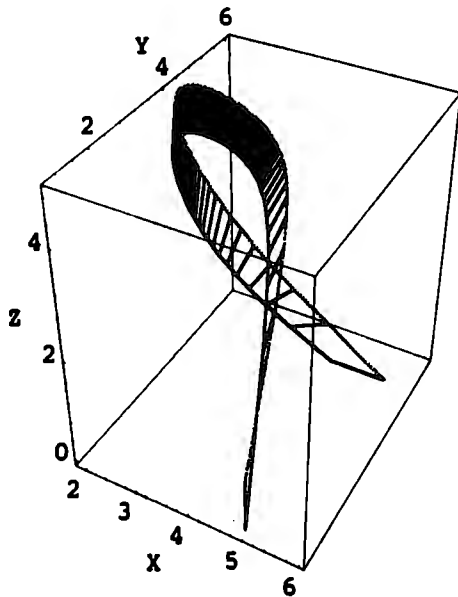
ds: Resolución del B-spline racional.

V: Factor de acercamiento.

P={{5,1,0},{3,6,3},{1,4,5},{3,3,3},{5,4,1}};

Di={{5,1,1},{3,7,3},{1,4,6},{3,3,4},{6,4,1}};

PlotRBezier3DAII[P,Di,.01,.05];



# CAPÍTULO 4

## GENERACIÓN DE TRAYECTORIAS

### 4.1 INTRODUCCIÓN

En el Capítulo 2 se trataron algunos conceptos fundamentales sobre la generación de trayectorias. Se estableció que la *trayectoria* se refiere a la historia temporal de posición, velocidad y aceleración para cada grado de libertad del robot. Los métodos de generación de trayectorias "aproximan" la *ruta* deseada haciendo uso de diversos tipos de funciones (como las estudiadas en el Capítulo 3), y generan una secuencia de puntos de consigna o *parámetros de control* a lo largo del tiempo para llevar al manipulador desde su posición inicial hasta su destino.

Existen dos alternativas para planificar trayectorias:

- a) Planificación en espacio de articulación:  $\Theta = f(t)$ .
- b) Planificación en espacio cartesiano:  $\chi = f(t)$ .

La relación entre los espacios depende de la anatomía de los manipuladores, que a excepción de los cartesianos, en todos los demás es altamente no lineal; ver Figura 4.1. Este obstáculo ha obligado a los investigadores a desarrollar métodos específicos para cada espacio.

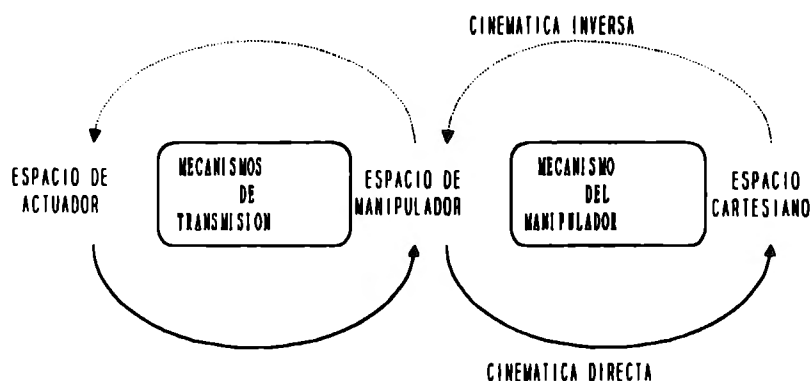


Figura 4.1 Espacios de trabajo de un robot manipulador.

Los puntos ligadura de la ruta se pueden especificar en coordenadas de articulación o en coordenadas cartesianas. Las coordenadas cartesianas no son adecuadas como coordenadas de trabajo porque los ejes de las articulaciones de la mayoría de los manipuladores no son ortogonales, y no separan posición de orientación. Si se requieren las coordenadas de articulación se debe recurrir a la *cinemática inversa* del robot.

Se han identificado dos problemas básicos que dependen de la información dada para planear la trayectoria, ver Figura 4.2

INFORMACIÓN	INTERFASE DE USUARIO	COORDENADAS UTILIZADAS	FORMA DE GENERACIÓN
Puntos de trabajo	Especificación de un conjunto de puntos ligadura junto con sus correspondientes restricciones (posición, velocidad, aceleración, jerk).	Restricciones: De articulación  Trayectoria: De articulación	Interpolación con algún tipo de función polinomial para satisfacer las restricciones planteadas
Rutas de trabajo	Especificación de la ruta a seguir por la herramienta del robot usando una función analítica	Restricciones: Cartesianas:  Trayectoria: De articulación	Aproximación a la ruta cartesiana mediante funciones en espacio de articulación.  Transformación de las restricciones cartesianas en restricciones de articulación via <i>cinemática inversa</i> , y uso del método anterior.

Figura 4.2 Problemas de la planeación de trayectorias.

Antes de revisar los métodos, es necesario situarse nuevamente dentro del problema general de la planeación de trayectorias. Las consideraciones generales sobre la planeación de trayectorias enunciadas por Fu[12] se compilan en la Figura 4.3.

## 4.2 ESPACIO DE ARTICULACIÓN

En esta sección se describen algunos métodos para la generación de rutas descritas en términos de funciones de ángulos de articulación. Utilizando la notación del modelado con *matrices transformadas homogéneas* usada por Craig[9], el problema puede describirse esquemáticamente de la siguiente manera:

$$\left. \begin{array}{l}
 {}^E\chi_H(t_0) \rightarrow [UTOI] \rightarrow {}^E_H T_0 \rightarrow [INVKIN] \rightarrow \theta(t_0) \\
 \vdots \\
 {}^E\chi_H(t_f) \rightarrow [UTOI] \rightarrow {}^E_H T_f \rightarrow [INVKIN] \rightarrow \theta(t_f)
 \end{array} \right\} \left\{ \begin{array}{l}
 \theta(t) \\
 \frac{d\theta(t)}{dt} \\
 \frac{d^2\theta(t)}{dt^2}
 \end{array} \right.$$

$${}^E\chi_H = [x \ y \ z \ \alpha \ \beta \ \gamma]^T$$

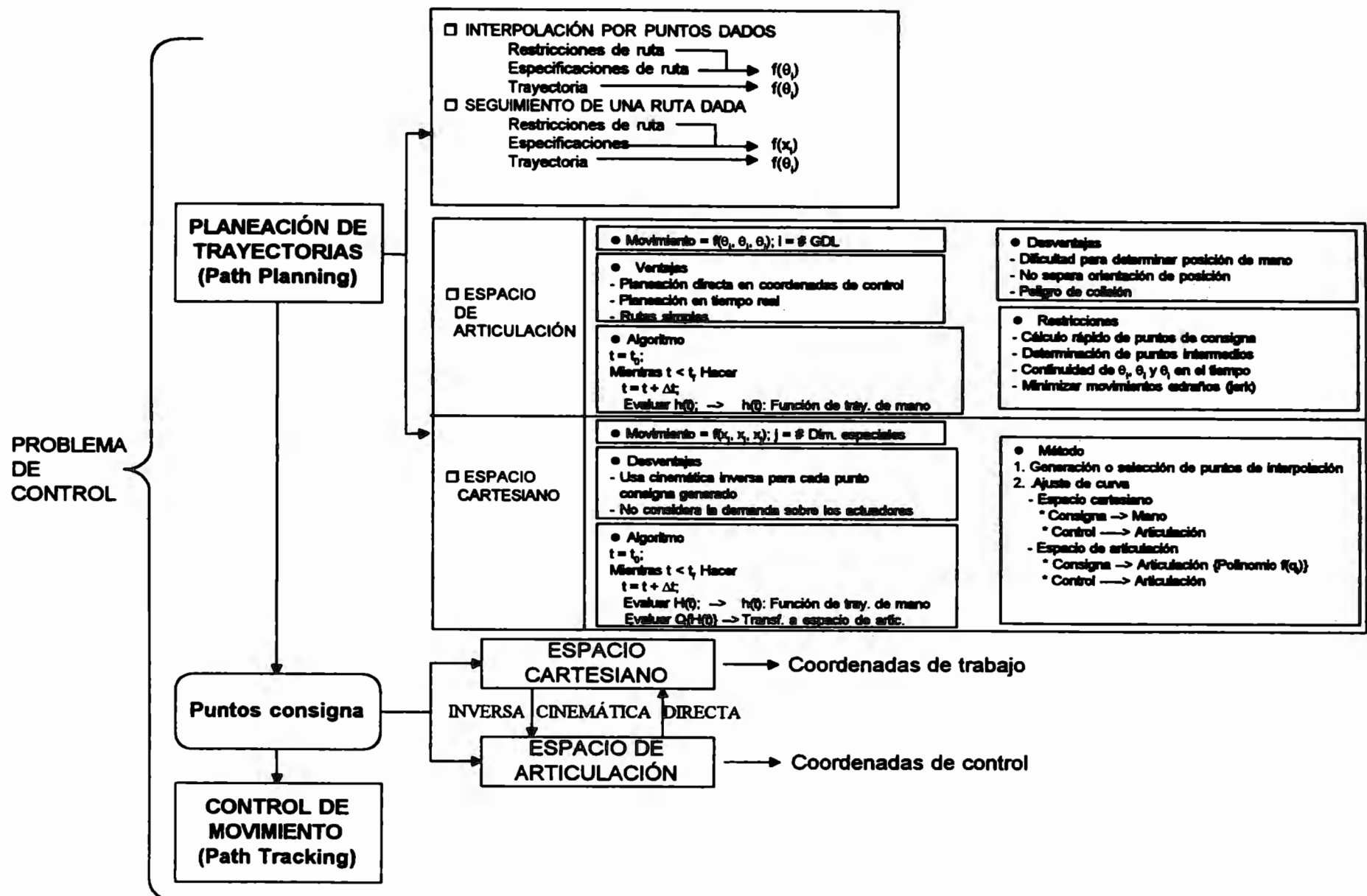


Figura 4.3 Consideraciones generales sobre la planeación de trayectorias.

Cada punto  ${}^E\chi_H$  es especificado por una posición y una orientación de la herramienta (efector final) respecto al marco de referencia de la estación de trabajo, y de un intervalo de tiempo  $[t_0, t_f]$ . Después se hace una transformación de la representación del usuario a la del sistema robótico mediante una rutina UTOI (User form TO Internal form). Posteriormente, todos los puntos son convertidos a un conjunto de ángulos de articulación aplicando la *cinemática inversa*.

Finalmente se encuentra una función de interpolación suave para cada una de las  $n$  articulaciones, que pase a través de los puntos ligadura hasta alcanzar la posición final. El tiempo requerido para moverse en cada segmento es el mismo para todas las articulaciones, por dos razones principales:

- a) Que todas alcancen los puntos vía correspondientes al mismo tiempo, resultando así en el movimiento cartesiano deseado.
- b) Que las funciones de articulación no sean dependientes entre sí.

A pesar de las desventajas mostradas en la Figura 4.3, el método tiene una característica que hace bastante útil su implementación: "Dada la suavidad de las funciones interpoladas para los movimientos de las articulaciones, no hay problemas con *singularidades* u otros problemas del mecanismo".

En la Figura 4.4 se muestran diversos métodos de generación de trayectorias en espacio de articulación. Cada uno tiene diferentes variantes, algunas de los cuales se estudiaron en el Capítulo 3. Se describe una técnica de aproximación polinomial como ejemplo, para familiarizarse con el problema de planeación en el espacio de articulación.

MÉTODO	VARIANTE	TÉCNICA
Polinomiales	Puntos inicial y final	
	Puntos vía	Trayectoria 3-5-3
		Trayectoria 4-3-4
		Trayectoria de 5 cúbicas
Funciones lineales con segmentos parabólicos	Puntos inicial y final	
	Puntos vía	
Splines	Generales	No racionales
		Racionales
	Locales	Splines cónicos
		Splines cúbicos

Figura 4.4 Métodos de planeación en el espacio de articulación.

### 4.2.1 Interpolación polinomial

Para mover al robot desde su posición inicial hasta el final, se puede hacer usando usando una ruta ruta directa o estableciendo puntos vía o puntos ligadura intermedios entre ambas posiciones. Finalmente la técnica de interpolación usada es la misma, y puede establecerse un algoritmo general.



4.2.4.1 Ligaduras para la planificación de trayectorias de articulación interpoladas.

Basándose en las ligaduras de una trayectoria de articulación típica dadas por Fu[12], el objetivo es seleccionar una clase de función polinomial de grado  $n$  o menor tal que la posición, velocidad y aceleración de la articulación requerida en estos puntos ligadura (inicial, despegue, asentamiento y final) se satisfacen y, en consecuencia, son continuas sobre todo el intervalo de tiempo  $[t_0, t_f]$ , ver Figura 4.5.

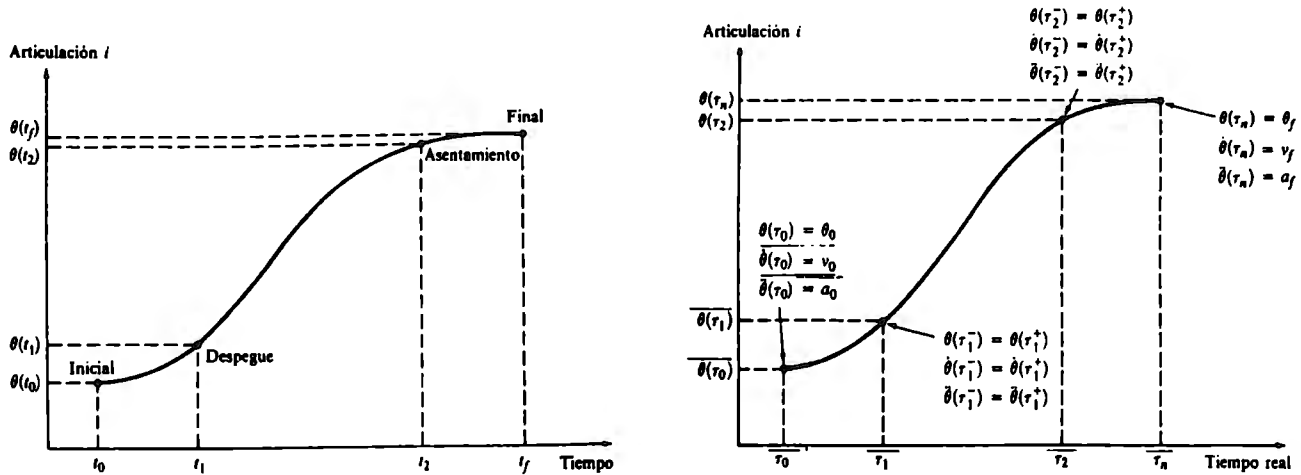


Figura 4.5 Condiciones de frontera para una trayectoria de articulación.

Posición inicial:

1. Posición [dada].
2. Velocidad [dada, normalmente nula].
3. Aceleración [dada, normalmente nula].

Posición intermedial:

4. Posición de despegue [dada].
5. Posición de despegue [continua con el segmento de trayectoria previa].
6. Velocidad [continua con el segmento de trayectoria previo].
7. Aceleración [continua con el segmento de trayectoria previo].
8. Posición de asentamiento [dada].
9. Posición de asentamiento [continua con el siguiente segmento de trayectoria].
10. Velocidad [continua con el siguiente segmento de trayectoria].
11. Aceleración [continua con el siguiente segmento de trayectoria].

Posición final:

12. Posición [dada].
13. Velocidad [dada, normalmente nula].
14. Aceleración [dada, normalmente nula].

4.2.1.1 Algoritmo

1. Establecer el conjunto de puntos ligadura o vía a través de los que debe pasar la curva a interpolar.
2. Establecer el tiempo para moverse entre dos puntos ligadura (segmento).
3. Normalizar el tiempo.

$$t = \frac{\tau - \tau_{i-1}}{\tau_i - \tau_{i-1}} \quad ; \tau \in [\tau_{i-1}, \tau_i] \quad ; t \in [0, 1]$$

t: Variable de tiempo normalizado.

$\tau$ : Tiempo real en segundos.

$\tau_i$ : Tiempo real al final del segmento de la trayectoria i-ésima.

$t_i = \tau_i - \tau_{i-1}$ : Tiempo real necesario para recorrer el segmento i-ésimo.

4. Establecer la ecuación del polinomio de interpolación para cada segmento de la trayectoria.

$$h_i(t) = c_{i,n} t^n + c_{i,n-1} t^{n-1} + \dots + c_{i,1} t + c_{i,0} \quad ; \text{segmento } i\text{-ésimo de la trayectoria.}$$

5. Determinar las ecuaciones para la velocidad y aceleración derivando la función de posición.

$$v_i(t) = \frac{1}{\tau_i - \tau_{i-1}} \frac{dh_i(t)}{dt} = \frac{1}{t_i} \frac{dh_i(t)}{dt}$$

$$a_i(t) = \frac{1}{(\tau_i - \tau_{i-1})^2} \frac{d^2h_i(t)}{dt^2} = \frac{1}{t_i^2} \frac{d^2h_i(t)}{dt^2}$$

6. Aplicar las condiciones iniciales/finales, considerando que:

$$\theta_i(t) = h_i(t)$$

$$\dot{\theta}_i(t) = v_i(t)$$

$$\ddot{\theta}_i(t) = a_i(t)$$

7. Escribir  $h_i(t)$ ,  $v_i(t)$  y  $a_i(t)$  sustituyendo las constantes por sus valores conocidos. En el último segmento hacer el cambio  $\tau = t - 1$ .
8. En los puntos de unión de segmentos (si los hay), considerar las condiciones de continuidad.

$$v_i(t_i^-) = v_i(t_i^+)$$

$$a_i(t_i^-) = a_i(t_i^+)$$

9. Igualar a cero cada una de las ecuaciones de segmentos.

10. Obtener  $\delta_i = \theta_i - \theta_{i-1} = h_i(t=1) - h_i(t=0)$  para cada segmento.

11. Compilar las ecuaciones de unión de segmentos junto con las de  $\delta_i$ .

12. Escribir las ecuaciones originales de cada segmento junto con las de sus constantes ya conocidas.

13. Expresar (11) en forma matricial:  $C \mathbf{x} = \mathbf{y}$

$\mathbf{x}$ : Vector de constantes  $a$ 's.

$\mathbf{y}$ : Vector de términos independientes.

$C$ : Matriz de coeficientes de las  $a$ 's.

14. Resolver  $\mathbf{x} = C^{-1} \mathbf{y}$ .

15. Crear la tabla de polinomios de la trayectoria, que representa la solución al problema de planeación de trayectorias

#### 4.2.1.2 Tipos de trayectorias polinomiales

Existen diferentes maneras de dividir la trayectoria de la articulación, y cada método posee propiedades diferentes. Las técnicas más comunes se describen a continuación. En las Figuras 4.6 y 4.7 se muestran las tablas de polinomios para estas técnicas mencionadas.

##### Trayectoria 4-3-4.

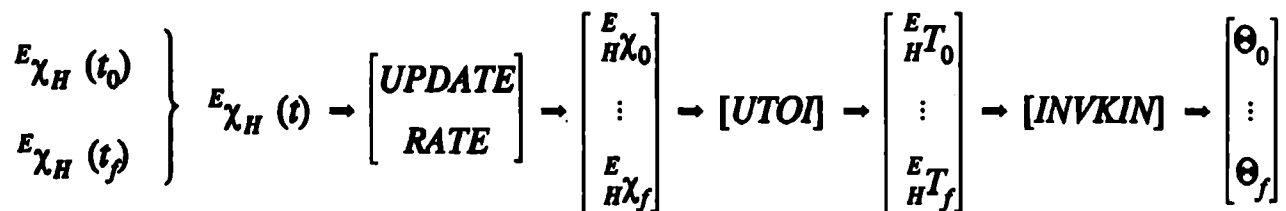
Cada articulación tiene los siguientes tres segmentos de trayectorias: el primer segmento es un polinomio de cuarto grado que especifica la trayectoria desde la posición inicial hasta la de despegue. El segundo segmento de trayectoria (o segmento medio de la trayectoria) es un polinomio de tercer grado que especifica la trayectoria desde la posición de despegue hasta la posición de asentamiento. El último segmento de la trayectoria es un polinomio de cuarto grado que especifica la trayectoria desde la posición de asentamiento hasta la posición final.

##### Trayectoria 3-5-3.

Es análoga a la trayectoria 4-3-4, pero utiliza polinomios de grados diferentes para cada segmento: un polinomio de tercer grado para el primer segmento, un polinomio de quinto grado para el segmento intermedio y un polinomio de tercer grado para el último segmento.

### 4.3 ESPACIO CARTESIANO

La planeación de trayectorias en el espacio cartesiano tiene la ventaja de ser un concepto directo; sin embargo, la estructura mecánica de los manipuladores tiene como característica ser altamente no lineal, lo que dificulta enormemente convertir restricciones torque/fuerza en el espacio de articulación en sus correspondientes límites de velocidad y aceleración en el espacio cartesiano. La descripción esquemática del problema es la siguiente:



$${}^E\chi_H = [x \ y \ z \ \alpha \ \beta \ \gamma]^T$$

<b>POLINOMIOS PARA UNA TRAYECTORIA 4 - 3 - 4</b>		
<b>Segmento 1</b>	$h_1(t) = a_{14}t^4 + a_{13}t^3 + a_{12}t^2 + a_{11}t + a_{10}$ $t \in [0, 1]$	$a_{10} = \theta_0$
		$a_{11} = (v_0 t_1)$
		$a_{12} = \left(\frac{a_0 t_1^2}{2}\right)$
		$a_{13} = \sigma$
		$a_{14} = \left(\delta_1 - v_0 t_1 - \frac{a_0 t_1^2}{2} - \sigma\right)$
$v_1 = \frac{4\delta_1}{t_1} - 3v_0 - a_0 t_1 - \frac{\sigma}{t_1}$		
$a_1 = \frac{12\delta_1}{t_1^2} - \frac{12v_0}{t_1} - 5a_0 - \frac{6\sigma}{t_1^2}$		
<b>Segmento 2</b>	$h_2(t) = a_{23}t^3 + a_{22}t^2 + a_{21}t + a_{20}$ $t \in [0, 1]$	$a_{20} = \theta_1$
		$a_{21} = (v_1 t_2)$
		$a_{22} = \left(\frac{a_1 t_2^2}{2}\right)$
		$a_{23} = \left(\delta_2 - v_1 t_2 - \frac{a_1 t_2^2}{2}\right)$
		$v_2 = \frac{3\delta_2}{t_2} - 2v_1 - \frac{a_1 t_2}{2}$
$a_2 = \frac{6\delta_2}{t_2^2} - \frac{6v_1}{t_2} - 2a_1 t_2$		
<b>Segmento 3</b>	$h_n(t) = a_{n4}t^4 + a_{n3}t^3 + a_{n2}t^2 + a_{n1}t + a_{n0}$ $t \in [0, 1]$	$a_{n0} = \theta_2$
		$a_{n1} = (v_2 t_n)$
		$a_{n2} = \left(\frac{a_2 t_n^2}{2}\right)$
		$a_{n3} = \left(-8\delta_n + 5v_2 t_n - \frac{a_2 t_n^2}{2} + 3v_2 t_n\right)$
		$a_{n4} = \left(9\delta_n - 4v_2 t_n - \frac{a_2 t_n^2}{2} - 5v_2 t_n + \frac{a_2 t_n^2}{2}\right)$
<p>donde <math>\sigma = f/g</math> y</p> <div style="display: flex; justify-content: space-between;"> <div style="width: 60%;"> <math display="block">f = 2\delta_1 \left(4 + \frac{2t_n}{t_2} + \frac{2t_n}{t_1} + \frac{3t_2}{t_1}\right) - \frac{\delta_2 t_1}{t_2} \left(3 + \frac{t_n}{t_2}\right) + \frac{2\delta_n t_1}{t_n} -</math> <math display="block">- v_0 t_1 \left(6 + \frac{6t_2}{t_1} + \frac{4t_n}{t_1} + \frac{3t_n}{t_2}\right) - v_2 t_1 - a_0 t_1 t_n \left(\frac{5}{3} + \frac{t_1}{t_2} + \frac{2t_1}{t_n} + \frac{5t_2}{2t_n}\right) + a_f t_1 t_n</math> </div> <div style="width: 35%; text-align: right;"> <math display="block">g = \frac{t_n}{t_2} + \frac{2t_n}{t_1} + 2 + \frac{3t_2}{t_1}</math> </div> </div>		

Figura 4.6 Polinomios para una trayectoria 4-3-4.

POLINOMIOS PARA UNA TRAYECTORIA 3 - 5 - 3		
Segmento 1	$h_1(t) = a_{13}t^3 + a_{12}t^2 + a_{11}t + a_{10}$	$a_{10} = \theta_0$
		$a_{11} = (v_0 t_1)$
		$a_{12} = \left(\frac{u_0 t_1^2}{2}\right)$
		$a_{13} = \left(\delta_1 - v_0 t_1 - \frac{a_0 t_1^2}{2}\right)$
	$v_1 = \frac{\dot{h}_1(1)}{t_1} = \frac{3\delta_1}{t_1} - 2v_0 - \frac{a_0 t_1}{2}$	
$a_1 = \frac{\ddot{h}_1(1)}{t_1^2} = \frac{6\delta_1}{t_1^2} - \frac{6v_0}{t_1} - 2a_0$		
Segmento 2	$h_2(t) = a_{25}t^5 + a_{24}t^4 + a_{23}t^3 + a_{22}t^2 + a_{21}t + a_{20}$	$a_{20} = \theta_1$
		$a_{21} = (v_1 t_2)$
		$a_{22} = \left(\frac{a_1 t_2^2}{2}\right)$
		$a_{23} = \left(10\delta_2 - 6v_1 t_2 - 4v_2 t_2 - \frac{3a_1 t_2^2}{2} + \frac{a_2 t_2^2}{2}\right)$
		$a_{24} = \left(-15\delta_2 + 8v_1 t_2 + 7v_2 t_2 + \frac{3a_1 t_2^2}{2} - a_2 t_2^2\right)$
		$a_{25} = \left(6\delta_2 - 3v_1 t_2 - 3v_2 t_2 - \frac{a_1 t_2^2}{2} + \frac{a_2 t_2^2}{2}\right)$
	$v_2 = \frac{\dot{h}_2(1)}{t_2} = \frac{3\delta_2}{t_n} - 2v_f + \frac{a_f t_n}{2}$	
$a_2 = \frac{\ddot{h}_2(1)}{t_2^2} = \frac{-6\delta_2}{t_n^2} + \frac{6v_f}{t_n} - 2a_f$		
Segmento 3	$h_3(t) = a_{n3}t^3 + a_{n2}t^2 + a_{n1}t + a_{n0}$	$a_{n0} = \theta_2$
		$a_{n1} = \left(3\delta_n - 2v_f t_n + \frac{a_f t_n^2}{2}\right)$
		$a_{n2} = (-3\delta_n + 3v_f t_n - a_f t_n^2)$
		$a_{n3} = \left(\delta_n - v_f t_n + \frac{a_f t_n^2}{2}\right)$

Figura 4.7 Polinomios para una trayectoria 3-5-3.

La planeación de la ruta de la trayectoria se hace en espacio cartesiano. Derivando la ecuación de la curva  ${}^E\chi_H(t)$  se obtienen las ecuaciones para velocidad y aceleración. Muestreando las tres curvas con un *update rate* adecuado, se discretiza la trayectoria obteniendo las tablas de puntos consigna en el espacio cartesiano.

La etapa final del problema es aplicar cinemática inversa, Jacobiano y Hessiano a las tablas de posición, velocidad y aceleración respectivamente para encontrar los puntos consigna para el control en espacio de articulación.

Existen diversos métodos para generar trayectorias cartesianas, por ejemplo:

- a) Movimiento cartesiano en línea recta
- b) Rutas complejas
  - Splines globales
  - Splines locales

Para familiarizarse con el problema, se describe la planeación del movimiento cartesiano en línea recta.

### 4.3.1 Movimiento cartesiano en línea recta

El movimiento más simple de un cuerpo es el rectilíneo. La descripción del movimiento rectilíneo de la herramienta de un robot en espacio cartesiano también es muy simple. La dificultad de este problema radica en la cantidad de trabajo de cómputo necesario para mapear la trayectoria cartesiana al espacio de articulación.

Para robots articulados una alternativa para resolver el problema es planificar una ruta con polinomios cúbicos en espacio de articulación, lo cual nos genera una trayectoria aproximadamente recta en el espacio cartesiano.

El algoritmo es el siguiente:

1. Establecer como interfase con el usuario, la representación ángulo-eje equivalente.

$${}^E\chi_A = \begin{bmatrix} {}^E P_{A_{ORG}} \\ {}^E K_A \end{bmatrix} ; \quad {}^E K_A = \begin{bmatrix} K_x \\ K_y \\ K_z \end{bmatrix} \theta_{E A}$$

2. Considerar los problemas de la representación interna relacionados con la representación del usuario.

- a) La representación ángulo-eje equivalente no es única:

$$({}^E K_A, \theta_{E A}) = ({}^E K_A, \theta_{E A} + n 360^\circ) ; n = 1, 2, \dots$$

b) En un movimiento  $\{A\} \rightarrow \{B\}$  la rotación debe ser mínima:

$$| {}^E K_B - {}^E K_A | \rightarrow \text{Mínima}$$

3. Mapear los puntos del espacio cartesiano al espacio de articulación utilizando la cinemática inversa del manipulador.

4. Generar la ruta en espacio de articulación

a) Generar splines para cada gdl con el método de la Sección 4.2.1.1.

b) Los tiempos de los segmentos son los mismos para cada gdl, para asegurar el movimiento en línea recta cartesiana.

c) La aceleración en los segmentos para cada gdl puede ser diferente debido a la relación entre tiempo y distancia recorrida en cada segmento.

### 4.3.2 Problemas geométricos con rutas cartesianas

Se mencionó anteriormente que aún cuando la generación de la trayectoria se haga en espacio cartesiano, el control siempre se hace en coordenadas de articulación. Las rutas cartesianas pueden presentar diversos problemas relacionados con el espacio de trabajo y las singularidades del mecanismo del robot, Craig [9].

#### 4.3.2.1 Puntos intermedios inalcanzables

Mientras que la localización inicial del manipulador y el punto de destino se sitúan dentro del espacio de trabajo, es posible que no todos los puntos de la ruta lo estén, ver Figura 4.8. Moverse de A a B en espacio de articulación puede no representar problema, pero un movimiento cartesiano en línea recta requiere pasar por puntos inalcanzables por el efector final del robot.

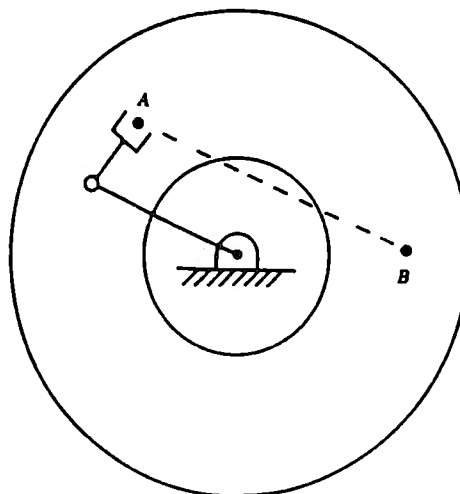


Figura 4.8 Puntos intermedios inalcanzables.

### 4.3.2.2 Altas velocidades de articulación cerca de singularidades

Existen rutas descritas en el espacio cartesiano que son imposibles de seguir por el robot, debido a la proximidad a una configuración singular, ver Figura 4.9. Un manipulador que sigue una ruta en línea recta y se aproxima a una configuración singular del mecanismo, puede provocar que una más de sus velocidades de articulación tiendan a incrementarse infinitamente. En una configuración singular, el manipulador pierde uno o más grados de libertad. Debido a los límites de los actuadores, esta situación usualmente resulta en una desviación de la ruta deseada, pudiendo ocasionar daños al robot mismo o a los objetos del entorno.

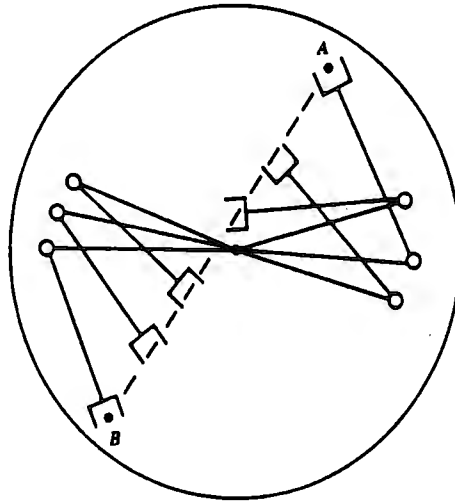


Figura 4.9 Altas velocidades de articulación cerca de las singularidades.

### 4.2.2.3 Puntos inicial y final alcanzables en diferentes soluciones

Debido a las restricciones mecánicas de sus mecanismos, pocos manipuladores tienen rangos de movimiento mayores o cercanos a los  $360^\circ$  en sus articulaciones. Esto puede significar que la ruta a seguir por el robot tenga puntos que no puedan alcanzarse, sino en un montaje diferente del mismo, ver Figura 4.10. En esta situación, el sistema de planeación de trayectorias debe detectar el problema antes de iniciar el cálculo de la ruta.

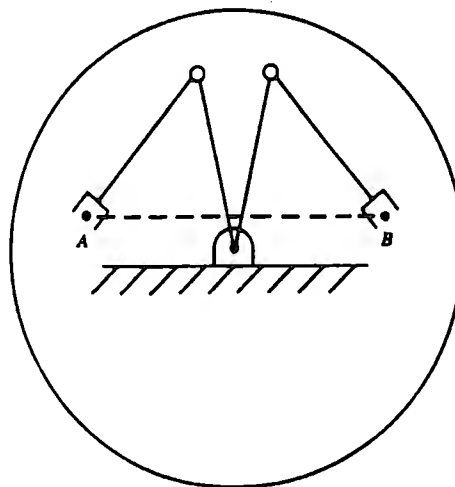


Figura 4.10 Puntos inicial y final alcanzables en diferentes soluciones.



### 4.4 APROXIMACIÓN CARTESIANA EN ESPACIO DE ARTICULACIÓN

Una alternativa para la planeación de movimientos en línea recta es la aproximación de rutas cartesianas en el espacio de la articulación. Chang-Lee[8] derivan un algoritmo para generar rutas de articulación que se aproximan a cualquier curva en el espacio cartesiano proporcionando puntos ligadura apropiados. La descripción esquemática del problema es la siguiente:

$$\left. \begin{matrix} {}^E\chi_H \\ {}^E\chi_H \end{matrix} \right\} \xrightarrow{{}^E\chi_H} \begin{bmatrix} \text{UPDATE} \\ \text{RATE} \end{bmatrix} \rightarrow \begin{bmatrix} {}^E\chi_{H_0} \\ \vdots \\ {}^E\chi_{H_j} \end{bmatrix} \xrightarrow{\begin{bmatrix} \text{APROX} \\ \text{KNOTS} \end{bmatrix}} \begin{bmatrix} {}^E_H\chi_{l_1} \\ {}^E_H\chi_{l,j} \\ {}^E_H\chi_{l,j, \text{center}} \end{bmatrix} \xrightarrow{[\text{INVKIN}]} \begin{bmatrix} \theta_0 \\ \vdots \\ \theta_f \end{bmatrix} \xrightarrow{\begin{bmatrix} \text{MQSI} \\ \text{R-DINAMIC} \end{bmatrix}} \begin{cases} \theta(t) \\ \frac{d\theta(t)}{dt} \\ \frac{d^2\theta(t)}{dt^2} \end{cases}$$

$${}^E\chi_H = [x \ y \ z \ \alpha \ \beta \ \gamma]^T$$

La interpolación con los puntos ligadura se hace en espacio cartesiano. Con una frecuencia de muestreo adecuada se discretiza la función, y usando el algoritmo de Chang-Lee[8] se obtiene un nuevo conjunto de puntos ligadura. Estos puntos se mapean al espacio de articulación, en donde las rutas de articulación se obtienen interpolando mediante splines cuárticos simples (SQSI) o modificados (MSQI). Los esquemas se extienden para incluir las condiciones de movimiento en mínimo tiempo sujetas a las restricciones de velocidad, aceleración y sacudimiento (jerk) en cada articulación.

El funcionamiento del método se puede observar en la Figura 4.11.

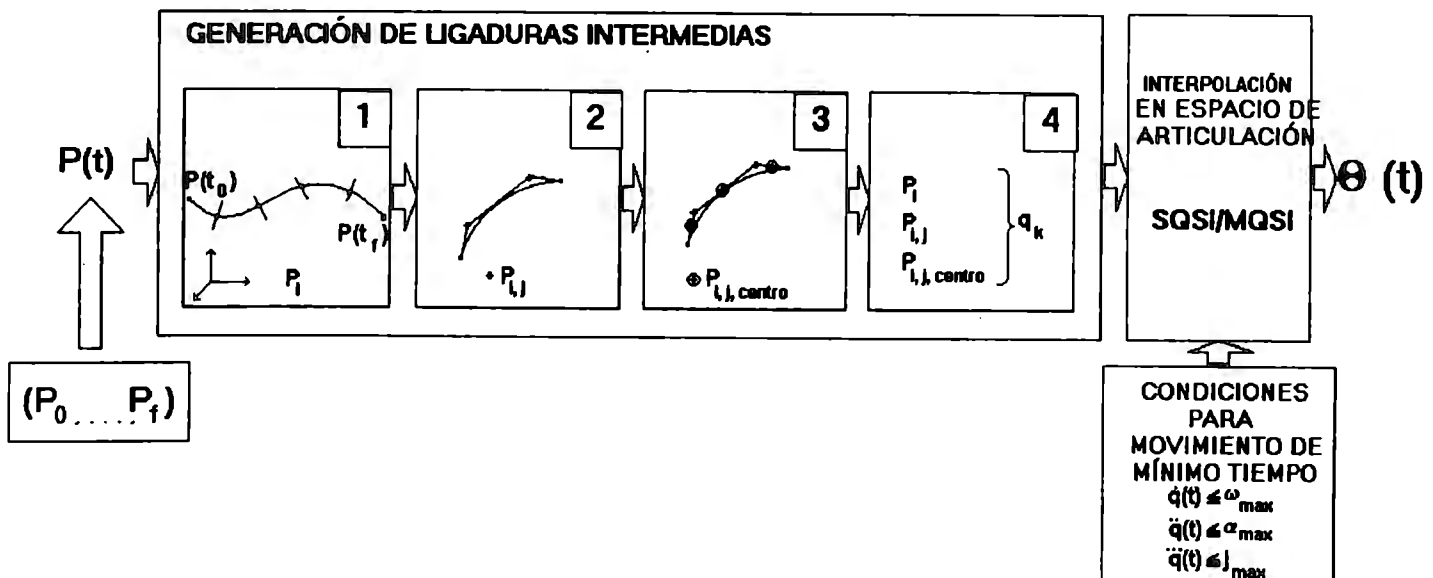


Figura 4.11 Aproximación de ruta cartesiana en espacio de articulación.

## 4 GENERACIÓN DE TRAYECTORIAS

1. Generación secuencial de ligaduras intermedias a lo largo de cualquier curva. Se requiere que el error entre la trayectoria deseada y la ruta aproximada sea menor que la tolerancia preespecificada.
  - a) Dividir la ruta en un conjunto de subdivisiones apropiado ( $P_i$ ).
  - b) Aproximación de cada segmento con un arco de círculo, considerando un error de curvatura mínimo.
  - c) Aproximar cada arco con un conjunto de segmentos de línea recta ( $P_i, P_{ij}$ ).
  - d) Generar ligaduras intermedias apropiadas a lo largo de cada segmento de línea ( $P_{ij, center}$ ).
2. Transformar las ligaduras cartesianas en ligaduras de articulación  $q_k$  mediante cinemática inversa.
3. Interpolarse la ruta en espacio de articulación considerando las restricciones de velocidad, aceleración y jerk.

### 4.5 TOPOLOGÍA DEL ESPACIO DE CONFIGURACIÓN

La representación del espacio de configuración o c-espacio se refiere a una relación pseudogeométrica donde las coordenadas de localización del extremo, de la muñeca del robot, son las variables de articulación. El c-espacio se usa principalmente en la planeación de trayectorias libres de obstáculos para robots manipuladores. La tarea se realiza en dos partes:

- a) Construcción de una base de datos para representar las restricciones geométricas, mapeando los obstáculos desde el espacio de trabajo del robot al c-espacio.
- b) Búsqueda de la solución mediante una técnica heurística.

La transformación de puntos (obstáculos) desde el espacio de trabajo del robot al espacio de configuración, está fuertemente relacionado con la cinemática inversa del mecanismo.

El problema de transformación, para un robot plano de dos grados de libertad, se describe en la Figura 4.12; donde con la función cinemática inversa se mapea la ruta del espacio cartesiano al de configuración.

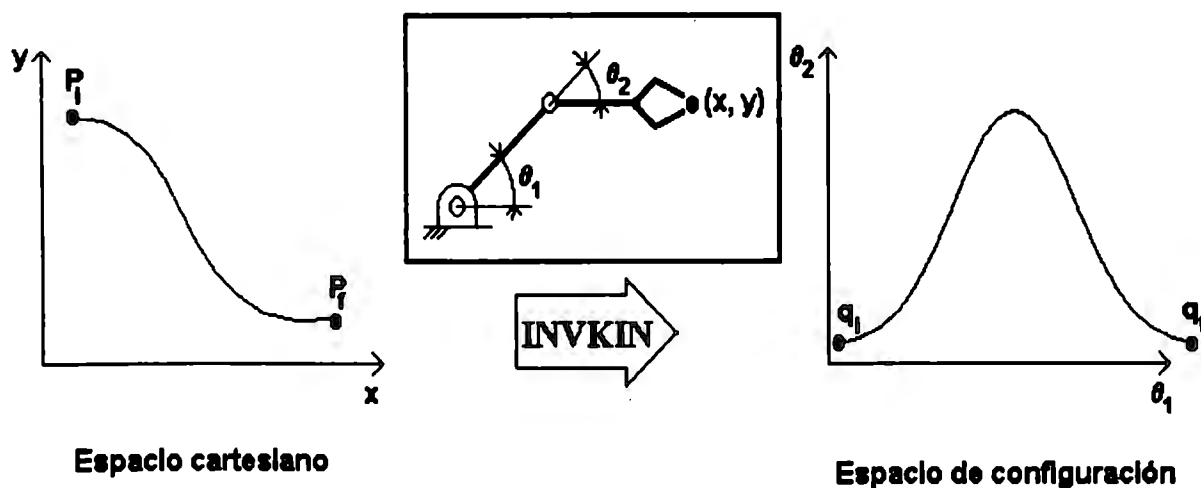


Figura 4.12 El problema del mapeo entre espacios.

## 4 GENERACIÓN DE TRAYECTORIAS

El mapeo es complicado debido a la multiplicidad de soluciones posibles para  $\theta_1$  y  $\theta_2$  arrojadas por la función INVKIN. Se hace necesario aplicar un criterio de selección, generalmente dicho criterio es el de mínimo desplazamiento.

En las secciones 4.2 y 4.3, se trataron los métodos de planeación de trayectorias en espacio de articulación y espacio cartesiano por separado. Existe una posible solución a esta incompatibilidad de espacios, que se puede plantear en los siguientes puntos:

1. La cinemática directa representa la función de transformación (mapeo) del espacio de articulación al espacio cartesiano.
2. La cinemática inversa mapea puntos o rutas del espacio cartesiano al espacio de articulación.
3. Maciejewski-Fox[11] demuestran que para un robot plano de dos gdl el mapeo de líneas rectas y circunferencias del espacio cartesiano al de articulación tiene un patrón definido, ver Figura 4.13.

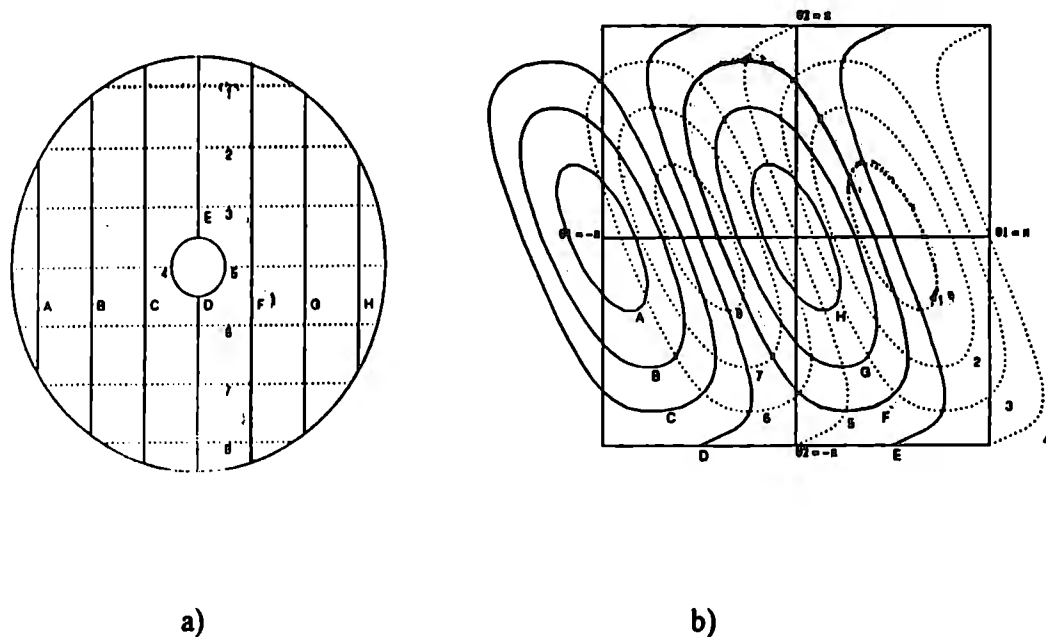


Figura 4.13 Transformación de posición del efector final en espacio cartesiano (a), al espacio de articulación (b). Los números y letras corresponden a los mismos elementos en ambos espacios.

4. Se pueden establecer relaciones análogas entre curvas en ambos espacios, de manera que no sea necesario hacer el mapeo para determinar la forma que tendrá la curva trazada en uno de los espacios, en el otro.
5. Conociendo la topología del espacio de configuración, pueden trazarse trayectorias en este espacio con la seguridad de obtener movimientos cartesianos perfectamente conocidos.

#### 4 GENERACIÓN DE TRAYECTORIAS

El problema de planeación de trayectorias de acuerdo a la topología del espacio de configuración puede esquematizarse como sigue:

$$\begin{array}{l}
 {}^E\chi_H(t_0) \rightarrow [UTOI] + [INVKIN] \rightarrow \Theta(t_0) \\
 \vdots \\
 {}^E\chi_H(t_f) \rightarrow [UTOI] + [INVKIN] \rightarrow \Theta(t_f)
 \end{array}
 \left. \vphantom{\begin{array}{l} \\ \\ \end{array}} \right\} \rightarrow \left[ \begin{array}{c} \text{INTERPOLACION} \\ \text{CON FUNCIONES} \\ \text{CARACTERIZADAS} \end{array} \right] \rightarrow \left\{ \begin{array}{l} \Theta(t) \\ \frac{d\Theta(t)}{dt} \\ \frac{d^2\Theta(t)}{dt^2} \end{array} \right.$$

$${}^E\chi_H = [x \ y \ z \ \alpha \ \beta \ \gamma]^T$$

Los puntos de ligadura se transforman al c-espacio, y con ellos se interpola usando alguna función "caracterizada" según la topología del espacio de configuración del tobot. Por ejemplo, en la Sección 4.3.1 se planteó el uso de polinomios cúbicos en c-espacio para obtener trayectorias aproximadamente rectas en espacio cartesiano.

La planeación de trayectorias de acuerdo a la topología del c-espacio conjuga las ventajas de la planeación en espacios separados, y elimina casi todas las desventajas:

- a) Permite conocer exactamente la posición y orientación del efector final en el espacio de trabajo.
- b) Conocer la forma de la ruta cartesiana (vía mapeo entre espacios), previene los peligros de colisión.
- c) Usa cinemática inversa solamente para puntos ligadura.
- d) La planeación se hace directamente en las coordenadas de control.

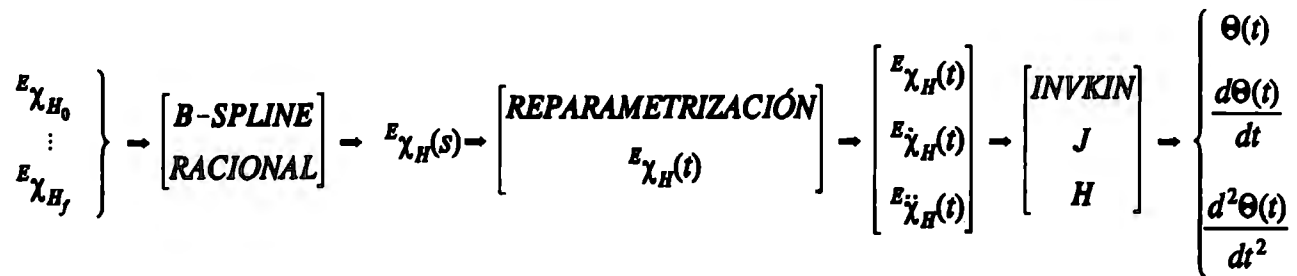
Las desventajas tienen que ver con la complejidad de la forma de las rutas planeadas y con los problemas en el espacio cartesiano enunciados en la Sección 4.3.3.

Actualmente, el problema principal, es la dificultad para caracterizar las curvas entre ambos espacios. Como se mencionó anteriormente, el mapeo está estrechamente ligado a la cinemática inversa (y por lo tanto a la anatomía del robot), con todas las dificultades que esto implica, ver Sección 2.3.

Tratar a fondo este problema no está dentro de los objetivos de esta tesis, sin embargo, se consideró de suma importancia plantearlo claramente para futuras investigaciones.

## 4.6 GENERACIÓN DE TRAYECTORIAS USANDO B-SPLINES RACIONALES

Una vez analizados los principales métodos para generar trayectorias para robots, se puede ahora plantear la metodología para planear trayectorias usando *B-splines racionales*. la representación esquemática del problema es la siguiente:



$${}^E\chi_H = [x \ y \ z \ \alpha \ \beta \ \gamma]^T$$

Al conjunto de puntos  ${}^E\chi_H$  que definen posición y orientación del efector final, se aplica el método de Bézier desarrollado en el Capítulo 3 para obtener una trayectoria cartesiana compleja. Como el B-spline resultante es una función parametrizada por la longitud de arco  $s$ , es necesario reparametrizarlo con respecto al tiempo, para así poder derivarlo y obtener las funciones de velocidad y aceleración cartesianas. Las tres ecuaciones (posición, velocidad y aceleración) son muestreadas y transformadas al espacio de articulación utilizando la cinemática inversa, el Jacobiano y el Hessiano respectivamente; para el robot particular que efectuará el trabajo siguiendo la trayectoria generada.

Para detallar la metodología se adopta el esquema de la Sección 2.2, para el cálculo del movimiento de un manipulador.

### 4.6.1 Metodología para la planeación de la trayectoria

#### 1. Interface con el usuario.

Para establecer una forma sencilla y útil para que el usuario defina los datos para el sistema de planeación de trayectorias, es necesario auxiliarse de la Figura 4.14. Tomando el caso de un manipulador plano de tres grados de libertad, se sabe que son necesarios tres datos para definir completamente la posición y orientación de su efector final: coordenada  $y$ , coordenada  $x$  y ángulo de orientación  $\varphi$ . Analizando la figura encontramos que el problema puede ser resuelto usando dos puntos:

- a) Punto de localización espacial de la punta de la herramienta del robot:  $\chi_{H_i}$ .
- b) Punto para trazar el vector de orientación  $k$  de la herramienta:  $\chi_{W_i}$ . este punto debe localizarse sobre el eje longitudinal de la herramienta o gripper del robot.

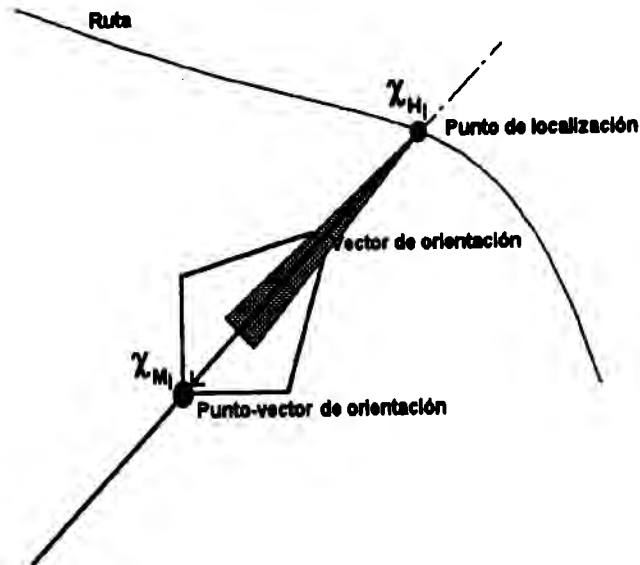


Figura 4.14 Definición de la posición y orientación del efector de un manipulador plano de tres *gdl*.

En este caso el punto para el vector  $k$  puede estar en la herramienta misma, o en el extremo de la muñeca del manipulador. Para robots espaciales y/o redundantes, la selección del punto-vector puede ser útil en extremo para simplificar la solución de la cinemática inversa y de la planeación de trayectorias; como se verá en el punto siguiente.

En la Figura 4.15 se observa como, con dos puntos, se define completamente la posición y orientación del manipulador plano considerado; a lo largo de una trayectoria cualquiera. En la figura se identifica la enorme ventaja de adoptar esta forma de definición de datos: Se pueden trazar dos trayectorias cuasiparalelas, dividiendo el problema de planeación de trayectorias en dos partes:

- Trayectoria de posicionamiento para las articulaciones menores ( 1 a 2 para robots planos y 1 a 3 para robots espaciales) → Trayectoria de la muñeca.
- Seudotrayectoria de orientación para las articulaciones mayores (  $j \geq 3$  para robots planos y  $j \geq 4$  para robots espaciales) → Trayectoria de la herramienta.

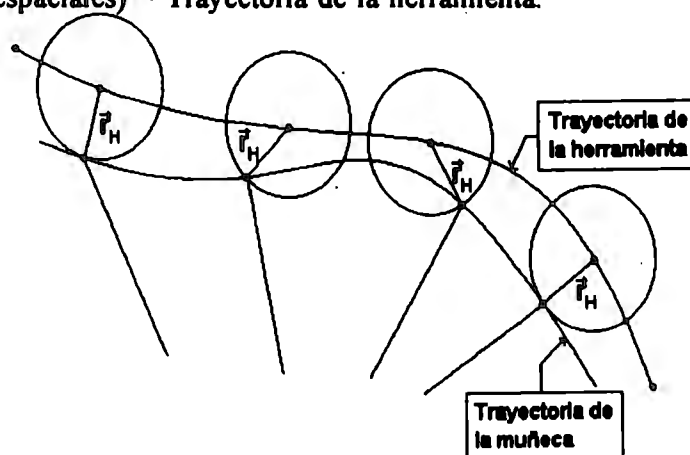


Figura 4.15 Trayectoria de herramienta y trayectoria de muñeca.

## 2. Representación interna.

Es necesario establecer una representación interna congruente con los siguientes aspectos del sistema:

- El módulo de planeación de la ruta geométrica.
- El modelo cinemático y dinámico.
- El módulo de obtención de la historia temporal para posición, velocidad y aceleración.

En la implementación del generador de la ruta, se estableció que la información que requerida son dos conjuntos de puntos ligadura para la obtención de los B-splines de posición y orientación. El manejo de los datos para posición no involucra problemas; la mayoría de los módulos son capaces de trabajar con puntos expresados con coordenadas cartesianas. Sin embargo, para trabajar con el modelo cinemático la orientación debe expresarse en términos de convenciones de ángulos (ángulos fijos o ángulos de Euler), o por medio de matrices de rotación.

Para transformar la forma de representar la orientación necesitamos analizar la Figura 4.16. Cada configuración del robot esta determinada por un par de puntos; y la orientación es dada por el vector que une ambos puntos.

$$\vec{k}_i = \vec{r}_{H_i} - \vec{r}_{M_i}$$

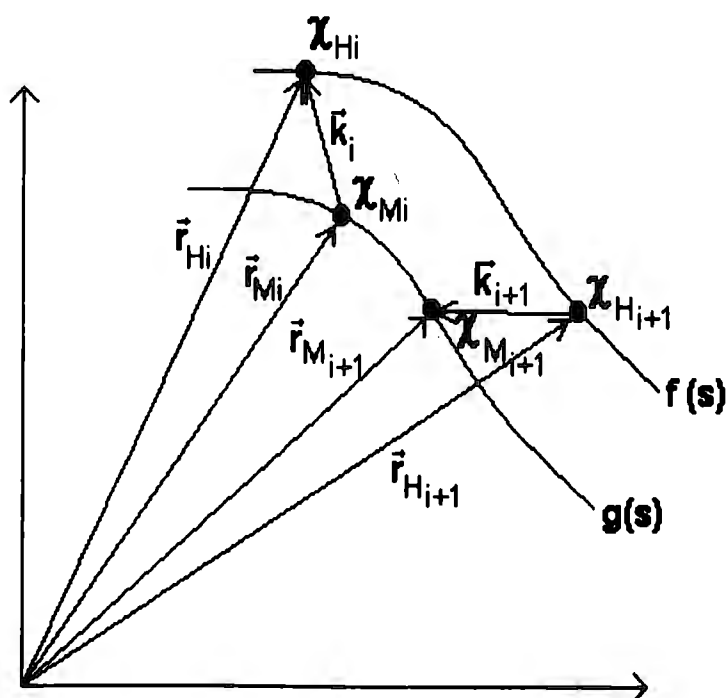


Figura 4.16 Vectores de posición y orientación.

Habiendo determinado el vector de orientación en función de la información de los B-splines, ahora debemos traducirla a una forma manejable, esta forma es una matriz de rotación. Auxiliándonos de la Figura 4.17 obtenemos la representación de ángulos fijos.

Datos:  $\vec{k}(x, y, z)$

$$\gamma = \tan^{-1} \left( \frac{x}{y} \right) \rightarrow R_z(\gamma)$$

$$\alpha = \tan^{-1} \left( \frac{y}{z} \right) \rightarrow R_x(\alpha)$$

$$\beta = \tan^{-1} \left( \frac{x}{z} \right) \rightarrow R_y(\beta)$$

De acuerdo con lo anterior, la matriz de rotación es:

$${}^A_B R_{XYZ}(\alpha, \beta, \gamma) = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta c\gamma - s\alpha c\gamma & c\alpha s\beta s\gamma + s\alpha c\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix}$$

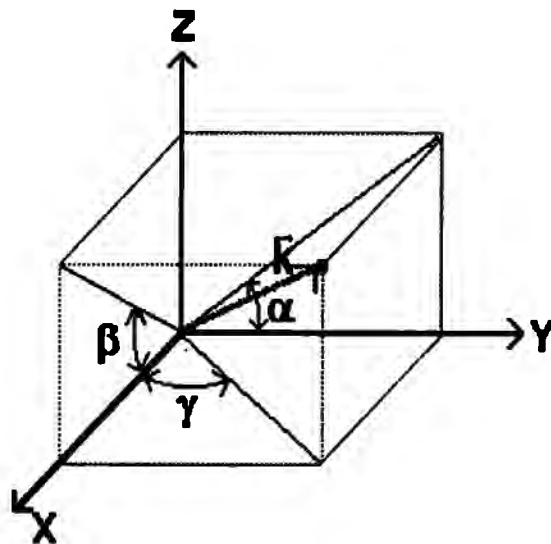


Figura 4.16 Representación de la orientación con ángulos fijos ( $\alpha$ ,  $\beta$ ,  $\gamma$ ).



### 3. Planeación de la ruta

La planeación de la ruta geométrica fue desarrollada en el Capítulo 3. Como se recordará, los únicos datos que solicita los programas *BezierPoints* y *RBezierPoints* son: el conjunto de puntos a interpolar y la definición que se necesita para tener una curva suave, para los dos; y el factor de acercamiento hacia el polígono de definición para el segundo.

### 4. Historia temporal

Una vez que se ha definido la ruta, el problema restante es ajustar los intervalos entre las ligaduras de manera que no se excedan las velocidades y aceleraciones máximas de cada articulación. Para ello se necesita obtener las curvas de velocidad y aceleración derivando con respecto al tiempo la ecuación del B-spline.

Para obtener ecuaciones de velocidad y aceleración completamente definidas es necesario considerar algunas condiciones de frontera, como se planteó en la Sección 4.2.4.1. Normalmente se considera que un robot inicia su movimiento desde una posición de reposo ( $v_o = a_o = 0$ ), y alcanza su meta haciendo alto total ( $v_f = a_f = 0$ ).

El problema en esta etapa lo podemos resumir con el siguiente esquema:

$$\begin{array}{c} \left[ \begin{array}{l} B\text{-spline} \\ \text{racional} \end{array} \right] \rightarrow f(s) \rightarrow \left[ \begin{array}{c} \text{REPARAMETRIZACIÓN} \\ f(t) \end{array} \right] \rightarrow [\text{INVKIN}] \rightarrow \Theta(t) \\ \\ \frac{df(s)}{dt} \rightarrow \dots \rightarrow \frac{d\Theta(t)}{dt} \\ \\ \frac{d^2f(s)}{dt^2} \rightarrow \dots \rightarrow \frac{d^2\Theta(t)}{dt^2} \end{array}$$

El punto de partida es la función de posición en espacio cartesiano parametrizada con la longitud de arco  $s$ . El primer objetivo es obtener una función de posición en espacio de articulación parametrizada en el tiempo. Además hay que obtener las funciones para velocidad y aceleración cartesianas y, de ser posible, mapearlas al espacio de articulación.

Con la información de la caracterización de tareas, ver Capítulo 2, se pueden establecer dos casos para la planeación de trayectorias usando B-splines, de acuerdo con sus atributos temporales.

## a) Consigna tiempo.

La mayoría de los robots industriales poseen funciones de movimiento para establecer tiempos de recorrido. Los intervalos de tiempo pueden ajustarse por escalamiento lineal o no lineal. El escalamiento lineal es fácil de implementar, pero el no lineal resulta en un movimiento más rápido del robot.

Datos:

$$P = (P_0, P_1, \dots, P_n)$$

$$[t_0, t_f] \rightarrow (t_0, t_1, \dots, t_f)$$

Restricciones:

$v$ : Velocidad a lo largo de la ruta.

$a$ : Aceleración.

Resultado:

$$\chi = f(s) \rightarrow \chi = f(t) \rightarrow \chi(t), \dot{\chi}(t)$$

## b) Consigna velocidad.

Es común programar movimientos de robots industriales dejando que el sistema de control utilice una velocidad por default. Esta función se comprende mejor en las máquinas de control numérico, cuando se le indica realizar avances cortando material. En tales casos, la velocidad de avance es función de los parámetros mecánicos de corte (material a maquinar, dimensiones de la pieza, material de la herramienta, etc.); y el tiempo de recorrido queda relegado a un segundo término. En algunas aplicaciones de robots se presentan casos similares.

Datos:

$$P = (P_0, P_1, \dots, P_n)$$

$$v = (v_0, v_1, \dots, v_n)$$

Restricciones:

$v$ : Velocidad a lo largo de la ruta.

$a$ : Aceleración.

Resultado:

$$\chi = \chi(s) \rightarrow d\chi(s) / dt$$

Analizando el problema de planeación de trayectorias con B-splines, lo identificamos como un problema con consigna de velocidad; por lo tanto, no es necesario encontrar una función explícita que relacione posición y tiempo.

**Obtención de la función de velocidad cartesiana.**

Si expresamos a la función de posición como  $f(s)$ , para derivarla en el tiempo se debe usar la regla de la cadena:

$$\frac{df(s)}{dt} = \frac{df(s)}{ds} \cdot \frac{ds}{dt}$$

Del cálculo vectorial, ver Hsu[28], se conoce que

$$\frac{dg(s)}{ds} = g'(s) = T$$

$$T = \frac{f'(t)}{|f'(t)|}$$

y de la Sección 3.1.4 conocemos la parametrización de una función por longitud de arco

$$s(t) = \int |f'(t)| dt$$

de donde

$$\frac{ds(t)}{dt} = |f'(t)|$$

por lo tanto la ecuación de la velocidad cartesiana de una trayectoria cuya ruta es un B-spline es:

$$\boxed{\frac{df(s)}{dt} = \frac{df(t)}{dt} = f'(t) = v(t)}$$

Analizando la función, se observa que tiene dos componentes obtenidos al aplicar la regla de la cadena. El primero define la razón de cambio de la curva al ir variando el parámetro  $s$ , y es un vector que representa la dirección instantánea de la velocidad. El segundo es un número que representa la magnitud de la velocidad para un valor específico de  $s$  y en un instante de tiempo dado.

Esto significa que podemos armar la función de velocidad derivando la función de posición (B-spline) con respecto a  $s$  y multiplicandola por una magnitud de velocidad cartesiana determinada por la tarea que el robot realizará.

**Obtención de la función de aceleración cartesiana.**

Si derivamos la ecuación de velocidad recién obtenida, obtendremos la función de aceleración.

$$\begin{aligned}
 a(t) &= \frac{d}{dt} f'(t) = \frac{d}{dt} \left[ \frac{df(s)}{ds} \cdot \frac{ds}{dt} \right] \\
 &= \frac{d}{dt} \left[ \frac{df(s)}{ds} \right] \cdot \frac{ds}{dt} \\
 &= \left( \frac{d}{ds} \left[ \frac{df(s)}{ds} \right] \cdot \frac{ds}{dt} \right) \cdot \frac{ds}{dt} \\
 &= \frac{d^2}{ds^2} f(s) \cdot \left( \frac{ds}{dt} \right)^2
 \end{aligned}$$

Con la última ecuación comprobamos que nuestro problema efectivamente tiene como consigna a la velocidad. La función de aceleración involucra al perfil de velocidad definido por la tarea, y no requiere de perfil de aceleración alguno. En el capítulo siguiente se implementa el método desarrollado, y se realizan algunas corridas de ejemplo.

# CAPÍTULO 5

## SISTEMA GENERADOR DE TRAYECTORIAS COMPLEJAS

### 5.1 DESCRIPCIÓN DEL PROBLEMA

En el Capítulo 3 se desarrolló un método para generar rutas complejas usando B-splines racionales. En el Capítulo 4 se completó el método para generar trayectorias desarrollando las matemáticas de la historia temporal para velocidad y aceleración. En este capítulo se usarán todas las herramientas implementadas para armar un sistema de generación de trayectorias complejas cartesianas usando B-splines racionales.

El funcionamiento general del sistema se ilustra en la Figura 5.1. Para describir tal funcionamiento, en este capítulo se soluciona un problema de planeación.

Se tiene un robot SCARA como el mostrado en la Figura 5.2a, con un área de trabajo como la mostrada en la Figura 5.2b. Se necesita que el robot describa una trayectoria pasando por los 10 puntos señalados en la misma figura.

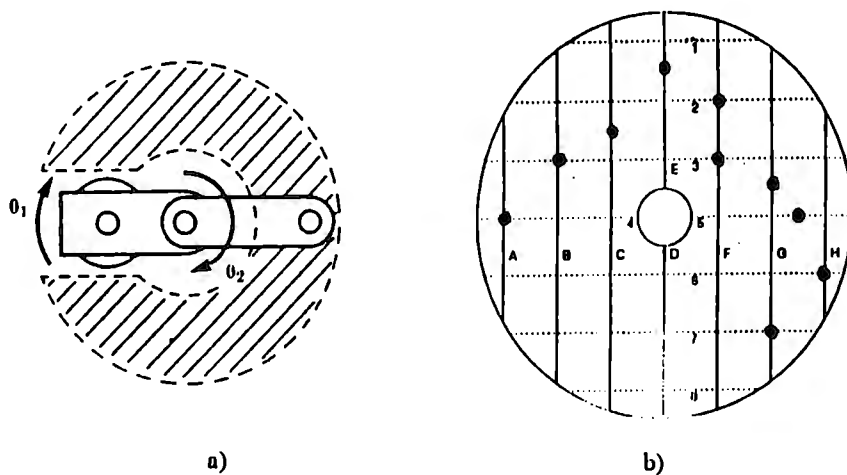


Figura 5.2 Descripción del problema.

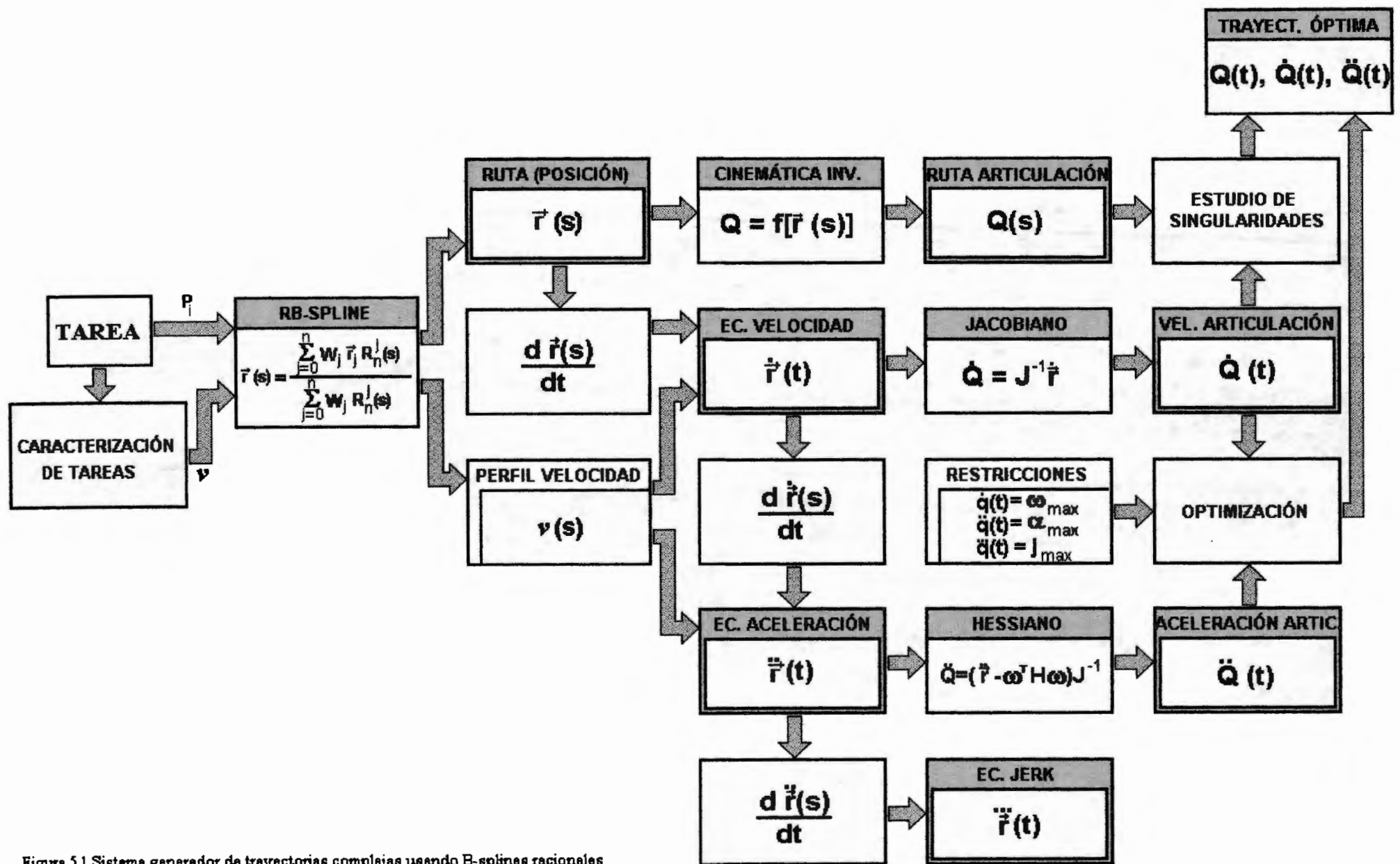


Figura 5.1 Sistema generador de trayectorias complejas usando B-splines racionales

## 5.2 PLANEACIÓN DE LA RUTA

Para planear la ruta con el método de Bézier implementado en el Capítulo 3, únicamente se necesita plantear el vector de puntos ligadura, la resolución de la curva y el factor de acercamiento. El problema es encontrar la ruta que mejor se aproxime, por lo cual se proponen dos alternativas. Los datos son:

Vector de puntos:

$$P = [ (-3, 0), (-2, 1), (-1, 1.5), (0, 2.5), (1, 2), (1, 1), (2, 0.5), (2.5, 0), (3, -1), (2, -2) ]$$

Resolución:

$$ds = 0.01$$

Factores de acercamiento:

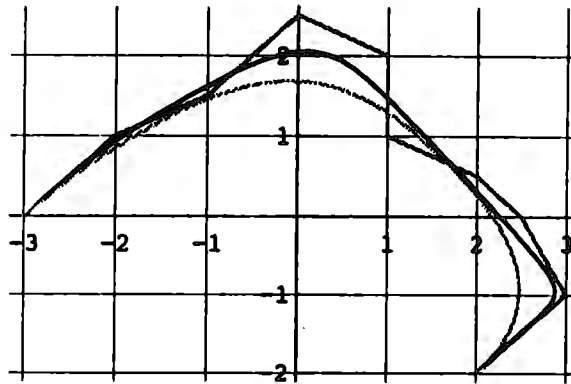
$$V_1 = 0.1$$

$$V_2 = 0.02$$

La solución del problema se muestra en la Figura 5.3.

`In[56]: =`

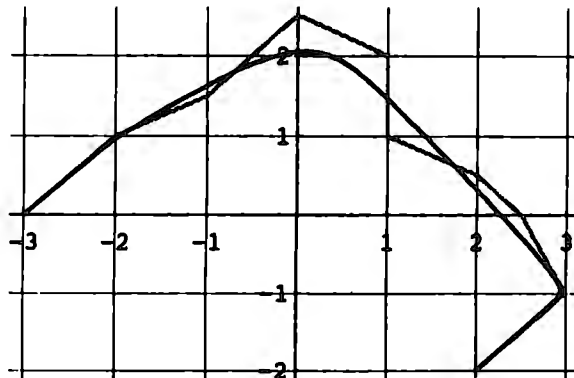
`PlotRBezier[PA,.01,.1]`



Ruta para un factor de acercamiento  $V = 0.1$

`In[75]: =`

`PlotRBezier[PA,.01,.02]`



Ruta para un factor de acercamiento  $V = 0.02$

Figura 5.3 Rutas alternativas para la trayectoria buscada.

### 5.3 GENERACIÓN DE LA TRAYECTORIA

Una vez obtenida la ruta geométrica, se pueden obtener las funciones de velocidad y aceleración con el método desarrollado en el Capítulo 4; para hacerlo se necesitan además de los datos de la ruta, un conjunto de velocidades asociadas a los puntos ligadura. Las velocidades son:

$$v = [ 0, 0.9, 0.6, 0.4, 0.5, 0.9, 0.7, 0.6, 0.5, 0 ]$$

Los resultados se muestran en la Figura 5.4

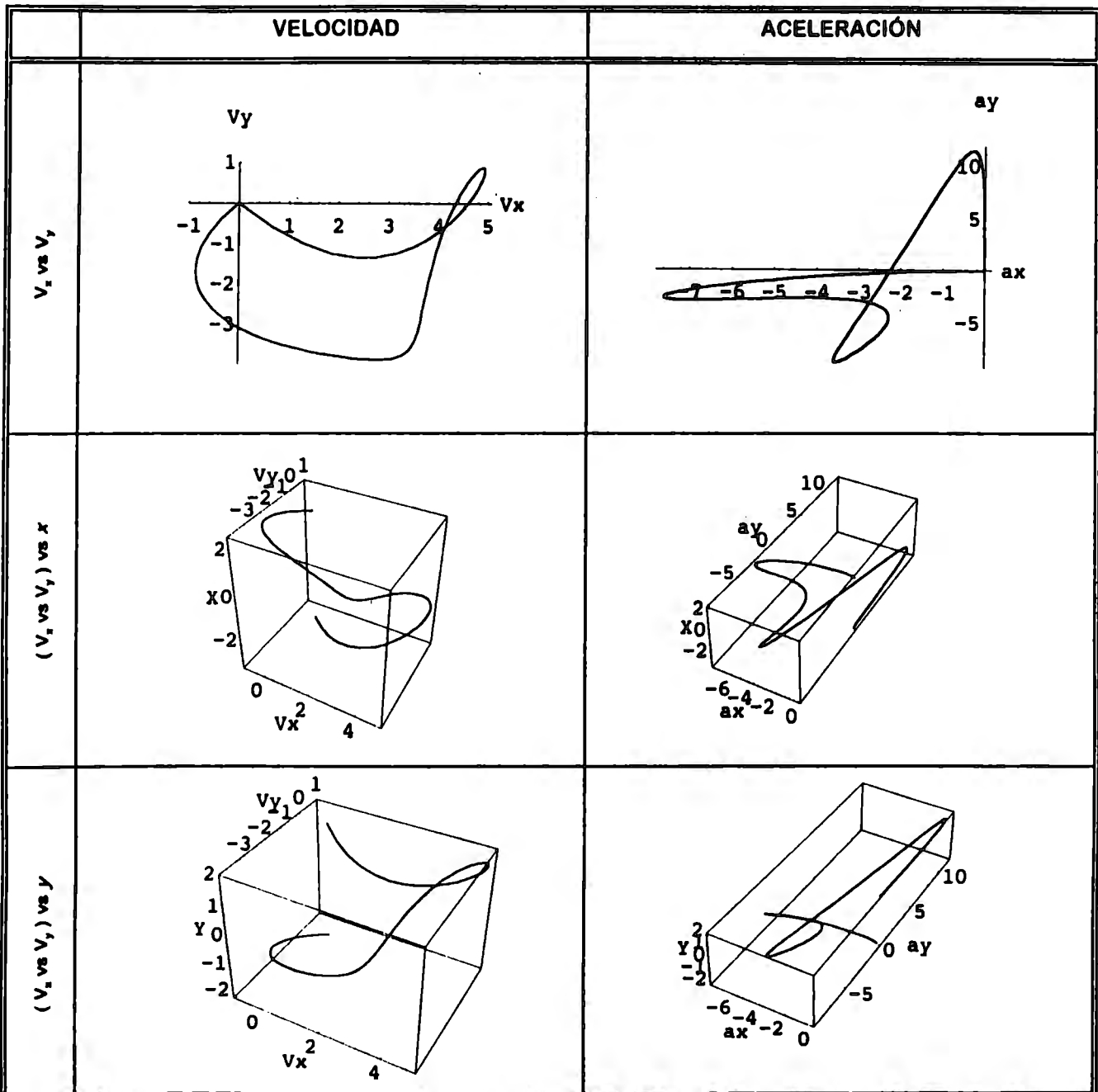


Figura 5.4 Perfiles de velocidad y aceleración para la trayectoria.



### 5.4 CONSIDERACIONES CINEMÁTICAS Y DINÁMICAS

Obtenidas la ruta geométrica de la trayectoria con sus correspondientes perfiles de velocidad y aceleración, ahora se debe introducir el modelo dinámico y cinemático del robot SCARA, para validar la trayectoria encontrada.

Los parámetros dinámicos y cinemáticos se obtuvieron utilizando un programa elaborado por alumnos de la Maestría en Sistemas de Manufactura del ITESM-CEM. El programa es simbólico/numérico, está escrito en lenguaje Mathematica y utiliza como base el *modelo general de robots con coeficientes de influencia* desarrollado por Vásquez[25]. El diagrama general de funcionamiento se muestra en la Figura 5.6. La descripción detallada se hace en las tesis de los autores del programa mencionados en las referencias.

La información obtenida del programa de modelado para el robot SCARA se muestra en la Figura 5.5.

```

MatrixRot[ listaalf, listath, listaan, listadn ]
Cos[th1] -Sin[th1] 0
Sin[th1] Cos[th1] 0
0 0 1

Cos[th2] -Sin[th2] 0
Sin[th2] Cos[th2] 0
0 0 1

Pos[vectorX, vectorZ, listaan, listadn, Ron, rf]
L1 Cos[th1] + L2 Cos[th1 + th2]
L1 Sin[th1] + L2 Sin[th1 + th2]
0

JacobianoT[vectorZ, posicion, vectorR, listath]
{-(L1 Sin[th1]) - L2 Sin[th1 + th2], -(L2 Sin[th1 + th2])}
{L1 Cos[th1] + L2 Cos[th1 + th2], L2 Cos[th1 + th2]}
{0, 0}

HessiaR[vectorZ, listath]
0 0
0 0
0 0

0 0
0 0
0 0

HessianoT[Gt, vectorZ, listath]
-(L1 Cos[th1]) - L2 Cos[th1 + th2] -(L2 Cos[th1 + th2])
-(L1 Sin[th1]) - L2 Sin[th1 + th2] -(L2 Sin[th1 + th2])
0 0

-(L2 Cos[th1 + th2]) -(L2 Cos[th1 + th2])
-(L2 Sin[th1 + th2]) -(L2 Sin[th1 + th2])
0 0

Preaccion[Masa, Gt, vectorZ, Hr, Hessi]
0 -(L1 L2 (m1 + m2) Sin[th2])
L1 L2 (m1 + m2) Sin[th2] 0

-(L1 L2 (m1 + m2) Sin[th2]) -(L1 L2 (m1 + m2) Sin[th2])
0 0

Virtual[Masa,Rota,vectorZ,Gravedad,Carga,CentroM,product= lon
gtudes];
{-(gutil L1 Cos[th1]) - 0.5 g L1 m1 Cos[th1] - 0.5 g L1 m2 Cos[th1]
gutil L2 Cos[th1 + th2] - 0.5 g L2 m2 Cos[th1 + th2],
-(gutil L2 Cos[th1 + th2]) - 0.5 g L2 m2 Cos[th1 + th2]}

TotalT[React, Itotal]
2 2
{alfa[2] (2 + L2 m1 + L2 m2 + L1 L2 m1 Cos[th2] + L1 L2 m2
Cos[th2]) +
2 2 2 2
alfa[1] (2 + L1 m1 + L2 m1 + L1 m2 + L2 m2 + 2 L1 L2 m1
Cos[th2] +
2 L1 L2 m2 Cos[th2]) - L1 L2 (m1 + m2) omega[1] omega[2]
Sin[th2] -
L1 L2 (m1 + m2) omega[2] (omega[1] + omega[2]) Sin[th2],
2 2
(2 + L2 m1 + L2 m2) alfa[2] +
    
```

Figura 5.5 Resultados del modelado dinámico y cinemático del robot SCARA.

## 5 SISTEMA GENERADOR DE TRAYECTORIAS COMPLEJAS

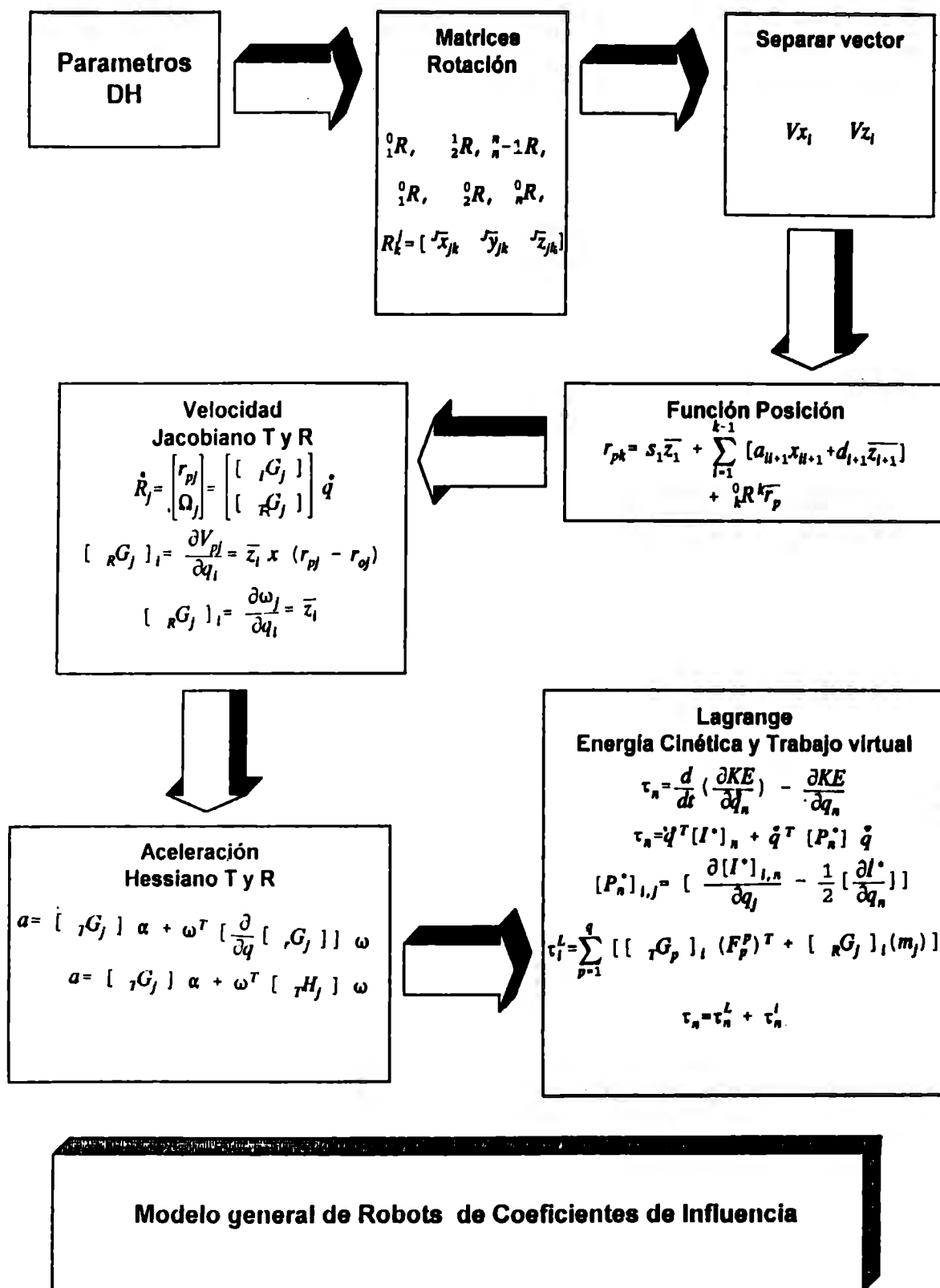


Figura 5.6 Diagrama de flujo del programa de modelado general de robots con coeficientes de influencia.

## 5 SISTEMA GENERADOR DE TRAYECTORIAS COMPLEJAS

Además de la información del modelo dinámico y cinemático, se necesita la cinemática inversa robot. Para obtener esta información, se escribió una pequeña rutina en Mathematica para calcular tanto la cinemática directa como la inversa del manipulador SCARA. Los resultados se muestran en la Figuras 5.7 y 5.8.

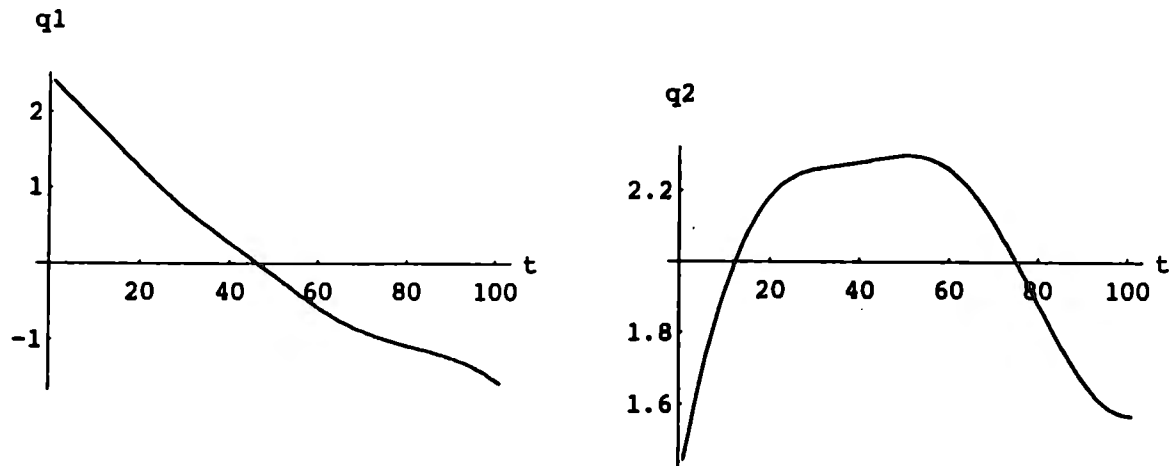


Figura 5.7 Rutas de articulación para generar la trayectoria especificada obtenida vía cinemática inversa.

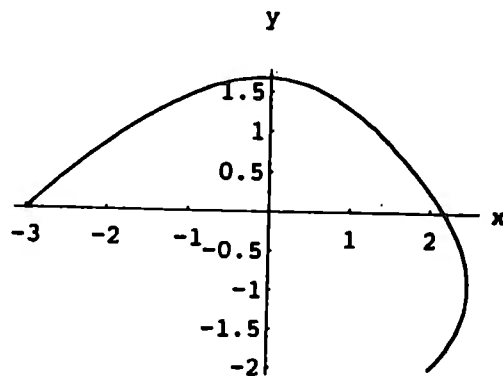


Figura 5.8 Cinemática directa de las rutas de articulación.

# CAPÍTULO 6

---

---

## CONCLUSIONES

---

---

### 6.1 CONCLUSIONES

---

Para terminar la presente tesis se hace una recopilación de los logros alcanzados y de los objetivos cumplidos.

1. Se sintetizó la información sobre la problemática de diversos tópicos relacionados con la robótica, para llegar a establecer el "*problema fundamental de la robótica*", al cual se considera como integrado por tres partes:
  - a) Caracterización de tareas (task planning).
  - b) Planeación de la trayectoria (path planning).
  - c) Seguimiento de la trayectoria (path tracking).
2. Se estableció una metodología para realizar la caracterización de tareas para robots. Tal metodología se dividió en dos partes:
  - a) Desarrollo de las herramientas: *Perfiles de desempeño*, *Redes de Petri* para tareas y definición de matrices *QFD de criterios*.
  - b) Caracterización de tareas específicas: Aplicación de las herramientas a casos específicos.
3. Se reformuló el problema de planeación de trayectorias óptimas (OPPP), para incluir las etapas de elección de interfase para el usuario, representación interna, planeación de la ruta, historia temporal y Fault Tolerance.
4. Se implementó un método computacional simbólico/numérico en Mathematica para generar curvas suaves y complejas usando la teoría de *Splines racionales de Bézier*. Conjuntamente se desarrollaron herramientas matemáticas y gráficas para manipular directamente los polinomios de Bernstein-Bézier. Además se desarrolló una técnica simple pero funcional para evaluar los parámetros de forma de los B-splines racionales, y poder hacer aproximaciones controladas al polígono de definición.

5. Se identificaron las ventajas derivadas de desarrollar una técnica de planeación de trayectorias de acuerdo a la topología del espacio de configuración. y se realizaron algunos primeros intentos para *caracterizar* curvas entre espacios.
6. Se desarrolló una metodología completa para generar trayectorias cartesianas usando B-splines racionales, adoptando el esquema del OPPP antes planteado. La implementación se hizo en forma numérica/simbólica en Mathematica.

De acuerdo con estos logros, las conclusiones sobre la tesis son las siguientes:

1. Los objetivos planteados para la tesis, enunciados en la Sección 1.6, se cumplieron en su totalidad. Sin embargo, debido a la complejidad del problema y al tiempo disponible, el grado de desarrollo de algunos puntos tuvo que limitarse al mínimo, ver Figura 6.1.

OBJETIVO	GRADO DE DESARROLLO	OBSERVACIONES
a) Establecer una metodología para caracterizar tareas de robot.	100 %	La metodología planteada requiere de una etapa de desarrollo de herramientas, al final de la cual se obtendrán los criterios para la elección del método de planeación de trayectorias más conveniente.
b) Establecer un método para generar curvas complejas y suaves para trayectorias de robot.	100 %	El método seleccionado utiliza B-splines racionales cuya exactitud depende del número de puntos dados para su generación, de la posición de los puntos y de los parámetros de forma o pesos. Tal método fue desarrollado por Vázquez[17].
c) Implementar un modelo computacional del método de generación de curvas complejas y suaves.	100 %	La implementación del método se hizo creando una jerarquía de funciones, de manera que se tiene toda una infraestructura para manipulación matemática y/o gráfica de los polinomios de Bernstein-Bézier.
d) Integrar las restricciones dinámicas del robot de planeación de trayectorias complejas y suaves.	80 %	La integración total no se completó debido a la falta de un método general para la cinemática inversa, y a que es necesario construir una base de datos con la información de robots y/o actuadores comerciales.
e) Generar un modelo computacional para el método de planeación de trayectorias.	90 %	El modelo computacional fue definido completamente. Para la implementación, se desarrollaron la mayor parte de las funciones necesarias para ejecutar de manera global el método.

2. El uso del modelo computacional para generar rutas complejas es bastante simple, sólo requiere que se le proporcione el vector de puntos a través de los cuales se requiere que pase la trayectoria, la definición necesaria del B-spline y el parámetro de acercamiento de la curva hacia su polígono de definición.

3. El método para generar trayectorias complejas y suaves está hecho para trabajar fuera de línea. Las razones son dos principalmente: la primera es que el tipo de tareas que requieren tales trayectorias no demandan planeación en tiempo real, la segunda es que la información manipulada para obtener trayectorias cinemática y dinámicamente óptimas es bastante considerable.
4. El modelo computacional para generar trayectorias complejas puede usarse en dos maneras diferentes:
  - a) Para selección del tipo de robot más adecuado para realizar una tarea específica.  
Se genera la trayectoria completa y se utilizan los parámetros arrojados por el modelo cinemático y dinámico referentes a velocidad máxima, torque máximo, rango de movimiento de las articulaciones, etc.; para elegir el modelo de robot que pueda ejecutarla.
  - b) Para programar una trayectoria de trabajo para robots específicos.  
En este caso, los parámetros arrojados por el modelo dinámico y cinemático del robot para toda la trayectoria son evaluados para asegurar que estén dentro de sus capacidades. En caso de que haya valores inalcanzables (de posición, velocidad y aceleración), es necesario modificar algunos de los parámetros usados para especificar la trayectoria, preferentemente las velocidades de trabajo.

# APENDICE

## A.1 CORRIDAS DEL PROGRAMA GENERADOR DE TRAYECTORIAS

### EJEMPLO 1.

Puntos:  $P = \{ \{0,1\}, \{2,2\}, \{3,6\}, \{4,3\}, \{6,2\} \}$

Velocidades:  $v = \{ \{0,5,1,5,0\} \}$

Pos=Bn[4]

$$(1 - s)^4 P_j[0] + 4 (1 - s)^3 s P_j[1] + 6 (1 - s)^2 s^2 P_j[2] + 4 (1 - s) s^3 P_j[3] +$$

Vel= D[Pos/. s-> s[t],t]

$$-4 P_j[0] (1 - s(t))^3 s'(t) + 4 P_j[1] (1 - s(t))^3 s'(t) -$$

$$12 P_j[1] (1 - s(t))^2 s(t) s'(t) + 12 P_j[2] (1 - s(t))^2 s(t) s'(t) -$$

$$12 P_j[2] (1 - s(t)) s(t)^2 s'(t) + 12 P_j[3] (1 - s(t)) s(t)^2 s'(t) -$$

3

3

Acel=D[Vel,t]/. s"[t] -> 0

$$12 P_j[0] (1 - s(t))^2 s'(t)^2 - 24 P_j[1] (1 - s(t))^2 s'(t)^2 + 12 P_j[2] (1 - s(t))^2$$

$$24 P_j[1] (1 - s(t)) s(t) s'(t)^2 - 48 P_j[2] (1 - s(t)) s(t) s'(t)^2 +$$

$$24 P_j[3] (1 - s(t)) s(t) s'(t)^2 + 12 P_j[2] s(t)^2 s'(t)^2 - 24 P_j[3] s(t)^2 s'(t)^2$$

$$12 P_j[4] s(t)^2 s'(t)^2$$

Pj[0]={0,1};

Pj[1]={2,2};

Pj[2]={3,6};

Pj[3]={4,3};

Pj[4]={6,2};

pp=BezierPoints[{{0,1},{2,2},{3,6},{4,3},{6,2}},.05]

{0, 1}, {0.3855, 1.24012}, {0.744, 1.5419}, {1.0785, 1.87962}, {1.392, 2.2304},  
 {1.6875, 2.57422}, {1.968, 2.8939}, {2.2365, 3.17512}, {2.496, 3.4064},  
 {2.7495, 3.57912}, {3., 3.6875}, {3.2505, 3.72862}, {3.504, 3.7024}, {3.7635, 3.  
 {4.032, 3.4619}, {4.3125, 3.26172}, {4.608, 3.0224}, {4.9215, 2.75812}, {5.256,

In[43]: =

Vel

Out[43]=

$$\{8 (1 - s[t])^3 s'[t] + 12 (1 - s[t])^2 s[t] s'[t] + 12 (1 - s[t]) s[t]^2 s'[t] + 8 s[t]^3 s'[t], 4 (1 - s[t])^3 s'[t] + 48 (1 - s[t])^2 s[t] s'[t] - 36 (1 - s[t]) s[t]^2 s'[t] - 4 s[t]^3 s'[t]\}$$

In[44]: =

Acel

Out[44]=

$$\{-12 (1 - s[t])^2 s'[t]^2 + 12 s[t]^2 s'[t]^2, 36 (1 - s[t])^2 s'[t]^2 - 168 (1 - s[t]) s[t] s'[t]^2 + 24 s[t]^2 s'[t]^2\}$$

In[45]: =

vv=BezierPoints[{0,.5,1,.5,0},.05]

Out[45]=

{0, 0.0995125, 0.1962, 0.287513, 0.3712, 0.445312, 0.5082, 0.558512, 0.5952, 0.61  
0.625, 0.617513, 0.5952, 0.558512, 0.5082, 0.445312, 0.3712, 0.287512, 0.1962,  
0.0995125, -4.44089 10<sup>-16</sup>}

In[46]: =

Vperfil=Table[Vel /. {s[t]-> i/20, s'[t]-> vv[[i+1]]}, {i,0,20}]

Out[46]=

{{0, 0}, {0.739378, 0.548264}, {1.3577, 1.27059}, {1.86021, 2.00008}, {2.2569, 2.  
2.56055, 2.97803}, {2.78494, 3.07563}, {2.94336, 2.88109}, {3.04742, 2.41889},  
{3.10609, 1.74478}, {3.125, 0.9375}, {3.10609, 0.0892306}, {3.04742, -0.704717},  
{2.94336, -1.35635}, {2.78494, -1.79496}, {2.56055, -1.97607}, {2.2569, -1.88867  
{1.86021, -1.56019}, {1.3577, -1.0587}, {0.739378, -0.491542},  
{-3.55271 10<sup>-15</sup>, 1.77636 10<sup>-15</sup>}



In[47]:=

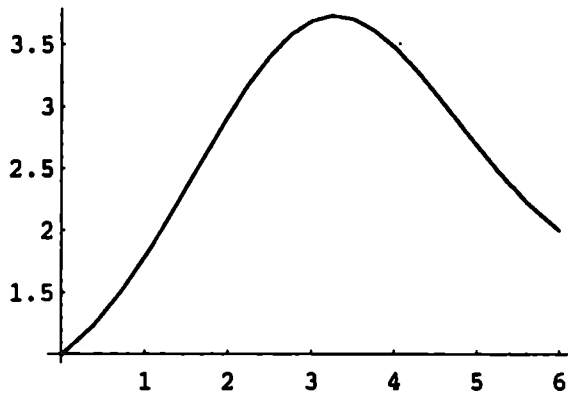
```
Aperfil=Table[Acel /. {s[t]-> i/20, s'[t]-> vv[[i+1]]},{i,0,20}]
```

Out[47]=

```
{0, 0}, {-0.10695, 0.24331}, {-0.369547, 0.549701}, {-0.694373, 0.424063},
{-0.992084, -0.396834}, {-1.18982, -1.93346}, {-1.23968, -3.99798},
{-1.12297, -6.26056}, {-0.850231, -8.33227}, {-0.457586, -9.84954}, {0., -10.546
{0.457586, -10.3071}, {0.850231, -9.1825}, {1.12297, -7.38353}, {1.23968, -5.237
{1.18982, -3.12328}, {0.992084, -1.38892}, {0.694373, -0.270309}, {0.369547, 0.1
{0.10695, 0.136361}, {2.36658 10-30, 4.73317 10-30 }
```

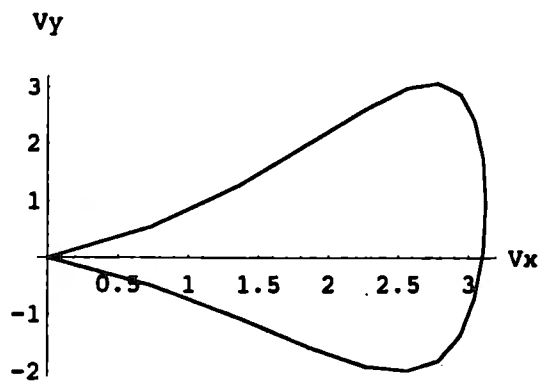
In[48]:=

```
PlotBezier[pp,.05]
```



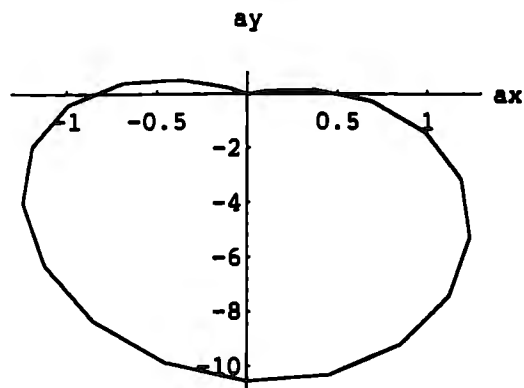
In[49]:=

```
ListPlot[Vperfil,PlotJoined->True, AxesLabel->{"Vx","Vy"}];
```



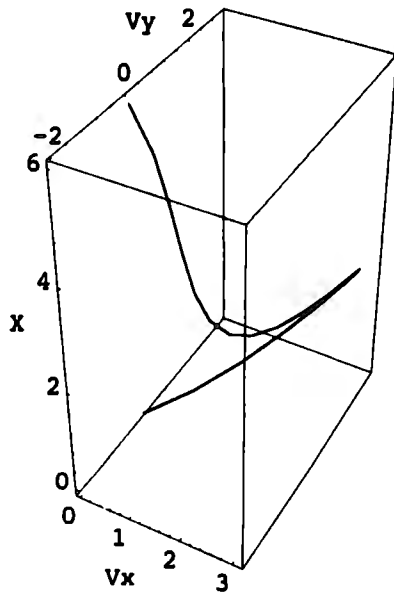
In[50]: =

```
ListPlot[Aperfil, PlotJoined->True, AxesLabel->{"ax", "ay"}];
```



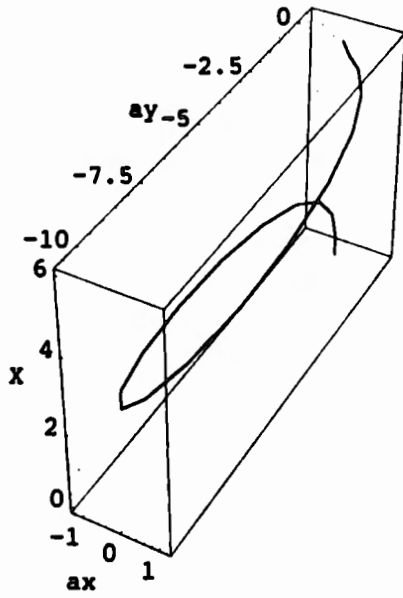
In[51]: =

```
Newperfil=N[Table[Append[Vperfil[[i]],pp[[i,1]]],{i,1,21}];  
Show[Graphics3D[Line[Newperfil],PlotRange->All, Axes->True,AxesLabel->{"Vx","Vy","X"}];
```



In[53]: =

```
Newperfil=N[Table[Append[Aperfil[[i]],pp[[i,1]]],{i,1,21}];
Show[Graphics3D[Line[Newperfil],PlotRange->All, Axes->True,AxesLabel->{"ax","ay","X"}];
```



**EJEMPLO 1. ROBOT PLANO DE DOS GRADOS DE LIBERTAD**

**CINEMÁTICA INVERSA**

**Seguimiento de la trayectoria.**

*In[64]:* =

l1=4;

l2=3;

*In[66]:* =

P={{0,1},{2,2},{3,6},{4,3},{6,2}};

*In[67]:* =

pp=BezierPoints[P,.05]

*Out[67]:* =

```
{0, 1}, {0.3855, 1.24012}, {0.744, 1.5419}, {1.0785, 1.87962},
{1.392, 2.2304}, {1.6875, 2.57422}, {1.968, 2.8939},
{2.2365, 3.17512}, {2.496, 3.4064}, {2.7495, 3.57912},
{3., 3.6875}, {3.2505, 3.72862}, {3.504, 3.7024},
{3.7635, 3.61162}, {4.032, 3.4619}, {4.3125, 3.26172},
{4.608, 3.0224}, {4.9215, 2.75812}, {5.256, 2.4859},
{5.6145, 2.22562}, {6., 2.}
```

*In[68]:* =

Q=InvKin[pp]

*Out[68]:* =

```
{1.5708, 3.14159}, {0.688758, 2.90184}, {0.361606, 2.73771},
{0.219576, 2.57922}, {0.164798, 2.4244}, {0.152639, 2.27432},
{0.160805, 2.13099}, {0.176923, 1.99676}, {0.193411, 1.87402},
{0.205241, 1.76494}, {0.208967, 1.67106}, {0.202355, 1.59295},
{0.184323, 1.52971}, {0.155007, 1.47868}, {0.115878, 1.43531},
{0.0698258, 1.39335}, {0.0212123, 1.3452}, {-0.0241205, 1.28229},
{-0.0588629, 1.19514}, {-0.0740343, 1.07223},
```

In[69]:=

pq=Kin[Q]

Out[69]=

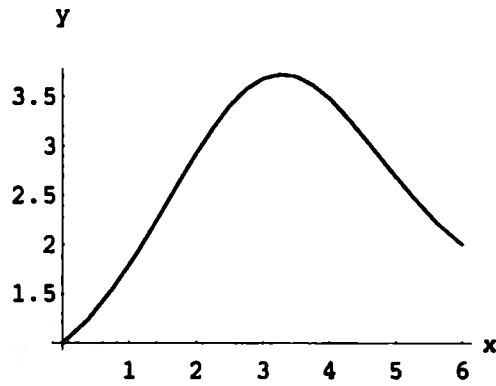
```

      -16
{{-3.06287 1016, 1.}, {0.3855, 1.24012}, {0.744, 1.5419},
 {1.0785, 1.87962}, {1.392, 2.2304}, {1.6875, 2.57422},
 {1.968, 2.8939}, {2.2365, 3.17512}, {2.496, 3.4064},
 {2.7495, 3.57912}, {3., 3.6875}, {3.2505, 3.72862},
 {3.504, 3.7024}, {3.7635, 3.61162}, {4.032, 3.4619},
 {4.3125, 3.26172}, {4.608, 3.0224}, {4.9215, 2.75812},
 {5.256, 2.4859}, {5.6145, 2.22562}, {6., 2.}}

```

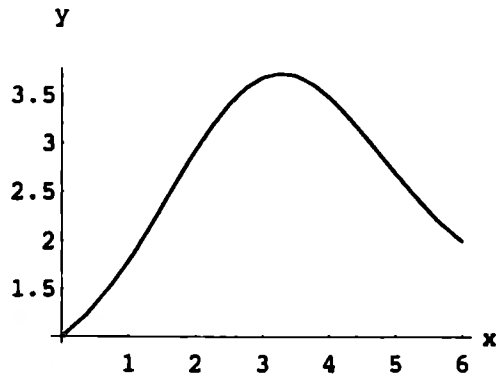
In[70]:=

ListPlot[pp,PlotJoined->True, AxesLabel->{"x","y"}];



In[71]:=

ListPlot[pq,PlotJoined->True, AxesLabel->{"x","y"}];



In[72]:=

```
q1=Table[{i,Q[[i,1]]},{i,1,21}]
q2=Table[{i,Q[[i,2]]},{i,1,21}]
```

Out[72]=

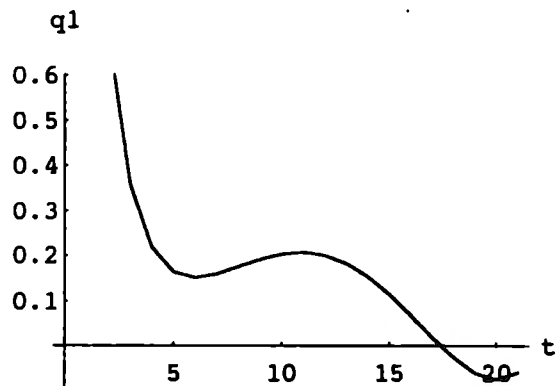
- ```
{1, 1.5708}, {2, 0.688758}, {3, 0.361606}, {4, 0.219576},
{5, 0.164798}, {6, 0.152639}, {7, 0.160805}, {8, 0.176923},
{9, 0.193411}, {10, 0.205241}, {11, 0.208967}, {12, 0.202355},
{13, 0.184323}, {14, 0.155007}, {15, 0.115878}, {16, 0.0698258},
{17, 0.0212123}, {18, -0.0241205}, {19, -0.0588629},
{20, -0.0740343}, {21, -0.0575629}}
```

Out[73]=

- ```
{1, 3.14159}, {2, 2.90184}, {3, 2.73771}, {4, 2.57922},
{5, 2.4244}, {6, 2.27432}, {7, 2.13099}, {8, 1.99676},
{9, 1.87402}, {10, 1.76494}, {11, 1.67106}, {12, 1.59295},
{13, 1.52971}, {14, 1.47868}, {15, 1.43531}, {16, 1.39335},
{17, 1.3452}, {18, 1.28229}, {19, 1.19514}, {20, 1.07223},
{21, 0.895665}}
```

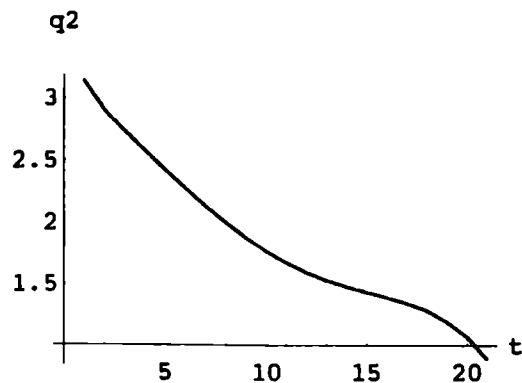
In[74]:=

```
ListPlot[q1,PlotJoined->True, AxesLabel->{"t","q1"}];
```



In[75]:=

```
ListPlot[q2,PlotJoined->True, AxesLabel->{"t","q2"}];
```



**EJEMPLO 2.**

**Puntos:**  $P = \{-3,0\},\{-2,2\},\{-1,1\},\{0,3\},\{2,1\},\{3,0\},\{1,-2\},\{-1,-3\}$

**Velocidades:**  $v = \{0,5,1,3,8,5,3,0\}$

*In[32]:=*

Pos=Bn[7]

*Out[32]=*

$$(1 - s)^7 Pj[0] + 7 (1 - s)^6 s Pj[1] + 21 (1 - s)^5 s^2 Pj[2] + 35 (1 - s)^4 s^3 Pj[3] + 35 (1 - s)^3 s^4 Pj[4] + 21 (1 - s)^2 s^5 Pj[5] + 7 (1 - s) s^6 Pj[6] + s^7 Pj[7]$$

*In[33]:=*

Vel= D[Pos/. s-> s[t],t]

*Out[33]=*

$$\begin{aligned} & -7 Pj[0] (1 - s(t))^6 s'(t) + 7 Pj[1] (1 - s(t))^6 s'(t) - \\ & 42 Pj[1] (1 - s(t))^5 s(t) s'(t) + 42 Pj[2] (1 - s(t))^5 s(t) s'(t) - \\ & 105 Pj[2] (1 - s(t))^4 s(t)^2 s'(t) + 105 Pj[3] (1 - s(t))^4 s(t)^2 s'(t) - \\ & 140 Pj[3] (1 - s(t))^3 s(t)^3 s'(t) + 140 Pj[4] (1 - s(t))^3 s(t)^3 s'(t) - \\ & 105 Pj[4] (1 - s(t))^2 s(t)^4 s'(t) + 105 Pj[5] (1 - s(t))^2 s(t)^4 s'(t) - \\ & 42 Pj[5] (1 - s(t)) s(t)^5 s'(t) + 42 Pj[6] (1 - s(t)) s(t)^5 s'(t) - \\ & 7 Pj[6] s(t)^6 s'(t) + 7 Pj[7] s(t)^6 s'(t) \end{aligned}$$

In[34]:=

Acel=D[Vel,t]/. s''[t] -> 0

Out[34]=

$$\begin{aligned}
 & 42 P_j[0] (1 - s(t))^5 s'(t)^2 - 84 P_j[1] (1 - s(t))^5 s'(t)^2 + 42 P_j[2] (1 - s(t))^5 \\
 & 210 P_j[1] (1 - s(t))^4 s(t) s'(t)^2 - 420 P_j[2] (1 - s(t))^4 s(t) s'(t)^2 + \\
 & 210 P_j[3] (1 - s(t))^4 s(t) s'(t)^2 + 420 P_j[2] (1 - s(t))^3 s(t)^2 s'(t)^2 - \\
 & 840 P_j[3] (1 - s(t))^3 s(t)^2 s'(t)^2 + 420 P_j[4] (1 - s(t))^3 s(t)^2 s'(t)^2 + \\
 & 420 P_j[3] (1 - s(t))^2 s(t)^3 s'(t)^2 - 840 P_j[4] (1 - s(t))^2 s(t)^3 s'(t)^2 + \\
 & 420 P_j[5] (1 - s(t))^2 s(t)^3 s'(t)^2 + 210 P_j[4] (1 - s(t))^4 s(t)^2 s'(t)^2 - \\
 & 420 P_j[5] (1 - s(t))^4 s(t)^2 s'(t)^2 + 210 P_j[6] (1 - s(t))^4 s(t)^2 s'(t)^2 + \\
 & 42 P_j[5] s(t)^5 s'(t)^2 - 84 P_j[6] s(t)^5 s'(t)^2 + 42 P_j[7] s(t)^5 s'(t)^2
 \end{aligned}$$

In[35]:=

Pj[0]={-3,0};  
 Pj[1]={-2,2};  
 Pj[2]={-1,1};  
 Pj[3]={0,3};  
 Pj[4]={2,1};  
 Pj[5]={3,0};  
 Pj[6]={1,-2};  
 Pj[7]={-1,-3};

In[43]:=

pp=BezierPoints[{{-3,0},{-2,2},{-1,1},{0,3},{2,1},{3,0},{1,-2},{-1,-3}},.05]

Out[43]=

```
{(-3, 0), {-2.64981, 0.566066}, {-2.29729, 0.939449}, {-1.93811, 1.19739},
{-1.56781, 1.38124}, {-1.18365, 1.50842}, {-0.785992, 1.5819}, {-0.379107, 1.597
{0.028352, 1.54767}, {0.423384, 1.42759}, {0.789062, 1.23437}, {1.10536, 0.96930
{1.35034, 0.637478}, {1.50169, 0.247294}, {1.53865, -0.190424}, {1.44415, -0.663
{1.20736, -1.15743}, {0.826413, -1.65815}, {0.311464, -2.14811}, {-0.312064, -2.
{-1., -3.}}
```



In[44]:=

Vel

Out[44]=

$$\begin{aligned} & \{7 (1 - s[t])^6 s'[t] + 42 (1 - s[t])^5 s[t] s'[t] + 105 (1 - s[t])^4 s[t]^2 s'[t] + \\ & 280 (1 - s[t])^3 s[t]^3 s'[t] + 105 (1 - s[t])^2 s[t]^4 s'[t] - 84 (1 - s[t]) s[t]^5 s'[t] \\ & 14 s[t]^6 s'[t], 14 (1 - s[t])^6 s'[t] - 42 (1 - s[t])^5 s[t] s'[t] + \\ & 210 (1 - s[t])^4 s[t]^2 s'[t] - 280 (1 - s[t])^3 s[t]^3 s'[t] - \\ & 105 (1 - s[t])^2 s[t]^4 s'[t] - 84 (1 - s[t]) s[t]^5 s'[t] - 7 s[t]^6 s'[t]\} \end{aligned}$$

In[45]:=

Acel

Out[45]=

$$\begin{aligned} & \{420 (1 - s[t])^3 s[t]^2 s'[t]^2 - 420 (1 - s[t])^2 s[t]^3 s'[t]^2 - \\ & 630 (1 - s[t]) s[t]^4 s'[t]^2, -126 (1 - s[t])^5 s'[t]^2 + 630 (1 - s[t])^4 s[t] s'[t]^2 \\ & 1680 (1 - s[t])^3 s[t]^2 s'[t]^2 + 420 (1 - s[t])^2 s[t]^3 s'[t]^2 - \\ & 210 (1 - s[t]) s[t]^4 s'[t]^2 + 42 s[t]^5 s'[t]^2 \} \end{aligned}$$

In[47]:=

vv=BezierPoints{{0,.5,.1,.3,.8,.5,.3,0},.05}

Out[47]=

{0, 0.133926, 0.207422, 0.246769, 0.270628, 0.291083, 0.314731, 0.343798, 0.37723  
0.411783, 0.442969, 0.465989, 0.476475, 0.471086, 0.447919, 0.406696, 0.348687,  
0.276371, 0.19276, 0.100404, -4.66294 10<sup>-16</sup> }

In[48]:=

Vperfil=Table[Vel /. {s[t] -> i/20, s'[t] -> vv[[i+1]]}, {i,0,20}]

Out[48]=

{(0, 0), {0.939483, 1.21381}, {1.47288, 1.27036}, {1.79692, 1.06776},  
{2.04049, 0.834019}, {2.27785, 0.586003}, {2.53891, 0.286243}, {2.81215, -0.1099  
{3.04624, -0.634429}, {3.15811, -1.2891}, {3.05233, -2.03489}, {2.65037, -2.7922  
{1.92328, -3.45572}, {0.917561, -3.91864}, {-0.236114, -4.09787}, {-1.33794, -3.  
{-2.15807, -3.48759}, {-2.49176, -2.75543}, {-2.21643, -1.84213}, {-1.33505, -0.  
{6.52811 10<sup>-15</sup>, 3.26406 10<sup>-15</sup>}}

In[49]: =

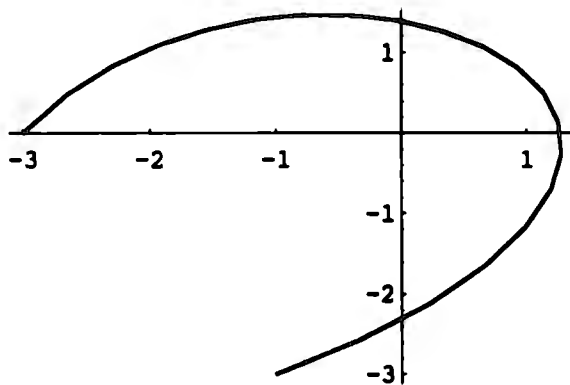
```
Aperfil=Table[Acel /. {s[t] -> i/20, s'[t]-> vv[[i+1]] },{i,0,20}]
```

Out[49]=

```
{0, 0}, {0.0152299, -1.35227}, {0.114654, -1.93577}, {0.274529, -1.75707},
{0.413422, -1.62514}, {0.469155, -1.80016}, {0.380048, -2.29726}, {0.044469, -3.
-0.688531, -4.02982}, {-1.97304, -4.98248}, {-3.86311, -5.66589}, {-6.19188, -5
-8.51302, -5.31835}, {-10.1822, -4.23136}, {-10.5909, -2.82137}, {-9.46373, -1.
-7.0592, -0.325181}, {-4.13288, 0.31781}, {-1.63697, 0.474821}, {-0.267245, 0.2
-30
{0, 9.13205 10  } }
```

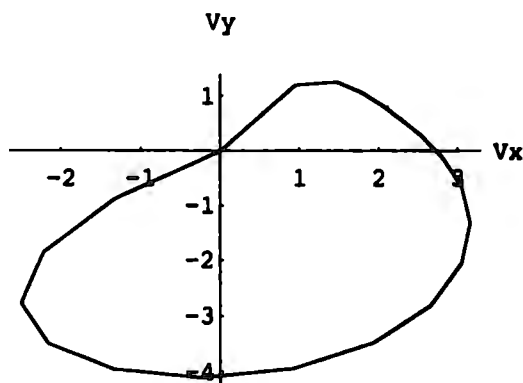
In[50]: =

```
PlotBezier[pp,.05]
```



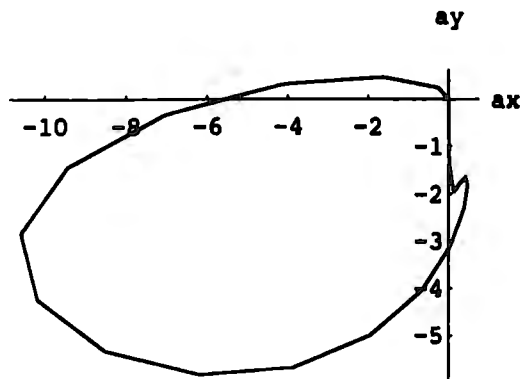
In[51]: =

```
ListPlot[Vperfil,PlotJoined->True, AxesLabel->{"Vx","Vy"}];
```



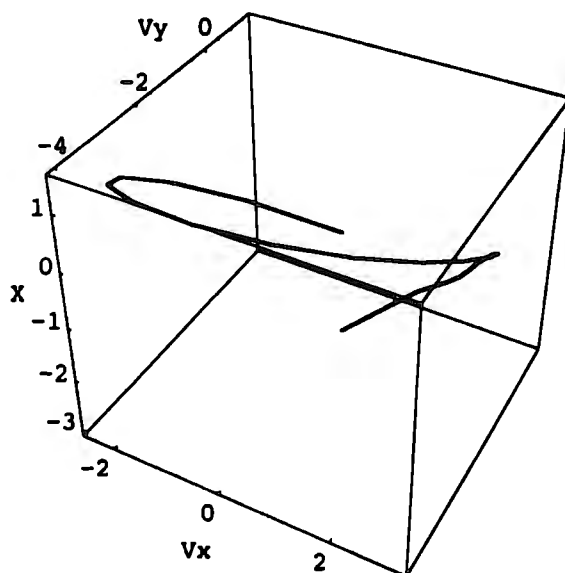
In[52]: =

```
ListPlot[Aperfil, PlotJoined->True, AxesLabel->{"ax", "ay"}];
```



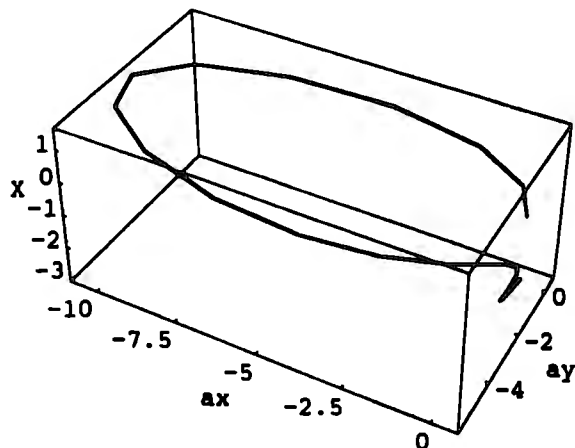
In[53]: =

```
Newperfil=N[Table[Append[Vperfil[[i]],pp[[i,1]]],{i,1,21}];
Show[Graphics3D[Line[Newperfil],PlotRange->All, Axes->True,AxesLabel->{"Vx","Vy","X"}];
```



In[55]: =

```
Newperfil=N[Table[Append[Aperfil[[i]],pp[[i,1]]],{i,1,21}];
Show[Graphics3D[Line[Newperfil],PlotRange->All, Axes->True,AxesLabel->{"ax","ay","X"}];
```



**EJEMPLO 2. ROBOT PLANO DE DOS GRADOS DE LIBERTAD**

**CINEMÁTICA INVERSA**

**Seguimiento de la trayectoria**

*In[66]:* =

l1=3;

l2=2;

*In[68]:* =

P= {{-3,0},{-2,2},{-1,1},{0,3},{2,1},{3,0},{1,-2},{-1,-3}};

*In[69]:* =

pp=BezierPoints[P,.05]

*Out[69]:* =

```
{-3, 0}, {-2.64981, 0.566066}, {-2.29729, 0.939449},
{-1.93811, 1.19739}, {-1.56781, 1.38124}, {-1.18365, 1.50842},
{-0.785992, 1.5819}, {-0.379107, 1.59723}, {0.028352, 1.54767},
{0.423384, 1.42759}, {0.789062, 1.23437}, {1.10536, 0.969301},
{1.35034, 0.637478}, {1.50169, 0.247294}, {1.53865, -0.190424},
{1.44415, -0.663208}, {1.20736, -1.15743}, {0.826413, -1.65815},
{0.311464, -2.14811}, {-0.312064, -2.60544}, {-1., -3.}}
```

*In[70]:* =

Q=InvKin[pp]

*Out[70]:* =

```
{2.46192, 1.91063}, {2.22234, 2.0618}, {2.02979, 2.17729},
{1.85864, 2.27948}, {1.69222, 2.3739}, {1.51975, 2.46063},
{1.33399, 2.53803}, {1.12968, 2.60364}, {0.902905, 2.65442},
{0.652129, 2.68728}, {0.380493, 2.70094}, {0.0964432, 2.69801},
{-0.190769, 2.6849}, {-0.478273, 2.66883}, {-0.773441, 2.65306},
{-1.09172, 2.63187}, {-1.44519, 2.5872}, {-1.8181, 2.4936},
{-2.15604, 2.33328}, {-2.40527, 2.10543}, {-2.5516, 1.82348}}
```

In[71]: =

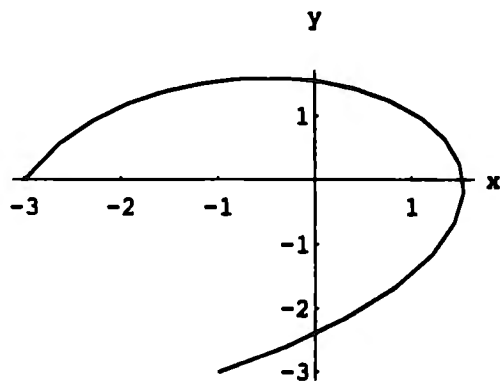
pq=Kin[Q]

Out[71]=

```
{-3., 0.}, {-2.64981, 0.566066}, {-2.29729, 0.939449},
{-1.93811, 1.19739}, {-1.56781, 1.38124}, {-1.18365, 1.50842},
{-0.785993, 1.5819}, {-0.379107, 1.59723}, {0.028352, 1.54767},
{0.423384, 1.42759}, {0.789062, 1.23437}, {1.10536, 0.969301},
{1.35034, 0.637478}, {1.50169, 0.247294}, {1.53865, -0.190424},
{1.44415, -0.663208}, {1.20736, -1.15743}, {0.826413, -1.65815},
{0.311464, -2.14811}, {-0.312064, -2.60544}, {-1., -3.}}
```

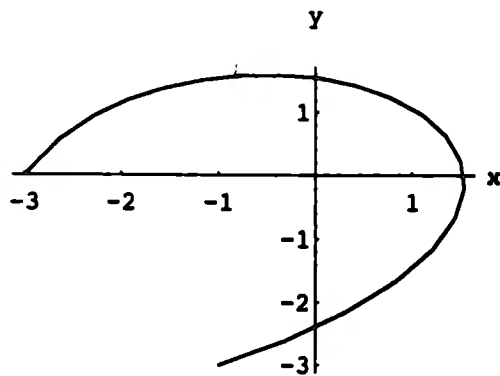
In[72]: =

ListPlot[pp,PlotJoined->True, AxesLabel->{"x","y"}];



In[73]: =

ListPlot[pq,PlotJoined->True, AxesLabel->{"x","y"}];



In[75]:=

```
q1=Table[{i,Q[[i,1]]},{i,1,21}]
q2=Table[{i,Q[[i,2]]},{i,1,21}]
```

Out[75]=

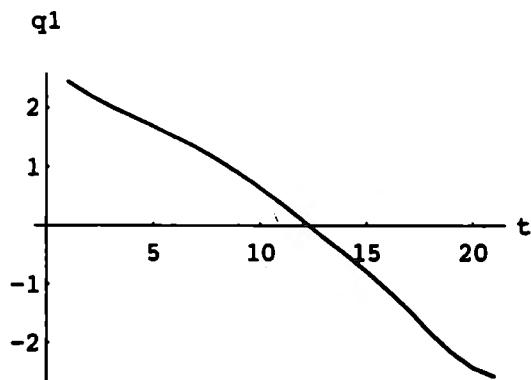
- {1, 2.46192}, {2, 2.22234}, {3, 2.02979}, {4, 1.85864},
- {5, 1.69222}, {6, 1.51975}, {7, 1.33399}, {8, 1.12968},
- {9, 0.902905}, {10, 0.652129}, {11, 0.380493}, {12, 0.0964432},
- {13, -0.190769}, {14, -0.478273}, {15, -0.773441},
- {16, -1.09172}, {17, -1.44519}, {18, -1.8181}, {19, -2.15604},
- {20, -2.40527}, {21, -2.5516}

Out[76]=

- {1, 1.91063}, {2, 2.0618}, {3, 2.17729}, {4, 2.27948},
- {5, 2.3739}, {6, 2.46063}, {7, 2.53803}, {8, 2.60364},
- {9, 2.65442}, {10, 2.68728}, {11, 2.70094}, {12, 2.69801},
- {13, 2.6849}, {14, 2.66883}, {15, 2.65306}, {16, 2.63187},
- {17, 2.5872}, {18, 2.4936}, {19, 2.33328}, {20, 2.10543},
- {21, 1.82348}

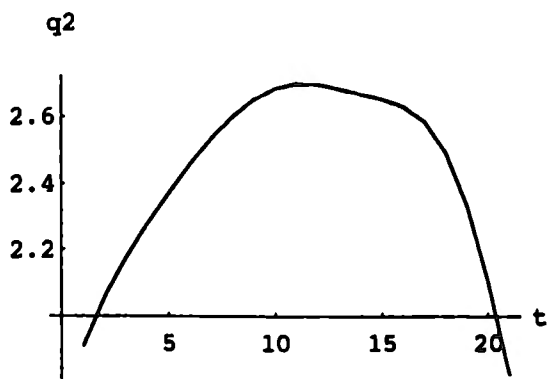
In[77]:=

```
ListPlot[q1,PlotJoined->True, AxesLabel->{"t","q1"}];
```



In[78]:=

```
ListPlot[q2,PlotJoined->True, AxesLabel->{"t","q2"}];
```



---

---

# BIBLIOGRAFÍA Y REFERENCIAS

---

---

- [1] A. Barrera Torres, B. de la Merced Sánchez  
Proyecto de Robot Manipulador Actuado Hidraulicamente de 3 Grados de Libertad  
Proyecto Terminal para obtener el título en Ingeniería Mecánica  
Universidad Autónoma Metropolitana, Unidad Azcapotzalco  
México, D. F. 1994
- [2] Mikel P. Groover, Mitchell Weiss y otros  
Robótica Industrial, Tecnología, Programación y Aplicaciones  
Mc Graw-Hill  
Madrid, España. 1989
- [3] G. Ferráte y otros  
Robótica industrial  
Marcombo editores  
Barcelona, España. 1986
- [4] Stuart E. Thompson, Rajnikant V. Patel  
Formulation of joint trayectories for industrial robots using B-splines  
Department of Electrical Research Council, Concordia University  
Montreal, Quebec. Canada. 1987
- [5] J. Hartley  
Concurrent Engineering: shortening lead times, raising quality and lowering costs  
Productivity press  
Boston, U. S. A. 1992
- [6] A. Çela, Y. Hamam  
Decomposition method for the constrained path planning of articulated systems  
Control Laboratoty. Group ESIEE  
Noisy-Le-Grand, Cedex-France  
IEEE Transactions on Robotics and Automation. 1991
- [7] R.V. Mayorga, A. K. C. Wong y K. S. Ma  
Un método local eficiente para la generación de rutas de robots manipuladores  
Department of Systems Design Engineering, University of Waterloo  
Waterloo Ontario, Canada  
Journal of Robotics Systems. 1990

- [8] H. Chang, T. Lee, H. Liu  
On-line cartesian path trajectory planning for robot manipulator  
Chung-Cheng Institute of Technology- University of Kentucky  
Taiwan, R. O. C.- Lexington Kentucky  
IEEE Transactions on Robotics and Automation. 1988
- [9] Jonh J. Craig  
Introduction to Robotics  
Addison - Wesley Publishing Company  
U. S. A. 1989
- [10] J. Y. Wu  
A systematic approach to robot inverse kinematics  
IEEE Transactions on Robotics and Automation. 1990
- [11] Anthony A. Maciejewsky, John J. Fox  
Path planingng and the topology of configuration space  
IEEE Transactions on Robotics and Automation. 1993
- [12] K.S. Fu y otros  
Robótica: Control, Detección, Visión e Inteligencia. Primera edición  
Mc Graw-Hill  
Madrid, España. 1987
- [13] J. M. Angulo y otros  
Curso de Robótica. Primera edición  
Editorial Paraninfo  
Madrid, España. 1983
- [14] Brady, Hollerbach, Jonhson, Lozano-Pérez and Mason  
Robot Motion: Planning and Control  
MIT Press. 1984
- [15] John F. Canny  
The Complexity of Robot Motion Planning  
MIT Press.  
Boston, U. S. A. 1988
- [16] F. S. Hill Jr.  
Computer Graphics  
Maxwell MacMillan International Editions, 1990
- [17] Agustín Vásquez Arvalo  
Sistema de asistencia por computadora para manufactura: Definición y programación de formas complejas  
Proyecto Terminal para obtener el título en Ingeniería Mecánica  
Universidad Autónoma Metropolitana, Unidad Azcapotzalco  
México, D. F. 1989



- [18] D. N. Moreton, D. B. Parkinson and W. K. Wu  
The application of a biarc technique in CNC machining  
University of Liverpool  
Computer-Aided Engineering Journal  
Liverpool, England. 1991
- [19] Gerald Farin  
From conics to NURBS: a tutorial and Survey  
Arizona State University  
IEEE Computer Graphics & Applications  
Arizona, U. S. A. 1992
- [20] Jin J. Chou, Les A. Piegl  
Data reduction using cubic rational B-splines  
Computers Sciences Corporation, NASA Ames - University of South Florida  
IEEE Computer Graphics & Applications  
Florida, U. S. A. 1992
- [21] S. Tehrani, T. E. Weymouth, B. Schunck  
Interpolating cubic spline contours by minimizing second derivative discontinuity  
Department of Electrical Engineering and Computer Science  
University of Michigan  
IEEE Computer Graphics & Applications  
Ann Arbor, Michigan. U. S. A. 1995
- [22] Stephen Wolfram  
MATHEMATICA, A system for Doing Mathematics by Computers  
Wolfram Research, Inc.  
Addison - Wesley Publishing Company  
Champaign, Illinois. U. S. A. 1991
- [23] Theodore W. Gray and Jerry Glynn  
The Begginer's guide to Mathematica version 2  
Wolfram Research, Inc.  
Addison - Wesley Publishing Company  
Champaign, Illinois. U. S. A. 1992
- [24] Roman E. Maeder  
Programming in Mathematica. Second Edition  
Addison - Wesley Publishing Company  
Redwood City, California. U. S. A. 1991
- [25] Agustín Vásquez Arvallo  
Modelo Dinámico y Cinemático General para Robots Manipuladores  
Tesis para obtener el grado de Maestro en Ciencias  
ITESM - CEM, University of Texas - Austin  
Atizapán, Edo. de México. México. 1993

**[26] John F. Nethery and Mark W. Spong**  
**Robotica: A Mathematica package for robot analysis**  
**University of Illinois**  
**Robotics & Automation Magazine**  
**Urbana-Champaign, Illinois. U. S. A. 1994**

**[27] Herbert Schildt**  
**Turbo C/C++. Manual de Referencia**  
**Mc Graw-Hill**  
**1992**