



UNIVERSIDAD MICHOACANA DE SAN  
NICOLÁS DE HIDALGO

FACULTAD DE INGENIERÍA ELÉCTRICA  
DIVISIÓN DE ESTUDIOS DE POSGRADO

“PLANEACIÓN DE TRAYECTORIAS ÓPTIMAS  
PARA ROBOTS MANIPULADORES UTILIZANDO  
POLINOMIOS DE OCTAVO GRADO, ALGORITMOS  
GENÉTICOS Y TÉCNICAS DE PROCESAMIENTO EN  
PARALELO”

TESIS

QUE PARA OBTENER EL GRADO DE  
DOCTOR EN CIENCIAS EN INGENIERÍA ELÉCTRICA

PRESENTA

WALDEMAR PÉREZ BAILÓN

ASESOR

DR. EDMUNDO BARRERA CARDIEL

COASESOR

DR. IGNACIO JUÁREZ CAMPOS

MORELIA, MICHOACÁN

AGOSTO DEL 2011



## RESUMEN

Esta tesis presenta el desarrollo de una plataforma de software para la planeación y optimización de trayectorias de robots manipuladores. Muchas aplicaciones requieren de trayectorias suaves y de la minimización de un índice de eficiencia. El algoritmo de optimización presentado, minimiza el consumo de energía mecánica o minimiza el tiempo de traslado de un robot manipulador. Se introduce una función polinomial para generar trayectorias suaves, para la representación paramétrica de una ruta dada. Se implementa un algoritmo genético para solucionar el modelo de optimización de consumo de energía en el robot manipulador. Para la solución del modelo de optimización del tiempo de traslado, se implementa un método que combina un algoritmo genético con el algoritmo numérico conocido como método de bisecciones sucesivas. El software incluye módulos para la solución de la cinemática directa, la cinemática inversa y la dinámica del robot manipulador. Para mejorar el tiempo de ejecución del algoritmo de optimización, la plataforma de software utiliza una técnica de procesamiento en paralelo basada en multithreading. El software incluye una interfase gráfica para simular la ejecución de una trayectoria dada.

## ABSTRACT

This thesis presents the development of a software platform for planning and optimization trajectories for robot manipulators. Many applications require smooth trajectories and the minimization of a performance index. The optimization algorithm presented minimizes the mechanical energy consumed in the robot manipulator or minimizes the path traveling time of a robot manipulator. It is introduced eighth-degree polynomial functions to generate smooth trajectories for the parametric representation of a given path. To solve the optimization model of the mechanical energy consumed in the robot manipulator, a genetic algorithm is implemented. To solve the optimization model of the path traveling time of a robot manipulator, a method based on a combination of a genetic algorithm and the numerical algorithm known as successive bisection method has been implemented. The software includes modules to solve the direct kinematics, the inverse kinematics, and the dynamics of the robot manipulator. To improve the execution time of the optimization algorithm, the developed software platform uses a parallel processing technique based on multithreading. This thesis includes a graphical interface to simulate the execution of a given trajectory.

# CONTENIDO

Resumen .....	iv
Abstract .....	v
Lista de Figuras .....	viii
Lista de Tablas.....	x
Lista de Publicaciones .....	xi
<b>CAPÍTULO 1. INTRODUCCIÓN.....</b>	<b>1</b>
1.1 Antecedentes .....	1
1.2 Revisión del Estado del Arte.....	3
1.3 Objetivos .....	14
1.3.1 Objetivo General.....	14
1.3.2 Objetivos Específicos.....	14
1.4 Justificación .....	15
1.5 Contribuciones Potenciales .....	18
1.6 Descripción de los Capítulos.....	18
<b>CAPÍTULO 2. PRINCIPIOS BÁSICOS PARA EL MODELADO, ANÁLISIS Y PLANEACIÓN DE TRAYECTORIAS DE ROBOTS MANIPULADORES .....</b>	<b>20</b>
2.1 Modelado de la Cinemática de un Robot Manipulador .....	21
2.1.1 Cinemática Directa .....	22
2.1.2 Cinemática Inversa .....	29
2.2 Modelado de la Dinámica de un Robot Manipulador .....	31
2.2.1 Distribución de Masa.....	33
2.2.2 Método de Newton Euler.....	49
2.3 Planeación de Trayectorias.....	53
2.3.1 La Recta en Forma Paramétrica .....	57
2.3.2 Perfil de Trayectoria .....	58
2.3.2.1 Perfil Trapezoidal.....	58
2.3.2.2 Perfil Polinomial de Tercer Grado .....	62
2.3.2.3 Perfil Polinomial de Quinto Grado .....	66
2.3.2.4 Perfil Polinomial de Octavo Grado .....	70
2.4 Conclusiones.....	75
<b>CAPÍTULO 3. CONCEPTOS BÁSICOS PARA OPTIMIZACIÓN MEDIANTE ALGORITMOS GENÉTICOS.....</b>	<b>76</b>
3.1 Estrategia Evolutiva .....	76
3.1.1 Algoritmos Genéticos.....	77
<b>CAPÍTULO 4. CONCEPTOS BÁSICOS DE PROCESAMIENTO EN PARALELO .....</b>	<b>84</b>
4.1 Introducción .....	84
4.2 Procesamiento en Paralelo.....	85
4.3 Arquitecturas para el Procesamiento en Paralelo.....	86
4.3.1 Multicomputadoras.....	86

4.3.2 Multiprocesadores.....	87
4.4 Plataformas para el Procesamiento en Paralelo.....	87
4.5 Diseño de Algoritmos en Paralelo.....	89
4.6 Medición de la Eficiencia de un Algoritmos en Paralelo.....	90
<b>CAPÍTULO 5. PLANEACIÓN Y OPTIMIZACIÓN DE TRAYECTORIAS</b>	
<b>DE ROBOTS MANIPULADORES .....</b>	<b>92</b>
5.1 Proceso de Planeación y Optimización de Trayectorias .....	92
5.2 Singularidades .....	94
5.3 Optimización de Energía Mecánica en la Planeación de Trayectorias	
Aplicando un Algoritmo Genético.....	96
5.4 Optimización del Tiempo de Traslado en la Planeación de Trayectorias	
Aplicando un Algoritmo Genético.....	100
5.5 Aplicación de una Técnica de Procesamiento en Paralelo a la Planeación	
de Trayectorias Óptimas .....	104
<b>CAPÍTULO 6. CASOS DE ESTUDIO .....</b>	<b>108</b>
6.1 Procedimiento de las Pruebas .....	108
6.1.1 Procedimiento de Validación: Solución Mediante Técnica	
de Búsqueda Exhaustiva.....	111
6.2 Planeación de Trayectorias con Optimización del Gasto de Energía .....	113
6.2.1 Ruta 1 .....	114
6.2.2 Ruta 2 .....	118
6.2.3 Ruta 3 .....	122
6.3 Planeación de Trayectorias con Optimización del Tiempo de Traslado.....	124
6.3.1 Ruta 1 .....	125
6.3.2 Ruta 2 .....	130
6.3.3 Ruta 3 .....	132
6.4 Conclusiones del Capítulo .....	134
<b>CAPÍTULO 7. CONCLUSIONES .....</b>	<b>136</b>
7.1 Conclusiones.....	136
7.2 Trabajos Futuros .....	137
<b>REFERENCIAS .....</b>	<b>138</b>

## LISTA DE FIGURAS

<b>Figura 2.1</b>	Relación entre la cinemática directa y la cinemática inversa.....	22
<b>Figura 2.2</b>	Robot planar con dos grados de libertad.....	22
<b>Figura 2.3</b>	Sistemas de referencias de coordenadas sujetos al eslabón (i-1).....	24
<b>Figura 2.4</b>	Asignación de los sistemas de referencia de coordenadas a los eslabones del robot a analizar. ....	27
<b>Figura 2.5</b>	Eslabones del robot manipulador analizado en esta tesis. ....	36
<b>Figura 2.6</b>	Partes del eslabón 2 para el cálculo del tensor de inercia. ....	37
<b>Figura 2.7</b>	Dimensiones de las partes del eslabón 2.....	38
<b>Figura 2.8</b>	Centros de gravedad de las piezas del eslabón 2 (acotaciones en m) .....	40
<b>Figura 2.9</b>	Fuerza $F$ actuando en el centro de masa de un cuerpo causa una aceleración $\dot{v}_c$ .....	50
<b>Figura 2.10</b>	Momento $N$ actuando sobre un cuerpo provocando que éste rote con velocidad $\omega$ y una aceleración $\dot{\omega}$ .....	51
<b>Figura 2.11</b>	Componentes de la trayectoria.....	56
<b>Figura 2.12</b>	Relación que mantiene el lugar geométrico con el perfil de trayectoria.....	56
<b>Figura 2.13</b>	Caracterización de un perfil trapezoidal .....	60
<b>Figura 2.14</b>	Perfil de trayectoria polinomial cúbica, a) Perfil de posición, b) Perfil de velocidad, c) Perfil de aceleración .....	66
<b>Figura 2.15</b>	Perfil de posición del polinomio de quinto grado .....	67
<b>Figura 2.16</b>	Perfil de velocidad del polinomio de quinto grado .....	67
<b>Figura 2.17</b>	Perfil de aceleración del polinomio de quinto grado .....	68
<b>Figura 2.18</b>	Perfiles de velocidad del robot manipulador dentro del intervalo de movimientos naturales.....	74
<b>Figura 2.19</b>	Perfiles de velocidad del robot manipulador fuera del intervalo de movimientos naturales.....	74
<b>Figura 3.1</b>	Diagrama a bloques del algoritmo genético.....	79
<b>Figura 3.2</b>	Representación binaria de la población genética (genotipo) .....	80
<b>Figura 3.3</b>	Esquema de la ruleta para la selección de individuos con mayor aptitud.....	81
<b>Figura 3.4</b>	Cruzamiento, operador genético para obtener la nueva generación .....	82
<b>Figura 4.1</b>	Arquitectura multicomputadora .....	86
<b>Figura 4.2</b>	Arquitectura multiprocesador .....	88
<b>Figura 4.3</b>	Plataforma multithreading .....	89
<b>Figura 4.4</b>	Etapas de diseño de algoritmos paralelo .....	90
<b>Figura 5.1</b>	Algoritmo propuesto para el proceso de planeación y optimización de trayectorias.....	93
<b>Figura 5.2</b>	Algoritmo de optimización del consumo de energía sobre una trayectoria de un robot manipulador en un tiempo dado.....	98
<b>Figura 5.3</b>	División de la ruta en $n$ puntos .....	99
<b>Figura 5.4</b>	Algoritmo de optimización del tiempo de traslado sobre una trayectoria de un robot manipulador .....	103
<b>Figura 5.5</b>	Esquema de procesamiento en paralelo basado en multithreading en la planeación de trayectorias optimas minimizando el gasto de energía .....	105

<b>Figura 5.6</b>	Esquema de procesamiento en paralelo basado en multithreading en la planeación de trayectorias óptimas, minimizando el tiempo de traslado.....	106
<b>Figura 6.1</b>	Interface de simulación del movimiento del robot manipulador .....	109
<b>Figura 6.2</b>	Equipo de computo utilizado para el desarrollo de esta tesis .....	110
<b>Figura 6.3</b>	Rutas propuestas para el análisis de los seis casos de estudio .....	111
<b>Figura 6.4</b>	Algoritmo de planeación de trayectorias con optimización del gasto de energía mediante técnica de búsqueda exhaustiva .....	113
<b>Figura 6.5</b>	Comportamiento del par y velocidad rotacional de las articulaciones del robot manipulador, con la trayectoria optima de la ruta 1.....	116
<b>Figura 6.6</b>	Comportamiento del par y velocidad rotacional de las articulaciones del robot manipulador, con la trayectoria optima de la ruta 2.....	120
<b>Figura 6.7</b>	Comportamiento del par y velocidad rotacional de las articulaciones del robot manipulador, con la trayectoria optima de la ruta 3.....	123
<b>Figura 6.8</b>	Comportamiento del par y velocidad rotacional de las articulaciones del robot manipulador, de la trayectoria optima de la ruta 1.....	127
<b>Figura 6.9</b>	Comportamiento del par y velocidad rotacional de las articulaciones del robot manipulador, de la trayectoria optima de la ruta 2.....	131
<b>Figura 6.10</b>	Comportamiento del par y velocidad de las articulaciones del robot manipulador, de la trayectoria optima de la ruta 3.....	133

## LISTA DE TABLAS

Tabla 2.1	Parámetros de Denavit-Hartenberg de cada uno de los eslabones del robot diseñado.....	27
Tabla 2.2	Ecuaciones de la solución de la cinemática inversa del robot manipulador usado en este proyecto de tesis.....	30
Tabla 2.3	Momentos de inercia de masa de dos formas geométricas comunes. ....	35
Tabla 2.4	Restricciones para evaluar los coeficientes del polinomio de quinto grado .....	68
Tabla 2.5	Restricciones para evaluar los coeficientes del polinomio de octavo grado. ...	70
Tabla 6.1	Coordenadas de los puntos extremos de las rutas a analizar .....	111
Tabla 6.2	Límites nominales de par y velocidad de las articulaciones del robot manipulador.....	112
Tabla 6.3	Trayectoria solución con consumo de energía mínima de la ruta 1 .....	115
Tabla 6.4	Trayectoria solución con consumo de energía mínima de la ruta 2 .....	119
Tabla 6.5	Trayectoria solución con consumo de energía mínima de la ruta 3 .....	122
Tabla 6.6	Trayectoria solución con tiempo mínimo de traslado de la ruta 1.....	126
Tabla 6.7	Comportamiento del par real al aumentar y disminuir el <i>índice<sub>m</sub></i> óptimo manteniendo fijo el tiempo mínimo óptimo, para la trayectoria óptima de la ruta 1 .....	129
Tabla 6.8	Comportamiento del par real al aumentar y disminuir el tiempo mínimo manteniendo fijo el <i>índice<sub>m</sub></i> para la trayectoria óptima de la ruta 1 .....	129
Tabla 6.9	Trayectoria solución con tiempo mínimo de traslado de la ruta 2 .....	130
Tabla 6.10	Comparación del par real aumentando y disminuyendo el <i>índice<sub>m</sub></i> óptimo manteniendo fijo el tiempo mínimo en la trayectoria óptima de la ruta 2 .....	132
Tabla 6.11	Comportamiento del par real aumentando y disminuyendo el tiempo mínimo manteniendo fijo el <i>índice<sub>m</sub></i> para la trayectoria óptima de la ruta 2 .....	132
Tabla 6.12	Trayectoria solución con tiempo mínimo de traslado de la ruta 3 .....	134



## **LISTA DE PUBLICACIONES**

### **CONGRESOS INTERNACIONALES**

1. W. Pérez Bailón, E. Barrera Cardiel, I. Juárez Campos, A. Ramos Paz. “Mechanical Energy Optimization in Trajectory Planning for Six DOF Robot Manipulators Based on Eighth-Degree Polynomial Functions, a Genetic Algorithm and a Parallel Processing Technique Based on Multithreading”. GEM2010, PROCEEDING OF THE 2010 INTERNATIONAL AND EVOLUTIONARY, Las Vegas, Nevada, USA. July 12.15, 2010.
2. W. Pérez Bailón, E. Barrera Cardiel, I. Juárez Campos, A. Ramos Paz. “Mechanical Energy Optimization in Trajectory Planning for Six DOF Robot Manipulators Based on Eighth-Degree Polynomial Functions and a Genetic Algorithm”. CEE2010. 20107th International Conference on Electrical Engineering, Computing Science and Automatic Control. Tuxtla Gutierrez, Chiapas, México. September 8-10, 2010.
3. W. Pérez Bailón, E. Barrera Cardiel, I. Juárez Campos, A. Ramos Paz. “Traveling Time Minimization in Trajectory Planning for Six DOF Robot Manipulators Using Eighth-Degree Polynomial Functions and a Genetic Algorithm”. XII Reunión de Otoño de Potencia, Electrónica y Computación, INTERNACIONAL ROPEC'2010. Manzanillo, Colima, México, 10-12 de noviembre 2010.

### **CONGRESOS NACIONALES Y REGIONALES**

1. E. Barrera Cardiel, I. Juárez Campos, W. Pérez Bailón. “Diseño de una Plataforma de Software para la Planeación de Trayectorias de un Robot Manipulador de Seis Grados de Libertad”. Congreso Nacional de Ingeniería y Arquitectura 08. IA08UMSNH, Morelia, Michoacán, México, 2008.

### **ARTICULO ENVIADO A REVISTA INDEXADA**

1. W. Pérez Bailón, E. Barrera Cardiel, I. Juárez Campos, A. Ramos Paz. “Optimal Trajectory Planning for Robot Manipulator Based on Eighth-Degree Polynomial Functions, a Genetic Algorithm and a Parallel Processing Technique”. International Journal of Nonlinear Sciences and Numerical Simulation.

# CAPÍTULO 1

## INTRODUCCIÓN

### 1.1 Antecedentes

A lo largo de la historia, el hombre ha sentido fascinación por las máquinas y dispositivos capaces de imitar las funciones y movimientos de los seres vivos. Los griegos llamaban a estas máquinas, autómatas; de esta palabra se deriva la actual autómata: máquina que imita la figura y movimientos de un ser animado [Barrientos *et al.*, 1998].

Los robots hicieron primeramente su aparición en la mente prolífica de los escritores de ciencia ficción. Desde el siglo XIX se describían las notables ventajas de unas máquinas destinadas al servicio del ser humano en multitud de tareas. Sin embargo, las imaginativas perspectivas del desarrollo de los robots se quedaron en su versión de “ficción”, ya que su componente “ciencia” tuvo que esperar hasta principios de 1950 para ver su aplicación tecnológica en la industria [Barrientos *et al.*, 1998].

Durante y después de la segunda guerra mundial, los servomecanismos y las computadoras dieron un gran impulso a la robótica. La computadora hacía los cálculos del movimiento de los dispositivos que deberían de ser posicionados por los servomecanismos para poder atacar con efectividad aviones enemigos descubiertos por el radar. Esto dio origen a la tecnología del control numérico, propuesto en 1940 por J. T. Parsons [J. Somló *et al.*, 1997]. La primer máquina con control numérico se desarrolló en 1948 y en 1952 se hace una demostración en el MIT (*Massachusetts Institute of Technology*) de un molino de tres ejes controlado por tecnología digital.

La primera patente de un dispositivo robótico fue solicitada por el inventor británico C. W. Kenwar, patente que le fue otorgada en 1957. Sin embargo, se considera que fue George

C. Devol, ingeniero norteamericano, quien estableció las bases del robot industrial moderno y concibió la idea de un dispositivo de transferencia de artículos programada, idea que fue patentada en 1961. En 1956 Devol pone esta idea en conocimiento de Joseph F. Engelberger, quien además de ser asiduo lector de Asimov, era director de ingeniería de la división aeroespacial de la empresa *Manning Maxwell y Moore en Stanford, Connecticut*, con quien forman la *Consolidated Controls Corporation*, que más tarde se conocería como *Unimation (Universal Automation)*, empresa que en 1960 instala su primer robot en la fábrica General Motors de Nueva Jersey, con una aplicación en la fundición por inyección.

Durante los últimos 45 años, las investigaciones sobre robótica, se han propuesto encontrar soluciones técnicas sobre la robótica aplicada. La evolución de los diferentes campos de aplicación y su sofisticación han influido en los tópicos de investigación en la comunidad robótica. Al inicio de la década de 1960, aparece el robot en la industria para liberar al ser humano de tareas arriesgadas y peligrosas. Posteriormente se incorpora al robot industrial dentro de otros procesos de producción. Procesos que requieren de mayor flexibilidad e inteligencia de los robots industriales.

Un robot industrial es una máquina programable de uso general que tiene algunas características antropomórficas o humanoides. Las características humanoides más típicas de los robots actuales es la de sus brazos móviles. Un robot manipulador, conocido como brazo robótico, se puede modelar como una cadena articulada en lazo abierto con algunos cuerpos rígidos conectados en serie por una articulación de revolución o prismática movida por actuadores. Un extremo de la cadena se une a una base fija mientras que el otro extremo está libre y unido a una herramienta para manipular objetos o realizar alguna tarea programada.

Actualmente, la creación de nuevos mercados y necesidades diferentes a las de mercados tradicionales de robots de fabricación tradicional y el agitado mundo en que vivimos, está demandando robots de campo y servicio para asistir al nuevo mercado y a las necesidades sociales humanas.

## 1.2 Revisión del Estado del Arte

Las líneas de investigación fuertemente relacionadas con aplicaciones de la robótica en el mundo real, son: Robots manipuladores, robots móviles y robots inspirados biológicamente. Los robots manipuladores incluyen investigaciones sobre robots industriales, robots médicos y robots de rehabilitación, así como aplicaciones de servicio como: soldadura, pintura, selección, peletizado, etc. Las investigaciones sobre robótica móvil incluyen a robots terrestres, acuáticos y vehículos aéreos. Los robots biológicos incluyen principalmente los humanoides.

Fue alrededor de 1960 cuando los robots industriales se introdujeron por primera vez en el proceso de producción y hasta 1990 los robots industriales dominaron las investigaciones en robótica. Al inicio, la industria automotriz dictó las especificaciones que deberían cumplir los robots industriales, debido principalmente a la influencia sobre el mercado industrial y a necesidades técnicas. Estas necesidades determinaron qué áreas de investigación eran predominantes. La cinemática del robot manipulador fue un área que predominó las investigaciones debido a la inexactitud de sus modelos basados en los parámetros industriales [García, *et al.*, 2007]. Una detallada discusión de los fundamentos del modelado de la cinemática puede encontrarse en la literatura [Craig, 1986] [FU, *et al.*, 1987]. La investigación sobre la cinemática del robot sigue siendo un problema de oportunidad y se esperan nuevos métodos que reduzcan la complejidad computacional [García, *et al.*, 2007].

Otro tópico de investigación importante es la planeación del movimiento. Además de la planeación del movimiento del robot, las leyes de control que aseguran la ejecución del plan son requeridas para completar la tarea del robot. Así, un tópico fundamental de investigación se enfoca sobre técnicas de control. Un robot manipulador es un sistema multivariable no lineal con amplio espectro de técnicas de control que pueden experimentarse, desde los métodos simples de control proporcional derivativo (PD) y proporcional integral derivativo (PID) [Slotine, *et al.*, 1991] y los controles adaptivos más sofisticados [Astrom, *et al.*, 1989].

Los robots industriales típicos se diseñaron para manipular objetos e interactuar con su medio ambiente. En el control entre el manipulador y su medioambiente, la fuerza de contacto del efector final del manipulador es controlada. Hay diferentes esquemas de control de la fuerza activa, tales como: control “*stiffness*”, control “*compliant*”, control de la impedancia, control de fuerza explícita y control híbrido de fuerza y posición [García, *et al.*, 2007]. Los tres primeros esquemas pertenecen a la categoría de control indirecto de fuerza que logra controlar la fuerza vía el control de movimiento, mientras que los dos últimos métodos realizan el control directo de la fuerza por medio de control de lazo cerrado con retroalimentación [Siciliano, *et al.*, 1999]. Una alternativa interesante para implementar las leyes de control de fuerza, es el uso de un dispositivo mecánico cuya trayectoria es modificada por la interacción de las propias fuerzas del robot [García, *et al.*, 2007].

Al inicio de 1990, nuevas áreas de aplicación de robots industriales impusieron nuevas especificaciones, con características principales de flexibilidad. Las industrias que introdujeron a los robots industriales en sus procesos fueron la industria alimenticia y la farmacéutica. El servicio postal buscaba los sistemas robóticos para automatizar su logística. El principal requisito era la capacidad de controlar un producto con características tan variables como son: tamaño, forma, rigidez, peso, etc. La habilidad de adaptarse por sí mismo al producto y su medio ambiente se vuelve un problema para las líneas de investigación en el área de la robótica industrial. La principal línea de investigación apunta a equipar los sistemas de control con suficiente inteligencia y capacidad de solución de problemas. Esto se obtiene aplicando las técnicas de inteligencia artificial.

Diferentes técnicas de inteligencia artificial son usadas para proporcionar al robot flexibilidad e inteligencia que pueda operar en ambiente dinámico y en presencia de incertidumbre. Esas técnicas pertenecen a tres áreas de la inteligencia artificial: Aprendizaje, razonamiento y solución de problemas [Russell, *et. al.*, 2003]. Entre los diversos algoritmos de aprendizaje, el aprendizaje inductivo es el más ampliamente usado en robótica, en el que el robot aprende de ejemplos preseleccionados [Michalski, *et al.*,

1983]. Paradigmas típicos de razonamiento en robótica incluyen el razonamiento difuso, principalmente usado en la planeación bajo incertidumbre, razonamiento espacial y razonamiento temporal. Las técnicas más comúnmente usadas en robótica para la solución de problemas son: El razonamiento “*means-end*”, razonamiento heurístico y el modelo “*blackboard*” (BB).

Otra solución al control de robots en ambientes no conocidos, consiste en introducir el operador en el lazo de control de tal manera que el robot es controlado remotamente. El éxito de un sistema tele operado consiste de una buena retroalimentación interactuando con el medio ambiente, ya sea viendo, tocando y aplicando su fuerza. La mayor desventaja de un sistema tele operado involucra detalles de la transmisión cuando la distancia entre el operador y el robot es significativamente grande, como en aplicaciones en el espacio. Algunas investigaciones han explorado soluciones a este problema. Otra solución es la tele-programación, en donde el operador manda comandos de alto nivel y el robot realiza la tarea en control de lazo cerrado. Recientemente, gran atención se ha consagrado a la tele-operación basada en el internet [Niemeyer, *et al.*, 2004].

Con la rápida modernización del primer mundo, se requieren nuevos tipos de servicios para mantener una cierta calidad de vida. Un nuevo y prometedor sector en robótica está haciendo investigación para servir al ser humano. Los robots industriales tradicionales están siendo modificados para responder a este nuevo mercado, Robots quirúrgicos, robots para alimentación de líquidos, robots para seleccionar productos, robots para alimentar personas, robots para la rehabilitación, robots como mascotas, etc. Dos de las aplicaciones de servicio más relevantes de los robots manipuladores son: los robots médicos y los robots en rehabilitación. Estos dos temas están llamando la atención de los investigadores actualmente en todo el mundo. En años recientes, el campo de la medicina ha incorporado en su práctica a los robots, no para remplazar personal capacitado como doctores y enfermeras, sino para asistir en trabajo rutinario y tareas de precisión. Los robots médicos es un campo prometedor desde 1990, a partir de ahí una gran variedad de aplicaciones han surgido: robots de laboratorio, tele cirugía, entrenamiento quirúrgico, cirugía remota, tele

medicina, tele consulta, rehabilitación, ayuda para ciegos y sordos, etc. Los Robots asistentes en operaciones, hacen posible el ajuste de prótesis milimétricas. Sin embargo, hay muchos desafíos en la implementación de robots en el campo de la medicina, debido principalmente a problemas como seguridad, precisión y rechazo para aceptar esta tecnología.

En junio del 2006 un grupo de investigadores de la Universidad de Washington, en Seattle, a 2000 kilómetros al sur de *Rangeland Nort de Simi Valley*, al Sudoeste de California, hacen las pruebas exitosas de un prototipo para realizar cirugías a distancia. Las pruebas no se hacen sobre un ser humano, sino sobre piezas hechas de látex, simulando los órganos humanos. El telemando quirúrgico trata de salvar la vida llevando el hospital al paciente en lugar de traer al paciente al hospital. [Rosen *et al.*, 2006].

La actividad de los robots en el campo de la rehabilitación inicia en 1960, actividad que se desarrolló lentamente a través de los años hasta llegar al punto de tener el primer producto exitoso y comercial. Actualmente, el concepto de robot de rehabilitación incluye un amplio rango de dispositivos mecatrónicos, que van desde los miembros artificiales hasta los robots para la terapia de la rehabilitación de apoyo o para proporcionar asistencia en hospitales. El campo de la robótica de rehabilitación está menos desarrollado que el de robots industriales.

Aunque la robótica es un hecho científico, falta mucho por investigar para llegar a tener un robot con las características descritas por los primeros escritores de ciencia ficción. Independientemente del tipo de robot y del área de aplicación, las trayectorias que deben de seguir los robots, llegan a ser de vital importancia. Aunque existe un gran número de aplicaciones sobre el problema de generación de trayectorias con movimientos suaves, las investigaciones sobre la planeación de trayectorias, sigue siendo un campo de oportunidades. Enseguida se describen de manera cronológica algunos trabajos relacionados a la investigación propuesta en esta tesis.

En [Kang, *et al.*, 1985], se presenta un método para la obtención de trayectorias para controlar un brazo mecánico en un tiempo mínimo dada una ruta geométrica deseada, con restricciones del par. Además, la trayectoria puede tocar el límite de la región admisible. Convencionalmente, el algoritmo de control está dividido en dos partes, planeación de trayectorias y seguimiento de la ruta. El robot usado en el ejemplo es un robot simple de dos grados de libertad, una articulación de rotación y otra prismática.

En [Olomski, 1989], se describe un sistema de control de un robot para el seguimiento de una trayectoria deseada con alta precisión. La estructura de control está separada en planeación y generación de trayectorias, y en el control de lazo cerrado. Se hace uso de una Spline cúbica para suavizar los cambios de aceleración. El tiempo de recorrido es programado por el usuario o por optimización, tomando en cuenta las restricciones dinámicas del robot. Se hace uso de un procesador especial de punto flotante (TMS32010) para la generación de la trayectoria en coordenadas cartesianas y transformación de las referencias dentro del nivel de las uniones con una velocidad de muestreo de 800Hz. El control de lazo cerrado es responsable de seguir las señales de referencia ordenadas. Se hace uso de un sensor óptico de captura rápida para suprimir el error de posición dinámico causado por el control, el modelo inverso del robot y la holgura de los engranajes.

En [Tagawa, *et al.*, 1993] se presenta un esquema de procesamiento en paralelo para el control de robots manipuladores, basado procesadores digitales de señal (DSP) de propósito general. Donde se considera que el par mecánico de cada unión será tratado por un DSP independientemente.

En [Kim, *et al.*, 1997], se presenta un método de planeación de trayectorias con mínimo tiempo de recorrido y con un esquema de control de seguimiento para robots manipuladores. Primeramente se encuentra el tiempo mínimo de recorrido de la trayectoria por la optimización de las trayectorias de las uniones con polinomios cúbicos usando estrategias evolutivas. Posteriormente se sintonizan los parámetros del controlador de modo deslizante, para que el robot manipulador siga las trayectorias encontradas previamente. El



problema se formula como un problema de control de tiempo óptimo con restricciones de límite de velocidad, aceleración y Jerk. La aplicación de este método se hizo sobre un manipulador de dos grados de libertad.

En [Monteiro, *et al.*, 1999], se aplican algoritmos genéticos para la planeación de las trayectorias de un brazo de robot llamado Jeca de cinco grados de libertad. Primeramente se hace la planeación de la trayectoria en el plano cartesiano evitando obstáculos. Posteriormente, la planeación en el espacio de las uniones se implementa usando los algoritmos genéticos clásicos con algunas modificaciones. Esta parte está dividida en dos: Posicionamiento inicial e incremento de posición. El posicionamiento inicial tiene el propósito de localizar al efector final en el primer punto de la trayectoria y el incremento de posición tiene el propósito de mover al efector final al siguiente punto de la trayectoria. El resultado es una trayectoria completa planeada con algoritmos genéticos que demuestra la flexibilidad de esta técnica de inteligencia artificial. El objetivo es determinar los ángulos de las uniones del robot, esto es, encontrar el modelo de la cinemática inversa del robot. Este modelo se encuentra de acuerdo a la convención de Denavit-Hartenberg y se válida para tres grados de libertad.

En [Ma, *et al.*, 2000], se propuso un esquema de control de tiempo óptimo de la cinemática de un manipulador redundante para seguir una ruta geométrica definida, con límite del par en las uniones. Este esquema hace uso completo de la redundancia para incrementar la velocidad de traslado sobre la ruta, resolviendo el problema de la planeación de la trayectoria en tiempo mínimo mediante el análisis en el plano de fase y técnicas de programación lineal. La simulación se ejecutó sobre un manipulador plano rotativo de tres grados de libertad, mostrando que la redundancia del manipulador fue completamente utilizada para incrementar la velocidad de traslado sobre la ruta propuesta y que en alguna de las uniones se llega al límite del par, mientras la tarea de traslado en tiempo mínimo se cumple. El movimiento del efector final del manipulador redundante se dio como tarea principal y como sub tareas la eliminación de singularidades, la eliminación de obstáculos o límites en las uniones, minimización de la velocidad en las uniones o par de las uniones se

realizó al mismo tiempo. Para demostrar la efectividad del modelo propuesto se hace simulación sobre un manipulador de tres grados de libertad con uniones rotativas. No se considera la orientación del efector final y se da un grado de redundancia. Los tres eslabones se consideraron idénticos, modelados como barras delgadas y uniformes de longitud de 1 m., masa de 10 Kg., y el límite del par de las uniones de 10Nm. La trayectoria propuesta fue una línea recta.

En [Amirabdollahian, *et al.*, 2002], se estudia la creación de movimientos parecidos a los humanos en las interfaces hápticas, basado en el estudio del movimiento del brazo humano. Estos movimientos tienen por objeto capacitar los movimientos de las extremidades superiores de los pacientes que pierden las funciones de manipulación después de un accidente. Se presenta un modelo que utiliza polinomio de séptimo grado para definir una trayectoria y controlar el brazo robótico para alcanzar movimientos de sacudida mínima (Jerk).

En [Juárez, 2004], se propone mejorar simultáneamente las dimensiones de los eslabones y el plan de movimiento mediante polinomios de octavo grado para lograr el tiempo óptimo sobre una trayectoria. Para la optimización del eslabonamiento, se establece un modelo matemático distinto a los tradicionales porque ofrece nuevos parámetros que permiten medir la aptitud del manipulador de un robot dedicado a transportar discos de silicio en un proceso de ambiente limpio. Como perfil de trayectorias se escoge un polinomio de octavo grado, poco apreciado por sus desventajas de ser un polinomio de grado alto y desplazado en el estudio por otros perfiles más populares, pero menos suaves en sus transiciones de aceleración. Sin embargo, es una alternativa posible, debido a la suavidad del movimiento que es importante para minimizar el desgaste de los elementos mecánicos de un manipulador empleado en cuartos limpios. El índice de control del polinomio es un escalar que deforma la curva y proporciona diferentes resultados dinámicos. En este proceso de maximización, sólo se tiene una restricción que indica hasta donde se puede mejorar. Esta limitación es el par de los motores que accionan las articulaciones. El trabajo consiste en encontrar, con ayuda del modelado matemático

denominado regiones geométricas articulares, funciones de optimización que ayuden a determinar el mejor manipulador con base en la suavidad de su movimiento.

Por razones de robustez y número de variables manejadas en el pretendido proceso de mejoramiento, como herramienta de maximización se emplea un método basado en un algoritmo genético canónico. Como resultado se encontró el tiempo de proceso más corto en el cual ni se desaprovecha ni sobrecargan los actuadores del manipulador; además, se presenta el número de ciclos de trabajo máximos que el manipulador puede realizar. Se aportan al conocimiento dos elementos: 1) Un modelo matemático nuevo y 2) La productividad máxima del manipulador mediante el mejoramiento combinado de las dimensiones de sus eslabones y de la trayectoria gobernada por un perfil de trayectoria de grado ocho. Este modelo se probó con un manipulador de tres grados de libertad (PRRRd). Donde P es una articulación independiente prismática, R una articulación independiente rotativa y Rd es una articulación dependiente de las dos rotativas anteriores, no actuada.

En [Luo, *et al.*, 2004], se trató la planeación de la trayectoria óptima de un robot manipulador basado en dos índices de eficiencia. Uno es el tiempo óptimo de traslado y el otro es la energía mecánica óptima de los actuadores. Aquí la trayectoria del robot está compuesta de una secuencia de puntos en el espacio cartesiano y dos puntos consecutivos están conectados por curvas polinomiales cúbicas. En la optimización se tiene en cuenta las restricciones de velocidad, aceleración, Jerk, fuerza/par y eliminación de colisiones para el robot manipulador. Se propone el algoritmo de programación evolutiva intensificada (*Intensified Evolutionary Programming*, IEP) para la solución del correspondiente modelo de optimización. Especialmente en el algoritmo de optimización, la definición de funciones de aptitud y la descripción de mutación, selección y cruce de operadores se discutieron en detalle. El programa para la implementación del algoritmo IEP ha sido escrito en visual C++. NET El experimento se lleva a cabo sobre un robot *Stanford* de seis ejes.

En [Kishimoto, *et al.*, 2004] se presenta un método para solución de la cinemática inversa para robot manipuladores de tres grados de libertad, usando el método de Newton para

encontrar los ángulos más convenientes para determinar la posición deseada. En condiciones de singularidad en los límites del espacio de trabajo no hay forma de solucionar la cinemática inversa por lo que proponen modificar la posición real minimizando en una millonésima la diferencia entre la posición deseada y la posición real.

En [Soto, *et al.*, 2006], se presenta una propuesta para generar trayectorias para un robot manipulador de cinco grados de libertad (*RV-2AJ de Mitsubishi*). A partir de una imagen que contenga un objeto, se le aplica un procesamiento de imagen para obtener el contorno del objeto. Con ayuda de este contorno y aplicando una interpolación por Splines cúbicas, se realizan los cálculos de la cinemática inversa para generar la trayectoria del contorno del objeto que debe de seguir el manipulador. Se desarrolló un software de simulación en 3D para las pruebas del algoritmo y la visualización de los movimientos del manipulador

En [Gasparetto y Zanotto, 2007] se presenta un método para planeación de trayectorias suaves, donde la función objetivo contiene un término proporcional a la integral del cuadrado del jerk (definido como la derivada de la aceleración) , un segundo término proporcional al tiempo de ejecución se adiciona a la función objetivo. La trayectoria se construye considerando spline de quinto orden, con restricciones de velocidad, aceleración y jerk. El algoritmo se probó en simulación.

En [Huang, *et al.*, 2007], se presenta un sistema simulador para la planificación de trayectorias con restricciones en los ángulos, la velocidad, la aceleración y el par motor en las uniones de las articulaciones, con el fin de disminuir la tercera derivada de la posición de la trayectoria (Jerk), se hace uso de un algoritmo genético para encontrar los parámetros óptimos que minimicen el Jerk, que afecta negativamente la eficiencia de los algoritmos de control y estabilización de todo el sistema del robot. Por otro lado, la importancia de minimizar el Jerk, es el de reducir las vibraciones de manipulador. En esta formulación, un enfoque genético global determina la trayectoria reduciendo al mínimo el Jerk en el espacio de las uniones del robot manipulador.

En [Liu, *et al.*, 2007] se discute el problema de asignación de tareas en sistemas paralelos basado en procesadores multi-núcleo. Se propone un algoritmo heurístico para el modelo de asignación de tareas. El resultado del algoritmo es evaluado con respecto a una búsqueda exhaustiva y un algoritmo genético, mostrando que el algoritmo propuesto puede encontrar soluciones cercanas a la óptima en un tiempo razonable, comportándose mejor que el algoritmo genético cuando el número de hilos incrementa, dado que puede encontrar soluciones en menor tiempo que el algoritmo genético.

En [Yussof, *et al.*, 2009] se propone un algoritmo genético paralelo para solución del problema de la ruta más corta, con el propósito de que el algoritmo genético sea capaz de mejorar su tiempo de cómputo. El algoritmo propuesto se implementó en un cluster MPI (message passing interface).

En [Hadjikyriacous, *et al.*, 2009] se propone un algoritmo genético paralelo usando un cluster MPI, con el objetivo de presentar una evaluación experimental sobre un conjunto de datos del mundo real, y demostrar que los algoritmos genéticos resultan ser un problema donde las técnicas de procesamiento en paralelo toman lugar. En una de las pruebas realizadas con 35 CPU's y 5000 individuos, se logró una aceleración (Speed-up) de 512 veces más rápido.

En [Gasparetto y Zanotto, 2009] se describe un algoritmo de optimización de trayectorias de robots manipuladores de 6 DOF, donde la función objetivo a minimizar es la integral del cuadrado del Jerk y el tiempo de ejecución. La trayectoria se construye considerando spline cúbicas y spline de quinto orden, con restricciones de velocidad, aceleración y jerk.

En base a lo expuesto anteriormente en el estado del arte, se ha observado que en ningún trabajo anterior (excepto en [Juárez, 2002]) se ha presentado la manipulación del polinomio

de octavo grado mediante un parámetro, el cual proporciona una infinidad de perfiles de trayectorias que, aplicado a un robot manipulador producirá diferentes desempeños de éste. En [Juárez, 2002] se aplicó el polinomio de octavo grado a un robot manipulador de tres grados de libertad con capacidad únicamente de posicionamiento en el espacio de trabajo. Se aplicó para determinar las longitudes óptimas que ese manipulador debería tener para que pudiera trasladar en el menor tiempo posible el objeto trasladado sin rebasar los límites de los pares máximos de los actuadores.

En la tesis presentada se busca determinar el parámetro de control mediante un perfil gobernado por un polinomio de grado ocho aplicado a un robot de seis grados de libertad, posicionable y orientable que (1) minimice el consumo de energía o (2) el tiempo de traslado, aprovechando al máximo los pares motores.

Para el primer caso, se escoge un tiempo de proceso fijo y se determina el parámetro de control que permite que el robot gaste menos energía, siendo esto una clara diferencia con el trabajo de [Juárez, 2002], aquí radica la primer aportación.

Para el segundo caso, la presente tesis no busca determinar las longitudes óptimas de los eslabones del manipulador, que era el caso del trabajo de [Juárez, 2002]. La estrategia de búsqueda propuesta en [Juárez, 2002] estaba basada únicamente en un algoritmo genético canónico, que englobaba tres variables principales: la relación de las dimensiones de los eslabones, el índice de control de grado ocho y el tiempo de proceso a minimizar, todo esto sometido a la restricciones de pares motores máximos entregados por los actuadores. Esa estrategia seguida, hacía que el tiempo de cálculo fuera muy grande porque el espacio de búsqueda estaba magnificado por las tres variables mencionadas. En la presente tesis se expone una estrategia de búsqueda diferente para la solución del tiempo de traslado mínimo, la cual consiste en analizar pequeñas regiones del espacio global, que permita dar indicios de por dónde se encuentra la solución y no analizar todo el espacio de búsqueda como en el caso de [Juárez, 2002], esta estrategia consiste en un método combinado de un

algoritmo genético canónico con el método de bisecciones sucesivas, aquí radica la segunda aportación.

En este trabajo se propone un algoritmo de optimización de trayectorias con movimientos suaves, minimizando el gasto de energía así como un algoritmo de optimización de trayectorias con movimientos suaves minimizando el tiempo de traslado del robot manipulador sobre una trayectoria con posición y orientación deseada de la herramienta final del manipulador. En este algoritmo de optimización de trayectorias se consideran restricciones en el espacio de las uniones de la velocidad rotacional y el par motor. Para la obtención de una trayectoria óptima con mínimo gasto de energía se propone un algoritmo genético. Para la obtención de una trayectoria óptima con mínimo tiempo de traslado se propone una combinación de un algoritmo genético con el método de bisecciones sucesivas. Además se propone paralelizar los algoritmos propuesto de minimización de tiempo de traslado y consumo de energía, con el objetivo de minimizar los tiempos de cómputo de los algoritmos propuestos.

### **1.3 Objetivos**

En este proyecto de Tesis Doctoral se plantean los siguientes objetivos.

#### **1.3.1 Objetivo General**

Desarrollar una plataforma de software para planeación de trayectorias óptimas para robots manipuladores.

#### **1.3.2 Objetivos Específicos**

1. Implementar un algoritmo para la generación de trayectorias suaves utilizando funciones polinomiales de octavo grado para realizar la representación paramétrica de las trayectorias deseadas.

2. Desarrollar un modelo de optimización de trayectorias, que permita minimizar el gasto de energía mecánica en el robot manipulador.
3. Desarrollar un modelo de optimización de trayectorias, que permita minimizar el tiempo de traslado de una trayectoria deseada en el robot manipulador.
4. Incluir una interfaz gráfica para simular la ejecución de una trayectoria dada. La interfaz gráfica incluirá la animación del movimiento del robot manipulador.
5. Implementar técnicas de procesamiento en paralelo para mejorar el tiempo de ejecución de los algoritmos de optimización en la planeación de trayectorias.

#### **1.4 Justificación**

La plataforma de software para la planeación de trayectorias óptimas presentada en esta tesis, tiene como objetivo optimizar el gasto de energía o el de optimizar el tiempo de traslado en el proceso de llevar la herramienta de un robot manipulador a través de un lugar geométrico determinado.

Esta plataforma de software contribuye con el ahorro de energía, basándose en el uso de una función polinomial de octavo grado como parámetro de las leyes de velocidad, aceleración y tiempo de una ruta, para la generación de una trayectoria con movimientos suaves. La función polinomial de octavo grado permite la generación de un gran número de trayectorias para una ruta determinada, de tal manera que se escoge la trayectoria de menor consumo de energía.

Esta plataforma, aparte de contribuir con el ahorro de energía, contribuye en el ahorro de recursos financieros a través de los recursos que se asignan para el mantenimiento preventivo y correctivo de los robots manipuladores que hagan uso de ella. Se sabe que en el cambio de sentido del movimiento se producen golpes en las articulaciones, golpes que van deteriorando a todos los componentes de las articulaciones del robot manipulador, este



deterioro es paulatino, por lo que se tiene que asignar recursos para su mantenimiento ya sea preventivo o correctivo. Como esta plataforma genera transiciones suaves en el movimiento de las articulaciones, se reducen en gran medida estos golpes, el deterioro de los componentes de la articulación prolongan su vida útil, y el gasto en mantenimiento correctivo y preventivo disminuye considerablemente.

Por otro lado, optimizar el tiempo de traslado en las tareas que realiza un robot manipulador, significa que éste está en condiciones de terminar un proceso más rápidamente. Si los robot de una línea de producción, realizan sus actividades en tiempos más cortos, significa que esta línea de producción tendrá una mayor cantidad de producto terminado en un periodo de tiempo más corto y por lo tanto se pueden tener mayores ganancias.

La mayoría de los trabajos propuestos en planeación de trayectorias, emplean perfiles de velocidad polinomiales de tercer y quinto grado, y otros el modelo de interpolación denominado Spline. La desventaja del perfil polinomial de tercer orden con respecto al perfil polinomial de octavo orden, es que el perfil polinomial de tercer orden es lineal en la segunda derivada del perfil de posición, no proporcionando continuidad en el perfil de aceleración en los cambios de velocidad. La desventaja del perfil polinomial de quinto orden con respecto al perfil polinomial de octavo orden, es que el perfil polinomial de quinto orden a pesar de proporcionar suavidad en la segunda derivada del perfil de posición, únicamente tiene una sola opción de solución, mientras que el perfil de velocidad de octavo grado, puede tener una gran cantidad de opciones.

Se ha observado que en trayectorias de robot manipuladores, los perfiles de velocidad polinomial proporcionan mayor suavidad cuando el grado del polinomio es mayor [Sullivan, *et al.*, 2000]. Pero hay que tomar en cuenta que las funciones de grado alto se vuelven inestables, por lo que se producen errores [Fu, *et al.*, 1987]. El polinomio de grado ocho no está exento de estos inconvenientes, sin embargo, en ciertos rangos funciona perfectamente [Juárez, 2004]. En el polinomio de octavo grado aparece un término llamado

*indice\_m*, que relaciona la máxima velocidad con la distancia y el tiempo de recorrido sobre la trayectoria. Este término hace que exista un gran número de trayectorias polinomiales de octavo grado diferentes. El rango de funcionamiento del *indice\_m* en el polinomio de octavo grado es de 1.458333 a 2.1875. Tomando en cuenta que una trayectoria es una ruta afectada por leyes de tiempo y velocidad. Para cada *indice\_m* existe una trayectoria diferente, donde un punto en el mismo tiempo tiene diferente velocidad y aceleración. Si se hace variar el *indice\_m* entre el rango establecido anteriormente, del valor mínimo al máximo, con incrementos de una millonésima se pueden tener 729,167 trayectorias diferentes.

La razón de utilizar un algoritmo genético en la optimización es que estos son un método global y robusto de búsqueda de las soluciones de problemas. La principal ventaja de estas características es el equilibrio alcanzado entre la eficiencia y eficacia para resolver diferentes y muy complejos problemas de grandes dimensiones. Cabe mencionar que una importante ventaja de los AG es que son intrínsecamente paralelos. La mayoría de los otros algoritmos son en serie y sólo pueden explorar el espacio de soluciones hacia una solución en una dirección al mismo tiempo, y si la solución que descubren resulta sub-óptima, no se puede hacer otra cosa que abandonar todo el trabajo hecho y empezar de nuevo. Sin embargo, ya que los AG tienen descendencia múltiple, pueden explorar el espacio de soluciones en múltiples direcciones a la vez. Si un camino resulta ser un callejón sin salida, pueden eliminarlo fácilmente y continuar el trabajo hacia caminos más prometedores, dándoles una mayor probabilidad en cada ejecución de encontrar la solución.

Las técnicas de procesamiento en paralelo tienen cada día mayor interés en mas áreas del conocimiento [Oyama *et al.*, 1990], [Stavarakakis *et al.*, 1990], [Werler y Glavitsch 1993], [Mariños *et al.* 1994], [Lemaitre y Thomas 1996], [García *et al.* 2001]. El procesamiento en paralelo consiste en resolver simultánea o concurrentemente un problema. El objetivo principal de usar técnicas de procesamiento en paralelo es el de mejorar el tiempo de computo del algoritmo de optimización propuesto en esta tesis. Hay dos mejoras que se esperan del aumento de la velocidad de cálculo asociada con el

procesamiento en paralelo: la aplicación a estudios en tiempo real y el incremento del tamaño de los problemas, [Lemaitre y Thomas 1996].

### **1.5 Contribuciones**

1. Aplicación de una estrategia evolutiva de optimización para determinar el mejor perfil de trayectoria en la minimización del gasto de energía, en el proceso de llevar la herramienta de un robot manipulador de seis grados de libertad a través de un lugar geométrico determinado.
2. Aplicación de una estrategia evolutiva de optimización en conjunto con el método de bisecciones sucesivas para determinar el mejor perfil de trayectoria en la minimización del tiempo de traslado, con el mejor aprovechamiento de los pares de los motores de las articulaciones, considerando tanto posición como orientación de la herramienta en el proceso de llevar la herramienta de un robot manipulador de seis grados de libertad a través de un lugar geométrico determinado.
3. Desarrollo de una plataforma de software para la simulación y análisis del perfil de trayectorias optimizadas y parametrizadas con una función polinomial de octavo grado, incluyendo la dinámica del manipulador.
4. Aplicación de técnicas de procesamiento en paralelo basadas en Multithreading a la optimización del perfil de trayectoria, tanto en la minimización del gasto de energía como en la minimización del tiempo en el proceso de llevar la herramienta de un robot manipulador de seis grados de libertad.

### **1.6 Descripción de los Capítulos**

En el Capítulo 1 se hace una reseña de los antecedentes y avances asociados en la robótica. Se describen los objetivos principales a lograrse con esta tesis así como las aportaciones

originales de la misma y se comenta de manera concisa el contenido de los capítulos que la conforman.

En el Capítulo 2 de esta Tesis se hace una descripción de los principios básicos para el modelado, análisis y planeación de trayectorias de los robots manipuladores, como el modelado de la cinemática, el modelado de la dinámica y el modelado de la planeación de trayectorias.

En el Capítulo 3 de esta Tesis se hace una descripción de los conceptos básicos de los algoritmos genéticos.

En el Capítulo 4 de esta Tesis se hace una descripción de conceptos básicos de procesamiento en paralelo, que serán utilizados con el objeto de optimizar los tiempos de cómputo de los algoritmos propuesto en esta tesis.

En el Capítulo 5 de esta Tesis se hace una descripción de los algoritmos propuestos para la optimización de una trayectoria con tiempo mínimo de traslado o mínimo consumo de energía.

En el Capítulo 6 se aplican los programas, modelos y herramientas desarrollados en esta tesis a diferentes casos de estudio.

En el Capítulo 7 se describen las conclusiones obtenidas de esta Tesis doctoral así como los trabajos de investigación futura que se pueden desarrollar a partir de esta investigación.

# **CAPÍTULO 2**

## **PRINCIPIOS BÁSICOS PARA EL MODELADO, ANÁLISIS Y PLANEACIÓN DE TRAYECTORIAS DE ROBOTS MANIPULADORES**

En este capítulo se describen los elementos básicos para modelar el movimiento de los robots manipuladores. Elementos como la cinemática, la dinámica y la trayectoria a seguir, son básicos para analizar la posición y orientación con respecto al tiempo, velocidad y aceleración del efector final de un robot manipulador.

La cinemática se divide en cinemática directa y cinemática inversa. La cinemática directa permite expresar la posición y orientación del efector final en función de las variables de las uniones de la estructura mecánica con respecto a un sistema de referencia, mientras que la cinemática inversa consiste en determinar las variables de las uniones correspondientes a una posición y orientación dada del efector final. Por lo tanto, el modelo que se requiere controlar para que el efector final del robot manipulador viaje sobre una trayectoria propuesta, es el modelo de la cinemática inversa.

El modelo de la dinámica del robot manipulador juega un rol muy importante para la simulación del movimiento tomando en cuenta las fuerzas necesarias para lograr el movimiento, el análisis de la estructura, así como del diseño de los algoritmos de control. La simulación del movimiento del manipulador permite probar las estrategias de control y las técnicas de planeación de trayectorias sin la necesidad de usar un sistema físico. El análisis del modelo de la dinámica, ayuda en el diseño del prototipo mecánico.

El objetivo de la planeación de la trayectoria consiste en generar las entradas de referencia para que los sistemas de control de movimiento aseguren que el manipulador

ejecute la trayectoria planeada. El usuario especifica un número de parámetros para describir la trayectoria. La planeación consiste en la generación de una secuencia de valores en el tiempo adquiridos por una función de interpolación polinomial de la trayectoria deseada.

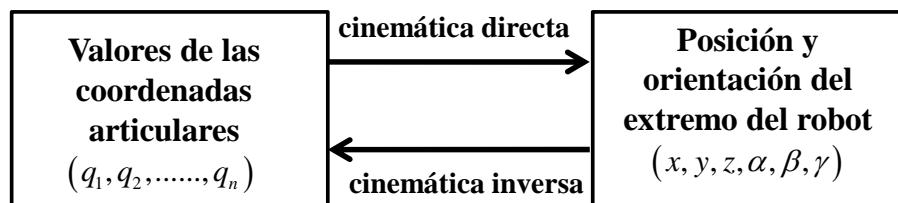
En este capítulo se describen las funciones de interpolación de trayectorias como el perfil trapezoidal, el perfil polinomial cúbico, los polinomios de interpolación Spline, el perfil polinomial de quinto grado, así como el perfil de interés de esta tesis: el perfil polinomial de octavo grado.

## **2.1 Modelado de la Cinemática de un Robot Manipulador**

En el análisis de los robots, es de gran interés la descripción analítica de su movimiento espacial como una función del tiempo, así como las relaciones entre la posición y orientación de su extremo final con los valores que toman sus coordenadas articulares. Considerando a un robot como una cadena cinemática formada por  $n$  objetos fijos o eslabones unidos entre sí, mediante articulaciones y estableciendo un sistema de coordenadas de referencia fijo situado en su base y en cada uno de sus eslabones, se puede describir la localización de cada uno de sus eslabones con respecto a la base.

Se entiende por cinemática el estudio del movimiento sin tomar en cuenta las fuerzas que lo producen [Craig. 1986]. En esta sección se describe la cinemática del robot, la cual estudia sus movimientos con respecto a un sistema de referencia. Existen dos problemas fundamentales a resolver en la cinemática del robot. El primero se conoce como el *problema de la cinemática directa* y consiste en determinar cuál es la posición y orientación del extremo final del robot con respecto a un sistema de coordenadas que se toma como referencia, conociendo los ángulos de sus articulaciones y sus parámetros geométricos. El segundo, denominado *problema de la cinemática inversa*, determina las variables articulares que debe de adoptar cada una de las articulaciones del robot para una

posición y orientación deseada del extremo final del robot [Craig. 1986]. La relación entre la cinemática directa y la cinemática inversa se ilustra en la Figura 2.1.

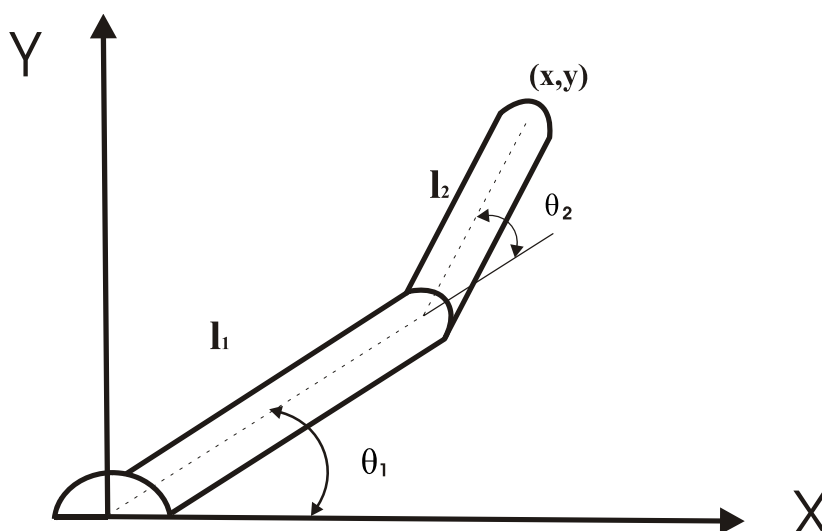


**Figura 2.1** Relación entre la cinemática directa y la cinemática inversa.

### 2.1.1 Cinemática Directa

La obtención de la posición y orientación del extremo final del robot no es complicada en robots de pocos grados de libertad, es fácil encontrarlas mediante consideraciones geométricas sencillas. Por ejemplo, en el caso del robot de la Figura 2.2 es fácil comprobar que:

$$\begin{aligned} x &= l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) \\ y &= l_1 \text{sen} \theta_1 + l_2 \text{sen}(\theta_1 + \theta_2) \end{aligned} \quad (2.1)$$



**Figura 2.2** Robot planar con dos grados de libertad.

Para robots con más grados de libertad este método puede complicarse; sin embargo, se puede plantear el método basado en la utilización de matrices de transformación homogénea que Denavit y Hartenberg propusieron en 1955 para describir la geometría espacial de los elementos de un robot, con respecto a un sistema de referencia fija, [Craig. 1986] [FU *et al.*, 1987]. El método hace uso de una matriz de transformación homogénea para describir la relación espacial entre los eslabones adyacentes del robot, por lo que el problema de la cinemática directa se reduce a encontrar una matriz de transformación homogénea de  $4 \times 4$ , que relaciona la localización del robot con respecto al sistema de coordenadas de su base.

Para describir la localización de cada eslabón en relación a sus vecinos, este método permite establecer un sistema de coordenadas ligado a cada uno de los eslabones del robot. En cada uno de los sistemas de referencia de coordenadas asignado, se definen cuatro parámetros, llamados parámetros de Denavit-Hartenberg [Craig. 1986] (ver Figura 2.3). Estos cuatro parámetros dependen únicamente de la geometría de los eslabones y de las articulaciones que los unen, dos describen al propio eslabón y dos describen la conexión entre eslabones vecinos, y son los siguientes:

$a_i$  = Es la distancia de  $\hat{Z}_i$  a  $\hat{Z}_{i+1}$  medida a lo largo de  $\hat{X}_i$ .

$\alpha_i$  = Es el ángulo de separación entre  $\hat{Z}_i$  y  $\hat{Z}_{i+1}$  medido alrededor de  $\hat{X}_i$ .

$d_i$  = Es la distancia de  $\hat{X}_{i-1}$  a  $\hat{X}_i$  medida a lo largo de  $\hat{Z}_i$ .

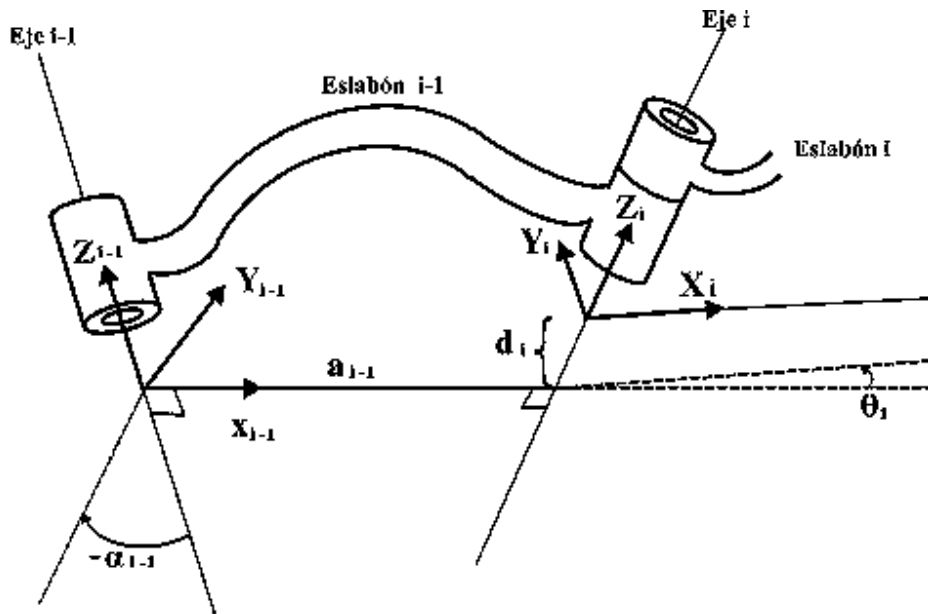
$\theta_i$  = Es el ángulo de separación entre  $\hat{X}_{i-1}$  y  $\hat{X}_i$  medido alrededor de  $\hat{Z}_i$ .

Los parámetros de Denavit-Hartenberg sirven para determinar la matriz de transformación homogénea de eslabonamiento, que relaciona la localización y la orientación del eslabón. Esta matriz que se determina con cuatro transformaciones básicas, consistentes en una sucesión de rotaciones y traslaciones que permiten relacionar el sistema de referencia del elemento (i) con el sistema de referencia del elemento (i-1). Normalmente



la matriz de transformación homogénea de  $4 \times 4$  que representa la posición y orientación relativa entre las coordenadas de los sistemas asociados a los eslabones consecutivos del robot se representa por:

$${}^{i-1}T_i = \begin{bmatrix} R_{3 \times 3} & P_{3 \times 1} \\ f_{1 \times 3} & w_{1 \times 1} \end{bmatrix} = \begin{bmatrix} \text{Rotación} & \text{Traslación} \\ \text{Perspectiva} & \text{Escala} \end{bmatrix} = \begin{bmatrix} {}^{i-1}R_i & {}^{i-1}P_{ORIG\ i} \\ 0 & 1 \end{bmatrix} \quad (2.2)$$



**Figura 2.3** Sistemas de referencias de coordenadas sujetos al eslabón  $i-1$ .

Donde  $R_{3 \times 3} = {}^{i-1}R_i$  es una matriz de rotación de  $3 \times 3$  que determina la orientación del sistema de coordenadas (i) con respecto al sistema de coordenadas (i-1).  $P_{3 \times 1} = {}^{i-1}P_{ORIG\ i}$  es un vector de translación de  $3 \times 1$  que representa la posición del origen del sistema (i) con respecto al sistema (i-1),  $f_{1 \times 3}$  una sub-matriz que representa una transformación de perspectiva y  $w_{1 \times 1}$  una sub-matriz que representa un escalado global. En robótica sólo interesará conocer las componentes de rotación y translación, considerándose las componentes de  $f_{1 \times 3}$  nulas y la de  $w_{1 \times 1}$  la unidad [Craig. 1986], [Barrientos *et al.*, 1998].

Considerando cada una de las transformaciones de estos parámetros y dado que el producto de matrices no es conmutativo, las transformaciones para representar  $\{i\}$  con respecto a  $\{i-1\}$  se han de realizar en el orden indicado por la siguiente ecuación, [Craig. 1986]:

$${}^{i-1}T_i = Rot(\hat{X}_{i-1}, \alpha_{i-1}) Trans(\hat{X}_{i-1}, a_{i-1}) Rot(\hat{Z}_i, \theta_i) Trans(\hat{Z}_i, d_i) \quad (2.3)$$

Cuando se escoge el eje de rotación sobre uno de los principales ejes del sistema, por ejemplo  $\{A\}$ , la matriz equivalente de rotación toma una de las siguientes formas, [Craig. 1986]:

$$Rot({}^A \hat{X}, \theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

$$Rot({}^A \hat{Y}, \theta) = \begin{bmatrix} \cos \theta & 0 & -\sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

$$Rot({}^A \hat{Z}, \theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

Planteando las transformaciones indicadas en (2.3) y sustituyendo en esta misma ecuación, se tiene la Ecuación (2.7).

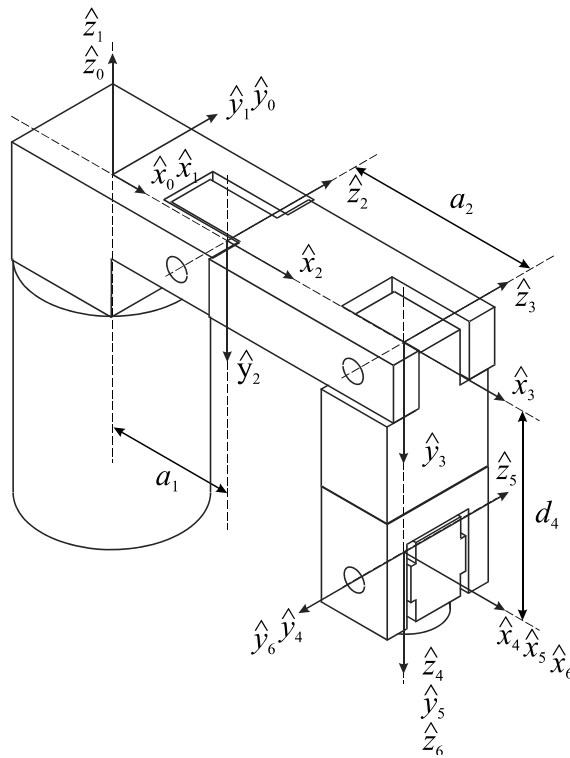
$${}^{i-1}T_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & a_{i-1} \\ \cos \alpha_{i-1} \sin \theta_i & \cos \alpha_{i-1} \cos \theta_i & -\sin \alpha_{i-1} & -\sin \alpha_{i-1} d_i \\ \sin \alpha_{i-1} \sin \theta_i & \sin \alpha_{i-1} \cos \theta_i & \cos \alpha_{i-1} & \cos \alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

En la ecuación. (2.7) se sustituyen los parámetros de Denavit y Hartenberg para obtener la matriz de transformación homogénea  ${}^{i-1}T_i$  que representa al sistema  $\{i\}$  con respecto al sistema  $\{i-1\}$ . Así  ${}^0T_1$  describe la posición y orientación del sistema de referencia fijo al primer eslabón con respecto al sistema de referencia fijo a la base. De la misma forma  ${}^1T_2$  describe la posición y orientación del sistema de referencia fijo al segundo eslabón con respecto al sistema de referencia fijo al primer eslabón. De tal forma que la posición y orientación del sistema de referencia fijo al tercer eslabón con respecto al sistema de coordenadas de la base se expresará como  ${}^0T_3$ .

Cuando se consideran todos los grados de libertad del robot se tiene que la matriz que expresa la posición y orientación del último eslabón con respecto a la base se obtiene por la multiplicación de las matrices homogéneas de cada una de las relaciones entre eslabones vecinos como muestra la ecuación (2.8). Con esta ecuación obtenemos la solución al problema de la cinemática directa de cualquier robot manipulador.

$${}^0T_n = {}^0T_1 {}^1T_2 {}^2T_3 \dots \dots {}^{n-1}T_n \quad (2.8)$$

Para ilustrar la metodología desarrollada en este trabajo, se ha propuesto un robot manipulador de seis grados de libertad. En la Figura 2.4 se muestra la estructura de este robot manipulador. En la Tabla 2.1 se muestran los parámetros de Denavit y Hartenberg del manipulador. En las ecuaciones (2.9) y (2.10) se tiene la solución de la cinemática directa para la estructura del robot manipulador. Estos resultados se obtuvieron en [Pérez, 2005].



**Figura 2.4** Asignación de los sistemas de referencia de coordenadas a los eslabones del robot a analizar.

**Tabla 2.1** Parámetros de Denavit-Hartenberg de cada uno de los eslabones del robot diseñado.

$i$	$\alpha_{i-1}$ (grados)	$a_{i-1}$ (metros)	$d_i$ (metros)	$\theta_i$ (grados)
1	0	0	0	$\theta_1$
2	-90	$a_1 = 0.138$	0	$\theta_2$
3	0	$a_2 = 0.210$	0	$\theta_3$
4	-90	0	$d_4 = 0.215$	$\theta_4$
5	90	0	0	$\theta_5$
6	-90	0	0	$\theta_6$

$${}^0_6T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.9)$$

En la Ecuación (2.10) se muestran los resultados de cada elemento de la matriz de la ecuación (2.9).

$$\begin{aligned} n_x &= c_1 c_{23} (c_4 c_5 c_6 - s_4 s_6) - c_1 s_{23} s_5 c_6 - s_1 (-s_4 c_5 c_6 - c_4 s_6) \\ n_y &= s_1 c_{23} (c_4 c_5 c_6 - s_4 s_6) - s_1 s_{23} s_5 c_6 + c_1 (-s_4 c_5 c_6 - c_4 s_6) \\ n_z &= -s_{23} (c_4 c_5 c_6 - s_4 s_6) - c_{23} s_5 c_6 \\ o_x &= c_1 c_{23} (-c_4 c_5 s_6 - s_4 c_6) + c_1 s_{23} s_5 s_6 - s_1 (-s_4 c_5 s_6 - c_4 c_6) \\ o_y &= s_1 c_{23} (-c_4 c_5 s_6 - s_4 c_6) + s_1 s_{23} s_5 s_6 + c_1 (s_4 c_5 s_6 - c_4 c_6) \\ o_z &= -s_{23} (-c_4 c_5 s_6 - s_4 c_6) + c_{23} s_5 s_6 \\ a_x &= -c_1 c_{23} c_4 s_5 - c_1 s_{23} c_5 - s_1 s_4 s_5 \\ a_y &= -s_1 c_{23} c_4 s_5 - s_1 s_{23} c_5 + c_1 s_4 s_5 \\ a_z &= s_{23} c_4 s_5 - c_{23} c_5 \\ p_x &= -c_1 s_{23} d_4 + c_1 (c_2 a_2 + a_1) \\ p_y &= -s_1 s_{23} d_4 + s_1 (c_2 a_2 + a_1) \\ p_z &= -c_{23} d_4 - s_2 a_2 \end{aligned} \quad (2.10)$$

Donde:

$$c_A = \cos(\theta_A)$$

$$c_{AB} = \cos(\theta_A + \theta_B)$$

$$s_A = \sin(\theta_A)$$

$$s_{AB} = \sin(\theta_A + \theta_B)$$

### 2.1.2 Cinemática Inversa

La solución de la cinemática inversa es importante para controlar la posición y orientación del extremo final de un robot manipulador y que éste pueda alcanzar el objetivo deseado. El método de la transformada inversa es la técnica que se usó para el cálculo de la cinemática inversa, ésta hace uso de las matrices de transformación homogénea para su solución.

Iniciamos por determinar la posición y orientación deseada que representa la ecuación (2.9). La relación existente entre la matriz que representa la posición y orientación del manipulador y estas matrices de transformación está dada por la ecuación (2.8), de esta forma se tiene que:

$${}^0T = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 \quad (2.11)$$

Posteriormente se invierte la matriz homogénea  ${}^0T_1$  y se premultiplica (2.11) por esta matriz inversa, por lo que tenemos una nueva ecuación con la variable de la primera articulación  $\theta_1$  aislada del resto de las demás articulaciones y tal vez sea posible encontrar su valor a través de las igualdades que se presenten en ella:

$$([{}^0T_1]^{-1}) {}^0T = {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 \quad (2.12)$$

Posteriormente, una vez obtenida  $\theta_1$ , se invierte la matriz homogénea  ${}^1T_2$  y se premultiplica (2.12) por esta matriz inversa, con lo que se tiene:

$$([{}^1T_2]^{-1} [{}^0T_1]^{-1}) {}^0T = {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 \quad (2.13)$$

Nuevamente se trata de encontrar  $\theta_2$ ,  $\theta_3$  o la suma de dos ángulos que nos ayude a encontrar cualquier valor articular. Se continúa de esta forma hasta encontrar todos los valores de los ángulos de las articulaciones del manipulador.

Las ecuaciones mostradas en la Tabla 2.2 son la solución a la cinemática inversa del robot manipulador usado en este trabajo de tesis. Estos resultados se obtuvieron en [Pérez, 2005].

**Tabla 2.2** Ecuaciones de la solución de la cinemática inversa del robot manipulador usado en este proyecto de tesis.

$\theta_1 = \text{arc tan} \left( \frac{p_y}{p_x} \right)$
$\theta_3 = \text{arc tan} \left( \frac{s_3}{c_3} \right)$ $s_3 = \frac{p_x^2 + p_y^2 + p_z^2 + a_1^2 - a_2^2 - d_4^2 - 2s_1a_1p_y - 2c_1a_1p_x}{-2a_2d_4}$ $c_3 = \pm \sqrt{1 - s_3^2}$
$\theta_2 = \text{arc tan} \left( \frac{K}{\pm \sqrt{K_1^2 + K_2^2 - K^2}} \right) - \text{arc tan} \left( \frac{K_1}{K_2} \right)$ $K = p_x^2 + p_y^2 + p_z^2 + a_1^2 + a_2^2 - d_4^2 - 2c_1a_1p_x - 2s_1a_1p_y$ $K_1 = -2a_1a_2 + 2s_1a_2p_y + 2c_1a_2p_x$ $K_2 = -2a_2p_z$
$\theta_5 = \text{arc tan} \left( \frac{s_5}{c_5} \right)$ $c_5 = -(c_2s_3 + s_2c_3)c_1a_x - (c_2s_3 + s_2c_3)s_1a_y + (s_2s_3 - c_2c_3)a_z$ $s_5 = \pm \sqrt{1 - c_5^2}$
$\theta_4 = \text{arc tan} \left( \frac{s_4}{c_4} \right)$ $s_4 = \frac{-s_1a_x + c_1a_y}{s_5}$

$c_4 = -\frac{(c_2c_3 - s_2s_3)c_1a_x + (c_2c_3 - s_2s_3)s_1a_y - (c_2s_3 + s_2c_3)a_z}{s_5}$
$\theta_6 = \arctan\left(\frac{s_6}{c_6}\right)$
$c_6 = ((-s_4c_3c_2 + s_4s_3s_2)c_1 + c_4s_1)o_x + ((-s_4c_3c_2 + s_4s_3s_2)s_1 - c_4c_1)o_y + (s_4c_3s_2 + s_4s_3c_2)o_z$
$s_6 = ((-s_4c_3c_2 + s_4s_3s_2)c_1 + c_4s_1)n_x + ((-s_4c_3c_2 + s_4s_3s_2)s_1 - c_4c_1)n_y + (s_4c_3s_2 + s_4s_3c_2)n_z$

## 2.2 Modelado de la Dinámica de un Robot Manipulador

La dinámica se ocupa de la relación entre las fuerzas que actúan sobre un cuerpo y el movimiento que en él se origina. El modelo dinámico de un robot tiene por objetivo conocer la relación entre el movimiento de los eslabones del robot y las fuerzas implicadas en el mismo. Esta relación se obtiene mediante el denominado modelo dinámico, que relaciona matemáticamente:

- a) La localización del robot definida por sus variables articulares o por las coordenadas de localización de su extremo y sus derivadas (velocidad y aceleración).
- b) Las fuerzas y pares aplicados en las articulaciones o en el extremo del robot.
- c) Los parámetros dimensionales del robot, como longitud, masa e inercias de sus elementos.

La obtención de este modelo para mecanismos de uno o dos grados de libertad no es excesivamente compleja, pero a medida que el número de grados de libertad aumenta, el planteamiento y obtención del modelo dinámico se complica enormemente. Por este motivo no siempre es posible obtener un modelo dinámico expresado en forma cerrada, esto es mediante una serie de ecuaciones, normalmente de tipo diferencial de segundo orden, cuya



integración permita conocer qué movimiento surge al aplicar unas fuerzas o que fuerzas hay que aplicar para obtener un movimiento determinado. El modelo dinámico debe ser resuelto entonces de manera iterativa mediante la utilización de un procedimiento numérico [Craig, 1986].

El problema de la obtención del modelo dinámico de un robot es uno de los aspectos más complejos de la robótica, este modelo dinámico es imprescindible para conseguir los siguientes fines:

- a) Simulación del movimiento del robot.
- b) Diseño y evaluación de la estructura mecánica del robot.
- c) Dimensionamiento de los actuadores.
- d) Diseño y evaluación del control dinámico del robot.
- e) Planeación de trayectorias

Este último fin es evidentemente de gran importancia, pues de la calidad del control dinámico del robot depende la precisión y velocidad de sus movimientos. La gran complejidad existente en la obtención del modelo dinámico del robot, ha motivado que se realicen ciertas simplificaciones, de manera que así pueda ser utilizado en el diseño del controlador. Es importante hacer notar que el modelo dinámico completo de un robot debe de incluir no sólo la dinámica de sus elementos, sino también la propia de sus sistemas de transmisión, de los actuadores y sus equipos electrónicos de mando. Estos elementos incorporan al modelo dinámico nuevas inercias, rozamientos, saturaciones de los circuitos electrónicos, etc., aumentando aún más su complejidad. Por último es preciso señalar que si bien en la mayor parte de las aplicaciones reales de la robótica, las cargas e inercias manejadas no son suficientes para originar deformaciones en los eslabones del robot, en determinadas ocasiones no ocurre así, siendo preciso considerar al robot como un conjunto de eslabones no rígidos. Aplicaciones de este tipo pueden encontrarse en la robótica espacial o en robots de grandes dimensiones. Este trabajo presta especial atención al modelo dinámico considerando a los eslabones del robot como rígidos y aplicando las ecuaciones iterativas de Newton-Euler para uniones rotacionales.

### 2.2.1 Distribución de Masa

El momento de inercia o inercia rotacional es una magnitud que da información de la distribución de masas de un cuerpo o un sistema de partículas alrededor de uno de sus puntos. Dado un eje arbitrario para un sistema de partículas, se define como la suma de los productos entre las masas de las partículas que componen un sistema y el cuadrado de la distancia  $r$  de cada partícula al eje escogido. El momento de inercia representa la inercia de un cuerpo a rotar. Matemáticamente se expresa como:

$$I = \sum m_i r^2 \quad (2.14)$$

Para un cuerpo de masa continua de densidad homogénea, lo anterior se generaliza como:

$$I = \int_V r^2 dm = \int_V \rho r^2 dV \quad (2.15)$$

El subíndice  $V$  de la integral indica que hay que integrar sobre todo el volumen del cuerpo. Este concepto desempeña en el movimiento de rotación un papel análogo al de masa inercial en el caso del movimiento rectilíneo y uniforme. Así, por ejemplo, la segunda ley de Newton:  $F = ma$  tiene como equivalente para la rotación:

$$\tau = I\alpha \quad (2.16)$$

donde:

$\tau$  es el momento aplicado al cuerpo.

$I$  es el momento de inercia del cuerpo con respecto al eje de rotación y

$\alpha = \frac{d^2\theta}{dt^2}$  es la aceleración angular.

La energía cinética de un cuerpo en movimiento con velocidad lineal  $v$  es  $mv^2/2$ , mientras que la energía cinética de un cuerpo en rotación con velocidad angular  $\omega$  es  $I\omega^2/2$ , donde  $I$  es el momento de inercia con respecto al eje de rotación. El tensor de inercia de un sólido rígido expresado en una base ortonormal viene dado por una matriz

simétrica de  $3 \times 3$ , como en la Ecuación (2.17) y representa la inercia rotacional de un sólido rígido. El tensor de inercia se forma a partir de los momentos de inercia ( $I_{xx}, I_{yy}, I_{zz}$ ) según tres ejes perpendiculares y tres productos de inercia ( $I_{xy} = I_{yx}, I_{yz} = I_{zy}, I_{zx} = I_{xz}$ ). Para un sólido rígido tridimensional pueden definirse momentos de inercia según diversos ejes o bases, en particular pueden definirse tres ejes perpendiculares independientes llamados X, Y y Z [Craig, 1986].

$$I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \quad (2.17)$$

Donde los elementos escalares de la matriz están dados por:

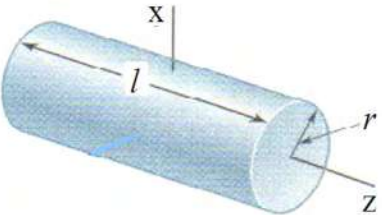
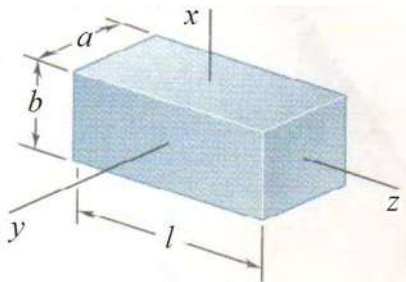
$$\begin{aligned} I_{xx} &= \int_M d_x^2 dm = \int_V \rho (y^2 + z^2) dx dy dz \\ I_{yy} &= \int_M d_y^2 dm = \int_V \rho (z^2 + x^2) dx dy dz \\ I_{zz} &= \int_M d_z^2 dm = \int_V \rho (x^2 + y^2) dx dy dz \\ I_{xy} &= I_{yx} = \int_M (-xy) dm = \int_V (-\rho xy) dx dy dz \\ I_{yz} &= I_{zy} = \int_M (-yz) dm = \int_V (-\rho yz) dx dy dz \\ I_{zx} &= I_{xz} = \int_M (-zx) dm = \int_V (-\rho zx) dx dy dz \end{aligned}$$

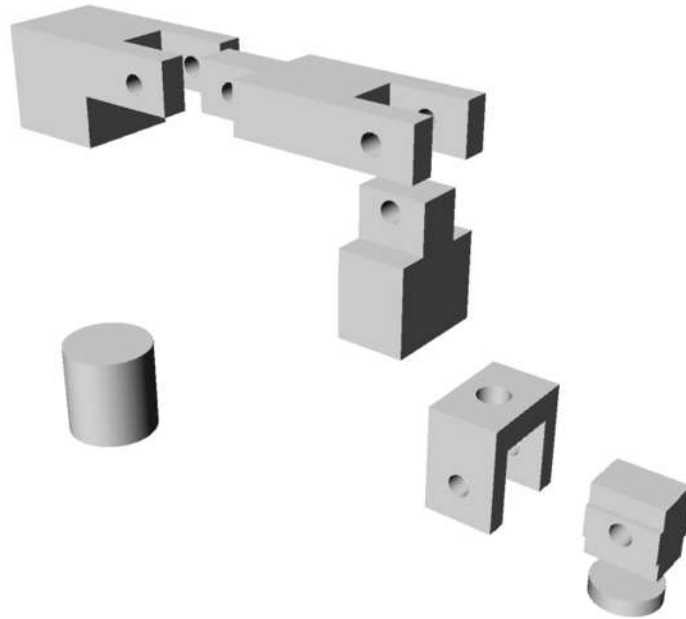
El cálculo de los tensores de inercia se facilita si los eslabones del robot manipulador se pueden descomponer en formas geométricas que coincidan con las mostradas en la Tabla 2.3. Los eslabones del robot manipulador considerado en este trabajo se muestran en la Figura 2.5. La Tabla 2.3 muestra formas geométricas con la fórmula de los momentos y productos de inercia respectivos. El tensor de inercia de una articulación obtenido por varias de estas formas geométricas comunes puede calcularse, con los momentos y/o

productos de inercia de cada una de sus partes y posteriormente haciendo la suma algebraica de todo el conjunto [Craig, 1986]. Enseguida se describen los pasos a seguir para calcular el tensor de inercia de cada eslabón:

- i) Determinar un prisma rectangular que cumpla con la longitud y anchura de la articulación.
- ii) Separar partes, (prismas y/o cilindros) hasta dejar la forma de la articulación.
- iii) Determinar las dimensiones de cada parte, así como su centro de gravedad.
- iv) Calcular los momentos de inercia de cada parte.
- v) El momento de inercia de cada parte se traslada al centro de gravedad del eslabón.
- vi) Hacer la suma algebraica de los tensores de inercia de cada parte. La sumatoria será el tensor de inercia del eslabón.

**Tabla 2.3** Momentos de inercia de masa de dos formas geométricas comunes.

Cuerpo	Momento de inercia
	$I_x = \frac{1}{4}Mr^2 + \frac{1}{12}Ml^2$ $I_y = I_x$ $I_z = \frac{1}{2}Mr^2$
	$I_x = \frac{1}{12}M(b^2 + l^2)$ $I_y = \frac{1}{12}M(a^2 + l^2)$ $I_z = \frac{1}{12}M(a^2 + b^2)$ $M = \text{Masa}$

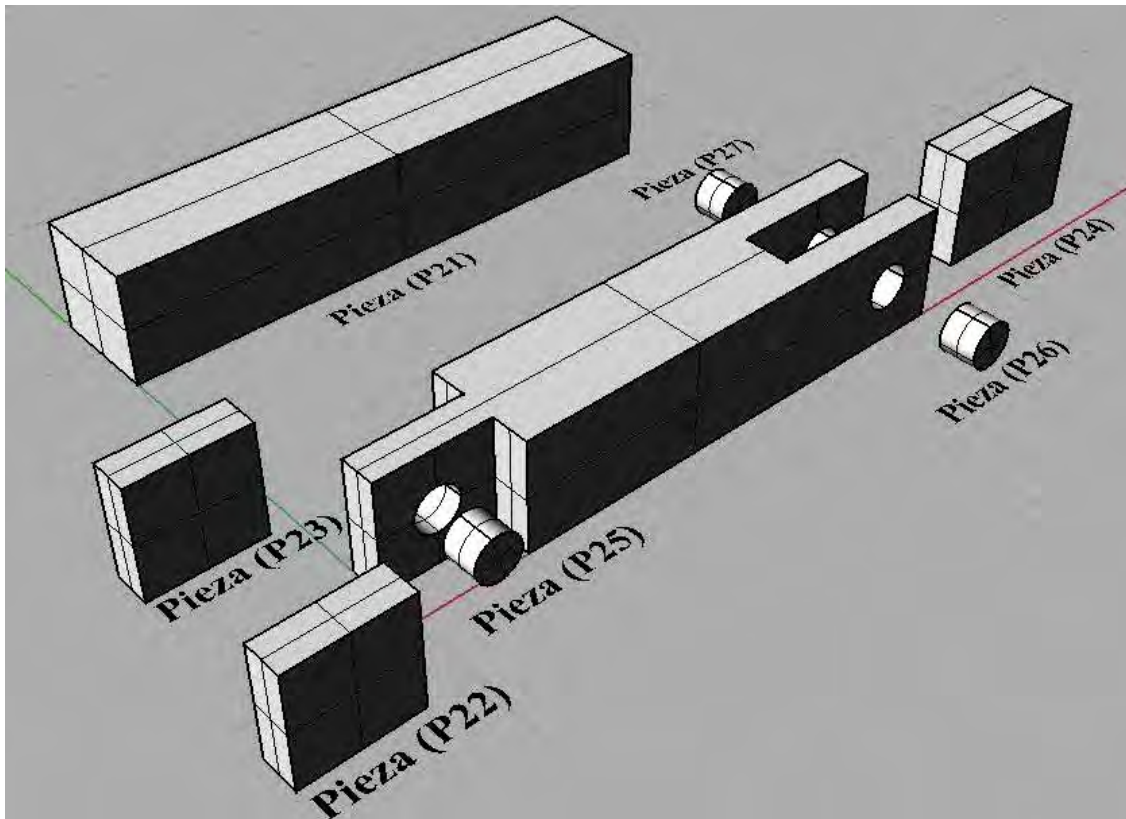


**Figura 2.5** Eslabones del robot manipulador analizado en esta tesis.

Para ilustrar este procedimiento, enseguida se presenta en detalle el cálculo del tensor de inercia del eslabón 2. En la Figura 2.6 se muestran las partes para calcular el tensor de inercia de la articulación 2 del robot manipulador y en la Figura 2.7 se muestran las dimensiones de estas partes. A un prisma rectangular (Pieza P21) se le quitan tres partes en forma de prisma (Pieza P22, Pieza P23, Pieza P24) y tres partes en forma de cilindro (Pieza P25, Pieza P26, Pieza P27).

Para cada pieza se determinan sus dimensiones, el sistema de coordenadas del tensor de inercia se pone coincidente con su centro de gravedad y con la misma orientación que el sistema de coordenadas del eslabón.

El material usado en los eslabones para el análisis del robot de este trabajo fue acero. La densidad del acero es de  $\rho_A = 7800 \text{ Kg}/m^3$ . Primeramente se hace el cálculo del volumen de cada una de las piezas del eslabón, así como el volumen total, de acuerdo a las dimensiones presentadas en la Figura 2.7.



**Figura 2.6** Partes del eslabón 2 para el cálculo del tensor de inercia.

$$V_{P21} = 0.26m * 0.05m * 0.05m$$

$$V_{P22} = V_{P23} = 0.05m * 0.05m * 0.0175m$$

$$V_{P24} = 0.07m * 0.05m * 0.02m$$

$$V_{P25} = V_{P26} = V_{P27} = \pi * r^2 * l = 3.1416 * 0.0001m^2 * 0.015m$$

Posteriormente se hace el cálculo de la masa del eslabón pieza por pieza, donde la masa es igual al producto del volumen por la densidad del material .

$$M_{P21} = V_{P21} * \rho_A = 5.07Kg$$

$$M_{P22} = V_{P22} * \rho_A = 0.3413kg$$

$$M_{P23} = V_{P23} * \rho_A = 0.4213Kg$$

$$M_{P24} = V_{P24} * \rho_A = 0.546Kg$$

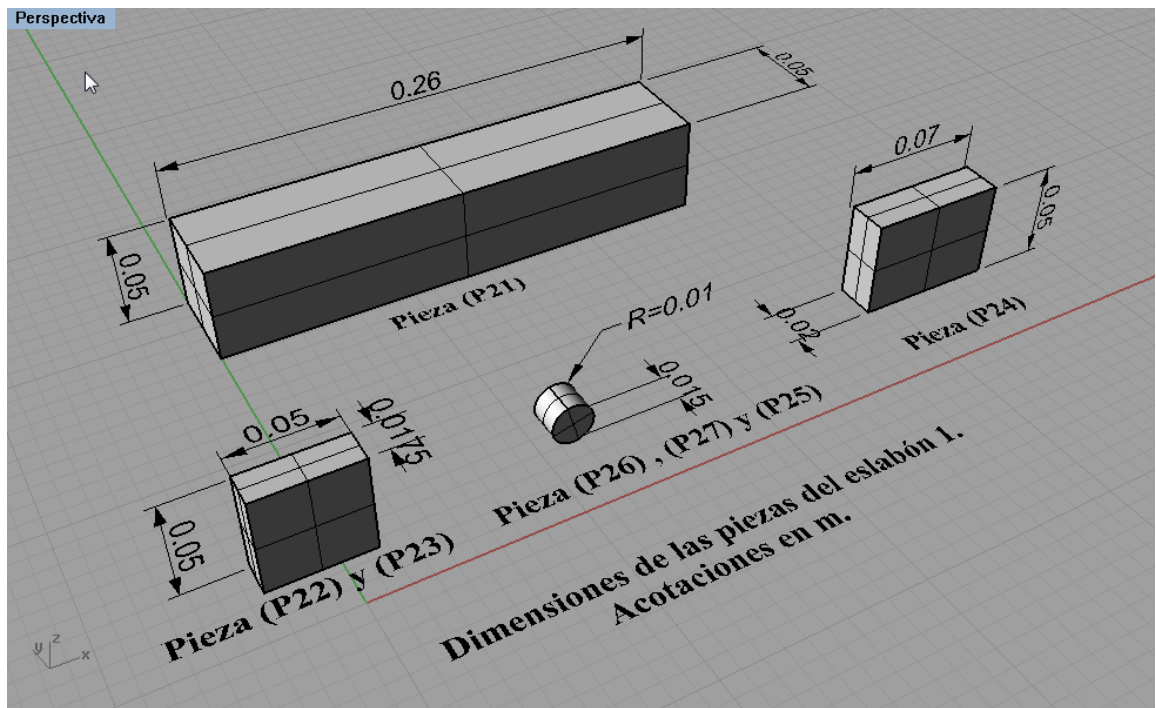
$$M_{P25} = V_{P25} * \rho_A = 0.0368Kg$$

$$M_{P26} = V_{P26} * \rho_A = 0.0368Kg$$

$$m_{P27} = V_{P27} * \rho_A = 0.0368Kg$$

Del tal forma que la masa del eslabón es:

$$M_{e2} = M_{P21} - M_{P22} - M_{P23} - M_{P24} - M_{P25} - M_{P26} - M_{P27} = 3.7312Kg$$



**Figura 2.7** Dimensiones de las partes del eslabón 2.

El centro de gravedad de cada una de las piezas del eslabón 2 se determinó gráficamente, ver la Figura 2.8. Con las coordenadas del centro de gravedad de cada pieza se hace el cálculo del centro de gravedad del eslabón  $CG_{P2n} = [x \quad y \quad z]$ .

$$CG_{P21} = [0.105 \quad 0 \quad 0]$$

$$CG_{P22} = [0 \quad 0 \quad 0.01625]$$

$$CG_{P23} = [0 \quad 0 \quad -0.01625]$$

$$CG_{P24} = [0.2 \quad 0 \quad 0]$$

$$CG_{P25} = [0 \quad 0 \quad 0]$$

$$CG_{P26} = [0.21 \quad 0 \quad 0.0175]$$

$$CG_{P27} = [0.21 \quad 0 \quad -0.0175]$$

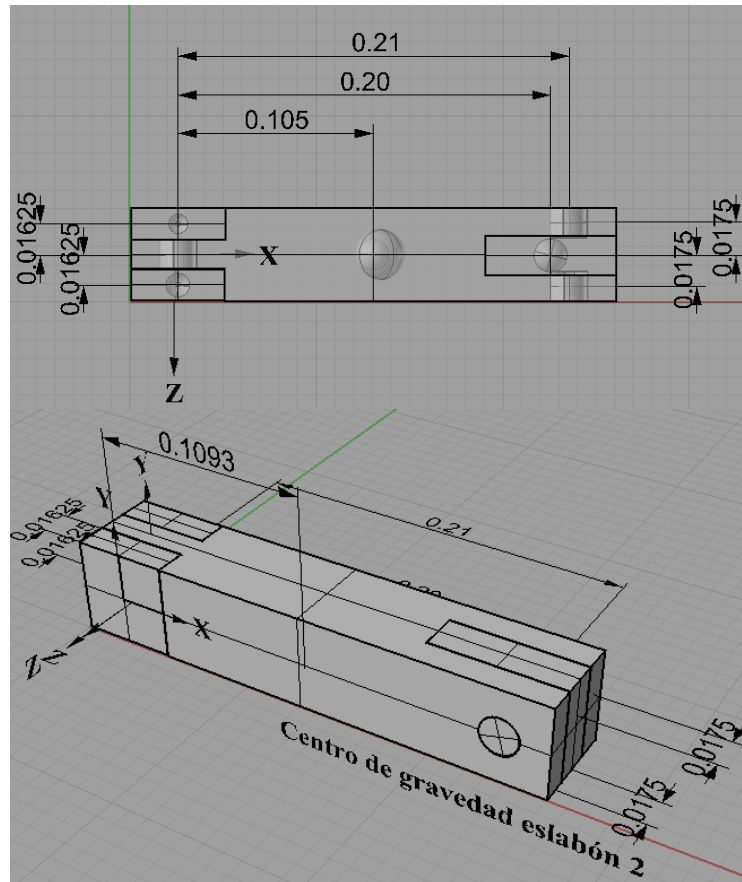
Por lo tanto el centro de gravedad del eslabón 2  $CG_{e2} = [x_{e2} \quad y_{e2} \quad z_{e2}]$  es:

$$x_{e2} = \frac{(CG_{P21x} * M_{P21}) - (CG_{P22x} * M_{P22}) - (CG_{P23x} * M_{P23}) - (CG_{P24x} * M_{P24}) - (CG_{P25x} * M_{P25}) - (CG_{P26x} * M_{P26}) - (CG_{P27x} * M_{P27})}{Me2} = 0.1093m$$

$$y_{e2} = \frac{(CG_{P21y} * M_{P21}) - (CG_{P22y} * M_{P22}) - (CG_{P23y} * M_{P23}) - (CG_{P24y} * M_{P24}) - (CG_{P25y} * M_{P25}) - (CG_{P26y} * M_{P26}) - (CG_{P27y} * M_{P27})}{Me2} = 0$$

$$z_{e2} = \frac{(CG_{P21z} * M_{P21}) - (CG_{P22z} * M_{P22}) - (CG_{P23z} * M_{P23}) - (CG_{P24z} * M_{P24}) - (CG_{P25z} * M_{P25}) - (CG_{P26z} * M_{P26}) - (CG_{P27z} * M_{P27})}{Me2} = 0$$





**Figura 2.8** Centros de gravedad de las piezas del eslabón 2 (acotaciones en m).

Los momentos de inercia de la pieza (P21) son:

$$i_{xx_{p21}} = \frac{1}{12} M_{p21} (b^2 + l^2)$$

$$i_{yy_{p21}} = \frac{1}{12} M_{p21} (a^2 + l^2)$$

$$i_{zz_{p21}} = \frac{1}{12} M_{p21} (a^2 + b^2)$$

Donde:

$$M_{p21} = 5.07 \text{ Kg}$$

$$a = 0.05 \text{ m}$$

$$b = 0.26 \text{ m}$$

$$l = 0.05 \text{ m}$$

Por lo tanto la matriz del tensor de inercia en el centro de gravedad de la pieza (P21) es:

$$I_{P21} = \begin{bmatrix} ixx_{P21} & 0 & 0 \\ 0 & iyy_{P21} & 0 \\ 0 & 0 & Izz_{P21} \end{bmatrix}$$

Como el sistema de coordenadas de referencia del eslabón 2 tiene el mismo sentido del sistema de referencia del tensor de inercia, éste se traslada al centro de gravedad del eslabón 2 haciendo uso del teorema de ejes paralelos [Beer *et al.*, 2005]. El centro de gravedad del eslabón 2 es  $CG_{e2} = [0.1093 \ 0 \ 0]$  y el centro de gravedad de la pieza (P21) es  $CG_{P21} = [0.105 \ 0 \ 0]$ . Por el teorema de ejes paralelos o teorema de Steiner se tiene:

$$ixx_{P21CG} = ixx_{P21} - M_{P21}(y^2 + z^2)$$

$$iyy_{P21CG} = iyy_{P21} - M_{P21}(x^2 + z^2)$$

$$izz_{P21CG} = izz_{P21} - M_{P21}(y^2 + x^2)$$

Donde:

$x = 0.0043 \text{ m}$  es la distancia que hay del centro de gravedad de la pieza P21 al centro de gravedad del eslabón 2 sobre el eje  $x$ .

$y = 0$  es la distancia que hay del centro de gravedad de la pieza P21 al centro de gravedad del eslabón 2 sobre el eje  $y$ .

$z = 0$  es la distancia que hay del centro de gravedad de la pieza P21 al centro de gravedad del eslabón 2 sobre el eje  $z$ .

Por lo tanto el tensor de inercia de la pieza P21 en el centro de gravedad del eslabón 2 es:

$$I_{P21CG} = \begin{bmatrix} i_{xx_{P21CG}} & 0 & 0 \\ 0 & i_{yy_{P21CG}} & 0 \\ 0 & 0 & i_{zz_{P21CG}} \end{bmatrix}$$

Aplicando las fórmulas del prisma rectangular de la Tabla 2.3 a la pieza (P22), los momentos de inercia de esta pieza (P22) son:

$$i_{xx_{P22}} = \frac{1}{12} M_{P22} (b^2 + l^2)$$

$$i_{yy_{P22}} = \frac{1}{12} M_{P22} (a^2 + l^2)$$

$$i_{zz_{P22}} = \frac{1}{12} M_{P22} (a^2 + b^2)$$

Donde:

$$M_{P22} = 0.3413 \text{Kg}$$

$$a = 0.05 \text{ m}$$

$$b = 0.05 \text{ m}$$

$$l = 0.0175 \text{ m}$$

Por lo tanto la matriz del tensor de inercia en el centro de gravedad de la pieza (P22) es:

$$I_{P22} = \begin{bmatrix} i_{xx_{P22}} & 0 & 0 \\ 0 & i_{yy_{P22}} & 0 \\ 0 & 0 & i_{zz_{P22}} \end{bmatrix}$$

El centro de gravedad del eslabón 2 es  $CG_{e2} = [0.1093 \ 0 \ 0]$  y el centro de gravedad de la pieza (P22) es  $CG_{P22} = [0 \ 0 \ 0.01625]$ . Por el teorema de ejes paralelos se tiene:

$$i_{xx_{P22CG}} = i_{xx_{P22}} - M_{P21}(y^2 + z^2)$$

$$i_{yy_{P22CG}} = i_{yy_{P22}} - M_{P22}(x^2 + z^2)$$

$$i_{zz_{P22CG}} = i_{zz_{P22}} - M_{P22}(y^2 + x^2)$$

Donde:

$x = 0.1093 \text{ m}$  es la distancia que hay del centro de gravedad de la pieza P22 al centro de gravedad del eslabón 2 sobre el eje  $x$ .

$y = 0$  es la distancia que hay del centro de gravedad de la pieza P22 al centro de gravedad del eslabón 2 sobre el eje  $y$ .

$z = 0.01625 \text{ m}$  es la distancia que hay del centro de gravedad de la pieza P22 al centro de gravedad del eslabón 2 sobre el eje  $z$ .

Por lo tanto el tensor de inercia de la pieza P22 en el centro de gravedad del eslabón 2 es:

$$I_{P22CG} = \begin{bmatrix} i_{xx_{P22CG}} & 0 & 0 \\ 0 & i_{yy_{P22CG}} & 0 \\ 0 & 0 & i_{zz_{P22CG}} \end{bmatrix}$$

El tensor de inercia de la pieza (P23) es igual al tensor de inercia de la pieza (P22). Las dimensiones de la figura son iguales, la única diferencia está en la posición con respecto al centro de gravedad del eslabón 1, esto hace que la distancia de  $z$  sea negativa, pero como se eleva al cuadrado no afecta. Por lo tanto la matriz del tensor de inercia de la pieza (P23) en el centro de gravedad del eslabón 2 es:

$$I_{P23CG} = I_{P22CG}$$

Los momentos de inercia de la pieza (P24) son:

$$i_{xx_{P24}} = \frac{1}{12} M_{P24} (b^2 + l^2)$$

$$i_{yy_{P24}} = \frac{1}{12} M_{P24} (a^2 + l^2)$$

$$i_{zz_{P24}} = \frac{1}{12} M_{P24} (a^2 + b^2)$$

Donde:

$$M_{P24} = 0.546 \text{Kg}$$

$$a = 0.05 \text{ m}$$

$$b = 0.07 \text{ m}$$

$$l = 0.02 \text{ m}$$

Por lo tanto la matriz del tensor de inercia en el centro de gravedad de la pieza (P24) es:

$$I_{P24} = \begin{bmatrix} i_{xx_{P24}} & 0 & 0 \\ 0 & i_{yy_{P24}} & 0 \\ 0 & 0 & i_{zz_{P24}} \end{bmatrix}$$

El centro de gravedad del eslabón 2 es  $CG_{e2} = [0.1093 \ 0 \ 0]$  y el centro de gravedad de la pieza (P24) es  $CG_{P24} = [0.2 \ 0 \ 0]$ . Por el teorema de ejes paralelos se tiene:

$$i_{xx_{P24CG}} = i_{xx_{P24}} - M_{P24} (y^2 + z^2)$$

$$i_{yy_{P24CG}} = i_{yy_{P24}} - M_{P24} (x^2 + z^2)$$

$$i_{zz_{P24CG}} = i_{zz_{P24}} - M_{P24} (y^2 + x^2)$$

Donde:

$x = 0.0907 \text{ m}$  es la distancia que hay del centro de gravedad de la pieza P24 al centro de gravedad del eslabón 2 sobre el eje  $x$

$y = 0$  es la distancia que hay del centro de gravedad de la pieza P24 al centro de gravedad del eslabón 2 sobre el eje  $y$

$z = 0$  es la distancia que hay del centro de gravedad de la pieza P24 al centro de gravedad del eslabón 2 sobre el eje  $z$

Por lo tanto el tensor de inercia de la pieza P24 en el centro de gravedad del eslabón 2 es:

$$I_{P24CG} = \begin{bmatrix} i_{xx_{P24CG}} & 0 & 0 \\ 0 & i_{yy_{P24CG}} & 0 \\ 0 & 0 & i_{zz_{P24CG}} \end{bmatrix}$$

Los momentos de inercia de la pieza (P25) son:

$$i_{xx_{P25}} = \frac{1}{4} M_{P25} r^2 + \frac{1}{12} M_{P25} l^2$$

$$i_{yy_{P25}} = i_{xx_{P25}}$$

$$i_{zz_{P25}} = \frac{1}{2} M_{P25} r^2$$

Donde:

$$M_{P25} = 0.0368 \text{ Kg}$$

$$r = 0.01m$$

$$l = 0.015 m$$

Por lo tanto la matriz del tensor de inercia en el centro de gravedad de la pieza (P25) es:

$$I_{P25} = \begin{bmatrix} ixx_{P25} & 0 & 0 \\ 0 & iyy_{P25} & 0 \\ 0 & 0 & Izz_{P25} \end{bmatrix}$$

El centro de gravedad del eslabón 2 es  $CG_{e2} = [0.1093 \ 0 \ 0]$  y el centro de gravedad de la pieza (P25) es  $CG_{P25} = [0 \ 0 \ 0]$ . Por el teorema de ejes paralelos se tiene:

$$ixx_{P25CG} = ixx_{P25} - M_{P25}(y^2 + z^2)$$

$$iyy_{P25CG} = iyy_{P25} - M_{P25}(x^2 + z^2)$$

$$izz_{P25CG} = izz_{P25} - M_{P25}(y^2 + x^2)$$

Donde:

$x = 0.1093 m$  es la distancia que hay del centro de gravedad de la pieza P25 al centro de gravedad del eslabón 2 sobre el eje  $x$

$y = 0$  es la distancia que hay del centro de gravedad de la pieza P25 al centro de gravedad del eslabón 2 sobre el eje  $y$

$z = 0$  es la distancia que hay del centro de gravedad de la pieza P25 al centro de gravedad del eslabón 2 sobre el eje  $z$

Por lo tanto el tensor de inercia de la pieza P25 en el centro de gravedad del eslabón 2 es:

$$I_{P25CG} = \begin{bmatrix} ixx_{P25CG} & 0 & 0 \\ 0 & iyy_{P25CG} & 0 \\ 0 & 0 & izz_{P25CG} \end{bmatrix}$$

Los momentos de inercia de la pieza (P26) son:

$$ixx_{P26} = \frac{1}{4} M_{P26} r^2 + \frac{1}{12} M_{P26} l^2$$

$$iyy_{P26} = ixx_{P26}$$

$$izz_{P26} = \frac{1}{2} M_{P26} r^2$$

Donde:

$$M_{P26} = 0.0368 \text{Kg}$$

$$r = 0.01 \text{m}$$

$$l = 0.015 \text{ m}$$

Por lo tanto la matriz del tensor de inercia en el centro de gravedad de la pieza (P26) es:

$$I_{P26} = \begin{bmatrix} ixx_{P26} & 0 & 0 \\ 0 & iyy_{P26} & 0 \\ 0 & 0 & izz_{P26} \end{bmatrix}$$

El centro de gravedad del eslabón 2 es  $CG_{e2} = [0.1093 \quad 0 \quad 0]$  y el centro de gravedad de la pieza (P26) es  $CG_{P26} = [0.21 \quad 0 \quad 0.0175]$ . Por el teorema de ejes paralelos se tiene:

$$ixx_{P26CG} = ixx_{P26} - M_{P26}(y^2 + z^2)$$



$$i_{yy_{P26CG}} = i_{yy_{P26}} - M_{P26}(x^2 + z^2)$$

$$i_{zz_{P26CG}} = i_{zz_{P26}} - M_{P26}(y^2 + x^2)$$

Donde:

$x = -0.1007 \text{ m}$  es la distancia que hay del centro de gravedad de la pieza P26 al centro de gravedad del eslabón 2 sobre el eje  $x$

$y = 0$  es la distancia que hay del centro de gravedad de la pieza P26 al centro de gravedad del eslabón 2 sobre el eje  $y$

$z = 0.0175$  es la distancia que hay del centro de gravedad de la pieza P26 al centro de gravedad del eslabón 2 sobre el eje  $z$

Por lo tanto, el tensor de inercia de la pieza P26 en el centro de gravedad del eslabón 2 es:

$$I_{P26CG} = \begin{bmatrix} i_{xx_{P26CG}} & 0 & 0 \\ 0 & i_{yy_{P26CG}} & 0 \\ 0 & 0 & i_{zz_{P26CG}} \end{bmatrix}$$

La pieza (P27) es igual que la pieza (P26), la única diferencia es que su centro de gravedad es diferente en signo en el eje  $z$  con la misma magnitud. Como este valor esta elevado al cuadrado no se ve afectado. Por lo tanto:

$$I_{P26CG} = I_{P27CG}$$

Después de calcular los tensores de inercia de cada una de las piezas y transportarlas al centro de gravedad del eslabón 2, se calcula el tensor de inercia del eslabón con la ecuación:

$$I_{e2} = I_{P21CG} - I_{P22CG} - I_{P23CG} - I_{P24CG} - I_{P25CG} - I_{P26CG} - I_{P27CG}$$

$$Ie2 = \begin{bmatrix} 0.0021 & 0 & 0 \\ 0 & 0.0434 & 0 \\ 0 & 0 & 0.0429 \end{bmatrix}$$

Siguiendo la misma metodología se calculan los tensores de inercia de los demás eslabones. Estos tensores de inercia son:

$$Ie1 = \begin{bmatrix} 0.0012 & 0 & 0 \\ 0 & -0.034 & 0 \\ 0 & 0 & -0.0047 \end{bmatrix}$$

$$Ie3 = \begin{bmatrix} -0.004 & 0 & 0 \\ 0 & 0.0006 & 0 \\ 0 & 0 & -0.0039 \end{bmatrix}$$

$$Ie4 = \begin{bmatrix} 0.003 & 0 & 0 \\ 0 & 0.0007 & 0 \\ 0 & 0 & 0.0029 \end{bmatrix}$$

$$Ie5 = \begin{bmatrix} 0.000065 & 0 & 0 \\ 0 & 0.000002 & 0 \\ 0 & 0 & 0.00007 \end{bmatrix}$$

$$Ie6 = \begin{bmatrix} 0.00025 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0.0004 \end{bmatrix}$$

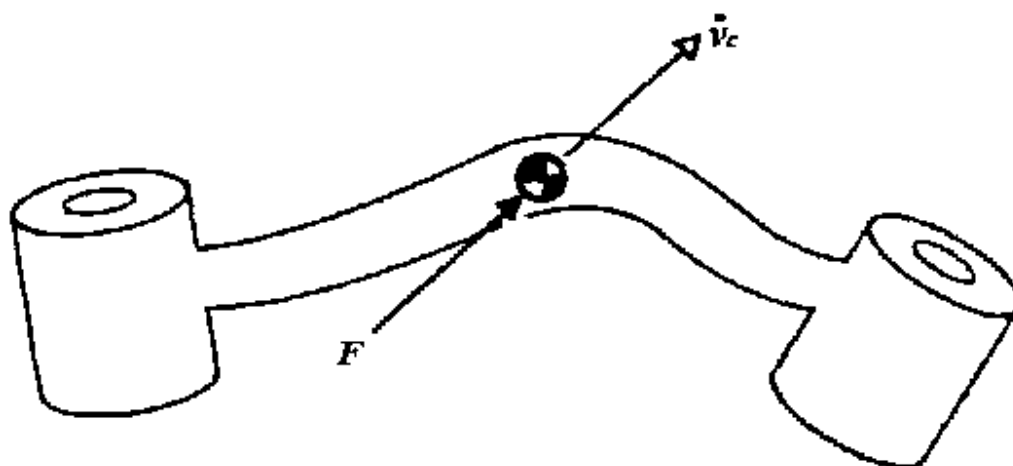
### 2.2.2 Método de Newton y Euler

En el procedimiento que aquí se presenta, se considera cada eslabón del robot manipulador como un cuerpo rígido de centro de masa y tensor de inercia conocidos, de tal manera que

la distribución de masa está completamente caracterizada. Para mover los eslabones, estos se deben de acelerar y desacelerar. Las fuerzas requeridas para tal movimiento están en función de la aceleración deseada y de la distribución de masa del eslabón. Las ecuaciones de Newton junto con las ecuaciones de Euler, describen como se relacionan las fuerzas, inercias y aceleraciones. La Figura 2.9 muestra un cuerpo rígido cuyo centro de masa tiene una aceleración  $\dot{v}_c$ . En tal situación, la fuerza  $F$  que causa la aceleración del centro de masa está dado por la ecuación de Newton:

$$F = m \dot{v}_c$$

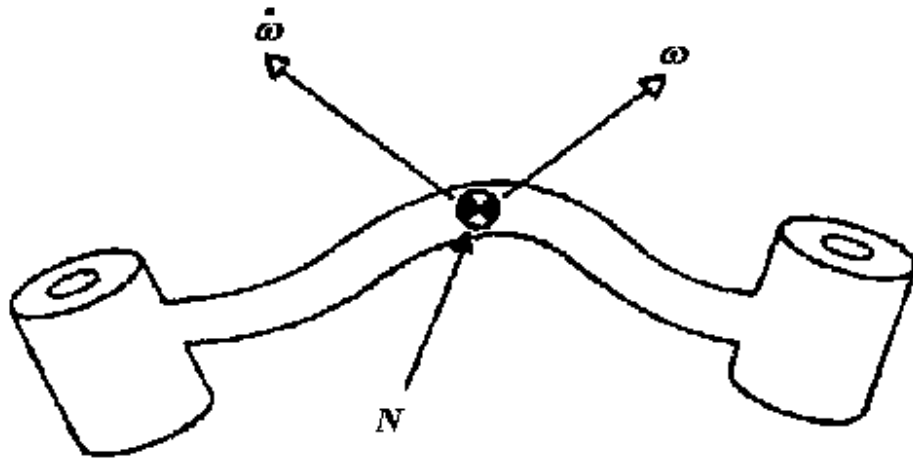
Donde  $m$  es la masa total del cuerpo.



**Figura 2.9** Fuerza  $F$  actuando en el centro de masa de un cuerpo causa una aceleración  $\dot{v}_c$ .

La Figura 2.10 muestra un cuerpo rígido rotando con velocidad angular  $\omega$  y con aceleración angular  $\dot{\omega}$ . De esta forma el momento  $N$  que debe de actuar sobre el cuerpo para provocar el movimiento, está dado por la ecuación de Euler (2.18), donde  ${}^c I$  es el tensor de inercia del cuerpo en su sistema de referencia  $\{C\}$ , cuyo origen se localiza en su centro de masa [Craig, 1986].

$$N = {}^c I \dot{\omega} + \omega \times {}^c I \omega \quad (2.18)$$



**Figura 2.10** Momento  $N$  actuando sobre un cuerpo provocando que éste rote con velocidad  $\omega$  y una aceleración  $\dot{\omega}$

Como resultado del análisis de la dinámica del robot manipulador correspondiente a una trayectoria deseada, se obtendrá el par motor de cada una de las articulaciones del manipulador. Para este análisis es necesario conocer la posición ( $\theta$ ), la velocidad rotacional ( $\dot{\theta}$ ) y la aceleración rotacional ( $\ddot{\theta}$ ) de cada una de las uniones del manipulador, así como los tensores de inercia. La posición de cada una de las uniones del manipulador se obtiene con la cinemática inversa, la velocidad con la derivada de la posición y la aceleración con la derivada de la velocidad.

Para el análisis de la dinámica del robot manipulador utilizado en esta tesis se hace uso del método iterativo de Newton-Euler. El método de Newton-Euler está compuesto de dos partes. En la primera parte, la velocidad y aceleración son calculadas iterativamente del eslabón 1 al eslabón 6, aplicando de forma iterativa las ecuaciones (2.19) a la (2.24) a cada eslabón. En la segunda parte se obtiene de forma recursiva las fuerzas y pares actuantes desde el eslabón 6 al eslabón 1, aplicando las ecuaciones de la (2.25), a la (2.27), [Craig, 1986]. Obteniéndose como resultado vectores numéricos, con el valor de la velocidad

rotacional, la aceleración rotacional, aceleración lineal y par que se debe de desarrollar en cada punto de la trayectoria de las articulaciones del robot manipulador.

iteraciones hacia afuera:  $i = 0, 1, 2, 3, 4, 5$

$${}^{i+1}\omega_{i+1} = {}^{i+1}R^i \omega_i + \dot{\theta}_{i+1} \hat{Z}_{i+1} \quad (2.19)$$

$${}^{i+1}\dot{\omega}_{i+1} = {}^{i+1}R^i \dot{\omega}_i + {}^{i+1}R^i \omega_i \times \dot{\theta}_{i+1} \hat{Z}_{i+1} + \ddot{\theta}_{i+1} \hat{Z}_{i+1} \quad (2.20)$$

$${}^{i+1}\dot{v}_{i+1} = {}^{i+1}R \left( \dot{\omega}_i \times {}^i P_{i+1} + \omega_i \times (\omega_i \times {}^i P_{i+1}) + \dot{v}_i \right) \quad (2.21)$$

$${}^{i+1}\dot{v}_{Ci+1} = \dot{\omega}_{i+1} \times {}^{i+1}P_{Ci+1} + \omega_{i+1} \times (\omega_{i+1} \times {}^{i+1}P_{Ci+1}) + \dot{v}_{i+1} \quad (2.22)$$

$${}^{i+1}F_{i+1} = m_{i+1} \dot{v}_{Ci+1} \quad (2.23)$$

$${}^{i+1}N_{i+1} = {}^{Ci+1}I_{i+1} \dot{\omega}_{i+1} + \omega_{i+1} \times {}^{Ci+1}I_{i+1} \omega_{i+1} \quad (2.24)$$

iteraciones hacia adentro:  $i = 6, 5, 4, 3, 2, 1$

$${}^i f_i = {}^i R^{i+1} f_{i+1} + {}^i F_i \quad (2.25)$$

$${}^i n_i = {}^i N_i + {}^i R^{i+1} n_{i+1} + {}^i P_{Ci} \times {}^i F_i + {}^i P_{i+1} \times {}^i R^{i+1} f_{i+1} \quad (2.26)$$

$$\tau_i = {}^i n_i^T \hat{Z}_i \quad (2.27)$$

Donde:

${}^i \omega_i =$  Velocidad rotacional del sistema de referencia del eslabón  $\{i\}$ .

${}^{i+1}R =$  Matriz de rotación que relaciona el sistema de referencia del eslabón  $\{i\}$  con el sistema de referencia del eslabón  $\{i+1\}$ .

${}^{i+1}\hat{Z}_{i+1}$  Eje de rotación del sistema de referencia del eslabón  $\{i+1\}$  con respecto al sistema de referencia del eslabón  $\{i+1\}$ .

${}^i\bullet v_i$  = Aceleración lineal del eslabón  $\{i\}$ .

${}^iP_{i+1}$  = Vector de posición del sistema de referencia del eslabón  $\{i+1\}$  con respecto al sistema de referencia  $\{i\}$ .

${}^{i+1}P_{C_{i+1}}$  = Vector de posición del centro de masa del eslabón  $\{i+1\}$ .

${}^{i+1}F_{i+1}$  = Fuerza actuando en el centro de masa del eslabón  $\{i+1\}$ .

${}^{i+1}N_{i+1}$  = Momento actuando en el centro de masa del eslabón  $\{i+1\}$ .

${}^{C_{i+1}}I_{i+1}$  = Tensor de inercia en el centro de gravedad del eslabón  $\{i+1\}$ .

${}^if_i$  = Fuerza ejercida sobre el sistema de referencia del eslabón  $\{i\}$ .

${}^in_i$  = Par ejercido sobre el sistema de referencia del eslabón  $\{i\}$ .

$\tau_i$  = Par ejercido sobre la unión del eslabón  $\{i\}$

### 2.3 Planeación de Trayectorias

El objetivo de la planeación de trayectorias, es generar las entradas de referencia para los sistemas de control de movimiento que aseguren que el manipulador ejecute las trayectorias planeadas. Típicamente, el usuario especifica un número de parámetros para describir la trayectoria deseada. La planeación de una trayectoria consiste en generar una secuencia en el tiempo de los valores obtenidos mediante una función polinomial que interpola la trayectoria deseada, [Sciavicco, *et al.* 1976].

El requerimiento mínimo para un manipulador es la capacidad para moverlo de una posición inicial a una posición final asignada. La transición debe ser caracterizada por las leyes del movimiento que hagan que los actuadores ejerzan fuerzas generalizadas en las uniones que no violen los límites de saturación y no exciten los modos resonantes

típicamente no modelados de la estructura. Es entonces necesario buscar algoritmos de planeación que generen trayectorias suaves adecuadas.

Un movimiento del manipulador es asignado típicamente en el espacio operacional en términos de los parámetros de la trayectoria tales como las posiciones inicial y final del efector final, posibles posiciones intermedias y tiempo de viaje a lo largo de las rutas geométricas particulares. Si se desea planear una trayectoria en el espacio de las uniones, los valores de las variables de las uniones se tienen que determinar primero a partir de la posición y orientación del efector final especificadas por el usuario. Es entonces necesario recurrir a un algoritmo de cinemática inversa. La planeación de trayectorias en el espacio operacional natural, permite considerar la presencia de restricciones de la ruta; éstas son debidas a las regiones del espacio de trabajo prohibidas al manipulador

El algoritmo de planeación genera una función  $q(t)$  interpolando los vectores dados de las variables de las uniones en cada punto. En general, un algoritmo de planeación de trayectorias en el espacio de las uniones requiere tener las siguientes características:

- i) Las trayectorias generadas no serán muy demandantes desde el punto de vista computacional,
- ii) Las posiciones y velocidades de las uniones serán funciones continuas del tiempo (la continuidad de las aceleraciones puede ser también impuesta),
- iii) Los efectos indeseables serán minimizados, por ejemplo, trayectorias no suaves interpolando una secuencia de puntos en una ruta.

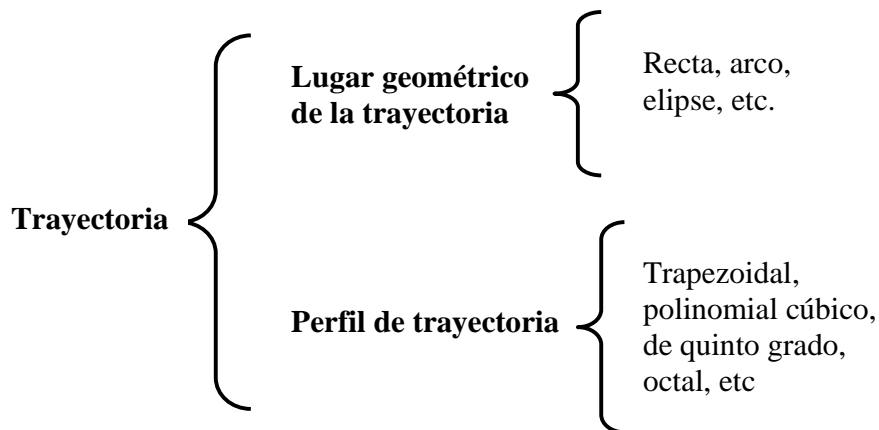
La trayectoria no sólo es el lugar geométrico de los puntos cartesianos por donde pasa algún punto del mecanismo u objeto transportado por él, con las correspondientes orientaciones en cada punto, también es la distribución en el tiempo de la distancia alcanzada a lo largo de ese lugar geométrico y del correspondiente ángulo alrededor del eje de giro para la orientación asociada. En la Figura 2.11 se puede observar que la trayectoria está compuesta de dos partes, el lugar geométrico de la trayectoria y el perfil de trayectoria.

Al hablar de trayectoria debe quedar claro que, aunque el efector final o cuerpo transportado no avance en posición cartesiana, puede estar rotando (cambiando su orientación). En este caso, el lugar geométrico es un punto por donde pasa el eje de giro, y el ángulo de rotación es el que cambiará con el tiempo hasta alcanzar la orientación deseada. La trayectoria de un mecanismo puede incluir cualquiera de los siguientes casos:

- i) El efector final u objeto transportado puede seguir cualquier lugar geométrico y cualquier perfil de trayectoria sin que cambie su orientación.
- ii) El efector final u objeto transportado puede seguir cualquier lugar geométrico y cualquier perfil de trayectoria. Mientras que la orientación asociada también puede hacerlo siguiendo algún perfil de trayectoria.
- iii) El efector final u objeto transportado no cambia de posición, pero sí de orientación, la cual sigue algún perfil de trayectoria.

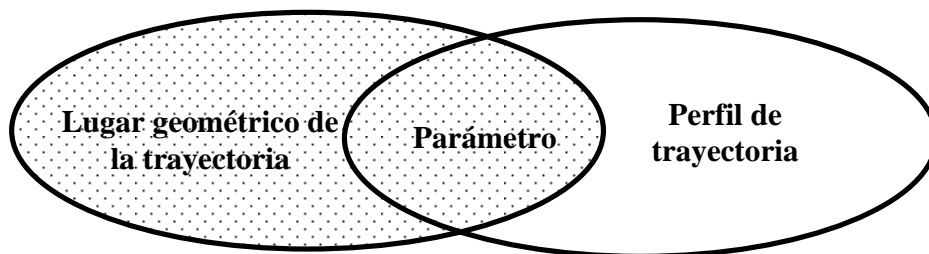
Una ruta denota el lugar de puntos en el espacio de las uniones, o en el espacio operacional, que el manipulador tiene que seguir en la ejecución del movimiento asignado. Una ruta es simplemente una descripción geométrica del movimiento. En cambio, una trayectoria es una ruta en la cual se especifica una ley en el tiempo, en términos, por ejemplo, de velocidades y aceleraciones en cada uno de los puntos. Una ruta puede ser definida en el espacio de las uniones o en el espacio operacional. Normalmente se prefiere este último, puesto que permite una descripción natural de la tarea que el manipulador tiene que realizar.





**Figura 2.11** Componentes de la trayectoria

El lugar geométrico de la trayectoria es el conjunto de puntos cartesianos por donde pasa algún punto del mecanismo. Puede ser una recta, un arco, una elipse, helicoide, etc., o la combinación de éstos. El lugar geométrico preferentemente debe quedar expresado en su forma paramétrica para poder realizar la unión con el perfil de trayectoria, a través de un parámetro, hecho que se ilustra en la Figura 2.12.



**Figura 2.12** Relación que mantiene el lugar geométrico con el perfil de trayectoria.

En esta tesis se ilustra el proceso de planeación de trayectorias para el lugar geométrico determinado por una recta, proceso que se describe en la siguiente sub-sección. Si se requiere una trayectoria con un lugar geométrico diferente, se sustituye en la ecuación (2.28), los términos correspondientes de la recta, por los de la región geométrica deseada,

en donde  $p(t)$ , es el perfil de trayectoria que en este caso, corresponde al polinomio de octavo grado, y que proporciona las leyes de tiempo y velocidad al lugar geométrico (ruta).

### 2.3.1 La Recta en Forma Paramétrica

Sean  $P_1(x_1, y_1, z_1)$  y  $P_2(x_2, y_2, z_2)$  el punto inicial y final de una recta,  $\bar{p}_1 \in \mathfrak{R}^3$  y  $\bar{p}_2 \in \mathfrak{R}^3$  los vectores que describen a cada uno, y  $p(t) \in \mathfrak{R}$  un parámetro variante con el tiempo y contenido en el intervalo  $[0, \|\bar{p}_2 - \bar{p}_1\|]$ . Con esta información es posible encontrar el vector  $\bar{p}(t) \in \mathfrak{R}^3$  asociado a cada punto de la recta, como se indica en la ecuación (2.28).

$$\bar{p}(t) = \bar{p}_1 + \left( \frac{\bar{p}_2 - \bar{p}_1}{\|\bar{p}_2 - \bar{p}_1\|} \right) p(t) \quad (2.28)$$

El parámetro  $p(t) \in \mathfrak{R}$  toma un valor para cada instante de tiempo  $t$ . Así, para  $t = 0$ ,  $p(0) = 0$  y  $\bar{p}(0) = \bar{p}_1$ . Mientras que para el tiempo final  $t = t_f$ ,  $p(t_f) = \|\bar{p}_2 - \bar{p}_1\|$  y, por lo tanto,  $\bar{p}(t_f) = \bar{p}_2$ . Desarrollando (2.28),

$$\bar{p}(t) = \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} + \frac{1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}} \begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{pmatrix} p(t) \quad (2.29)$$

Definiendo los elementos constitutivos de  $\bar{p}(t) = (x(t) \ y(t) \ z(t))^T$ , las ecuaciones de la recta en forma paramétrica son,

$$x(t) = x_1 + \left( \frac{x_2 - x_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}} \right) p(t) \quad (2.30)$$

$$y(t) = y_1 + \left( \frac{y_2 - y_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}} \right) p(t) \quad (2.31)$$

$$z(t) = z_1 + \left( \frac{z_2 - z_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}} \right) p(t) \quad (2.32)$$

### 2.3.2 Perfil de Trayectoria

El objeto que está siendo manipulado debe partir de una posición 1 y llegar a una posición 2. En ambas posiciones el estado del objeto es estático, esto es, no presenta velocidad ni aceleración. Para alcanzar el último punto, a partir del primero, se emplea un tiempo total de recorrido  $t_f$ .

El efector final ejerce una fuerza que sujeta al objeto, pero esta fuerza es finita con una magnitud determinada. Ahora bien, si la aceleración, escalada por la masa del objeto, es mayor a esa fuerza de sujeción, entonces, el objeto podría experimentar un movimiento relativo al órgano terminal que posiblemente modifique la posición adecuada. Entonces, es de suma importancia manejar un perfil de movimiento con pendientes “suavizadas”, tal que la transición del primer estado estático al último se presente con condiciones no tan severas ni difíciles para el mecanismo manipulador [Juárez, 2002]. El parámetro  $p(t)$  presente en la ecuación (2.28), es el que relaciona el lugar geométrico con el perfil de trayectoria. Este parámetro es un polinomio, que determina el perfil de movimiento, obteniéndose la suavidad de movimiento en la trayectoria. En seguida se describen los perfiles de trayectoria siguientes: Trapezoidal, Polinomio de tercer grado, Polinomio de quinto grado y polinomio de octavo grado.

#### 2.3.2.1 Perfil Trapezoidal

Un perfil de velocidad trapezoidal impone una aceleración constante en la fase inicial, una velocidad de cruce y una desaceleración constante en la fase de llegada. El perfil

resultante está formado por un segmento lineal conectado por dos segmentos parabólicos a las posiciones inicial y final. En la Figura 2.13 se muestra este perfil de velocidad en donde se considera que la velocidad inicial y final son cero y los segmentos con aceleraciones constantes tienen la misma duración; lo cual implica una magnitud igual  $\ddot{q}_c$  en los dos segmentos. Se puede notar que esta selección lleva a una trayectoria simétrica con respecto al punto promedio  $q_m = (q_f + q_i)/2$  en  $t_m = t_f/2$ .

El perfil de posición debe satisfacer algunas restricciones para asegurar la transición de  $q_i$  a  $q_f$  en un tiempo  $t_f$ . La velocidad al final del segmento parabólico debe ser igual a la velocidad (constante) del segmento lineal, expresado por la ecuación (2.33).

$$\ddot{q}_c t_c = \frac{q_m - q_c}{t_m - t_c} \quad (2.33)$$

Donde  $q_c$  es el valor alcanzado por la variable de la unión al final del segmento parabólico en el tiempo  $t_c$  con una aceleración constante  $\ddot{q}_c$ . Se tiene entonces:

$$q_c = q_i + \frac{1}{2} \ddot{q}_c t_c^2 \quad (2.34)$$

De (2.33)

$$q_c = q_m - \ddot{q}_c t_c t_m + \ddot{q}_c t_c^2 \quad (2.35)$$

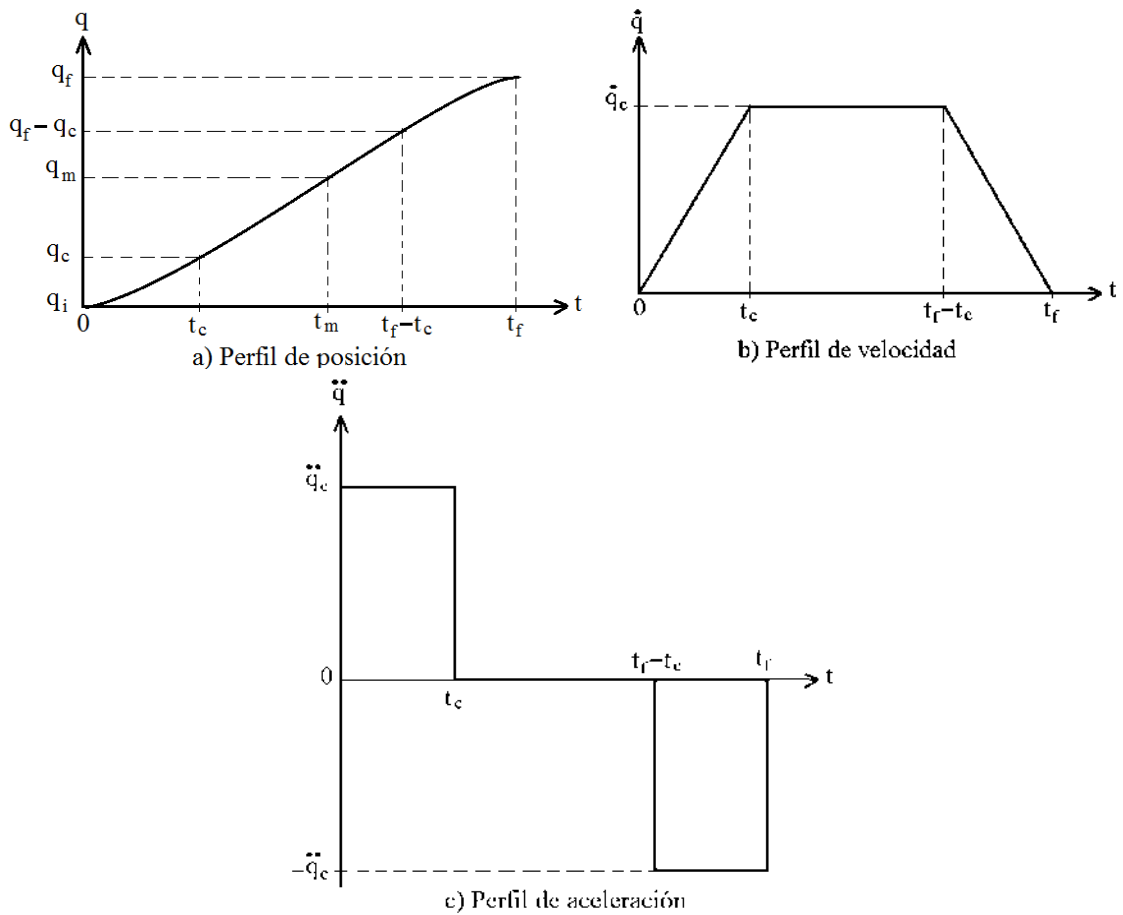
Sustituyendo (2.35) en (2.34)

$$q_m - \ddot{q}_c t_c t_m + \ddot{q}_c t_c^2 = q_i + \frac{1}{2} \ddot{q}_c t_c^2 \quad (2.36)$$

Eliminando  $q_m$  y  $t_m$

$$\frac{q_f + q_i}{2} - \ddot{q}_c t_c \frac{t_f}{2} + \frac{1}{2} \ddot{q}_c t_c^2 = q_i \quad (2.37)$$

$$-\ddot{q}_c t_f t_c + q_f - q_i = 0 \quad (2.38)$$



**Figura 2.13** Caracterización de un perfil trapezoidal

Generalmente,  $\ddot{q}_c$  se especifica con la restricción que  $\text{sgn}(\ddot{q}_c) = \text{sgn}(q_f - q_i)$ , [Sciaviaco, et al, 1976]; por lo que, para  $t_f$ ,  $q_i$  y  $q_f$  dados, la solución para  $t_c$  se calcula de (2.38) como:

$$t_c = \frac{\ddot{q}_c t_f - \sqrt{(\ddot{q}_c t_f)^2 - 4\ddot{q}_c(q_f - q_i)}}{2\ddot{q}_c} \quad (2.39)$$

$$t_c = \frac{t_f}{2} - \frac{1}{2} \sqrt{\frac{\ddot{q}_c t_f^2 - 4(q_f - q_i)}{\ddot{q}_c}} \quad (2.40)$$

Donde  $t_c \leq \frac{t_f}{2}$  y la aceleración cumple la restricción:

$$|\ddot{q}_c| \geq \frac{4|q_f - q_i|}{t_f^2} \quad (2.41)$$

Cuando la aceleración se escoge con el signo de igualdad en (2.41) la trayectoria resultante no incluye el segmento de velocidad constante y tiene solamente los segmentos de aceleración y desaceleración (perfil triangular). Dados  $q_i$ ,  $q_f$ ,  $t_f$  y una velocidad de transición promedio, la restricción en (2.41) permite imponer un valor de aceleración consistente con la trayectoria. Entonces  $t_c$  se calcula de (2.40) y la siguiente secuencia de polinomios es generada:

para  $0 < t \leq t_c$

$$q(t) = q_i + \frac{1}{2}\ddot{q}_c t^2 \quad (2.42)$$

para  $t_c < t \leq t_f - t_c$

$$q(t) = q_c + \dot{q}_c(t - t_c) \quad (2.43)$$

Pero

$$\dot{q}_c = \ddot{q}_c t_c$$

y

$$q_c = q_i + \frac{1}{2}\ddot{q}_c t_c^2$$

Así

$$q(t) = q_i + \frac{1}{2}\ddot{q}_c t_c^2 + \ddot{q}_c t_c(t - t_c) \quad (2.44)$$

$$q(t) = q_i + \ddot{q}_c t_c \left(t - \frac{t_c}{2}\right) \quad (2.45)$$

Para  $t_f - t_c < t \leq t_f$

$$q(t) = q_f + \frac{1}{2}\ddot{q}_c (t_f - t)^2 \quad (2.46)$$

Especificar la aceleración en el segmento parabólico no es la única manera de determinar trayectorias con perfil de velocidad trapezoidal. Además de  $q_i$ ,  $q_f$  y  $t_f$ , se puede especificar también la velocidad de cruce  $\dot{q}_c$  la cual está sujeta a la restricción:

$$\frac{q_f - q_i}{t_f} < \left| \dot{q}_c \right| \leq \frac{2|q_f - q_i|}{t_f} \quad (2.47)$$

Observando que  $\dot{q}_c = \ddot{q}_c t_c$  se puede calcular  $t_c$  de la ecuación (2.38)

$$\begin{aligned} \dot{q}_c t_c - \dot{q}_c t_f + q_f - q_i &= 0 \\ t_c &= \frac{q_i - q_f + \dot{q}_c t_f}{\dot{q}_c} \end{aligned} \quad (2.48)$$

Así la aceleración resultante es:

$$\ddot{q}_c = \frac{\dot{q}_c}{t_c} \quad (2.49)$$

Los valores calculados de  $t_c$  y  $\ddot{q}_c$  en (2.48) y (2.49) permiten la generación de la secuencia de polinomios (2.42), (2.45) y (2.46) en términos de  $\dot{q}_c$ . La adopción de un perfil de velocidad trapezoidal da como resultado un índice de funcionamiento peor que el correspondiente al polinomio cúbico.

### 2.3.2.2 Perfil Polinomial de Tercer Grado

En el movimiento de punto a punto, el manipulador se tiene que mover de una configuración inicial de las uniones a una configuración final en un tiempo dado  $t_f$ . El algoritmo de planeación generará una trayectoria que optimice algún índice de funcionamiento cuando la unión se mueva de una posición a otra. Sea:

$I_i$  = Momento de inercia del eslabón  $i$  respecto a su eje de rotación.

$q_i$  = Posición inicial de la unión  $i$ .  
 $q_f$  = Posición final de la unión  $i$ .  
 $t_f$  = tiempo de duración del movimiento

Es obvio que existe un número infinito de soluciones a este problema. Considerando que la rotación se ejecuta a través de un par  $\tau$  suministrado por un motor, se puede encontrar una solución que minimice la energía disipada en el motor. Esto es, resolviendo el siguiente problema de optimización con restricciones:

Siendo  $\dot{q} = \omega$ , determinar la solución de la ecuación diferencial siguiente, [Sciaviaco, et al, 1976];

$$I\dot{\omega} = \tau$$

Sujeta a la restricción

$$\int_0^{t_f} \omega(t) dt = q_f - q_i$$

Tal que se minimice el índice de funcionamiento

$$\int_0^{t_f} \tau^2(t) dt$$

Puede demostrarse [Sciaviaco, et al, 1976] que la solución es de tipo parabólico

$$\omega(t) = at^2 + bt + c$$

Por consiguiente, para determinar un movimiento de la unión, puede ser escogido el polinomio cúbico, como el de la ecuación (2.50).

$$q(t) = a_3t^3 + a_2t^2 + a_1t + a_0 \tag{2.50}$$



Derivando la ecuación (2.50), se obtiene la ecuación (2.51), que proporciona un perfil de velocidad parabólico.

$$\dot{q}(t) = 3a_3t^2 + 2a_2t + a_1 \quad (2.51)$$

Derivando la ecuación (2.51), se obtiene la ecuación (2.52) que proporciona un perfil de aceleración lineal.

$$\ddot{q}(t) = 6a_3t + 2a_2 \quad (2.52)$$

Puesto que se requiere evaluar cuatro coeficientes, es posible plantear, además de los valores inicial y final de la posición de las uniones  $q_i$  y  $q_f$ , también los valores inicial y final de las velocidades de las uniones  $\dot{q}_i$  y  $\dot{q}_f$ , a las cuales se les asigna el valor cero. La determinación de una trayectoria cualquiera, está dada por la solución del siguiente sistema de ecuaciones:

$$q(0) = q_i = a_0$$

$$\dot{q}(0) = \dot{q}_i = a_1$$

$$q(t_f) = q_f = a_3t_f^3 + a_2t_f^2 + a_1t_f + a_0 \quad (2.53)$$

$$\dot{q}(t_f) = \dot{q}_f = 3a_3t_f^2 + 2a_2t_f + a_1 \quad (2.54)$$

Que permiten calcular los coeficientes del polinomio (2.50). Enseguida se plantea un ejemplo de una trayectoria específica con las siguientes condiciones iniciales:

$q_i = 0$ ,  $q_f = 3rad$ ,  $t_f = 1s$ , y  $\dot{q}_i = \dot{q}_f = 0$ . Por lo tanto se tiene que:

$$a_0 = 0, \quad a_1 = 0$$

Sustituyendo  $a_1$  en la ecuación (2.54), se tiene:

$$\dot{q}_f = 3a_3 + 2a_2 = 0 \quad (2.55)$$

Despejando  $a_2$  de la ecuación (2.55), se tiene:

$$a_2 = -\frac{2}{3}a_3$$

Sustituyendo  $a_1$  y  $a_2$  en la ecuación (2.53) se tiene:

$$q_f = a_3 + a_2 = 3 \quad (2.56)$$

Sustituyendo  $a_2$  en la ecuación (2.56), se tiene:

$$a_3 - \frac{3}{2}a_3 = 3$$

Por lo tanto:

$$a_3 = -6$$

y

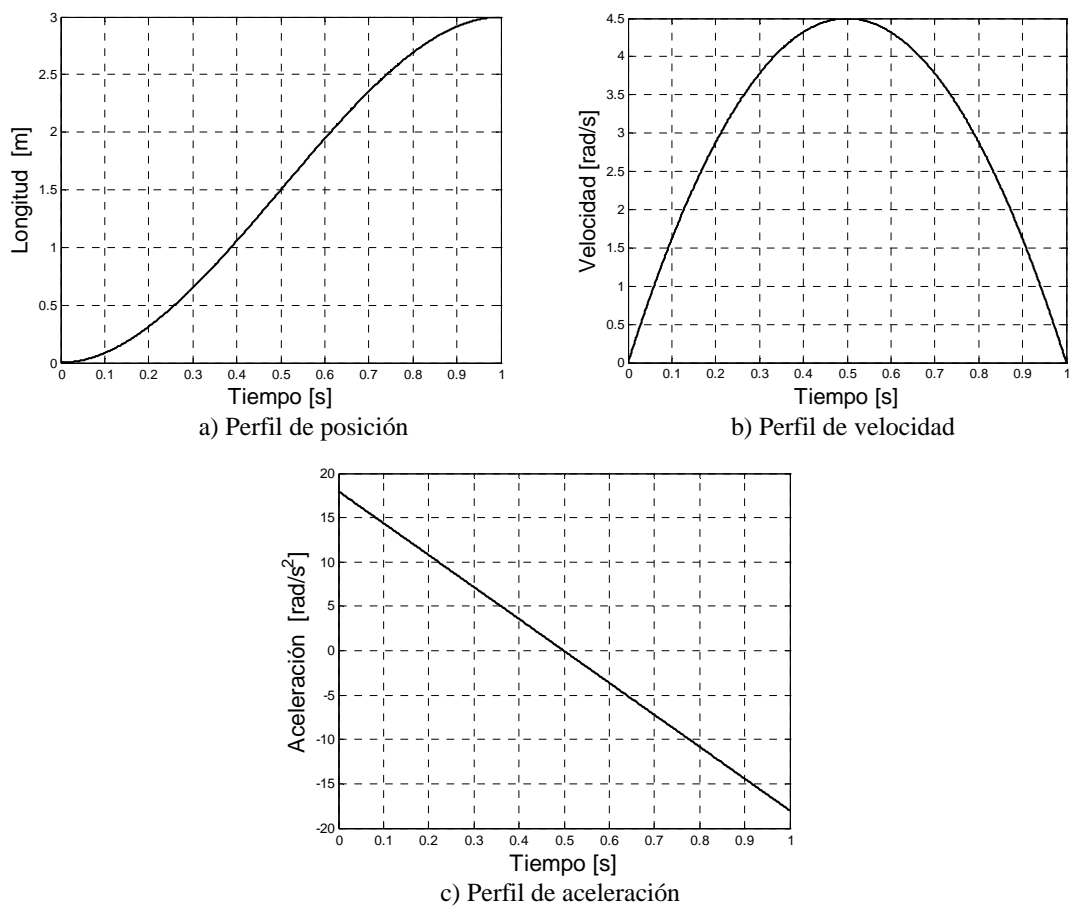
$$a_2 = 9$$

Sustituyendo los coeficientes anteriormente determinados, en las ecuaciones (2.50) , (2.51) y (2.52), se obtienen la ecuaciones que definen el perfil de posición, velocidad y aceleración, ilustrados en la Figura 2.14.

$$q(t) = -6t^3 + 9t^2$$

$$\dot{q}(t) = -12t^2 + 12t$$

$$\ddot{q}(t) = -36t + 18$$



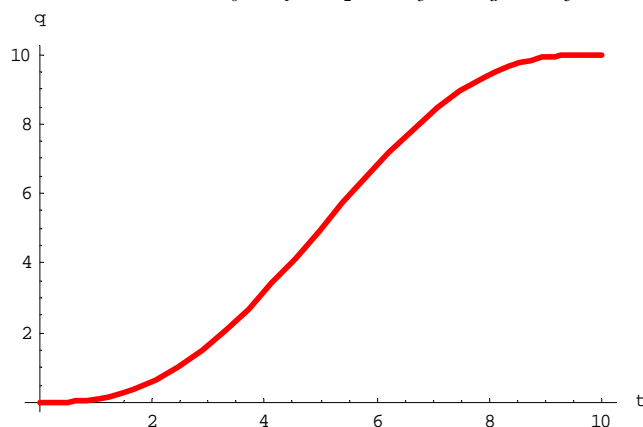
**Figura 2.14.** Perfil de trayectoria polinomial cúbica, a) perfil de posición, b) perfil de velocidad y c) perfil de aceleración

### 2.3.2.3 Perfil Polinomial de Quinto Grado

El espacio correspondiente a los perfiles de quinto grado, denominado  $P^5(q_f, t_f)$ , es un conjunto que agrupa perfiles de trayectoria basados en un polinomio de grado 5. A continuación se presentan las leyes matemáticas que determinan a un elemento  $q(t) \in P^5(q_f, t_f)$  que será empleado para regir la distribución, con el tiempo, de la posición y orientación del órgano terminal de un robot.

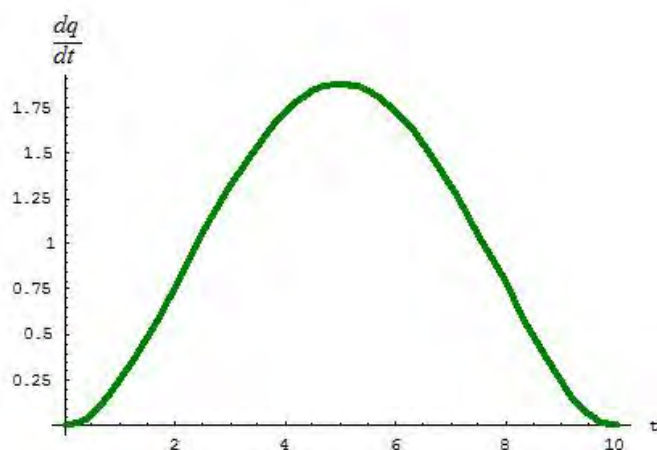
La ecuación que rige los anteriores perfiles de trayectoria es un polinomio de grado quinto expresado por la ecuación (2.58), sus primeras dos derivadas temporales están expresadas por las ecuaciones (2.59) y (2.60). La característica de este perfil es la “suavización” de los cambios de pendiente. Al inicio y fin del recorrido, el cambio de velocidad y aceleración es gradual, mejorando las características, como se puede observar en las gráficas de las Figuras (2.15), (2.16) y (2.17), que representan el perfil de posición, el perfil de velocidad y el perfil de aceleración respectivamente.

**POSICIÓN**  $p(t) = q(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5$  (2.57)



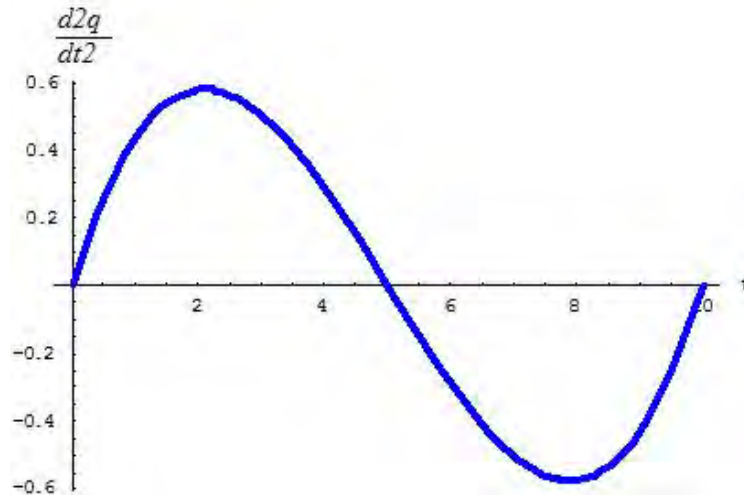
**Figura 2.15** Perfil de posición del polinomio de quinto grado.

**VELOCIDAD**  $v(t) = \dot{q}(t) = \frac{dq(t)}{dt} = a_1 + 2a_2t + 3a_3t^2 + 4a_4t^3 + 5a_5t^4$  (2.58)



**Figura 2.16** Perfil de velocidad del polinomio de quinto grado.

**ACELERACIÓN**  $a(t) = \ddot{q}(t) = \frac{d^2q(t)}{dt^2} = 2a_2 + 6a_3t + 12a_4t^2 + 20a_5t^3$  (2.59)



**Figura 2.17** Perfil de aceleración del polinomio de quinto grado

$$P(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5 \quad (2.60)$$

$$v(t) = a_1 + 2a_2t + 3a_3t^2 + 4a_4t^3 + 5a_5t^4 \quad (2.61)$$

$$a(t) = 2a_2 + 6a_3t + 12a_4t^2 + 20a_5t^3 \quad (2.62)$$

En la Tabla 2.4 se muestran las condiciones iniciales y finales que se proponen para la evaluación de las ecuaciones (2.60), (2.61), (2.62). Estas condiciones corresponden al comportamiento del perfil de posición, perfil de velocidad y perfil de aceleración de la función polinomial de quinto grado mostrado en las gráficas de las Figuras (2.15), (2.16) y (2.17) esto es,  $t = 0$  y  $t = t_f$ ,

**Tabla 2.4** Restricciones para evaluar los coeficientes del polinomio de quinto grado.

$t = 0$	$t = t_f$
$p(0) = 0$	$p(t_f) = \text{long. recta } (q_f)$
$v(0) = 0$	$v(t_f) = 0$
$a(0) = 0$	$a(t_f) = 0$

Sustituyendo los valores conocidos para  $t = 0$  y  $t = t_f$  en las ecuaciones (2.60), (2.61), (2.62), se tiene que:

Para  $t = 0$

$$\begin{aligned} P(0) &= a_0 = 0 \\ v(0) &= a_1 = 0 \\ a(0) &= 2a_2 = 0 \therefore a_2 = 0 \\ a_0 &= a_1 = a_2 = 0 \end{aligned} \tag{2.63}$$

Para  $t = t_f$

$$\begin{aligned} P(t_f) &= a_3 t_f^3 + a_4 t_f^4 + a_5 t_f^5 = q_f \\ v(t_f) &= 3a_3 t_f^2 + 4a_4 t_f^3 + 5a_5 t_f^4 = 0 \\ a(t_f) &= 6a_3 t_f + 12a_4 t_f^2 + 20a_5 t_f^3 = 0 \end{aligned}$$

Resolviendo este sistema de ecuaciones:

$$\begin{pmatrix} t_f^3 & t_f^4 & t_f^5 \\ 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 6t_f & 12t_f^2 & 20t_f^3 \end{pmatrix} \begin{pmatrix} a_3 \\ a_4 \\ a_5 \end{pmatrix} = \begin{pmatrix} q_f \\ 0 \\ 0 \end{pmatrix}$$

$$a_3 = 10 \left( \frac{q_f}{t_f^3} \right) \tag{2.64}$$

$$a_4 = -15 \left( \frac{q_f}{t_f^4} \right) \tag{2.65}$$

$$a_5 = 6 \left( \frac{q_f}{t_f^5} \right) \tag{2.66}$$

Entonces, el perfil de la posición queda definido por la ecuación (2.68).

$$P(t) = q^5(t) = q_f \left( 10 \left( \frac{t^3}{t_f^3} \right) - 15 \left( \frac{t^4}{t_f^4} \right) + 6 \left( \frac{t^5}{t_f^5} \right) \right) \quad (2.67)$$

El perfil de la velocidad con que viaja el órgano terminal está definido por la ecuación (2.69).

$$v(t) = \dot{q}^5(t) = q_f \left( 30 \left( \frac{t^2}{t_f^3} \right) - 60 \left( \frac{t^3}{t_f^4} \right) + 30 \left( \frac{t^4}{t_f^5} \right) \right) \quad (2.68)$$

En lo que respecta a la aceleración, ésta se determina por la ecuación (2.70).

$$a(t) = \ddot{q}^5(t) = q_f \left( 60 \left( \frac{t}{t_f^3} \right) - 180 \left( \frac{t^2}{t_f^4} \right) + 120 \left( \frac{t^3}{t_f^5} \right) \right) \quad (2.69)$$

#### 2.3.2.4 Perfil Polinomial de Octavo Grado

El perfil de trayectoria octal es una función polinomial de grado 8 como la representada por la Ecuación (2.71), la cual es una ecuación con 9 coeficientes ( $a_i$ ;  $i = 0, \dots, 8$ ) que son las incógnitas por determinar con las 9 ecuaciones correspondientes a cada restricción temporal presente en la Tabla 2.5 y manifestadas en las Ecs. (2.72) a la (2.80).

Tabla 2.5 Restricciones para evaluar los coeficientes del polinomio de octavo grado.

$t = 0$	$t = \frac{t_f}{2}$	$t = t_f$
$p(t) = 0$	$p(t) = \frac{q_f}{2}$	$p(t) = q_f$
$\dot{P}(t) = 0$	$\dot{P}(t) = v_{\max}$	$\dot{P}(t) = 0$
$\ddot{P}(t) = 0$	$\ddot{P}(t) = 0$	$\ddot{P}(t) = 0$

$$q(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5 + a_6t^6 + a_7t^7 + a_8t^8 \quad (2.70)$$

Para  $t = 0$ ,

$$q(0) = 0 = a_0 \quad (2.71)$$

$$\frac{dq(0)}{dt} = 0 = a_1 \quad (2.72)$$

$$\frac{d^2q(0)}{dt^2} = 0 = a_2 \quad (2.73)$$

Para  $t = \frac{t_f}{2}$ , donde  $t_f$  es el tiempo de proceso total,

$$\begin{aligned} q\left(\frac{t_f}{2}\right) = & a_0 + a_1\left(\frac{t_f}{2}\right) + a_2\left(\frac{t_f}{2}\right)^2 + a_3\left(\frac{t_f}{2}\right)^3 + a_4\left(\frac{t_f}{2}\right)^4 + a_5\left(\frac{t_f}{2}\right)^5 + \\ & + a_6\left(\frac{t_f}{2}\right)^6 + a_7\left(\frac{t_f}{2}\right)^7 + a_8\left(\frac{t_f}{2}\right)^8 \end{aligned} \quad (2.74)$$

$$\begin{aligned} \frac{dq\left(\frac{t_f}{2}\right)}{dt} = v_{\max} = & a_1 + 2a_2\left(\frac{t_f}{2}\right) + 3a_3\left(\frac{t_f}{2}\right)^2 + 4a_4\left(\frac{t_f}{2}\right)^3 + 5a_5\left(\frac{t_f}{2}\right)^4 + \\ & + 6a_6\left(\frac{t_f}{2}\right)^5 + 7a_7\left(\frac{t_f}{2}\right)^6 + 8a_8\left(\frac{t_f}{2}\right)^7 \end{aligned} \quad (2.75)$$

$$\begin{aligned} \frac{d^2q\left(\frac{t_f}{2}\right)}{dt^2} = 0 = & 2a_2 + 6a_3\left(\frac{t_f}{2}\right) + 12a_4\left(\frac{t_f}{2}\right)^2 + 20a_5\left(\frac{t_f}{2}\right)^3 + \\ & + 30a_6\left(\frac{t_f}{2}\right)^4 + 42a_7\left(\frac{t_f}{2}\right)^5 + 56a_8\left(\frac{t_f}{2}\right)^6 \end{aligned} \quad (2.76)$$



Para  $t = t_f$ ,

$$q(t_f) = q_f = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 + a_4 t_f^4 + a_5 t_f^5 + a_6 t_f^6 + a_7 t_f^7 + a_8 t_f^8 \quad (2.77)$$

$$\frac{dq(t_f)}{dt} = 0 = a_1 + 2a_2 t_f + 3a_3 t_f^2 + 4a_4 t_f^3 + 5a_5 t_f^4 + 6a_6 t_f^5 + 7a_7 t_f^6 + 8a_8 t_f^7 \quad (2.78)$$

$$\frac{d^2 q(t_f)}{dt^2} = 0 = 2a_2 + 6a_3 t_f + 12a_4 t_f^2 + 20a_5 t_f^3 + 30a_6 t_f^4 + 42a_7 t_f^5 + 56a_8 t_f^6 \quad (2.79)$$

Resolviendo este sistema de ecuaciones se obtienen las siguientes expresiones para los coeficientes  $a_i$ .

$$a_0 = 0 \quad (2.80)$$

$$a_1 = 0 \quad (2.81)$$

$$a_2 = 0 \quad (2.82)$$

$$a_3 = \frac{70q_f}{t_f^3} - \frac{32v_{\max}}{t_f^2} \quad (2.83)$$

$$a_4 = \frac{-315q_f}{t_f^4} + \frac{160v_{\max}}{t_f^3} \quad (2.84)$$

$$a_5 = \frac{546q_f}{t_f^4} - \frac{288v_{\max}}{t_f^4} \quad (2.85)$$

$$a_6 = \frac{-420q_f}{t_f^6} + \frac{224v_{\max}}{t_f^5} \quad (2.86)$$

$$a_7 = \frac{120q_f}{t_f^7} - \frac{64v_{\max}}{t_f^6} \quad (2.87)$$

$$a_8 = 0 \quad (2.88)$$

Sustituyendo los coeficientes obtenidos de las ecuaciones (2.80) a la (2.88) en la ecuación (2.70), se obtiene la expresión para  $q(t)$ , que corresponde al perfil de posición. La primera y segunda derivada de la ecuación (2.90), que corresponden al perfil de velocidad y aceleración respectivamente.

$$q(t) = q_f \left( \frac{70}{t_f^3} t^3 - \frac{315}{t_f^4} t^4 + \frac{546}{t_f^5} t^5 - \frac{420}{t_f^6} t^6 + \frac{120}{t_f^7} t^7 \right) + v_{\max} \left( \frac{-32}{t_f^2} t^3 + \frac{160}{t_f^3} t^4 - \frac{288}{t_f^4} t^5 + \frac{224}{t_f^5} t^6 - \frac{64}{t_f^6} t^7 \right) \quad (2.89)$$

A continuación se expresará la velocidad máxima ( $v_{\max}$ ) de la ecuación (2.89), mediante un parámetro de control adimensional, llamado *indice\_m*, como sigue:

$$v_{\max} = indice\_m \frac{q_f}{t_f} \quad (2.90)$$

Para tener trayectorias suaves (movimientos naturales), el rango de funcionamiento del *indice\_m* en el polinomio de octavo grado de la ecuación **¡Error! No se encuentra el origen de la referencia.**, es de 1.458333 a 2.1875, [Juárez, 2004]. Tomando en cuenta que una trayectoria es una ruta afectada por leyes de tiempo, velocidad y aceleración, para cada *indice\_m* existe una trayectoria diferente, como se muestra en la Figura 2.18. Si el valor del *indice\_m* se encuentra fuera del rango especificado se tendrán movimientos no naturales, como se muestra en la Figura 2.19.

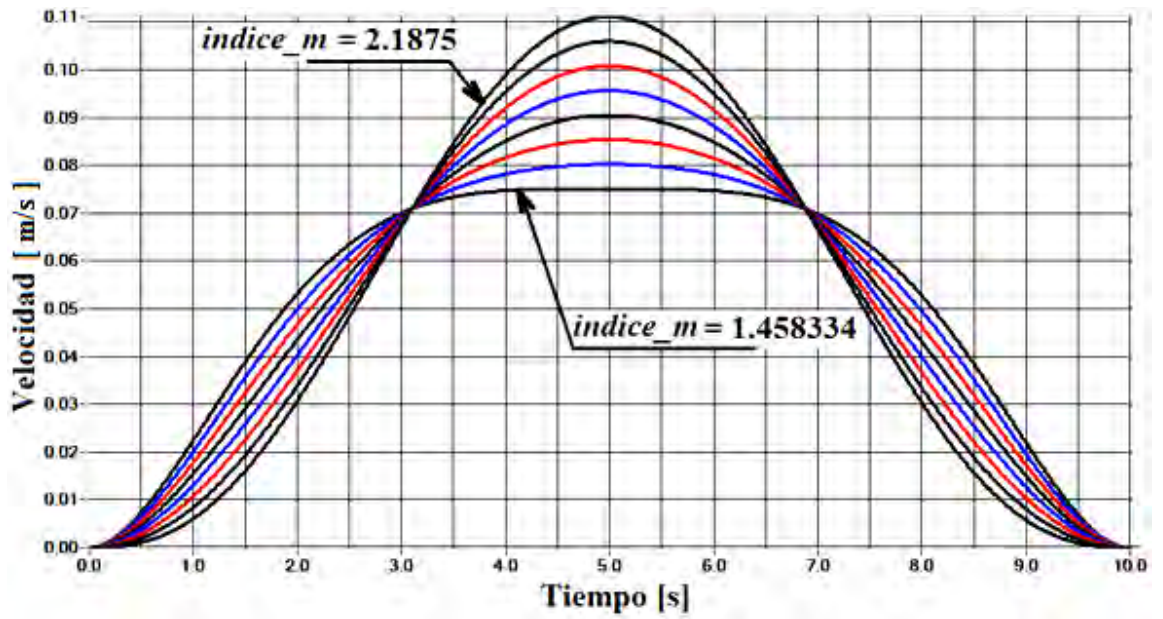


Figura 2.18 Perfiles de velocidad del robot manipulador dentro del intervalo de movimientos naturales.

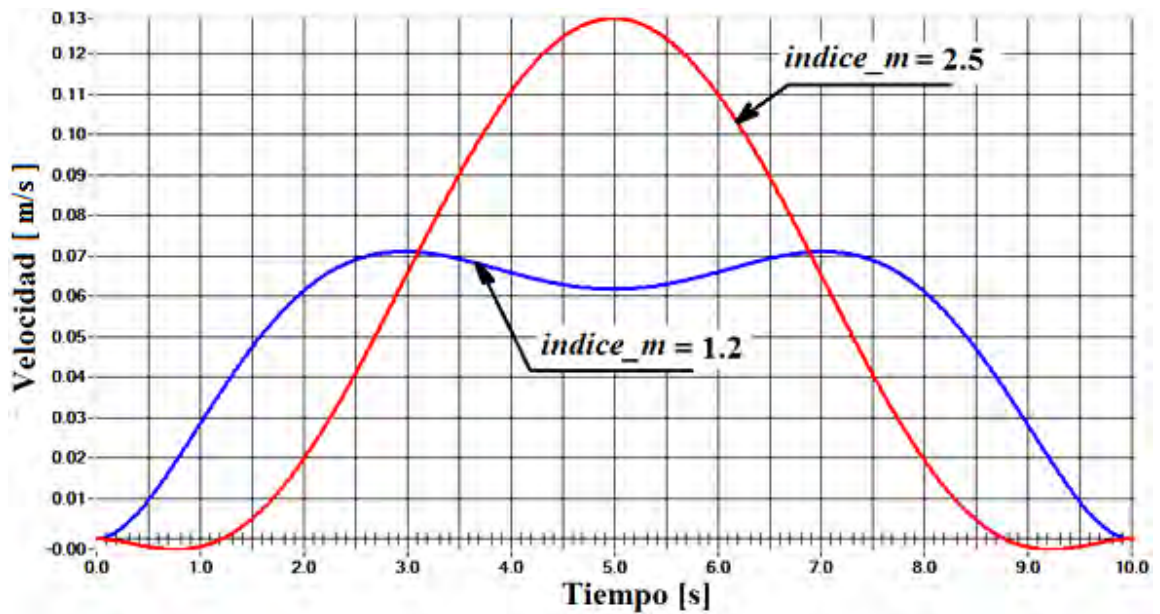


Figura 2.19 Perfiles de velocidad del robot manipulador fuera del intervalo de movimientos naturales.

## 2.4 Conclusiones

Los perfiles trapezoidales presentan cambios de velocidad y aceleración muy drásticos al inicio y final del recorrido y en los puntos de unión entre transitorios, o entre transitorio y estacionario. En esas posiciones se “sentirán” golpes debido al cambio de aceleración. Los golpes producidos por las transiciones en el perfil trapezoidal, pueden desgastar los elementos mecánicos del robot lo cual se traduce en mayor generación de partículas que pueden salir del cuerpo del robot, lo que puede traducirse en pérdidas económicas. En algunos procesos, por ejemplo, en los ambientes limpios empleados en la producción de semiconductores, productos farmacéuticos o alimenticios, estas micro partículas pueden dañar o arruinar la producción y/o los motores empleados en las articulaciones. Aumentando las actividades de mantenimiento correctivo.

El perfil polinomial cúbico presenta un perfil de velocidad parabólico, con un perfil de aceleración lineal con discontinuidades al inicio y final de la trayectoria, como puede verse en la Figura 2.14.

El perfil polinomial de quinto grado presenta un perfil de velocidad y aceleración con suavidad inherente al grado del polinomio. La característica de este perfil es la “suavización” de los cambios de pendiente. Al inicio y fin del recorrido el cambio de velocidad y aceleración es gradual, mejorando considerablemente las características dinámicas, desapareciendo el perfil de aceleración lineal.

El perfil polinomial de octavo grado aumenta la suavidad en los perfiles de velocidad y aceleración con respecto al perfil polinomial de quinto grado, con una característica adicional al grado del polinomio. Esta característica expresada por el parámetro de velocidad máxima del robot manipulador sobre la trayectoria, permite definir un parámetro adimensional llamado *indice\_m* que permite evaluar un gran número de trayectorias diferentes.

# CAPÍTULO 3

## CONCEPTOS BÁSICOS PARA OPTIMIZACIÓN MEDIANTE ALGORITMOS GENÉTICOS.

En este capítulo se hace una descripción general de los algoritmos genéticos (AG) como un subconjunto de las técnicas heurísticas conocidas como técnicas evolutivas. Estos algoritmos están basados en la teoría de la evolución de Darwin, imitando el comportamiento de los mecanismos de reproducción y selección dentro de una especie. Su gran versatilidad para resolver problemas en la búsqueda y optimización de muy diferentes campos ha hecho que hayan adquirido una gran popularidad.

### 3.1 Estrategia Evolutiva

Los Algoritmos Genéticos (AG) son una de las más conocidas y originales técnicas de resolución de problemas dentro de lo que se ha definido como "Computación Evolutiva" (o "Algoritmos Evolutivos"), término que agrupa a los Algoritmos Genéticos, las Estrategias Evolutivas y la Programación Evolutiva. En realidad todas estas técnicas son muy parecidas y comparten muchos aspectos.

Los AG son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización. Entre algunas de sus ventajas con respecto a las técnicas de optimización tradicionales, se pueden mencionar las siguientes: No necesitan conocimientos específicos sobre el problema que intentan resolver. Operan de forma simultánea con varias soluciones, en vez de trabajar de forma secuencial como las técnicas tradicionales. Cuando se usan para maximizar o minimizar una función objetivo, resultan menos afectados por los máximos o mínimos locales (falsas soluciones) que las técnicas tradicionales. Resulta sumamente fácil ejecutarlos en las modernas arquitecturas

masivamente paralelas. Usan operadores probabilísticos, en vez de los típicos operadores determinísticos de las otras técnicas. Entre algunas de sus desventajas se pueden mencionar las siguientes: Pueden tardar mucho en converger, o no converger en absoluto, dependiendo en cierta medida de los parámetros que se utilicen, tamaño de la población y número de generaciones.

El poder de los Algoritmos Genéticos proviene del hecho de que se trata de una técnica robusta, y pueden tratar con éxito una gran variedad de problemas provenientes de diferentes áreas, incluyendo aquellos en los que otros métodos encuentran dificultades. Si bien no se garantiza que el Algoritmo Genético encuentre la solución óptima del problema, existe evidencia empírica de que se encuentran soluciones de un nivel aceptable, en un tiempo competitivo con el resto de algoritmos de optimización combinatoria.

### **3.1.1 Algoritmos Genéticos**

El algoritmo genético (AG) es una técnica de búsqueda basada en la teoría de la evolución de Darwin, que ha cobrado tremenda popularidad en todo el mundo durante los últimos años. Esta técnica se basa en los mecanismos de selección que utiliza la naturaleza, de acuerdo a los cuales los individuos más aptos de una población son los que sobreviven, al adaptarse más fácilmente a los cambios que se producen en su entorno. Un investigador de la Universidad de Michigan llamado John Holland era consciente de la importancia de la selección natural, y a fines de los 60s desarrolló una técnica que permitió incorporarla a un programa. Su objetivo era lograr que las computadoras aprendieran por sí mismas. A la técnica que inventó Holland se le llamó originalmente "planes reproductivos", pero se hizo popular bajo el nombre "algoritmo genético" tras la publicación de su libro en 1975. Una definición bastante completa de un algoritmo genético es la propuesta por John Koza:

*“Es un algoritmo matemático altamente paralelo que transforma un conjunto de objetos matemáticos individuales con respecto al tiempo usando operaciones modeladas de acuerdo al principio Darwiniano de reproducción y supervivencia del más apto, y tras*

*haberse presentado de forma natural una serie de operaciones genéticas de entre las que destaca la recombinación sexual. Cada uno de estos objetos matemáticos suele ser una cadena de caracteres (letras o números) de longitud fija que se ajusta al modelo de las cadenas de cromosomas, y se les asocia con una cierta función matemática que refleja su aptitud”.*

Los algoritmos genéticos son de probada eficacia en caso de querer calcular funciones no derivables (o de derivación muy compleja), aunque su uso es posible con cualquier función. Deben tenerse en cuenta también las siguientes consideraciones:

- i) Si la función a optimizar tiene muchos máximos/mínimos locales se requerirán más iteraciones del algoritmo para "asegurar" el máximo/mínimo global.
- ii) Si la función a optimizar contiene varios puntos muy cercanos en valor al óptimo, solamente podemos "asegurar" que encontraremos uno de ellos (no necesariamente el óptimo).

Un algoritmo genético consiste en una función matemática o una rutina de software que toma como entradas a los ejemplares y retorna como salidas cuáles de ellos deben generar descendencia para la nueva generación. Los algoritmos genéticos establecen una analogía entre el conjunto de soluciones de un problema, llamado **fenotipo**, y el conjunto de individuos de una población natural, codificando la información de cada solución en una cadena, generalmente binaria, llamada cromosoma. Los símbolos que forman la cadena son llamados genes. Cuando la representación de los cromosomas se hace con cadenas de dígitos binarios se le conoce como **genotipo**. Los cromosomas evolucionan a través de iteraciones, llamadas generaciones. En cada generación, los cromosomas son evaluados usando alguna medida de aptitud. Las siguientes generaciones (nuevos cromosomas), llamada descendencia, se forman utilizando dos operadores, de cruzamiento y de mutación.

Un algoritmo genético puede presentar diversas variaciones, dependiendo de cómo se aplican los operadores genéticos (cruzamiento, mutación), de cómo se realiza la selección y

de cómo se decide el reemplazo de los individuos para formar la nueva población. En la Figura 3.1 se ilustran los pasos a seguir en el algoritmo genético propuesto en esta tesis.

**Figura 3.1** Diagrama a bloques del Algoritmo Genético.

A continuación se hace una breve descripción de cada uno de los pasos que componen el algoritmo genético.

- i) **Población inicial:** Se genera aleatoriamente la población inicial, que está constituida por un conjunto de cromosomas los cuales representan las posibles soluciones del problema. Esta población se representa utilizando un número binario de una longitud de 20 bits,

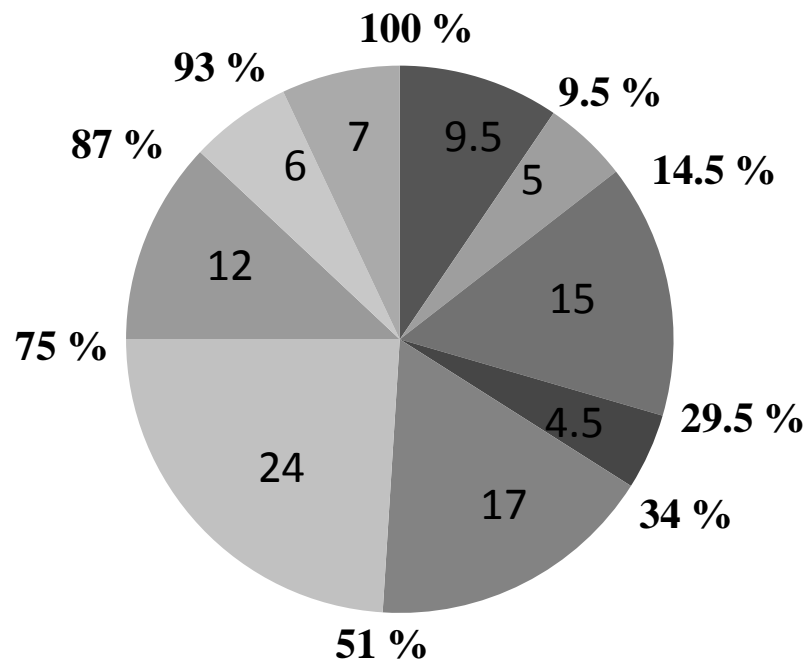


como se indica en la Figura 3.2. Donde cada bit tiene un valor correspondiente al lugar que ocupa dentro de la cadena que representa al individuo.

0	1	1	0	0	0	1	1	1	0	1	0	1	0	1	1	0	0	1	1
$2^{19}$																			$2^0$

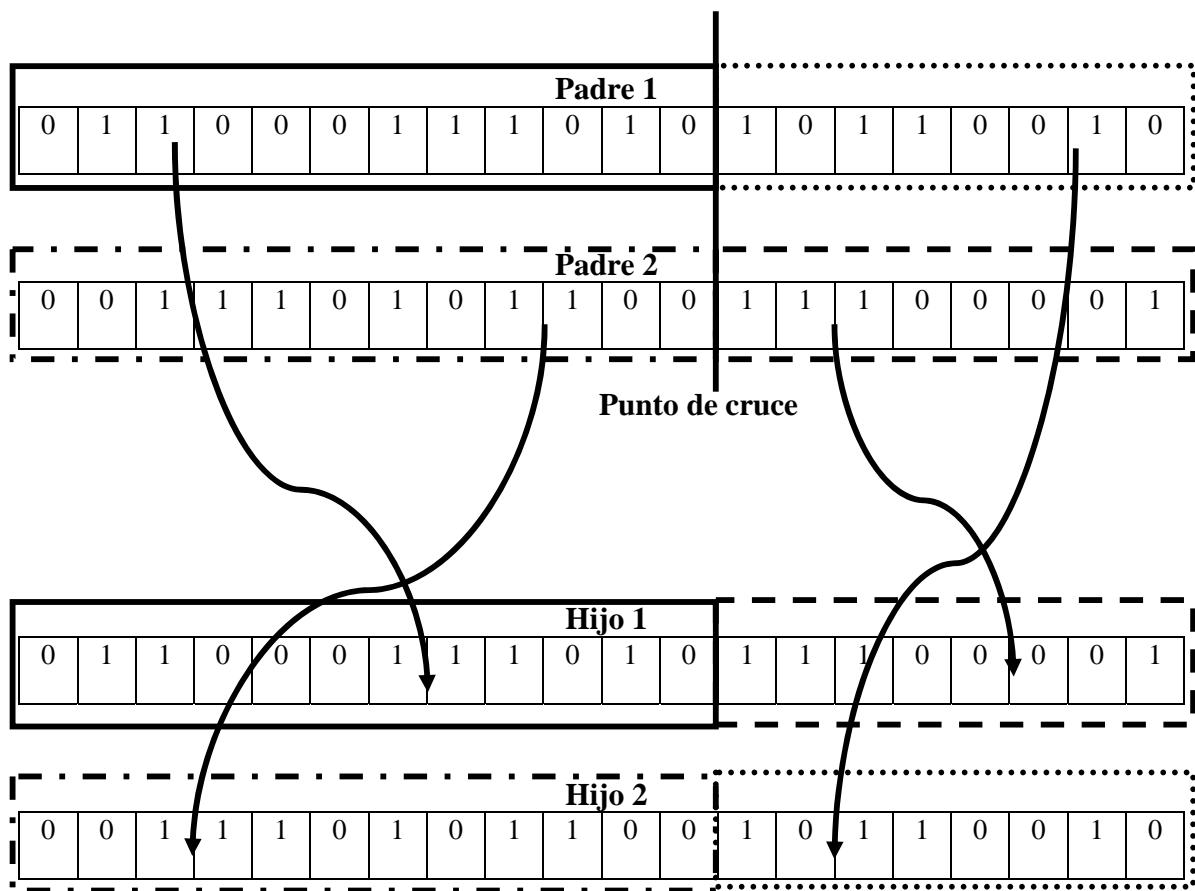
**Figura 3.2** Representación binaria de la población genética (genotipo)

- ii) **Evaluación de la función objetivo:** A cada uno de los cromosomas de la población se aplicará la función de aptitud para saber que tan "buena" es la solución que se está codificando. A cada individuo se le asigna un valor de aptitud de acuerdo con su proximidad a la solución.
- iii) **Criterio de terminación:** El AG se deberá detener cuando se alcance la solución óptima, pero ésta generalmente se desconoce, por lo que se deben utilizar otros criterios de terminación. Normalmente se usan dos criterios: correr el AG un número máximo de iteraciones (generaciones) o detenerlo cuando no haya cambios en la población.
- iv) **Selección:** Después de saber la aptitud de cada cromosoma se procede a elegir los cromosomas que serán cruzados en la siguiente generación, que serán los padres de la nueva generación. Los cromosomas con mejor aptitud tienen mayor probabilidad de ser seleccionados. Se hace uso de la técnica de selección por rueda de la ruleta. Con este método, la probabilidad que tiene un individuo de reproducirse es proporcional a su valor de función de evaluación, es decir, a su adaptación. En la Figura 3.3 se ilustra el esquema de la ruleta donde 9 individuos son mapeados de acuerdo a probabilidad de selección.



**Figura 3.3** Esquema de la ruleta para la selección de individuos con mayor aptitud.

- v) **Cruzamiento:** El cruzamiento es el principal operador genético, representa la reproducción sexual, opera sobre dos cromosomas (padres) a la vez para generar dos descendientes (hijos) donde se combinan las características de ambos cromosomas. Se elige al azar un punto dentro del intervalo del número binario que representa a cada padre, como se ilustra en la Figura 3.4.
- vi) **Mutación:** Modifica al azar parte del cromosoma de los individuos, y permite alcanzar zonas del espacio de búsqueda que no estaban cubiertas por los individuos de la población actual, permitiendo realizar una exploración de toda la información almacenada hasta el momento en la población y combinarla para crear mejores individuos. Esto consiste en hacer una selección aleatoria entre todos los bit de la población, para cambiar el estado de algunos de ellos.



**Figura 3.4** Cruzamiento, operador genético para obtener la nueva generación.

vii) **Reemplazo:** Una vez aplicados los operadores genéticos, se seleccionan los mejores individuos para conformar la población de la siguiente generación. El algoritmo opera iterativamente, en cada iteración, todos los miembros de la población son evaluados de acuerdo a una función de adaptación. Una nueva población es generada, seleccionando probabilísticamente los individuos de mayor adaptación en la población presente. Algunos de estos individuos pasan intactos a la siguiente generación. Otros son seleccionados para crear una nueva generación, aplicando operaciones genéticas como el cruce y la mutación.

En el capítulo 5 se desarrollan los algoritmos para la minimización del gasto de energía o de la minimización del tiempo de traslado del robot manipulador propuestos en esta tesis, haciendo una descripción detallada de cada uno de los términos expuestos en esta sección.

# CAPÍTULO 4

## CONCEPTOS BÁSICOS DE PROCESAMIENTO EN PARALELO

En éste capítulo se muestran las principales características del procesamiento en paralelo. Se analizan algunas de las principales arquitecturas computacionales usadas para el procesamiento en paralelo, así como las plataformas operativas usadas para el mismo. Además se muestran algunas de las métricas utilizadas para la medición de la eficiencia de los programas en paralelo.

### 4.1 Introducción

La evolución del hardware de las computadoras ha permitido que al día de hoy se tengan computadoras con grandes capacidades como: la memoria y la velocidad de cómputo. Estas dos características han sido utilizadas con el objeto de poder resolver problemas complejos en menor tiempo. A través de esto se ha podido aumentar el nivel de detalle de los modelos matemáticos utilizados para el análisis del comportamiento de los sistemas. Un modelo más adecuado permite obtener resultados más confiables que permitan ser usados en el diseño y planeación de otros sistemas o en las mejoras de los mismos.

El uso de computadoras con características avanzadas de hardware, tales como las computadoras multiprocesadores o multinúcleos, debe de ser acompañado con el uso de técnicas de programación avanzadas que permitan hacer un uso eficiente de los recursos computacionales disponibles.

El procesamiento en paralelo puede ser definido como una forma de procesamiento de información en la cual dos o más procesadores juntos o con algún tipo de esquema de

comunicación entre ellos, pueden cooperar para obtener la solución de un problema [Alvarado *et al.* 1992]. El auge del procesamiento en paralelo ha sido motivado por dos grandes factores, el primero es la posibilidad de resolver problemas de mayor envergadura, en tanto que el segundo está asociado con los avances en las tecnologías VLSI (*Very Large Scaled of Integration*).

El procesamiento en paralelo consiste en tener una gran cantidad de procesadores solucionando, en forma simultánea, un gran problema. Un proceso computacional paralelo en forma general divide un gran problema en muchos sub-problemas, los cuales son resueltos mediante el uso muchos procesadores interconectados [Lemaitre y Thomas 1996]. El procesamiento en paralelo se considera ahora un estándar para científicos que trabajan en diversas áreas tales como: evolución de las galaxias, modelado del clima, diseño de aviones, dinámica molecular, análisis de los sistemas eléctricos, etc.

La motivación del uso del procesamiento en paralelo, existiendo en la actualidad computadoras con velocidades de procesamiento muy altas, se debe a dos desventajas básicas que tienen las computadoras basadas en tecnologías de un solo procesador la frecuencia máxima del reloj de una computadora está limitada por con la velocidad de la luz. El procesamiento en paralelo es la única forma de vencer estas limitaciones [Jin, 1994].

En general se esperan dos grandes mejoras con la reducción de tiempo de cómputo mediante el uso de procesamiento en paralelo: la posibilidad de realizar estudios en tiempo real, así como incrementar el tamaño de los problemas a resolver y el nivel de detalle en los modelos utilizados en la representación de los elementos que constituyen a los sistemas [Lemaitre y Thomas, 1996].

#### **4.2 Procesamiento en Paralelo**

El procesamiento en paralelo consiste en resolver un problema usando dos o más elementos de proceso, de manera que cada uno de ellos coadyuve en la solución del problema. La mayoría de las computadoras de propósito general están diseñadas en base al uso de un sólo

procesador. Esta tecnología tiene una serie de limitaciones físicas, tecnológicas y económicas, por lo que las computadoras tradicionales uni-procesadores son incapaces de satisfacer las necesidades computacionales para aplicaciones cada vez más complejas [Jin 1994].

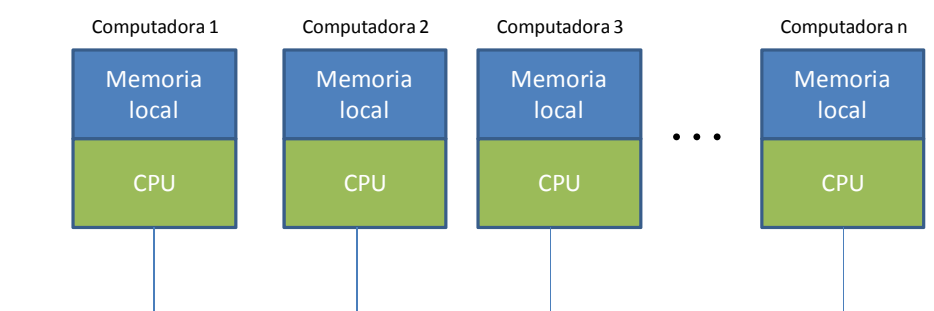
Una característica importante de la computación paralela es la estrecha relación entre los algoritmos y las arquitecturas computacionales paralelas. Dado un mismo problema, los algoritmos paralelos pueden ser desarrollados basados en diferentes modelos de arquitectura computacional paralela. El procesamiento en paralelo es una forma eficaz de procesamiento de información que favorece la explotación de los sucesos concurrentes.

### 4.3 Arquitecturas para el Procesamiento en Paralelo

Existen diferentes arquitecturas asociadas con el procesamiento en paralelo [Jin 1994]. Entre las principales se pueden mencionar los multiprocesadores y las multicomputadoras.

#### 4.3.1 Multicomputadoras

Una computadora multiprocesador consiste en múltiples procesadores interconectados compartiendo una memoria común, en cambio una multicomputadora no comparte memoria, pero interconecta varias computadoras mediante el uso de una red de alta velocidad [Jin 1994]. La Figura 4.1 muestra el esquema de una arquitectura multicomputadora.



**Figura 4.1** Arquitectura multicomputadora

La comunicación interprocesos y la sincronización es el requerimiento mayor para el buen funcionamiento de un algoritmo paralelo.

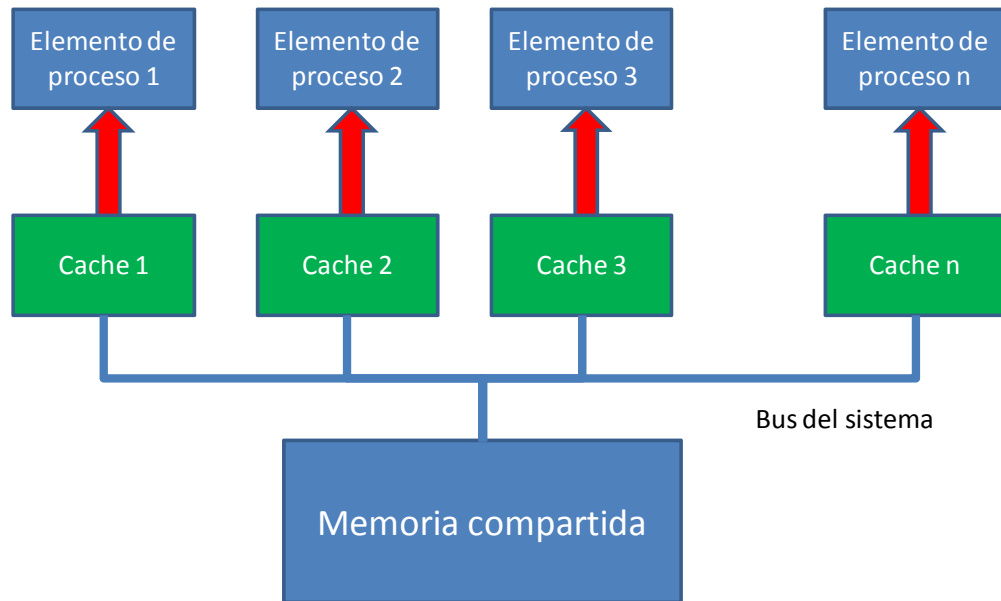
#### **4.3.2 Multiprocesadores**

En las computadoras multiprocesador, los procesos se comunican y sincronizan con los demás a través de variables compartidas; en tanto que en las multicomputadoras la comunicación y sincronización se realiza mediante el envío y recepción de mensajes. En base a esta distinción se pueden tener dos esquemas de programación: *programación con variables compartidas* y *programación con paso de mensajes* [Jin 1994]. La Figura 4.2 muestra el esquema de una computadora multiprocesador.

#### **4.4 Plataformas para el Procesamiento en Paralelo**

Además de existir diferentes arquitecturas asociadas con el procesamiento en paralelo, existen diferentes plataformas para el desarrollo de programas paralelos, entre las más populares se encuentran: multithreading, *parallel virtual machine* (PVM), *message passing interface* (MPI), openmp, cuda, etc., [Breshears, 2009].





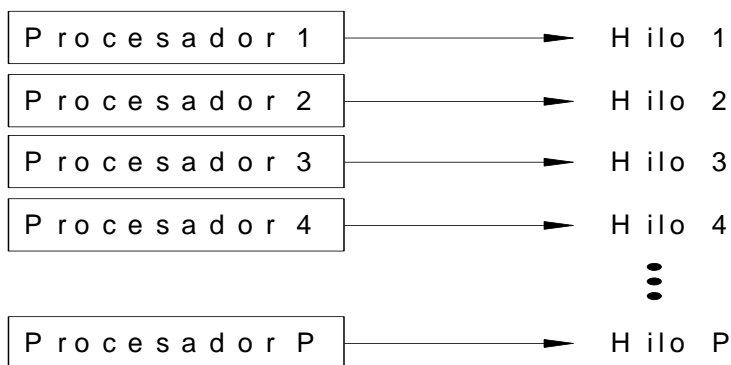
**Figura 4.2** Arquitectura multiprocesador

La palabra *multithreading* puede ser traducida como múltiples hilos de control o múltiples flujos de control. Mientras que un proceso tradicional UNIX siempre contiene un sólo hilo de control, multithreading (MT) separa un proceso en muchos hilos de ejecución, cada uno de los cuales se ejecuta por separado. En un ambiente de multiprocesadores de memoria compartida, cada hilo en el proceso puede ejecutarse en un procesador por separado al mismo tiempo, como se ilustra en la Figura 4.3. Siendo multithreading la plataforma utilizada en esta tesis

PVM es un paquete de software que permite a una colección heterogénea de computadoras UNIX o NT trabajar como una computadora paralela multiprocesador, con las siguientes características:

- i) Es fácil de instalar.
- ii) Muchas computadoras virtuales pueden coexistir en un mismo hardware.

- iii) El desarrollo de programas se basa en una librería de paso de mensajes.
- iv) Soporta C y Fortran.
- v) Permite maximizar los equipos de cómputo disponibles.
- vi) Permite controlar el flujo del programa y acceder a datos compartidos en hilos de ejecución concurrente.



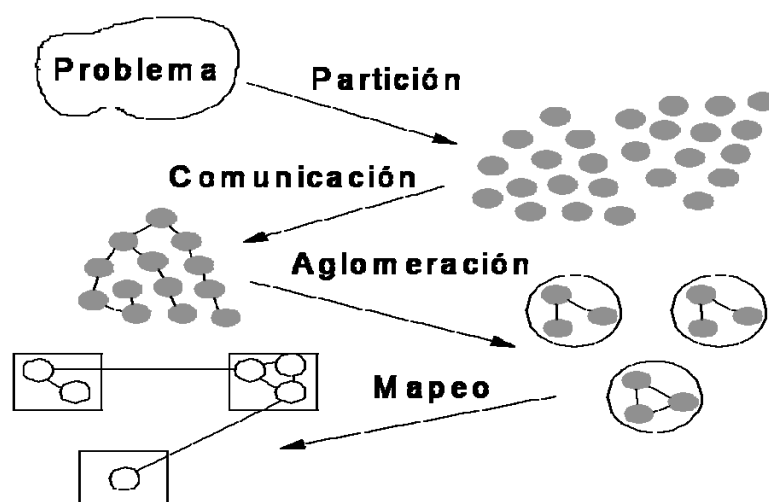
**Figura 4.3** Plataforma Multithreading.

#### 4.5 Diseño de Algoritmos en Paralelo

Diseñar algoritmos paralelos es un proceso altamente creativo, donde únicamente se deben considerar los aspectos independientes de la máquina y los aspectos específicos a la máquina deben ser dejados para más tarde. El diseño involucra cuatro etapas, las cuales se muestran en la Figura 4.4.

- i) **Partición:** El cómputo y los datos sobre los cuales se opera se descomponen en tareas. Se ignoran aspectos como el número de procesadores de la máquina a usar y se concentra la atención en explotar oportunidades de paralelismo.

- ii) **Comunicación:** Se determina la comunicación requerida para coordinar las tareas. Se definen estructuras y algoritmos de comunicación.
- iii) **Aglomeración:** El resultado de las dos etapas anteriores es evaluado en términos de eficiencia y costos de implementación. De ser necesario, se agrupan tareas pequeñas en tareas más grandes.
- iv) **Mapeo:** Cada tarea es asignada a un procesador tratando de maximizar la utilización de los procesadores y de reducir el costo de comunicación. La asignación puede ser estática, (se establece antes de la ejecución del programa) o a tiempo de ejecución mediante algoritmos de balanceo de carga.



**Figura 4.4** Etapas del Diseño de Algoritmos en Paralelo.

#### 4.6 Medición de la Eficiencia de un Algoritmo en Paralelo

Cuando se evalúa un sistema en paralelo, se tiene interés en conocer que tanto se ha mejorado la velocidad de cómputo, al paralelizar una aplicación determinada sobre la aplicación secuencial correspondiente. El Speed-up es una medida del beneficio relativo en la solución de un problema en paralelo y se define como la relación del tiempo que toma

resolver un problema en una computadora con un sólo elemento del proceso (procesador) contra el tiempo requerido para resolver el mismo problema en una computadora en paralelo con  $p$  elementos del proceso idénticos, Esta relación se muestra en la Ec. (4.1).

$$Speed - up = \frac{t_1}{t_p} \quad (4.1)$$

Donde

$t_1$  es el tiempo de ejecución obtenida en un sólo elemento de proceso

$t_p$  es el tiempo de ejecución obtenida en  $p$  elementos de proceso

# CAPÍTULO 5

## PLANEACIÓN Y OPTIMIZACIÓN DE TRAYECTORIAS DE ROBOTS MANIPULADORES

El objetivo de la planeación de trayectorias es generar las entradas de referencia para los sistemas de control de movimiento, a través de una secuencia en el tiempo de valores obtenidos mediante una función polinomial que interpola la trayectoria deseada. En este capítulo se describe el proceso de diseño de una plataforma de software en la planeación y optimización de trayectorias para robots manipuladores. Se describe un algoritmo de optimización que minimiza la energía mecánica haciendo uso de un algoritmo genético. También se describe un algoritmo que minimiza el tiempo de traslado haciendo uso de la combinación de un algoritmo genético y el método bisecciones sucesivas. Posteriormente se describe el uso de técnicas de procesamiento en paralelo basadas en multithreading con el objetivo de mejorar el tiempo de cómputo de los algoritmos de optimización.

### 5.1 Proceso de Planeación y Optimización de Trayectorias

Un robot manipulador debe tener la capacidad de moverse de una posición inicial hasta una posición final de una ruta deseada. La transición deberá ser caracterizada por leyes de movimiento que hagan que los actuadores ejerzan fuerzas generalizadas en las uniones que no violen los límites de saturación y no exciten los modos resonantes típicamente no modelados de la estructura. Es entonces necesario buscar algoritmos de planeación que generen trayectorias suaves adecuadas. Este algoritmo debe de incluir el cálculo de la cinemática y dinámica de los robots manipuladores, así como de un método de optimización que en este caso es un algoritmo genético.

Las variables a optimizar en el algoritmo de planeación de trayectorias que aquí se presentan son: el consumo de energía y el tiempo de traslado, considerando tanto posición como orientación de la herramienta en el proceso de llevar la herramienta del robot manipulador de seis grados de libertad a través de un lugar geométrico determinado. En términos generales el algoritmo de planeación y de optimización de trayectorias, propuesto para calcular el consumo de energía así como para determinar el tiempo mínimo de traslado del robot manipulador, se muestra en la Figura 5.1.

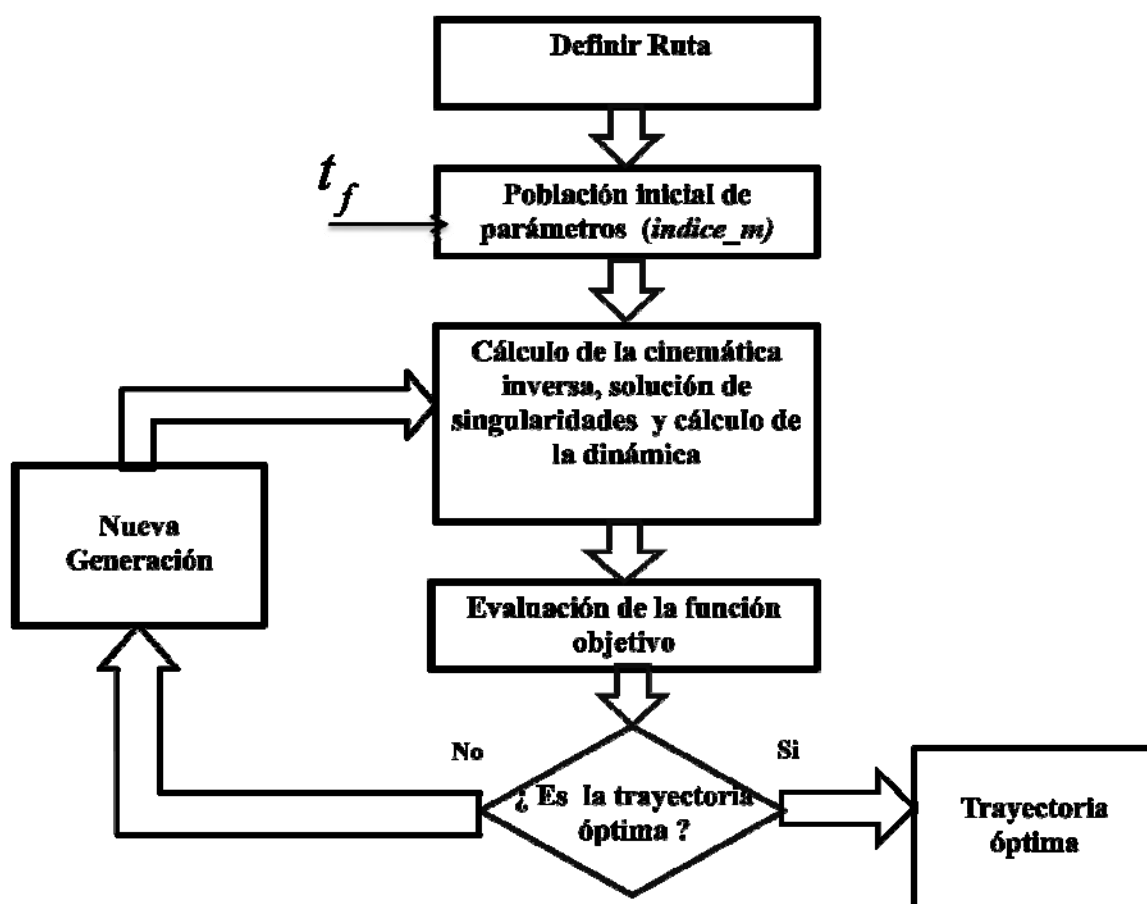


Figura 5.1 Algoritmo propuesto para el proceso de planeación y optimización de trayectorias.

El algoritmo inicia por definir la ruta a seguir por el robot manipulador, ruta que es parametrizada por una función polinomial de octavo grado. Posteriormente se genera la población inicial que requiere un algoritmo genético para la búsqueda de la función objetivo. Se continúa con el proceso calculando la cinemática, solución de las singularidades (si se presentan) y el cálculo de la dinámica, determinando una trayectoria por cada integrante de la población. Cada trayectoria generada se evalúa para determinar su aptitud, si la aptitud del conjunto de trayectoria, no cumple con la función objetivo, se genera una nueva población repitiéndose el ciclo hasta encontrar la aptitud de la función que determine la trayectoria óptima.

## 5.2 Singularidades

Si se tiene una matriz de transformación lineal que relacione la velocidad de las articulaciones con la velocidad en el espacio de trabajo del robot manipulador, una pregunta razonable sería: ¿es invertible no singular ésta matriz? Si la matriz es no singular, entonces se puede invertir para calcular las velocidades de las articulaciones partir de las velocidades cartesianas dadas a través de la ecuaciones (5.1), [Craig, 1986].

$$\dot{\Theta} = J^{-1}(\Theta)v \quad (5.1)$$

Donde

$\dot{\Theta}$  es el vector de velocidades de las articulaciones

$\Theta$  es el vector de ángulos de articulación del robot manipulador.

$v$  es el vector de velocidades en el espacio de trabajo del robot manipulador.

Es importante saber si se puede invertir el Jacobiano de las ecuaciones (5.1), para todos los valores de  $\Theta$ , si no ¿Dónde no es invertible? La mayoría de los manipuladores tienen valores de  $\Theta$  para los que los que el Jacobiano se vuelve singular. El peligro de aplicar la ecuación (5.1) en un sistema de control de robots es que, en un punto singular las

velocidades de las articulaciones se vuelven infinitas. Todos los manipuladores tienen singularidades en los límites o frontera de su espacio de trabajo y la mayoría tienen regiones de singularidades dentro del espacio de trabajo. Las singularidades en la frontera del espacio de trabajo ocurren cuando el manipulador está completamente extendido o doblado sobre sí mismo, de tal forma que el efector final se encuentra en (o muy cerca) de los límites del espacio de trabajo. Mientras que las singularidades dentro del espacio de trabajo, ocurren lejos de los límites del espacio de trabajo, generalmente se producen por el alineamiento de dos o más ejes de articulaciones.

La solución que se propone en esta tesis, para la solución de las singularidades está limitada a cada trayectoria y se da por medio de la observación, por ejemplo, en la trayectoria óptima de una ruta propuesta en los casos de estudio, se presenta una singularidad dentro del espacio de trabajo en el centro de la trayectoria. Los ejes de la articulación 5 se alinean con las de la articulación 4 y 6, produciéndose un giro de 180 grados en la articulación 4 y de -180 grados en la articulación 6. Se detecta que en el resto de la trayectoria conservan esta orientación. Para conservar la orientación inicial y no se produzca el giro en las articulaciones, se propone aplicar el siguiente algoritmo:

<pre> if (<math>\theta_{4(i)} - \theta_{4(i-1)} \geq 180</math>) { <math>\theta_{6(i)} = \theta_{6(i)} + \theta_{4(i)}</math> <math>\theta_{5(i)} = -\theta_{5(i)}</math> <math>\theta_{4(i)} = \theta_{4(i-1)}</math> } </pre>
---

De tal manera que no se permite el giro de  $\theta_6$  y se conserva la misma orientación de  $\theta_4$ , reorientando el sentido de  $\theta_5$ .



### 5.3 Optimización de Energía Mecánica en la Planeación de Trayectorias Aplicando un Algoritmo Genético.

Uno de los objetivos de este trabajo es minimizar el consumo de energía en el proceso de llevar la herramienta de trabajo de un robot manipulador de seis grados de libertad (DOF) a través de un lugar geométrico determinado, en un tiempo de traslado  $t_f$  dado considerando los límites de los pares motrices.

El algoritmo de optimización presentado en este trabajo minimiza la energía mecánica consumida por los actuadores del robot manipulador con la restricción que limita el par que satisface a los actuadores, de tal manera que la función objetivo está definida por:

$$Energia = \sum_{i=1}^6 \int_0^{t_f} |\tau_i| |\omega_i| dt \quad (5.2)$$

Sujeto a:

$$\{ |\tau_i| \leq \tau_{i,max} \} \quad (5.3)$$

Donde:

$i$  es el número del eslabón

$t_f$  es el tiempo de traslado sobre la trayectoria

$\tau_i$  es el par motor sobre el eslabón  $i$

$\omega_i$  es la velocidad rotacional del eslabón  $i$

$\tau_{i,max}$  es el límite del par motor para el eslabón  $i$

Para resolver el modelo de optimización se implementa un algoritmo genético que hace uso de una función polinomial de octavo grado. Función que permite la generación de un gran número de trayectorias través de la variable  $indice\_m$ , lo que es muy conveniente para el algoritmo genético y que se muestra en la ecuación (2.91). En la figura 5.2 se muestra a través de un diagrama de flujo el algoritmo de planeación de trayectorias óptimas.

El algoritmo inicia definiendo una ruta para un tiempo de traslado  $t_f$  dado, que consiste en establecer las coordenadas con la posición y orientación  $(x, y, z, \alpha, \beta, \gamma)$  de los puntos extremos de esta, la orientación proporcionada es fija. Esta ruta se divide en una cantidad  $n$  de puntos a través de la ecuación (5.4).

$$n = \frac{t_f}{\text{delta}} \quad (5.4)$$

Donde

$\text{delta} =$  Periodo de muestreo.

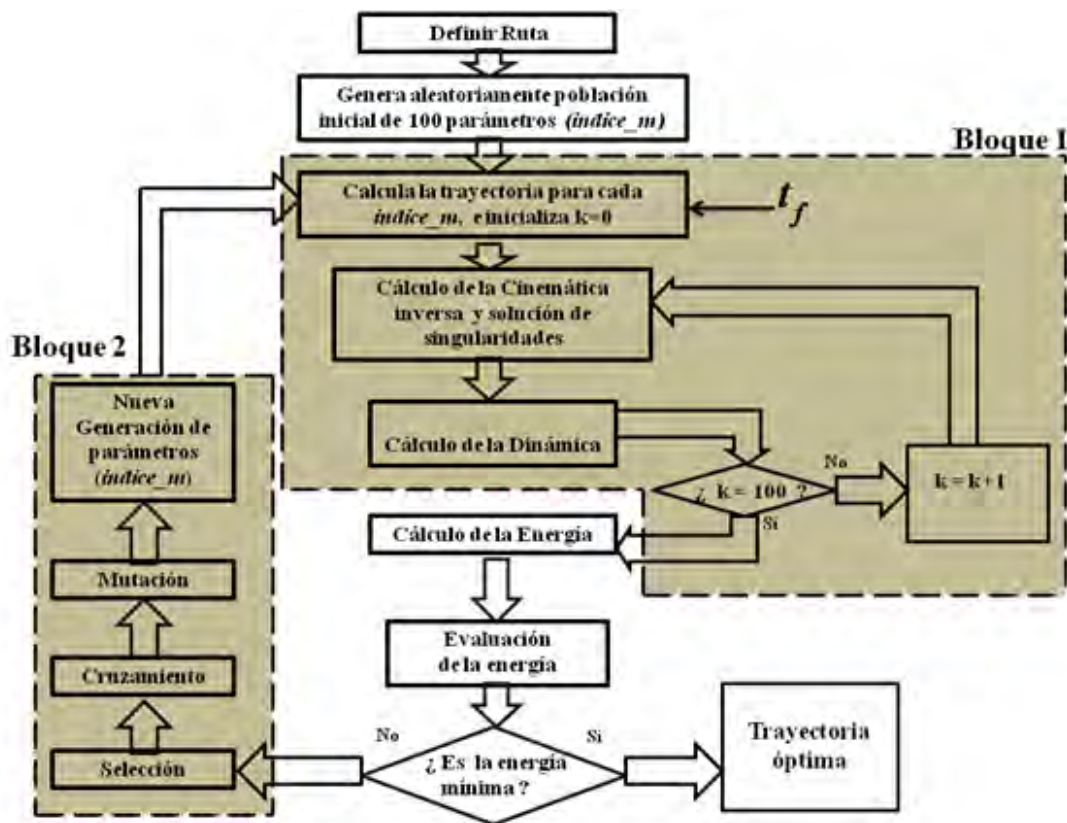
Posteriormente se genera aleatoriamente la población inicial de  $\text{indice}_m$  (dentro del rango 1.458334 y 2.1875) que en este caso son 100 individuos,. El parámetro  $\text{indice}_m$  es representado en un formato binario de veinte bits, convirtiendo este número binario en un número decimal. Con la representación decimal del  $\text{indice}_m$  se calculan las trayectorias en el espacio cartesiano, con las Ecs. (2.30), (2.31) y (2.32).

Posteriormente se hace el cálculo de la cinemática inversa, con lo que se obtiene la trayectoria en el espacio de las uniones del robot manipulador. La trayectoria en el espacio de las uniones, contiene el ángulo  $\theta_i$  de cada articulación en punto de la trayectoria, para que el efector final del robot manipulador pueda trasladarse sobre ésta en el espacio cartesiano, con la posición y orientación deseada. Posteriormente se calcula la dinámica del robot manipulador, obteniéndose los pares motrices  $\tau_i$  y velocidades rotacionales  $\omega_i$  de cada articulación. Una vez calculada la dinámica de las 100 trayectorias en el espacio de las uniones, se está en condiciones de determinar el consumo de energía de cada una de ellas.

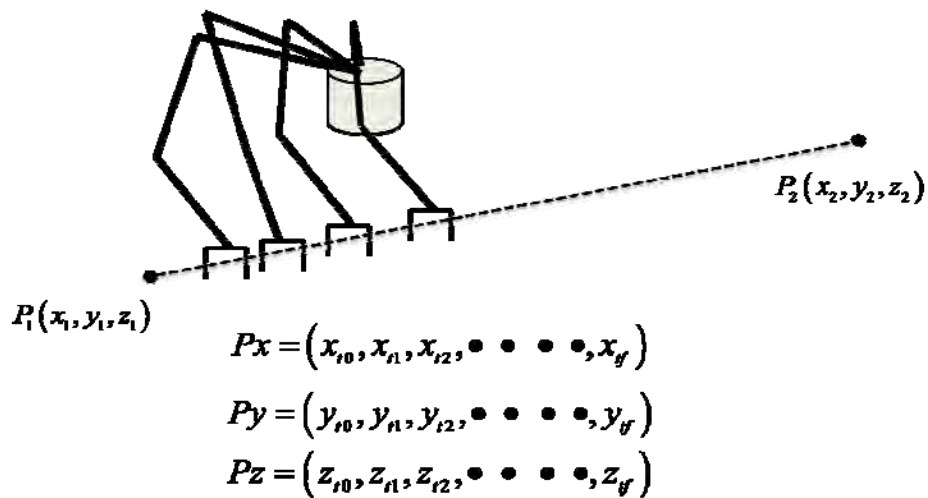
La ecuación (5.2) considera que la energía total consumida por los actuadores en las articulaciones del robot manipulador en una trayectoria es la suma de las energías de cada articulación (seis en este caso), la energía de cada articulación se obtiene mediante la integral definida de 0 a  $t_f$  del producto del valor absoluto de la potencia desarrollada por el

par motor correspondiente a cada articulación. Las energías de las 100 trayectorias se ordenan en un vector de forma ascendente para tener a la población clasificada de menor a mayor consumo de energía.

Los miembros de la población se clasifican en base a la menor energía mecánica consumida por los seis actuadores del robot manipulador, para saber que tan “buena” es la solución encontrada y preguntar si el criterio de convergencia del algoritmo se ha alcanzado. A la trayectoria de menor consumo de energía se le da mayor calificación y a la de mayor consumo de energía se le da menor calificación. El criterio de convergencia del algoritmo es que el proceso iterativo se detiene, cuando la diferencia de la energía consumida por las primeras 96 trayectorias, es menor de una millonésima.



**Figura 5.2** Algoritmo de optimización del consumo de energía sobre una trayectoria de un robot manipulador en un tiempo dado.



**Figura 5.3** División de la ruta en  $n$  puntos

Después de saber la aptitud de cada cromosoma se procede a elegir los cromosomas con mejor aptitud (mayor calificación), o sea con mayor probabilidad de ser seleccionados. Esta selección se hace de acuerdo a un proceso llamado “Ruleta Rusa”, (ver Figura 3.3) seleccionándose 50 parejas, llamados “Padre1” y “Padre2”. A cada pareja “Padre1” y “Padre2”, se les aplica el operador genético “cruzamiento”, (ver Figura 3.4) que representa la reproducción sexual para generar dos descendientes llamados “Hijo1” y “Hijo2”. Ambos descendientes heredan genes de cada uno de los padres. Este operador se conoce como operador de cruce basado en un punto aleatorio.

Posteriormente, se aplica el operador genético de mutación que modifica al azar, en este caso, el 0.25% de los bits que conforman la totalidad de los bits de los 100 nuevos individuos, esto permite alcanzar zonas del espacio de búsqueda que no están cubiertas por los individuos de la población actual. Esto significa que se complementan al azar 5 bits de los 2000 del total de la población. Una vez realizado el proceso de mutación a los 50 pares de hijos, queda conformada la generación de remplazo. “Hijo 1” ocupa el lugar de “Padre 1” e “Hijo 2” ocupa el lugar de “Padre 2”. El proceso se repite hasta que se cumple el criterio de de convergencia. El resultado como trayectoria óptima tiene un *indice\_m* para el

tiempo de traslado  $t_f$  propuesto, los pares motrices y la velocidad rotacional de cada articulación.

#### **5.4 Optimización del Tiempo de Traslado en la Planeación de Trayectorias Aplicando un Algoritmo Genético.**

Una de las metas a cumplir de cualquier empresa, es producir con calidad la mayor cantidad de producto en el menor tiempo posible. Para esto, requiere de hacer un buen estudio de sus tiempos de producción así como de los sistemas existentes que puedan satisfacer sus necesidades. Por lo que tener sistemas que aceleren su producción es esencial.

Otro de los objetivos de este trabajo de tesis es minimizar el tiempo de traslado en el proceso de llevar la herramienta de trabajo de un robot manipulador de seis grados de libertad a través de un lugar geométrico determinado, con restricción en los pares motrices y el máximo aprovechamiento de cuando menos uno de ellos. Bajo estas condiciones de plantea una función como la ecuación (5.5) dependiente del tiempo de traslado  $t_{f\min}$  y del *indice\_m* propuesto en la función polinomial de octavo grado, que generé el par motor en alguna de las articulaciones del robot manipulador igual al par máximo proporcionado por el fabricante.

$$f(\text{indice}_m, t_{f\min}) = \tau_i(t_{f\min}) = \tau_{iMAX} \quad (5.5)$$

$$t_{f\min} \in [0.2, t_f] \quad (5.6)$$

Por lo que se puede formar la función (5.7),

$$\text{relación de } par_i(\text{indice}_m, t_{i\min}) = \frac{par\_real_i(\text{indice}_m, t_{i\min})}{par\_maximo_i} \quad (5.7)$$

Donde  $i = 1, 2, \dots, 6$

La función (5.7) tiene tres condiciones, (1) que sea mayor de uno, (2) que sea mayor de uno y (3) que sea igual a uno.

En el caso (1) se dan las siguientes condiciones:

$$t_{f \min} = t_f \quad (5.8)$$

$$t_f = \frac{t_{f \max} - t_{f \min}}{2} + t_{f \min} \quad (5.9)$$

Para el caso (2) se dan las siguientes condiciones:

$$t_{f \max} = t_f \quad (5.10)$$

$$t_f = \frac{t_{f \max} - t_{f \min}}{2} + t_{f \min} \quad (5.11)$$

El caso (3) representa la solución  $f(\text{indice}_m, t_{f \min})$  perteneciente al intervalo del  $\text{ndice}_m$  y del tiempo de traslado. A esta condición tiene de tolerancia las siguientes condiciones:

$$t_{f \max} - t_{f \min} \leq 0.001 \quad (5.12)$$

$$1 - \text{relación de par}_i(\text{indice}_m, t_{i \min}) \leq 0.001 \quad (5.13)$$

El objetivo se logra a través del uso de un algoritmo genético, combinado con el método de bisecciones sucesivas. En la Figura 5.4, se muestra el diagrama de flujo de este algoritmo. La Figura 5.4, contiene dos bloques, que están relacionados con la Figura 5.2, el bloque 1 contiene el cálculo de la trayectoria para cada  $\text{indice}_m$ , el cálculo de la cinemática inversa, la solución de las singularidades así como el cálculo de la dinámica, el bloque 2 contiene los operadores del algoritmo genético como la selección, el cruzamiento, la mutación así la nueva generación del algoritmo.

De la misma manera que en la sección anterior, para la optimización del tiempo de traslado en la planeación de trayectorias, se hace uso de una función polinomial que, como ya se ha planteado con anterioridad, permite la generación de un gran número de trayectorias suaves, a través de la variable  $\text{indice}_m$  que muestra la ecuación (2.91).

Como resultado de este algoritmo se debe tener un *indice\_m* con un tiempo de traslado  $t_{f\min}$  que permita el máximo aprovechamiento de los pares desarrollados en las articulaciones del robot manipulador. El máximo aprovechamiento de los pares, significa que para ese tiempo mínimo e *indice\_m* en ninguna articulación se debe sobrepasar el límite del par máximo propuesto, pero al menos uno debe ser igual o menor al par nominal con una diferencia de una milésima de Joule, para todos los demás índices se debe de sobrepasar el límite del par nominal en cuando menos una articulación.

El algoritmo inicia definiendo una ruta, En este caso se inicia la optimización con tres variables de tiempo  $t_f$  que es el intento inicial del tiempo de traslado mínimo,  $t_{f\max}$  y  $t_{f\min}$ , que son el extremo superior e inferior respectivamente, del intervalo de búsqueda, del método de bisecciones sucesivas.

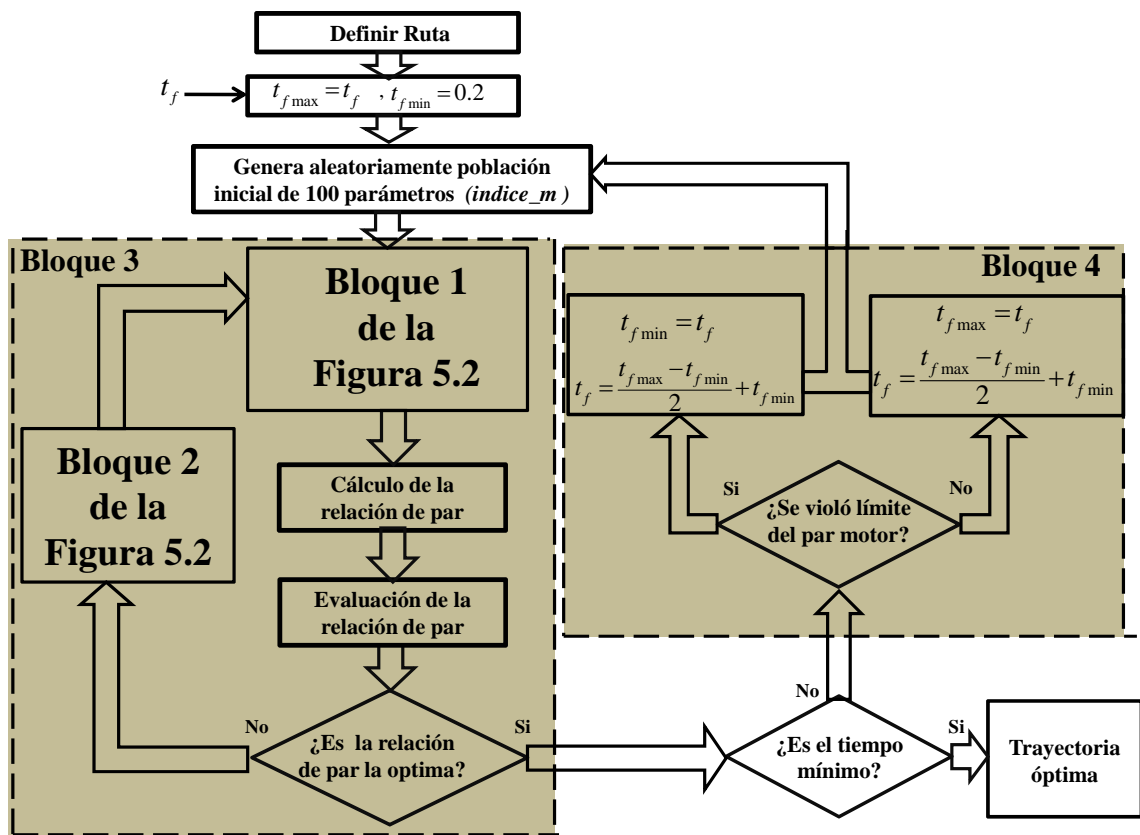
Posteriormente se genera aleatoriamente la población inicial, que en este caso son 100 individuos *indice\_m*, dentro del rango dado entre los valores 1.458334 y 2.1875. El parámetro *indice\_m* es representado en un formato binario de veinte bits, (genotipo), convirtiendo este número binario en un número decimal (fenotipo). En el bloque 1, con la representación decimal del *indice\_m* se calculan las trayectorias en el espacio cartesiano, base a las ecuaciones (2.30), (2.31) y (2.32) y el tiempo de traslado inicial  $t_f$  propuesto, quedando dividida cada trayectoria en una cantidad de puntos dada por ecuación (5.4).

Posteriormente se hace el cálculo de la cinemática inversa para cada punto de la trayectoria y la dinámica del robot manipulador, obteniéndose el par motriz  $\tau_i$  y velocidad rotacional  $\omega_i$  de cada articulación.

Una vez calculados los pares motrices de las 100 trayectorias en el espacio de las uniones, se está en condiciones de determinar la relación entre el par motriz real y el par

máximo, con la ecuación (5.7). De las seis articulaciones del robot manipulador se selecciona la articulación que tenga la mayor relación de par, de cada trayectoria.

Las relaciones de par motor seleccionadas de las cien trayectorias, se ordenan en un vector de menor a mayor, para su evaluación. La evaluación consiste en calificar la aptitud de las cien relaciones de par, dándole una mayor calificación normalizada a la de menor relación de par que a la de mayor relación de par.



**Figura 5.4** Algoritmo de optimización del tiempo de traslado sobre una trayectoria de un robot manipulador.

Una vez evaluados y calificadas las relaciones de par, es momento de preguntar si la relación de par es la adecuada para salir (esta se da cuando la diferencia de las primeras 96 relaciones de par es de una millonésima). Si la respuesta es negativa quiere decir que se tiene que generar una nueva población, mejorando la especie. Si la respuesta es positiva



quiere decir que existe un *indice\_m* óptimo para un tiempo de traslado  $t_f$  determinado, pero no necesariamente el tiempo  $t_f$  mínimo.

Si la respuesta fue negativa se tiene que buscar una nueva población, la dirección es hacia el bloque 2, (ver Figura 5.4). Esto es seleccionar, cruzar, mutar para obtener una nueva población que sustituya la población anterior, repitiéndose el ciclo hasta cumplir con la condición de salida. La selección, el cruzamiento y la mutación se hacen de acuerdo a lo expuesto en el capítulo 3 sobre algoritmos genéticos.

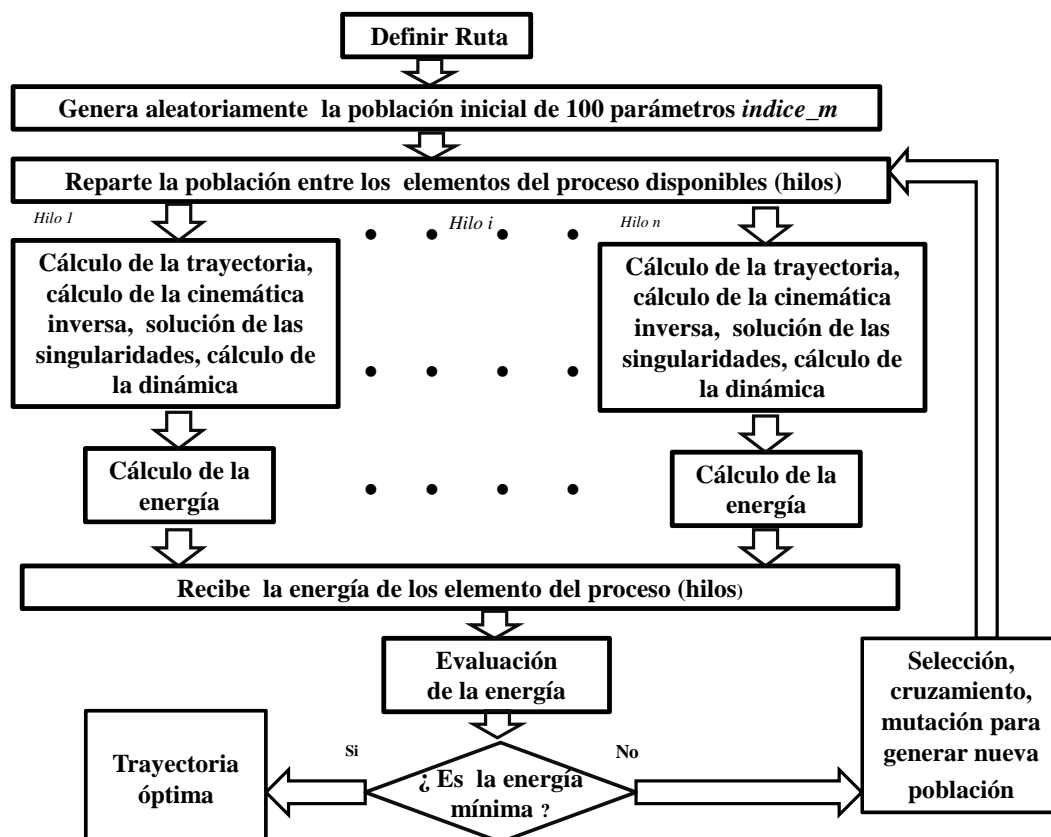
Si la respuesta es positiva, es porque se logró la condición de salida para ese tiempo. Por lo tanto se evalúa si se tiene el tiempo mínimo, dando se las condiciones de los casos (1) (2) .

Obtener una trayectoria óptima con el algoritmo optimización del tiempo de traslado en la planeación de trayectorias aplicando un algoritmo genético, es obtener un *indice\_m*, un tiempo de traslado  $t_f$ , el par motor y la velocidad rotacional de cada una de las seis articulaciones del robot manipulador.

## **5.5 Aplicación de una Técnica de Procesamiento en Paralelo a la Planeación de Trayectorias Óptimas**

El objetivo de la aplicación del procesamiento en paralelo es reducir el tiempo de cómputo utilizado en la solución de un problema, en este caso, la solución de trayectorias óptimas aplicando algoritmos genéticos para minimizar el consumo de energía así como el tiempo de traslado en el proceso de llevar la herramienta de un robot manipulador de seis grados de libertad a través de un lugar geométrico determinado. La Figura 5.5 se muestra en forma gráfica el esquema de procesamiento en paralelo basado en multithreading propuesto en esta tesis, para minimización del gasto de energía mecánica. Este algoritmo inicia con un elemento del proceso maestro definiendo una ruta, generando aleatoriamente la población

inicial de cien parámetros *indice\_m*, para posteriormente repartir la población entre el número de elementos de proceso disponibles. En este punto inicia la parte del algoritmo de minimización de energía que se paraleliza, y se soluciona de la misma forma que se hace en la sección 5.3.

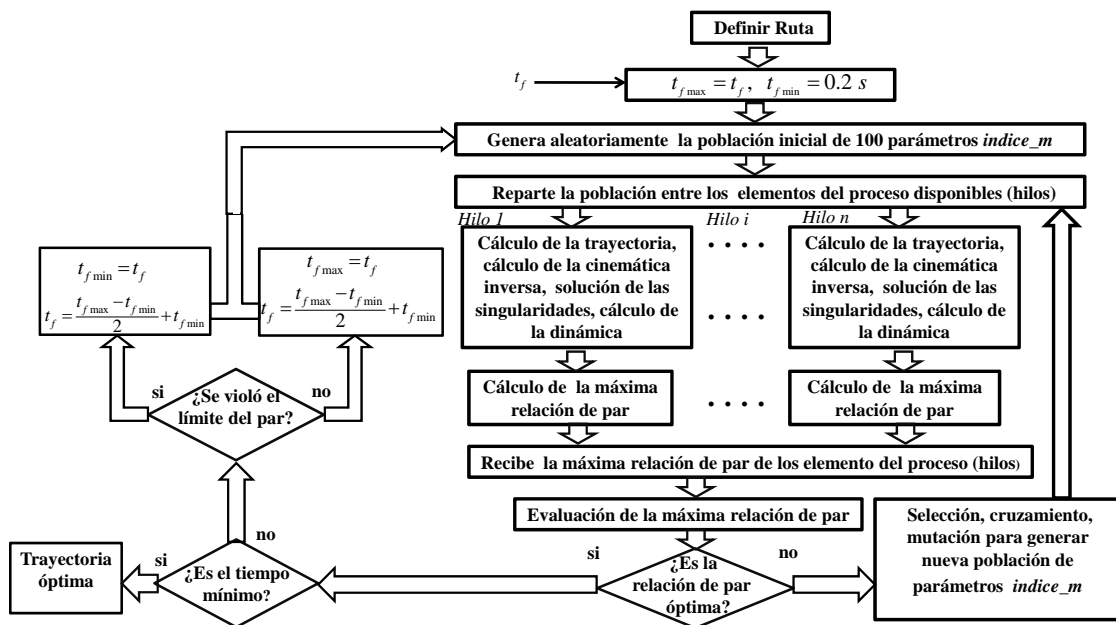


**Figura 5.5** Esquema de procesamiento en paralelo basado en multithreading en la planeación de trayectorias óptimas minimizando el gasto de energía.

Esto es, cada elemento del proceso recibe un determinado número de parámetros *indice\_m* y por cada uno de ellos calcula una trayectoria, Calcula la Cinemática inversa, soluciona las singularidades (si se presentan), calcula la dinámica, calculando el consumo de energía para cada una de las trayectorias, con lo que termina el proceso que se paraleliza del algoritmo de minimización de energía. Una vez que cada elemento de proceso ha calculado las energías de las trayectorias asignadas, el elemento de proceso maestro recibe las energías calculadas por cada hilo, las ordena y evalúa de la misma forma que se hizo en la sección

5.3 y se determina si se ha encontrado la energía mínima si esto ocurre el proceso termina, si no se busca una nueva población a través de los operadores genéticos, selección, cruzamiento y mutación. El proceso se repite hasta encontrar la energía mínima. Proceso que se realiza de la misma forma que se realizó en la sección 5.3.

La Figura 5.5 se muestra en forma gráfica el esquema de procesamiento en paralelo basado en multithreading propuesto en esta tesis, para minimización del tiempo de traslado sobre una trayectoria.



**Figura 5.6** Esquema de procesamiento en paralelo basado en multithreading en la planeación de trayectorias óptimas, minimizando el tiempo de traslado.

Este algoritmo inicia con un elemento del proceso maestro definiendo una ruta, generando aleatoriamente la población inicial de cien parámetros *indice\_m*, para posteriormente repartir la población entre el número de elementos de proceso disponibles. En este punto inicia la parte del algoritmo de minimización del tiempo de traslado que se paraleliza, y se soluciona de la misma forma que se hace en la sección 5.4. Esto es, Cada elemento del proceso recibe un determinado número de parámetros *indice\_m* y por cada

uno de ellos calcula una trayectoria, Calcula la cinemática inversa, soluciona las singularidades (si se presentan), calcula la dinámica y calculando la máxima relación de par para cada una de las trayectorias, con lo que termina el proceso que se paraleliza del algoritmo de minimización de energía. Una vez que cada elemento de proceso ha calculado la máxima relación de par de las trayectorias asignadas, el elemento de proceso maestro recibe las máximas relaciones de par calculadas continuando el proceso en la misma forma que se hizo en la sección 5.4, hasta encontrar el tiempo mínimo.

# CAPÍTULO 6

## CASOS DE ESTUDIO

En este capítulo se muestran los resultados obtenidos de tres casos de estudio de la metodología desarrollada para la planeación de trayectorias con optimización del gasto de energía u optimización del tiempo de traslado, se muestra un comparativo con los resultados del procesamiento en paralelo basadas en multithreading. Se muestran los procedimientos de validación para la optimización del gasto de energía y del tiempo de traslado.

### 6.1 Procedimiento de las Pruebas

Una vez que se ha desarrollado e implementado el algoritmo de optimización de trayectorias, haciendo uso de un algoritmo genético así como la paralelización de éste a través de técnicas de procesamiento en paralelo basadas en Multithreading, se presentan seis casos de estudio de tres rutas propuestas mostradas en la Figura 6.3. Tres casos de estudio para analizar el comportamiento del algoritmo genético, tanto en su forma secuencial como paralelo, en la planeación de trayectorias con optimización del gasto de energía y tres casos de estudio del mismo análisis en la planeación de trayectorias con optimización del tiempo de traslado.

El algoritmo de planeación de trayectorias se ha implementado en dos diferentes plataformas: En el sistema de desarrollo de LabWindows/CVI y en el software libre C++ de Linux. El programa de LabWindows/CVI, incluye un módulo para simular el movimiento del robot manipulador para una trayectoria dada. La simulación del movimiento del robot manipulador en la plataforma de LabWindows/CVI se visualiza a través de una interfaz hombre-máquina como la que se presenta en la Figura 6.1. El programa basado en C++ para Linux no incluye la simulación del movimiento del robot manipulador. Los algoritmos paralelizados se implementan únicamente bajo la plataforma de software libre C++ de

Linux y se ejecuta en un servidor de dos procesadores con cuatro núcleos cada uno, como el mostrado en la Figura 6.2.

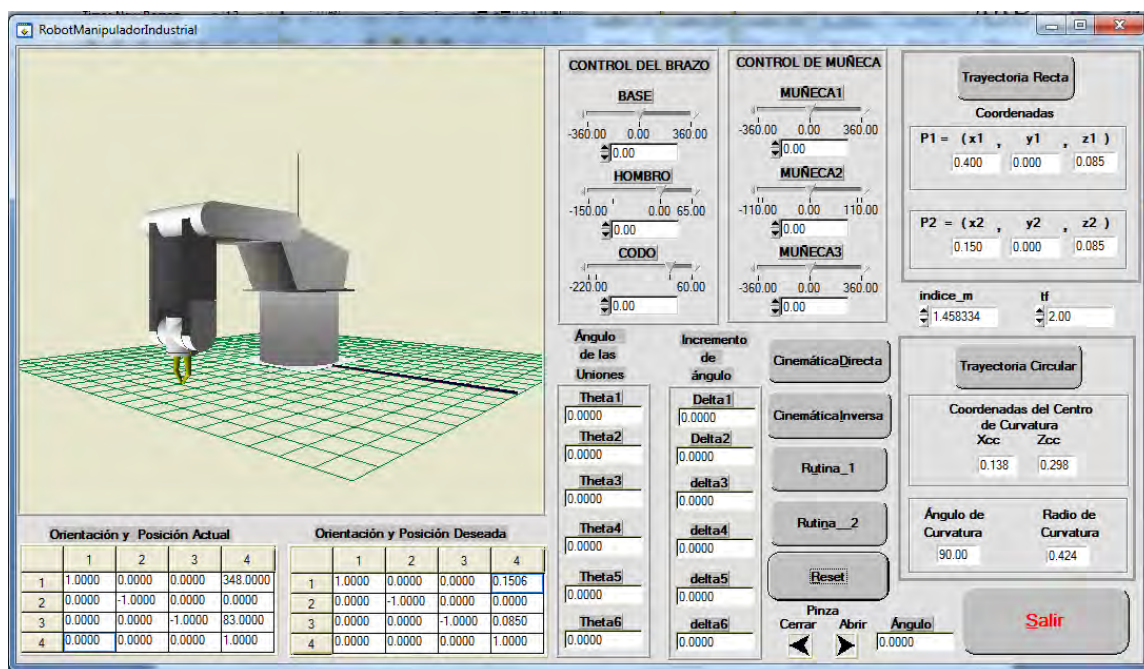


Figura 6.1 Interfaz de simulación del movimiento del robot manipulador

En la Figura 6.3 se ilustra el trazo de las rutas que se analizan tanto para la optimización del gasto de energía como para la optimización del tiempo de traslado del robot manipulador. Las coordenadas de los puntos extremos de estas rutas se muestran en la Tabla 6.1. En la Tabla 3 se muestran las coordenadas de los puntos extremos de las tres rutas propuestas, la orientación para estas rutas es fija.

La validación de los resultados del algoritmo de planeación de trayectorias con optimización del gasto de energía, se hace a través de la solución del algoritmo mediante técnica de búsqueda exhaustiva, como se explica en la siguiente sección, Este algoritmo tiene un tiempo de cómputo aproximado de dos horas y treinta minutos. En el caso del algoritmo de planeación de trayectorias con optimización del tiempo de traslado, no se hará uso del algoritmo exhaustivo para comprobar su solución, se hará bajo el razonamiento de

que el algoritmo debe de garantizar que los pares de las articulaciones del robot manipulador no sobrepasen el límite del par nominal propuesto en la Tabla 6.2.



a) Vista frontal del equipo de cómputo



b) Procesadores de la computadora



c) Sistema de refrigeración del equipo de cómputo

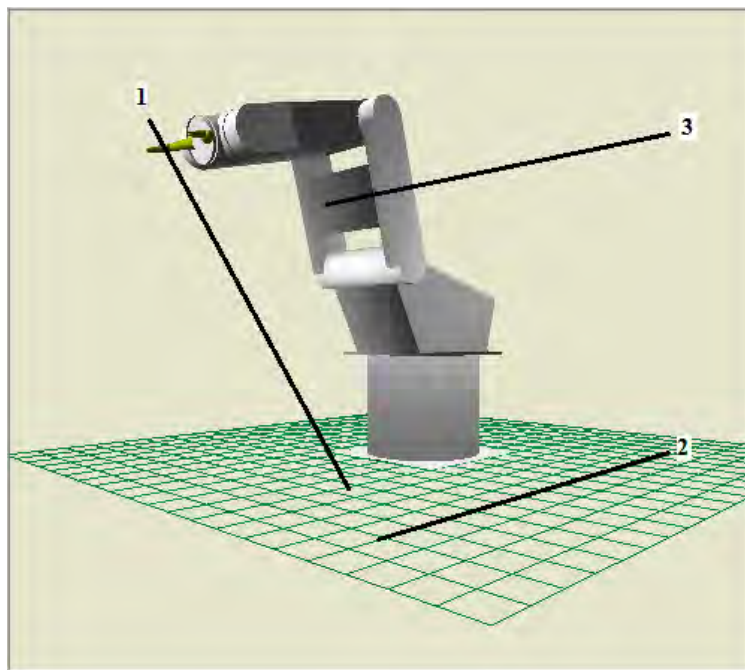


d) Disco duro del equipo de cómputo



e) Vista del equipo de cómputo en funcionamiento

**Figura 6.2** Equipo de cómputo utilizado para el desarrollo de esta tesis



**Figura 6.3** Rutas propuestas para el análisis en los seis casos de estudio.

**Tabla 6.1** Coordenadas de puntos extremos de las rutas a analizar.

Ruta	Coordenadas (x,y,z) del punto inicial			Coordenadas (x,y,z) del punto final		
	1	0.4	0.0	0.5	0.15	0.0
2	0.3	0.2	0.05	- 0.3	0.2	0.05
3	0.4	0.25	0.5	- 0.4	0.25	0.5

### 6.1.1 Procedimiento de Validación: Solución Mediante Técnica de Búsqueda Exhaustiva

Para tener una forma de validar los resultados del algoritmo de planeación de trayectorias con optimización del gasto de energía, se ha implementado la solución mediante una técnica de búsqueda secuencial. En la Figura 6.4 se ilustra el diagrama de

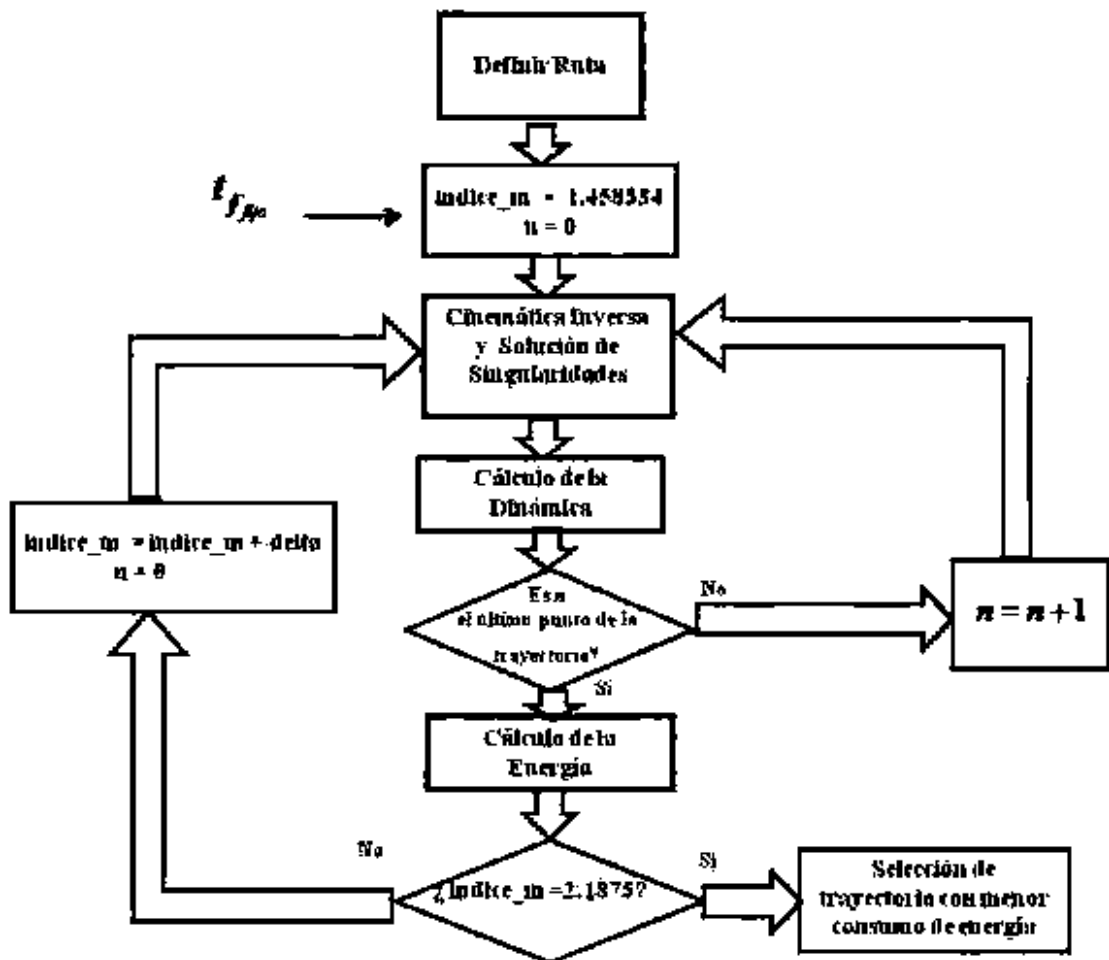


flujo del algoritmo de búsqueda exhaustiva implementado. En este algoritmo se calcula la energía de una ruta con tiempo de traslado fijo y un *indice\_m* que se va incrementando cada millonésima. De tal manera que se tiene que calcular la energía de 729,116 trayectorias y elegir la de menor gasto de energía.

Para calcular la energía se inicia por definir la ruta, posteriormente se calcula la trayectoria para el *indice\_m* = 1.458334 y se divide en *n* puntos. Posteriormente se calcula la cinemática inversa para cada punto de la trayectoria y se solucionan las singularidades si las hay. Después se calcula la dinámica para cada punto de la trayectoria. Después de haber calculado la dinámica, se calcula la energía para esa trayectoria. Si no se ha llegado al valor del *indice\_m* = 2.1875, se incrementa el valor del *indice\_m* con un *delta* = 0.000001, repitiendo el proceso como se indica en la Figura 6.4, hasta llegar al valor del *indice\_m* = 2.1875. Con este proceso se analizan 729,116 trayectorias diferentes y se escoge la de menor energía.

**Tabla 6.2** Límites nominales de par y velocidad de las articulaciones del robot manipulador

Articulación	Par Nominal (N-m)	Velocidad Nominal (rad/s)
1	± 20.0	± 49.6
2	± 25.0	± 52.4
3	± 7.0	± 41.9
4	± 0.32	± 14.9
5	± 1.1	± 24.7
6	± 0.32	± 14.9



**Figura 6.4** Algoritmo de planeación de trayectorias con optimización del gasto de energía mediante técnica de búsqueda exhaustiva.

## 6.2 Planeación de Trayectorias con Optimización del Gasto de Energía

La ejecución de los casos de estudio que se realizaron del algoritmo de planeación de trayectorias con optimización del gasto de energía, se hicieron en el equipo de cómputo presentado en la Figura 6.2, equipo que favorece el tiempo de cómputo del algoritmo. El algoritmo fue diseñado para ejecutar el programa a través de una línea de comandos proporcionada por el usuario desde el teclado. Los datos que se proporcionan en la línea de comandos del algoritmo de planeación de trayectorias que hace uso de un algoritmo

genético, diseñado para correrse en un solo procesador, son: las coordenadas  $(x_1, y_1, z_1)$  del punto inicial, las coordenadas  $(x_2, y_2, z_2)$  del punto final y el tiempo de traslado  $t_f$  de la trayectoria.

Los datos que se proporcionan en la línea de comandos del algoritmo de planeación de trayectorias que hace uso de un algoritmo genético, diseñado para correrse en un sistema paralelo, son: el número de procesadores, las coordenadas  $(x_1, y_1, z_1)$  del punto inicial, las coordenadas  $(x_2, y_2, z_2)$  del punto final y el tiempo de traslado  $t_f$  de la trayectoria.

Igualmente, los datos que se proporcionan en la línea de comandos del algoritmo de técnica de búsqueda exhaustiva son: las coordenadas  $(x_1, y_1, z_1)$  del punto inicial, las coordenadas  $(x_2, y_2, z_2)$  del punto final y el tiempo de traslado  $t_f$  de la trayectoria

### **6.2.1 Ruta1**

Los resultados que se presentan del algoritmo de planeación de trayectorias con optimización de energía para la ruta 1 son los presentados en la Tabla 6.3, que definen la trayectoria solución. En la Figura 6.5 se muestra el comportamiento del par y velocidad rotacional que se desarrollan en cada una de las seis articulaciones del robot manipulador para la trayectoria óptima que minimiza el gasto de energía. En esta grafica se puede ver que se mueven únicamente tres articulaciones, que el tiempo de traslado es de 1.0 s, que no hay violación de par y velocidad rotacional en ninguna articulación, que el par y velocidad muestran cambios suaves, reflejo del uso de la función polinomial de octavo grado para parametrizar la trayectoria.

Al hacer un comparativo del resultado de la trayectoria óptima, se observa que entre el algoritmo genético secuencial y algoritmo genético en paralelo no hay diferencias, como debe de esperarse. La diferencia entre el *indice\_m* obtenido con el algoritmo búsqueda

exhaustiva y *indice\_m* del algoritmo genético es de 4 millonésimas, diferencia muy aceptable. En lo que respecta al gasto de energía, se puede observar que son de la misma magnitud y esto se debe a que una diferencia tan pequeña en el *indice\_m* se refleja en la séptima cifra significativa para el gasto de energía del robot manipulador propuesto. Un cambio significativo y favorable, es en el tiempo de cómputo en la búsqueda de la solución, de 8672.102 segundos que se lleva el algoritmo exhaustivo con un solo procesador, se reduce a 45.142 segundos el algoritmo genético y hasta 7.236 segundos el algoritmo genético en paralelo.

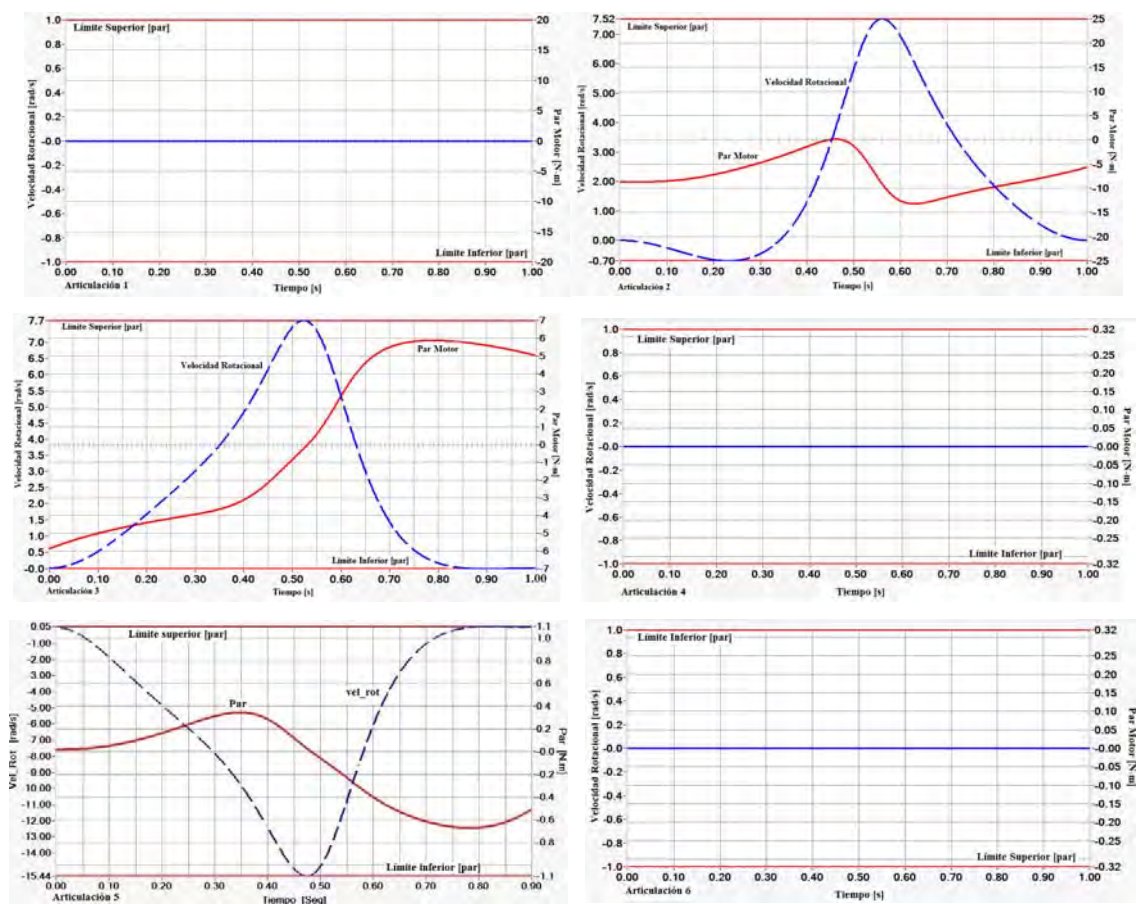
**Tabla 6.3** Trayectoria solución con consumo de energía mínimo de la ruta 1

	Tiempo de Cómputo (s)	<i>indice_m</i>	Gasto de energía (J)	<i>Speed-up</i>
Algoritmo de búsqueda exhaustiva	8672.102	1.925408	25.143368	
Algoritmo genético	45.142	1.925412	25.143368	1
Algoritmo genético paralelo	7.236	1.925412	25.143368	6.239

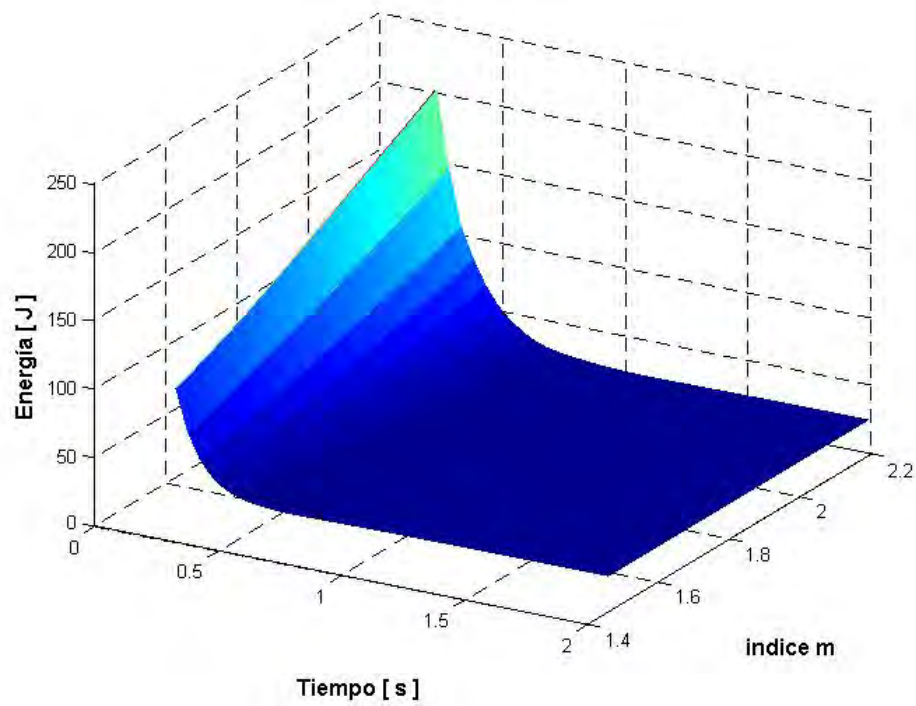
En la Figura 6.5a se muestra la forma del espacio de búsqueda de esta ruta 1, en la Figura 5.5b se muestra un acercamiento de una sección del espacio de búsqueda de la ruta 1, donde se puede observar imperfecciones en la superficie. Imperfecciones que el algoritmo genético pudo detectar, como se puede observar en la Figura 6.5c y en los resultados obtenidos de la trayectoria óptima. En la Figura 6.5c se tiene el comportamiento de la energía (J) con respecto al *indice\_m* de la trayectoria 1 con diferentes tiempos de traslado  $t_f$ , donde la trayectoria óptima fue la de un tiempo  $t_f = 1 s$ . Se pudo observar que el comportamiento de la energía a tiempos de traslado  $t_f$  superiores a 1.2 tiene una relación inversa con respecto al *indice\_m*, esto es a menor *indice\_m* mayor consumo de energía, a mayor *indice\_m* menor consumo de energía y el comportamiento de la energía a tiempos de traslado  $t_f$  inferiores a 0.7 tiene una relación directa con respecto al *indice\_m*,

esto es a menor *indice\_m* menor consumo de energía, a mayor índice\_m mayor consumo de energía.

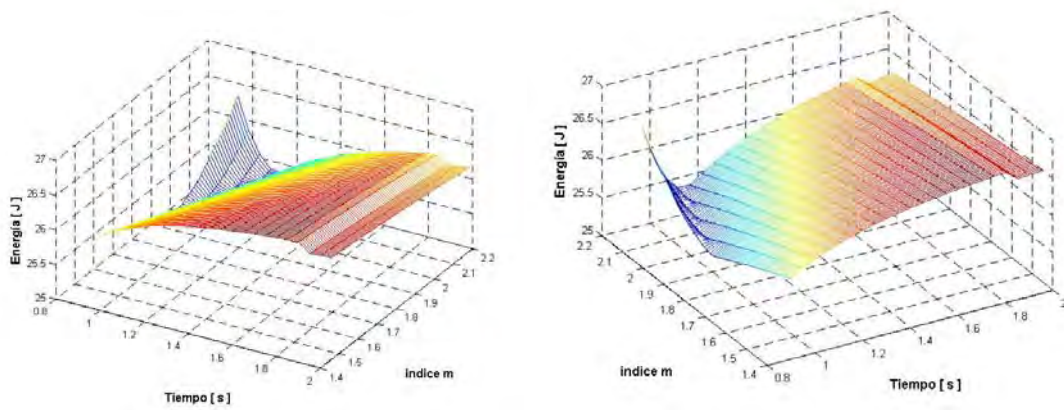
En la Figura 6.5d observa el comportamiento del tiempo de cómputo para uno, dos, tres, hasta ocho procesadores, con lo que se logra una reducción en el tiempo de cómputo del algoritmo. Si se tiene la misma tendencia, con 100 procesadores se podría tener un sistema de cómputo en línea. Este trabajo quedará como una propuesta de trabajo futuro.



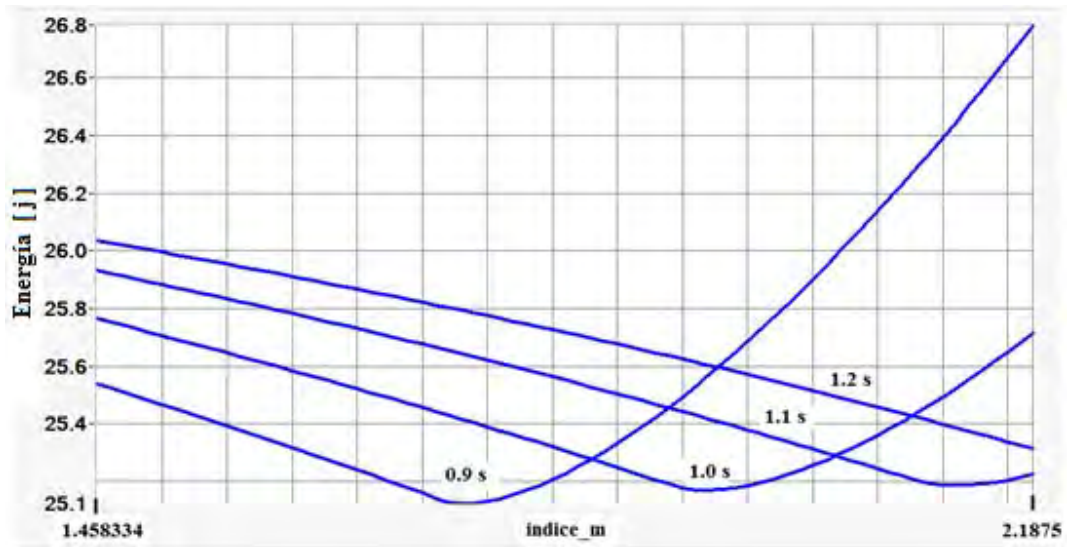
**Figura 6.5** Comportamiento del par y velocidad rotacional de las articulaciones del robot manipulador, con la trayectoria óptima de la ruta 1.



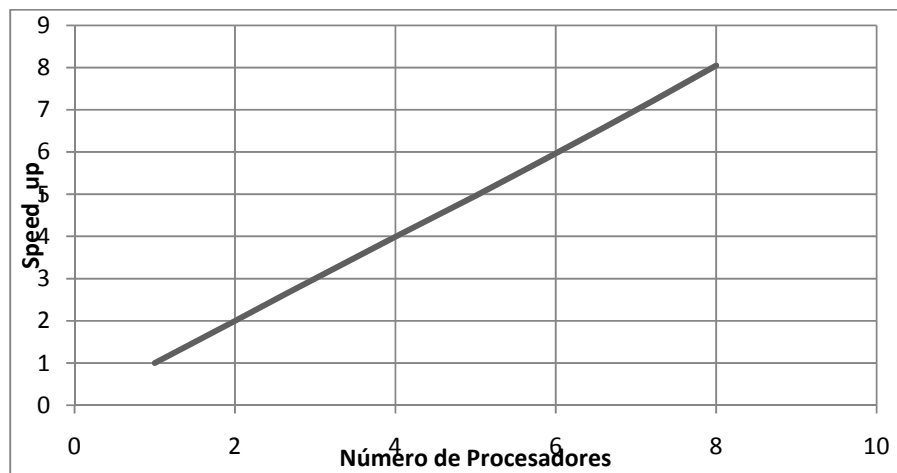
**Figura 6.5a** Forma del espacio de búsqueda de la ruta 1.



**Figura 6.5b** Acercamiento del espacio de búsqueda de la ruta 1, desde puntos de observación diferentes.



**Figura 6.5c** Comportamiento energía [ J ] con respecto al *idice\_m* de la ruta 1 para diferentes tiempos de traslado.



**Figura 6.5.d** Speed\_up del algoritmo exhaustivo paralelo correspondiente a la trayectoria óptima de la ruta1

### 6.2.2 Ruta 2

Los resultados que se presentan del algoritmo de planeación de trayectorias con optimización de energía para la ruta 2 son los presentados en la Tabla 6.4, que definen la

trayectoria solución. En la Figura 6.6 se muestra el comportamiento del par y velocidad rotacional que se desarrollan en cada una de las seis articulaciones del robot manipulador para la trayectoria óptima que minimiza el gasto de energía. En esta grafica se puede ver que se mueven las seis articulaciones, que el tiempo de traslado es de 1.0 s, que no hay violación de par y velocidad rotacional en ninguna articulación, que el par y velocidad muestran cambios suaves, reflejo del uso de la función polinomial de octavo grado para parametrizar la trayectoria.

Al hacer un comparativo del resultado de la trayectoria óptima, se observa que entre el algoritmo genético secuencial y algoritmo genético en paralelo no hay diferencias, como debe de esperarse. Un cambio significativo y favorable, es en el tiempo de cómputo en la búsqueda de la solución, de 9266.387 segundos que se lleva el algoritmo exhaustivo con un solo procesador, se reduce a 47.926 segundos el algoritmo genético y hasta 7.577 segundos el algoritmo genético en paralelo.

**Tabla 6.4** Trayectoria solución con consumo de energía mínimo de la ruta 2

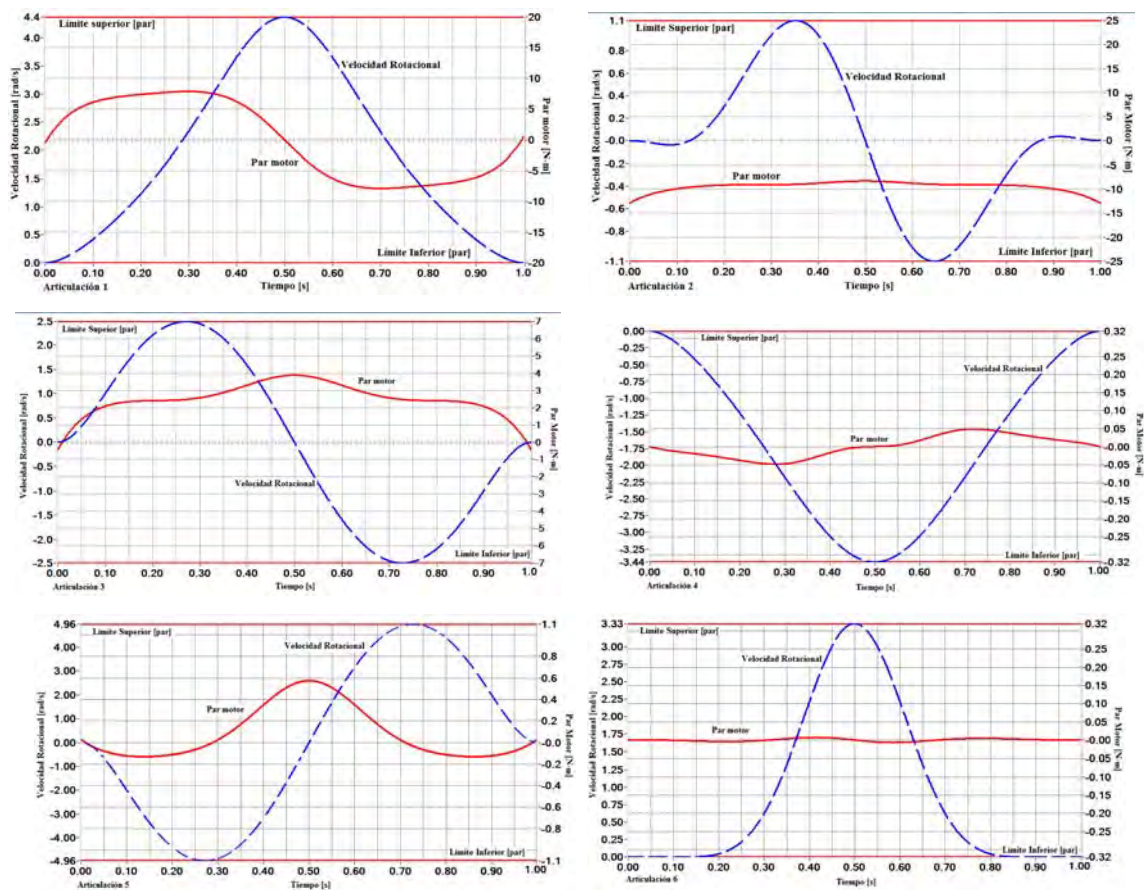
	Tiempo de Cómputo (s)	<i>indice_m</i>	Gasto de energía (J)	<i>Speed-up</i>
Algoritmo de búsqueda exhaustiva	9266.387	1.458334	19.577455	
Algoritmo genético	47.926	1.458334	19.577455	1
Algoritmo genético paralelo	7.577	1.458334	19.577455	6.325

En la Figura 6.6a se muestra la forma del espacio de búsqueda de esta ruta 2. Como puede observarse en la Figura 6.6a, esta superficie tiene una inclinación del menor *indice\_m* hacia el mayor *indice\_m*. En la Figura 6.6b se tiene el comportamiento de la energía (J) con respecto al *indice\_m* de la trayectoria 2 con diferentes tiempos de traslado  $t_f$ , donde la trayectoria óptima fue la de un tiempo  $t_f = 1$  s. Se pudo observar que el comportamiento de

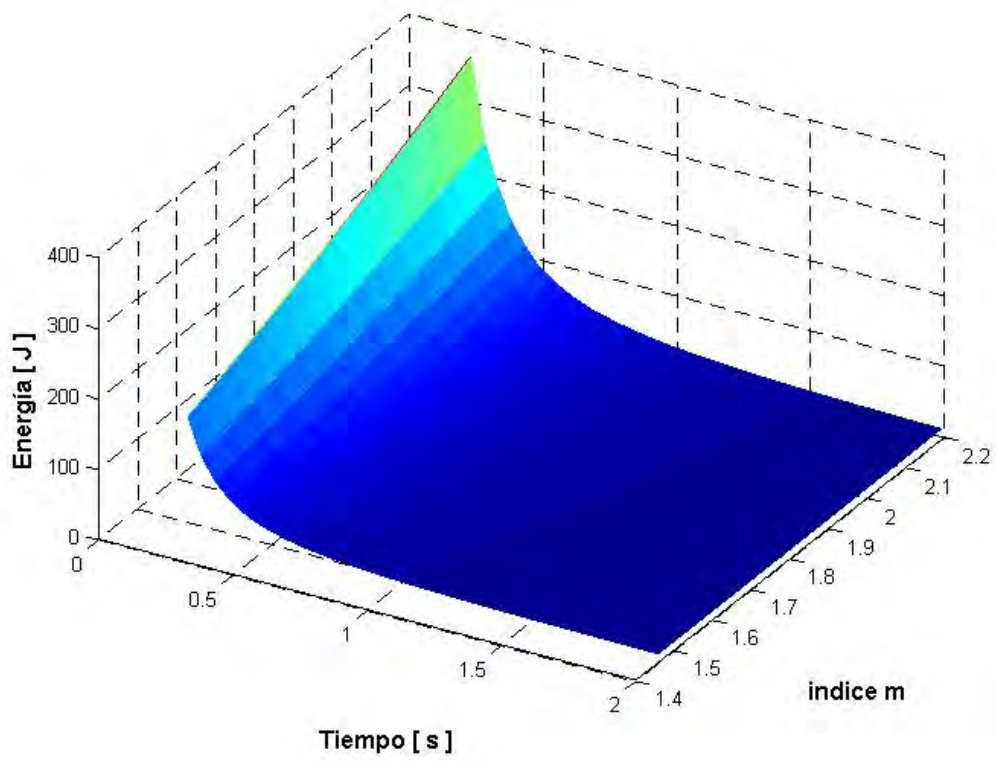


la energía para cualquier tiempo de traslado  $t_f$  tiene una relación directa con respecto al *indice\_m*, esto es a menor *indice\_m* menor consumo de energía, a mayor *indice\_m* mayor consumo de energía.

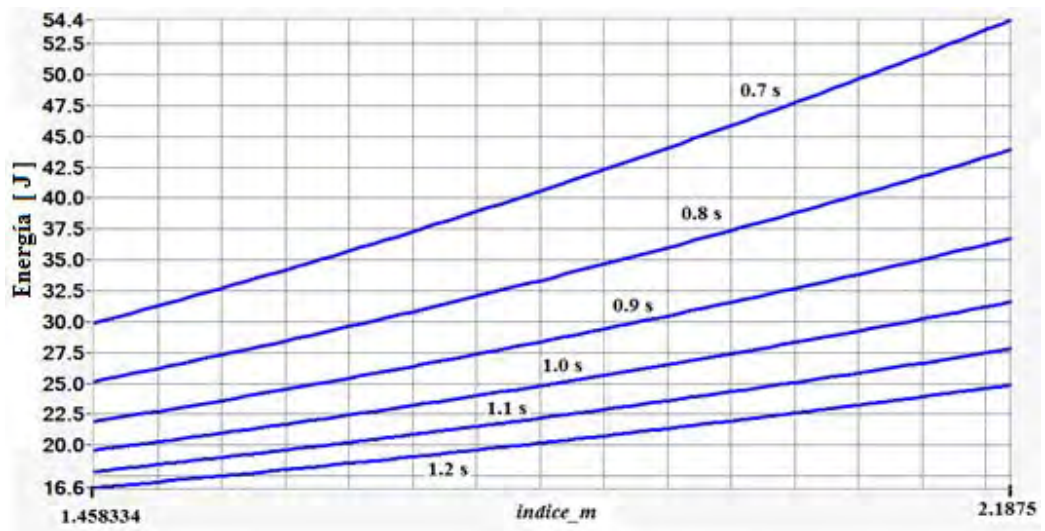
En la Figura 6.6c observa el comportamiento del tiempo de cómputo para uno, dos, tres, hasta ocho procesadores, con lo que se logro una reducción en el tiempo de cómputo del algoritmo.



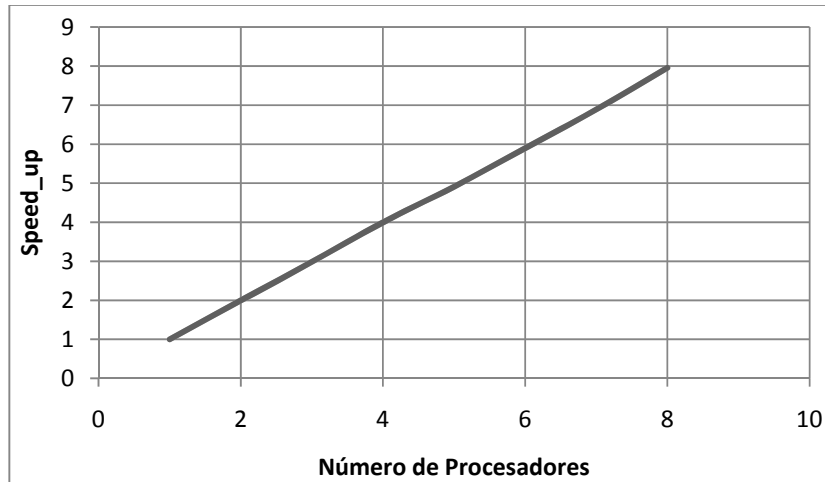
**Figura 6.6** Comportamiento del par y velocidad de las articulaciones del robot manipulador, de la trayectoria óptima de la ruta 2.



**Figura 6.6a** Forma del espacio de búsqueda de la ruta 1.



**Figura 6.6b** Comportamiento de la energía para diferentes  $t_f$  en la ruta 2



**Figura 6.6.c** Speed\_up del algoritmo exhaustivo paralelo correspondiente a la trayectoria óptima de la ruta 2

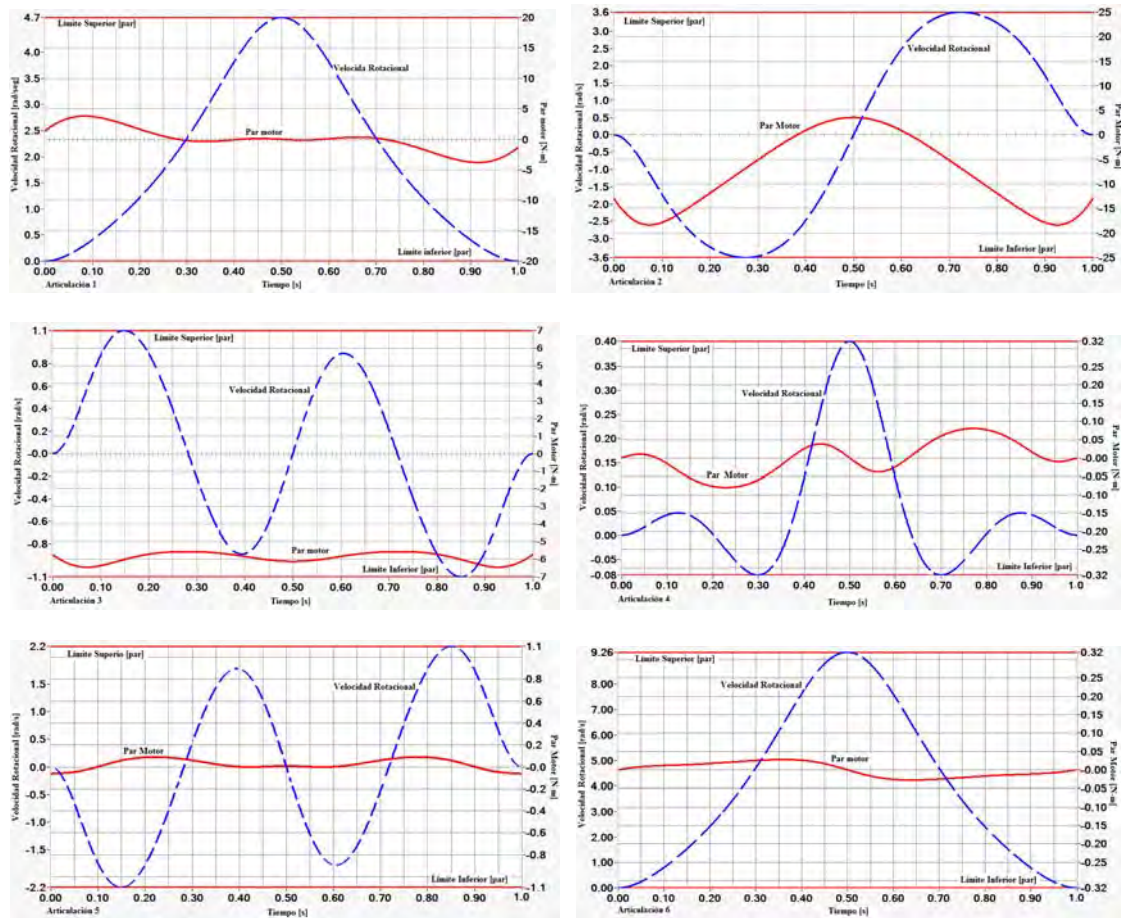
### 6.2.3 Ruta 3

De la misma manera que en el caso de las rutas 1 y 2, los resultados correspondientes a la ruta 3, se presentan en la Tabla 6.5. En la Figura 6.7 se ilustra el comportamiento del par y velocidad rotacional que se desarrollan en cada una de las seis articulaciones del robot manipulador para la trayectoria óptima que minimiza el gasto de energía. Este resultado es idéntico con los tres algoritmos, pero nuevamente se observa una diferencia favorable en el tiempo de cómputo hacia el algoritmo genético paralelo, como se puede observar en la Figura 6.7b.

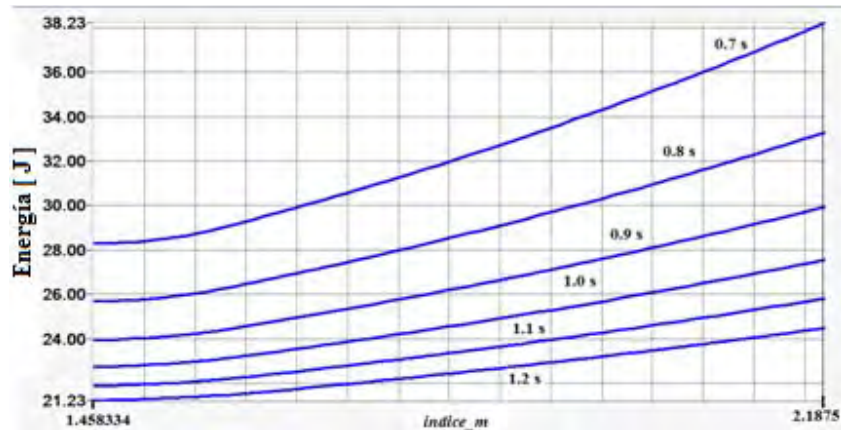
**Tabla 6.5** Trayectoria solución con consumo de energía mínimo de la ruta 3

	Tiempo de Cómputo (s)	<i>indice_m</i>	Gasto de energía (J)	<i>Speed-up</i>
Algoritmo de búsqueda exhaustiva	9034.103	1.458334	22.711851	
Algoritmo genético	47.926139	1.458334	22.711851	1
Algoritmo genético paralelo	7.59	1.458334	22.711851	6.212

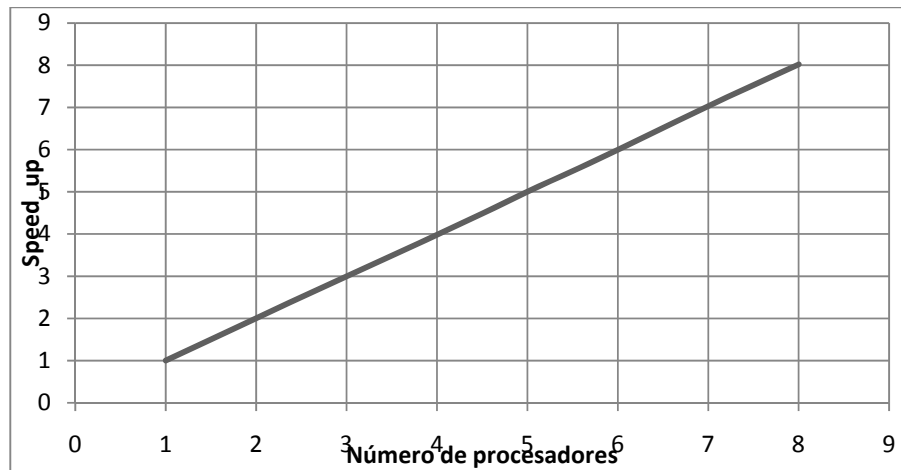
Con relación al espacio de búsqueda es similar al de la ruta 2, con la diferencia que en esta ruta el consumo de energía es mayor y en la Figura 6.7c se tiene una grafica del tiempo de cómputo de los algoritmos presentados, cuando estos se ejecutan con mas de un procesador.



**Figura 6.7** Comportamiento del par y velocidad de las articulaciones del robot manipulador, de la trayectoria optima de la ruta 3.



**Figura 6.7b** Comportamiento de la energía para diferentes  $t_f$  en la ruta 3



**Figura 6.7c.**  $Speed\_up$  del algoritmo exhaustivo paralelo correspondiente a la trayectoria óptima de la ruta 3

### 6.3 Planeación de Trayectorias con Optimización del Tiempo de Traslado

El análisis para la obtención del mínimo tiempo de traslado ( $t_f$ ), se hace con las mismas tres rutas propuestas en la Figura 6.3 y coordenadas de la Tabla 6.1. De la misma forma que en la sección 6.2, la ejecución de los casos de estudio que se realizaron del algoritmo de planeación de trayectorias con optimización del tiempo de traslado, se hicieron en el equipo de cómputo presentado en la Figura 6.3, equipo que favorece el tiempo de cómputo del algoritmo.

El tiempo mínimo de traslado debe garantizar que los pares y velocidades rotacionales de los motores no sobrepasen sus límites nominales establecidos en la Tabla 6.2. La opción de salida de los algoritmos genéticos es por un número máximo de iteraciones, con el objetivo de estandarizar el tiempo de ejecución, se hicieron pruebas, con 20, 30, 40, 60, 80 y 100 iteraciones máximo y se concluye que con 20 iteraciones es suficiente para garantizar el resultado, por lo tanto en la línea de comandos se introducen, el número de procesadores que ejecuta el programa, el número máximo de iteraciones, las coordenadas  $(x, y, z)$  de los puntos inicial y final de la ruta, además del tiempo de traslado, con que se inicia la búsqueda del tiempo mínimo de la trayectoria.

### 6.3.1 Ruta1

Los resultados del análisis hecho a la ruta 1, del algoritmo de planeación de trayectorias con optimización del tiempo de traslado se ilustran en la Tabla 6.6. De tal manera que la trayectoria óptima se da cuando el  $indice\_m = 1.760012$  y un tiempo mínimo de traslado  $t_{f\text{ minimo}} = 0.638971$  s.

En la Figura 6.8 se presenta la gráfica del comportamiento del par motor y velocidad rotacional de cada una de las articulaciones del robot manipulador. Se puede observar, que la articulación tres es la que cumple con la condición de las ecuaciones (6.1) y (6.2). Condiciones de salida necesarias para el cumplimiento del algoritmo. Además de que se tiene una suavidad en los cambios de velocidad y par inherentes al polinomio de octavo grado.

$$1 \geq \text{relacion de par optimo} \geq 0.999 \quad (6.1)$$

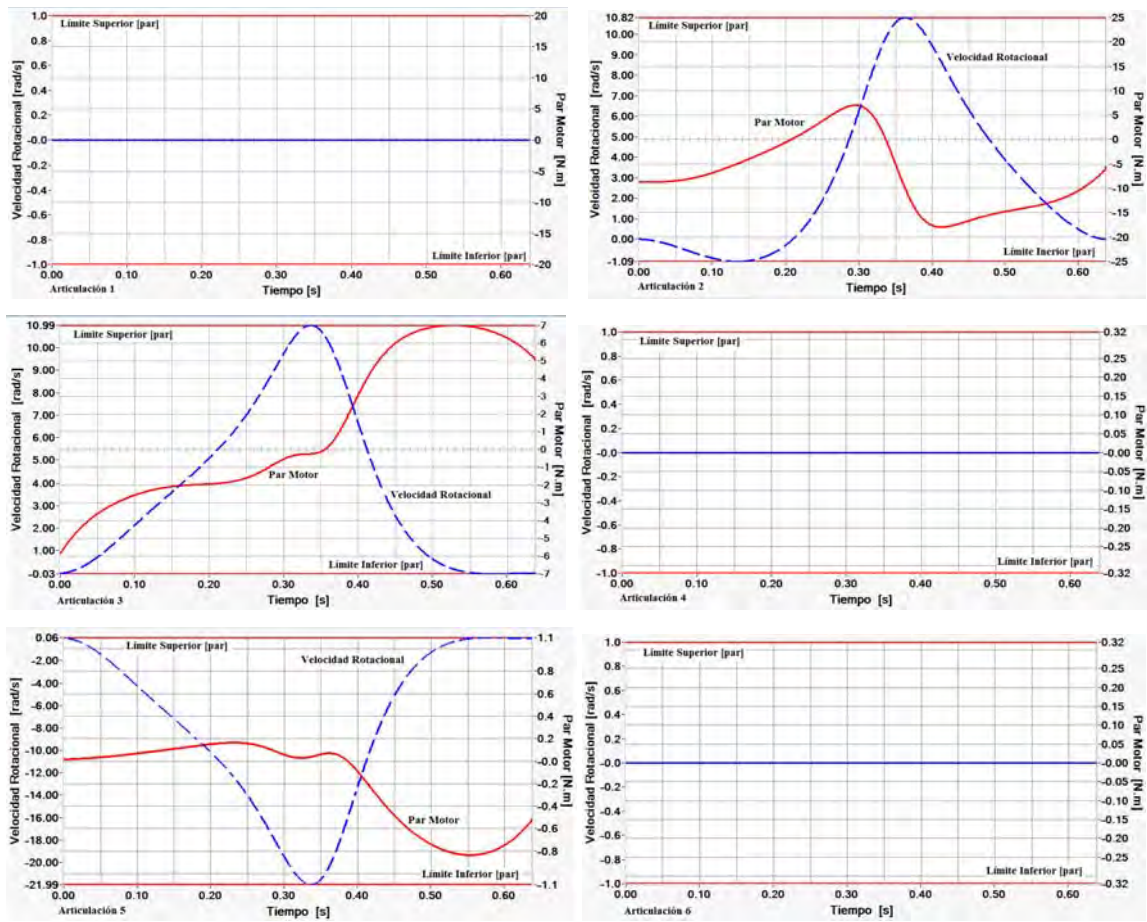
$$t_{f\text{ max}} - t_{f\text{ min}} \leq 0.001 \quad (6.2)$$

**Tabla 6.6** Trayectoria solución con tiempo mínimo de traslado de la ruta 1

	Tiempo de Cómputo (s)	<i>indice_m</i>	Tiempo mínimo de traslado $t_f$ (s)	<i>Speed-up</i>
Algoritmo genético	881.613	1.760012	0.638971	1
Algoritmo genético paralelo	116.850	1.760012	0.638971	7.523

En la Figura 6.8a se observa el espacio de búsqueda correspondiente a la ruta 1 y en la Figura 6.8b se observa el comportamiento del *Speed-up*. La comprobación del algoritmo se hace de acuerdo a los resultados obtenidos, de tiempo mínimo de traslado e *indice\_m*, de la trayectoria óptima. De tal manera que la trayectoria óptima queda representada por el valor del *indice\_m* y el valor del tiempo mínimo de traslado  $t_f$ . Las restricciones establecidas para la obtención de la trayectoria óptima, permite asegurar una sola trayectoria, dentro del rango permitido. Esto quiere decir, que para un tiempo de traslado mínimo, únicamente un *indice\_m* es permitido, cualquier otro *indice\_m*, hará que se violen los límites de par propuestos. El límite de par motor para la articulación tres es de  $\pm 7$  N-m.

En la Tabla 6.7, en la primer columna, se puede observar que el valor máximo del par en la trayectoria óptima que es 6.999992 N-m, con un valor de *indice\_m* = 1.760012 y un tiempo de traslado de 0.638971 segundos. Se sabe por las restricciones establecidas, que si se aumenta o disminuye el *indice\_m*, el máximo par real violará el límite establecido del par motor, casos que se ilustran en la segunda y tercer columna de la Tabla 6.7. Al incrementar el *indice\_m*, el par máximo que se desarrolla en la articulación es igual a 7.000001 N-m. Al disminuir el *indice\_m*, el par máximo que se desarrolla en la articulación es igual a 7.000017 N-m. Como se puede observar, se violan los límites en ambos casos. Se hacen pruebas con diferentes *indice\_m*, observándose la violación del límite nominal del par motor en la articulación 3.



**Figura 6.8** Comportamiento del par y velocidad de las articulaciones del robot manipulador, de la trayectoria óptima de la ruta 1.



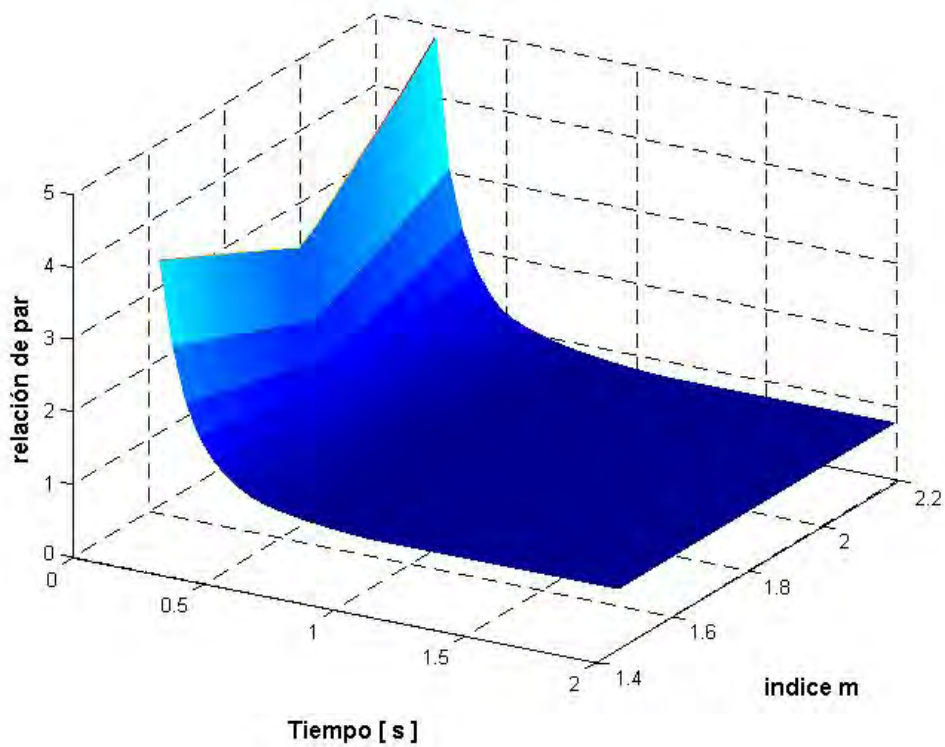


Figura 6.8 a Espacio de búsqueda de la relación de par con respecto al tiempo e índice\_m.

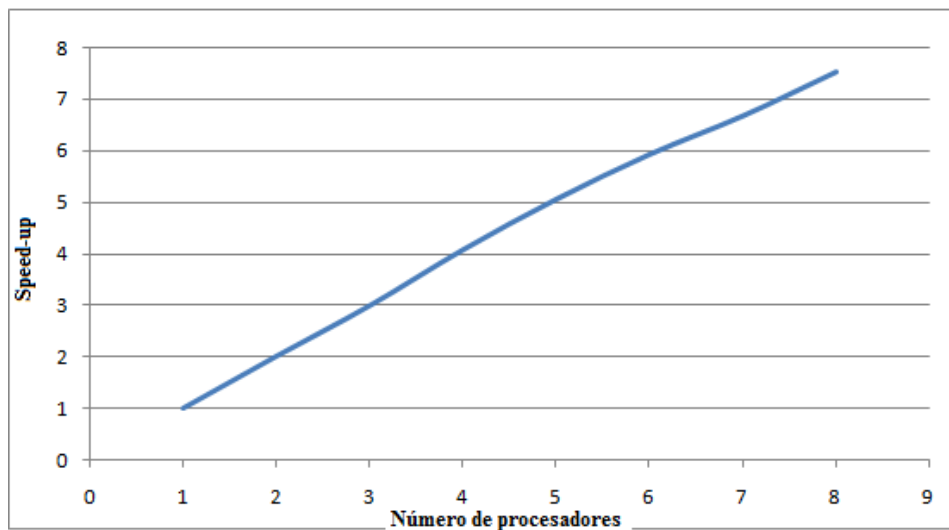


Figura 6.8 b *Speed-up* del algoritmo exhaustivo paralelo correspondiente a la trayectoria óptima de la ruta 1

**Tabla 6.7** Comportamiento del par real al aumentar y disminuir el *indice\_m* óptimo manteniendo fijo el tiempo mínimo óptimo, para la trayectoria óptima de la ruta 1.

Par real máximo articulación 3 con <i>indice_m</i> = 1.760012	Par real máximo articulación 3 con <i>indice_m</i> = 1.761012	Par real máximo articulación 3 con <i>indice_m</i> = 1.759012
6.999992 N-m	7.000001 N-m	7.000017 N-m

Otra forma de observar que el resultado de la trayectoria óptima con mínimo tiempo de traslado, es a través del siguiente razonamiento. Se sabe que el esfuerzo de un motor es mayor cuando hace un trabajo en menor tiempo. Esto significa, que en las articulaciones del robot manipulador, debe de incrementarse el par motor, cuando se disminuye el tiempo de traslado y decrementar el par motor, cuando se aumenta este mismo tiempo. Esta acción es la que se ilustra en la Tabla 6.8, en la primera columna de la Tabla 6.8, se tiene el valor máximo del par en la trayectoria óptima que es 6.999992 N-m, con un valor de *indice\_m* = 1.760012 y un tiempo de traslado de 0.638971 segundos.

En la segunda columna de la Tabla 6.8, se observa que al disminuir en una milésima el tiempo de traslado mínimo de la trayectoria óptima, el par máximo viola el límite. Cuando se aumenta el tiempo de traslado mínimo de la trayectoria óptima el par máximo en esta articulación es menor al par de la trayectoria óptima, como se observa en la tercera columna de la Tabla 6.8.

**Tabla 6.8** Comportamiento del par real al aumentar y disminuir el tiempo mínimo manteniendo fijo el *indice\_m* para la trayectoria óptima de la ruta 1.

<i>indice_m</i> = 1.760012		
Par real máximo articulación 3 con tiempo de trayectoria óptima $t_f = 0.638971$ s	Par real máximo articulación 3 con tiempo de trayectoria una milésima abajo del tiempo óptimo	Par real máximo articulación 3 con tiempo de trayectoria una milésima arriba del tiempo óptimo
6.999992 N-m	7.006180 N-m	6.993854 N-m

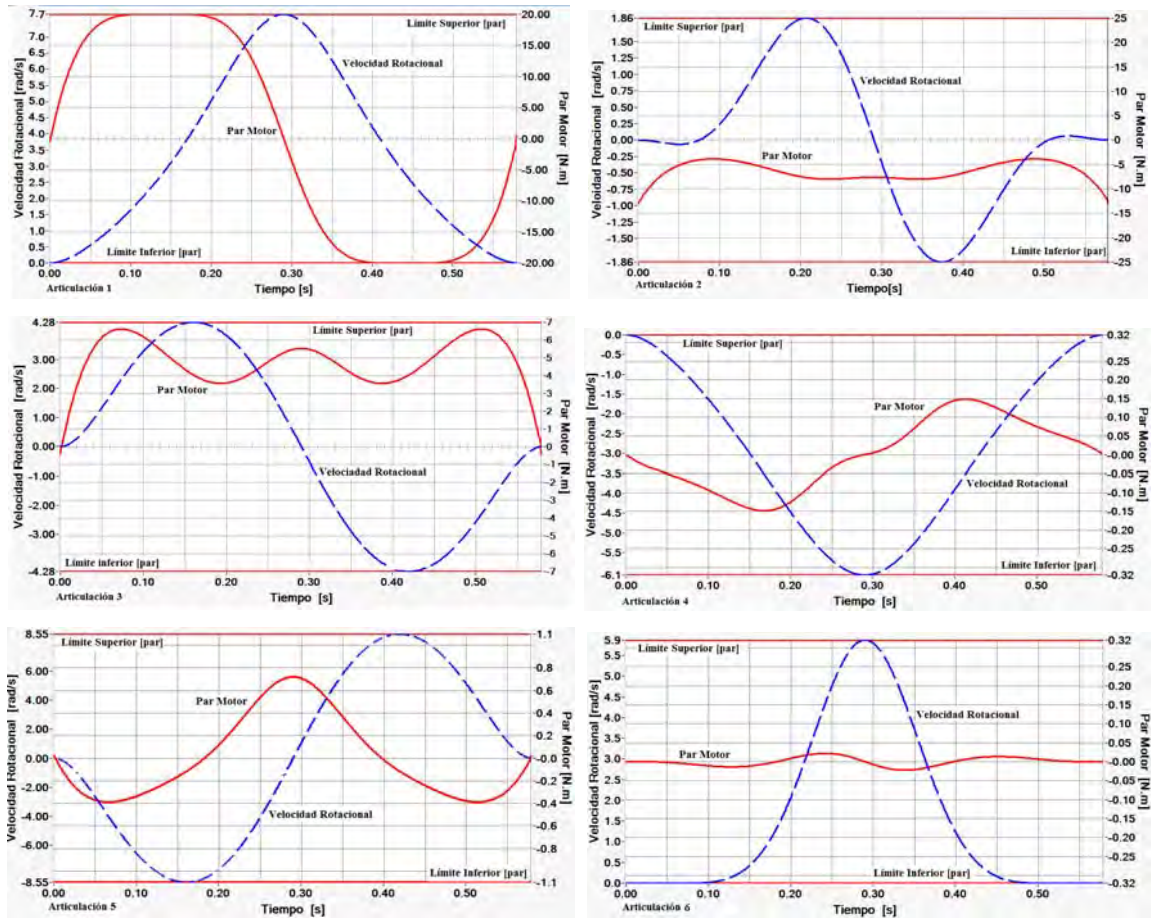
### 6.3.2 Ruta 2

En el caso de la ruta 2, la trayectoria óptima se da con un  $indice\_m$  de 1.490444 y un tiempo mínimo de traslado de 0.57997 s, resultado que se muestra en la Tabla 6.9. En la Figura 6.9 se puede observar el comportamiento del par motor desarrollado en cada una de las articulaciones. En este caso la articulación uno es la que cumple con las condiciones de las Ecs. (6.1) y (6.2), Existen dos espacios en el tiempo por donde se puede cumplir la restricción de par motor. Siendo la ruta un lugar en el espacio que inicia en el primer cuadrante y termina en el segundo cuadrante de coordenadas cartesianas, el movimiento del robot manipulador se genera de una forma simétrica. En los cambios de velocidad y par se observa que son con la suavidad que proporciona el polinomio de octavo grado. El espacio de búsqueda es semejante al de la ruta 1. En la Figura 6.9a se observa el comportamiento del tiempo de cómputo *Speed-up*.

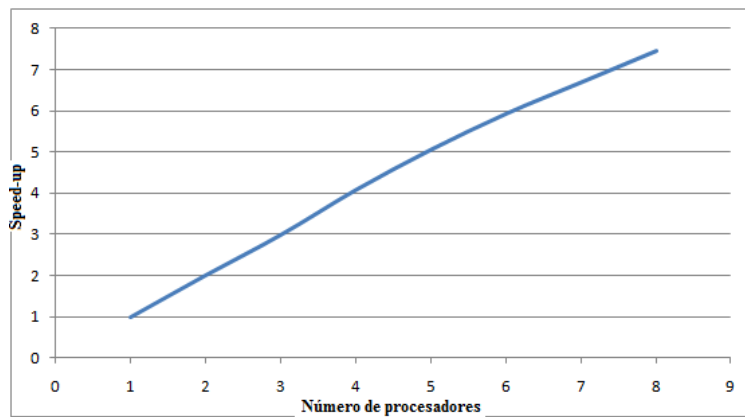
**Tabla 6.9** Trayectoria solución con tiempo mínimo de traslado de la ruta 2

	Tiempo de Cómputo (s)	$indice\_m$	Tiempo de traslado $t_f$ mínimo	<i>Speed-up</i>
Algoritmo genético	924.878	1.490444	0.579970	1
Algoritmo genético paralelo	123.794	1.490444	0.579970	7.458

Se puede observar en la Tabla 6.10, que al aumentar o disminuir el  $indice\_m$ , se violan los límites permitidos, en este caso  $\pm 20$  N-m para la articulación uno. Hay que observar que la trayectoria óptima se cumple acercándose más rápidamente al óptimo por el límite positivo, al aumentar el  $indice\_m$ , se viola el límite más rápidamente por el límite positivo, y al disminuir el  $indice\_m$ , se viola el límite más rápidamente por el límite negativo.



**Figura 6.9** Comportamiento del par y velocidad de las articulaciones del robot manipulador, de la trayectoria óptima de la ruta 2.



**Figura 6.9 b** *Speed-up* del algoritmo exhaustivo paralelo correspondiente a la trayectoria óptima de la ruta 2

**Tabla 6.10** Comparación del par real aumentando y disminuyendo el *indice\_m* óptimo manteniendo fijo el tiempo mínimo en la trayectoria óptima de la ruta 2.

$t_f = 0.57997 \text{ s}$		
Par real máximo con <i>indice_m</i> = 1.490444	Par real con <i>indice_m</i> = 1.491444	Par real con <i>indice_m</i> = 1.489444
19.997511 N-m -19.983697 N-m	20.001381 N-m -20.000521 N-m	20.000532 N-m -20.000928 N-m

El caso de mantener fijo el *indice\_m* cumple perfectamente el requisito, al aumentar el tiempo se observa que el par máximo es menor al par máximo óptimo, mientras que al disminuir el tiempo se violan los límites de par establecidos, características que se pueden ver en la Tabla 6.11.

**Tabla 6.11** Comparación del par real aumentando y disminuyendo el tiempo mínimo manteniendo fijo el *indice\_m* óptimo, trayectoria óptima ruta 2.

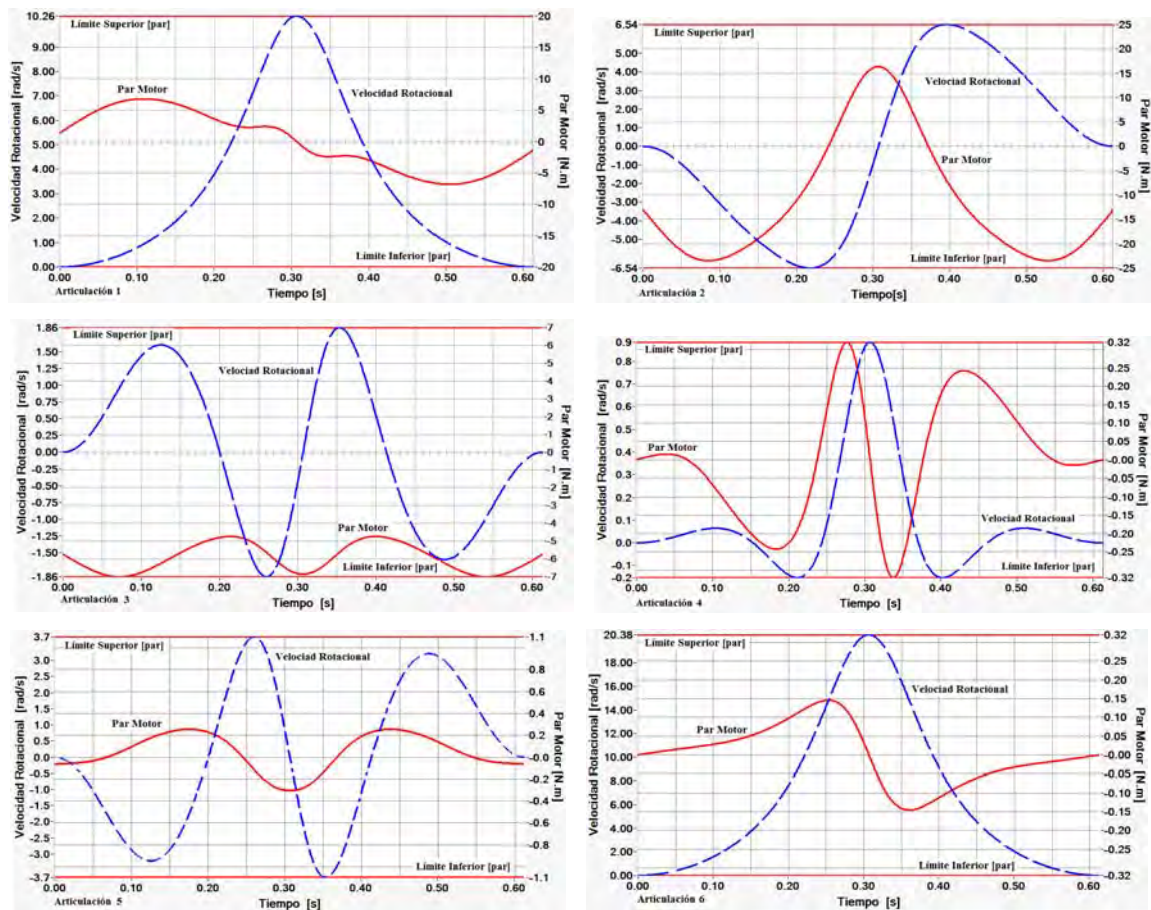
Par real máximo articulación 3 con tiempo de trayectoria óptima $t_f = 0.579970 \text{ s}$	Par real máximo articulación 3 con tiempo de trayectoria una milésima arriba del tiempo óptimo	Par real máximo articulación 3 con tiempo de trayectoria una milésima debajo del tiempo óptimo
19.997511 N-m -19.997508 N-m	19.931418 N-m -19.931362 N-m	20.063055 N-m -20.062159 N-m

### 6.3.3 Ruta 3

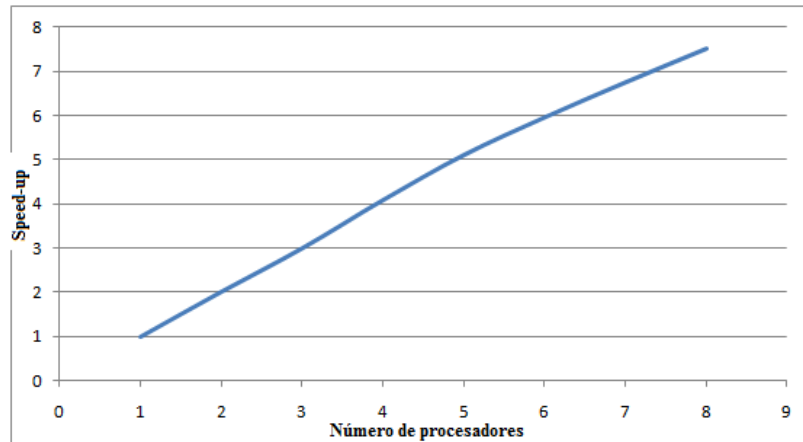
Para la ruta tres su trayectoria óptima se cumple con un *indice\_m* igual a 1.96816 y un tiempo de traslado mínimo igual a 0.613566 s, datos que se visualizan en la Tabla 6.12. Esta trayectoria presenta características especiales en los límites de par, como se puede ver en la Figura 6.10. Los límites se cumplen en dos articulaciones a la vez, las articulaciones tres y cuatro. En la articulación tres, en tres puntos intermedios y en la articulación cuatro

en dos puntos intermedios. Al hacer las mismas pruebas que en las rutas anteriores para la validación de la trayectoria óptima, se observa que estas cumplen las restricciones de par y velocidad. En los cambios de velocidad y par se continúa observando que son con la suavidad que proporciona el polinomio de octavo grado.

El espacio de búsqueda es semejante al de la ruta 1 y 2. En la Figura 6.10a se observa el comportamiento del tiempo de cómputo *Speed-up*, que como se observó en los dos casos de prueba anteriores se tiene una reducción en el tiempo de cómputo de los algoritmos presentados.



**Figura 6.10** Comportamiento del par y velocidad de las articulaciones del robot manipulador, de la trayectoria óptima de la ruta 3.



**Figura 6.10 b** *Speed-up* del algoritmo exhaustivo paralelo correspondiente a la trayectoria óptima de la ruta 2

**Tabla 6.12** Trayectoria solución con tiempo mínimo de traslado de la ruta 3

	Tiempo de Cómputo (s)	<i>indice_m</i>	Tiempo de traslado $t_f$ mínimo	<i>Speed- up</i>
Algoritmo genético	961.665	1.96816	0.613566	1
Algoritmo genético paralelo	128.086	1.96816	0.613566	7.512

#### 6.4 Conclusiones del Capítulo

Al hacer las pruebas de funcionalidad del algoritmo de Planeación de Trayectorias con Optimización del Gasto de Energía, así como del algoritmo de Planeación de Trayectorias con Optimización del Tiempo de Traslado. Se puede apreciar perfectamente el efecto que introduce el polinomio de octavo grado a los cambios en la velocidad y el par. Estos cambios son con una transición mucho muy suave. No se nota para nada cambios lineales.

Se puede observar que al arrancar y parar, lo hace de una forma paulatina, acelerando o desacelerando poco a poco, respectivamente.

Se comprueba la funcionalidad del algoritmo genético aplicado en el algoritmo de Planeación de Trayectorias con Optimización del Gasto de Energía, al hacer un comparativo con un algoritmo de búsqueda exhaustiva con incrementos de una millonésima.

Al ser similares los algoritmos de Planeación de Trayectorias con Optimización del Gasto de Energía, así como el algoritmo de Planeación de Trayectorias con Optimización del Tiempo de Traslado, su diferencia está en la función de optimización. Por lo tanto se comprueba la funcionalidad del algoritmo genético aplicado en el algoritmo de Planeación de Trayectorias con Optimización del Tiempo de traslado. Además de que las pruebas realizadas a este algoritmo, cumplen satisfactoriamente con los cuestionamientos teóricos.



# CAPÍTULO 7

## CONCLUSIONES Y TRABAJOS FUTUROS

### 7.1 Conclusiones

En base a la experiencia recabada con la realización de este trabajo de tesis sobre el desarrollo de una plataforma de software para la planeación de trayectorias óptimas para robots manipuladores, se definen las siguientes conclusiones generales:

1. Se ha presentado un algoritmo de planificación de trayectorias óptimas. El algoritmo se basa en funciones polinomiales de octavo grado para realizar la representación paramétrica de la trayectoria deseada, un algoritmo genético como método de optimización, así como técnicas de procesamiento en paralelo.
2. La función polinomial de octavo grado incluye un parámetro que define la forma de onda y la máxima velocidad, en las articulaciones del robot manipulador. El uso de este parámetro, permite la generación de un gran número de trayectorias suaves para una ruta dada, siendo esto muy conveniente para el uso del algoritmo genético.
3. El algoritmo de optimización presentado minimiza el consumo energía mecánica o minimiza el tiempo de traslado para una ruta determinada en el robot manipulador, con restricciones que limitan el par que satisface a los actuadores.
4. Los casos de estudio presentados demuestran la eficiencia de la plataforma de software para la planificación de trayectoria optimas tanto para minimizar el consumo de energía como para minimizar el tiempo de traslado de un robot manipulador sobre una ruta dada.

5. La plataforma de software desarrollada hace uso de una técnica de procesamiento en paralelo basada en multithreading con el objetivo de mejorar el tiempo de ejecución. Las pruebas realizadas demuestran un speed-up de 6, cuando se aplican las técnicas de procesamiento en paralelo.

## **7.2 Trabajos Futuros**

Trabajos de investigación futuros asociados con esta tesis se propone:

1. Como el objetivo de la planeación de trayectorias es generar las entradas de referencia para los sistemas de control de movimiento, se propone implementar un control de posición y velocidad, utilizando los resultados de este trabajo de tesis.
2. Aplicar un esquema de procesamiento híbrido formado por varias computadoras multiprocesadores, con el objeto de aumentar el Speed-up, experimentando sobre un proceso en tiempo real
3. Aplicación de esta técnica de planeación de trayectorias en tecnología de rehabilitación, por ejemplo: en sillas de ruedas y equipos para ejercitar al paciente sin la ayuda de una persona para su rehabilitación de extremidades superiores e inferiores.

## REFERENCIAS

[Alvarado *et al.* 1992]

F. Alvarado *et al.*, “*Parallel Processing in Power Systems Computation*”, IEEE *Transactions on Power Systems*, Vol. 7, No. 2, Mayo 1992, págs. 629-637.

[Amirabdollahian, *et al.* 2002]

Farshid Amirabdollahian, Rui Loureiro, and William Harwin, “*Minimum Jerk Trajectory Control For Rehabilitation And Haptic Application*”. Proceedings of the 2002 IEEE, International Conference on Robotics & Automation, Washinton, DC. May 2002.

[Aström, *et al.* 1989]

K.J. Aström and K.B Wittenmark, “*Adaptive Control*”, Reading, MA: Addison-Wesley, 1989.

[Barrientos *et al* 1998]

Barrientos Antonio, Peñín Luís Felipe, Balaguer Carlos, Aracil Rafael. “*Fundamentos de Robótica*”, McGraw-Hill, 1997.

[Chapra, 2009]

Steven C. Chapra. “*Applied Numerical Methods with Matlab, for Engineers and Scientists*”. Amazon, 2009

[Craig, 1986]

Craig John J. “*Introduction Robotics Mechanics & Control*”, Addison-Wesley Publishing Company, 1986.

[Foster, 1994]

I. Foster, “*Designing and Building Parallel Programs*”, Addison Wesley 1994.

[FU, *et al.* 1987]

K. S. FU, R. C. González, C. S. G. Lee. “*Robótica, Control, Detección, Visión e Inteligencia*”, McGraw-Hill, 1987.

[Gasparetto y Zanutto, 2007]

A.Gasparetto y V. Zanutto, “*A new method for smooth trajectory planning of robot manipulators*” *Mechanism and Machine Theory*, Volume 42, Issue 4, April 2007, Pages 455-471 Published journal article available from ScienceDirect. Elsevier Ltd All rights reserved.

[Gasparetto y Zanutto, 2010]

A.Gasparetto y V. Zanutto, “*Optimal trajectory planning for industrial robots*” *Advances in Engineering Software*, Volume 41, Issue 4, April 2010, Pages 548-556. Published journal article available from ScienceDirect. Elsevier Ltd All rights reserved.

[García, *et al.* 2007]

E. García, M.A. Jiménez, P. González de Santos and M. Armada, “*The evolution of Robotics Research*”, *IEEE Robotics Automation magazine* marzo 2007 vol. 14 No. 1. pp. 90-101.

[Hadjikyriacous, *et al.*, 2009]

Hadjikyriacous, E. Samaras, N. Margaritis, K. “*An Experimental Evaluation of a parallelGenetic Algorithm Using MPI*”. *Informatics, 2009.PCI '09. 13<sup>th</sup> Panhellenic Conference on.* Page 75-79, 2009.

[Jin, 1994]

L. Jin., “*Parallel Processing: Exploring the Architectures and Algorithms Close Relationship*”, *IEEE POTENTIALS*, Diciembre 94-Enero 95, págs. 17-20.

[Juárez, 2002]

Ignacio Juárez Campos. “*Sobre las Regiones Geométricas Articulares y su Desempeño de un Manipulador Mediante Estrategias Evolutivas*”. Tesis para obtener el grado de Doctor en Ingeniería. Universidad Nacional Autónoma de México, agosto 2004.

[J. Somló, et al. 1997]

J. Somló, B. Lantos, P. T. Cat, “*Advanced Robot Control*”, Akadémiai Kiadó, Budapest, 1997

[Kang, et al. 1985]

Kang G. Shin and Neil D. McKay. “*Minimum-Time Control of robotic Manipulator with Geometric path Constraints*”. IEEE Transaction on Automatic Control, vol. AC-30, N0 6, junio/1985.

[Kim, et al.. 1997]

Kee-Whan Kim, Hyun-Sink and Young-Kiu Choi. “*Optimization of Cubic Polynomial Joint Trajectories and Sliding Mode Controllers for Robots Using Evolution Strategy*”. IEEE International conference On Industrial Electronic Control and Instrumentation, 1997. IECON 97 23<sup>rd</sup>.

[Kishimoto, et al. 2004]

Tatsuya Kishimoto, Yutaka Fujimoto, “*Numerical Stable Inverse Kinematics Calculation of Robot Manipulators even in Singular Configuration*” The 30th Annual Conference of the IEEE Industrial Electronic Society, November 2-6, 2004, Busan Korea.

[Lemaitre y Thomas 1996]

Lemaitre C. y Thomas B., “*Two Applications of Parallel Processing in Power System Computation*”, IEEE Transactions on Power Systems, Vol. 11, No. 1, Febrero 1996, págs. 246-253.

[Liu, *et al.* 2007]

Yi Liu, Xin Zhang, He Li, Depei Qian, “*Allocasting Tasks in Multi-core Processor based parallels Systems*”, Network and Parallel Computing Workshops, 2007. NPC Workshops. IFIP International Conference on. IEECONFERENCE, 2007.

[Luo, *et al.* 2004]

Xiong Luo, Xiaoping Fan, Heng Zhang and Tefang Chen. “*Integrated Optimization of Trajectory Planning for Robot Manipulators Based on Intensified Evolutionary Programming*”. Proceeding of the 2004 IEEE. International Conference on Robotics and Biomimetics, August 2004. Shenyang, China.

[Ma, y Watanabe, 2000]

Shugen Ma and Mitsuru Watanabe. “*Minimum Time Path-tracking Control of Redundant Manipulator*”. Proceedings of the 2000 IEEE/RsJ International Conference on Intelligent Robots and Systems.

[Michalski, *et al.* 1983]

R.S. Michalski, J.C. Carbonell, and T.M. Mitchell, “*Machine Learning*”, Palo Alto, CA: Tioga, 1983.

[Mariños *et al.* 1994]

Mariños Z.A., Pereira J.L.R., Carneiro Jr., “*Fast Harmonic Power Flow Calculation Using Parallel Processing*”, IEE Proc.-Gener. Transm. Distrib., Vol 141, No. 1. Enero 1994, págs. 27-32.

[Monteiro, *et al.* 1999]

D. C. Monteiro and M. K. Madrid. “*Planning of Robot Trajectories with Genetic Algorithms*”. Robot Motion and Control, 1999. Proceeding of the First Worsohp. 28 Junio 1999.

[Niemeyer, *et al.* 2004]

G. Niemeyer and J.J.E. Slotine, “*Telemanipulation with time delays*”, Int. J. Robot, Res. Vol. 23, No. 9, pp 873-890, 2004.

[Olomski, 1989]

J. Olomski. “*Trajectory Planning, Optimization and Control for Industrial Robot*”. Proceedings ICCON 1989. IEEE International Conference On Control and Applications.

[Oyama *et al.* 1990]

Oyama T., Kitahara T., Serizawa Y., “*Parallel Processing for Power System Analysis Using Band Matrix*”, IEEE Transactions on Power Systems, Vol. 5, No. 3, Agosto 1990, págs. 1010-1016.

[Pérez, 2004]

Waldemar Pérez Bailón. “*Análisis y Simulación de un Robot Manipulador de Seis Grados de Libertad*”. Tesis para obtener el grado de Maestro en Ingeniería Eléctrica, División de estudios de posgrado, Facultad de Ingeniería Eléctrica, Universidad Michoacana de San Nicolás de Hidalgo. 2005.

[Rosen et al. 2006]

Jacob Rosen & Blake Hannaford, “*Doc a Distance*”, IEEE SPECTRUM, Octubre 2006, vol. 43. No 10. Pp 34-39.

[Russell, *et al.* 2003]

S. Russell and P. Norvig, “*Artificial Intelligence*”, 2<sup>nd</sup> ed. Upper Saddle River, NJ: Prentice Hall, 2003.

[Sciavico, *et al.* 1976]

Lorenzo Sciavico y Bruno Siciliano, “*Modeling and Control of Robot Manipulator*” McGraw-Hill, 1976. University of Naples, Naples, Italy

[Siciliano, *et al.* 1999]

B. Siciliano and L. Villani, “*Robot Force Control*”, Norwell, MA: Kluwer, 1999.

[Slotine, *et al.* 1991]

J.J.E. Slotine and W. Li, “*Applied Nonlinear Control*”, Upper Saddle River, NJ: Prentice-Hall, 1991.

[Soto, *et al.* 2006]

J. A. Soto Cajiga, J. E. Vargas Soto, J. C. Pedraza Ortega. “*Generación de Trayectorias para un Robot Manipulador Utilizando Procesamiento de Imágenes y Splines*”. Segundo Congreso Internacional de Ingeniería. Universidad Autónoma de Querétaro, marzo 2006.

[Stavrakakis *et al.* 1990]

Stavrakakis G.S., Lefas C., Pouliezios A., “*Parallel Processing Computer Implementation of a Real Time DC Motor Drive Fault Detection Algorithm*”, IEE Proceedings, Vol. 137, No. 5. Septiembre 1990. págs. 309-313.

[Tagawa, *et al.* 1993]

Kiyoharu Tagawa, Yoshihide Kanki, Yuzo Ohta and Hiromasa Haneda, “*A Parallel Scheme for Dynamic Control of Robotic Manipulators*”, International Conference



on Industrial Electronics, Control, and Instrumentation, 1993. Proceedings of the IECON '93. **IEEE CONFERENCES**

[Yussof, *et al.* 2009]

Salman Yussof, Rina Azlin Razali, Ong Hang See, “*A Parallel Genetic Algorithm for Path Routing Problem*”, International Conference on Future Computer and Communication, IEEE CONFERENCE, 2009.

[Werler y Glavitsch 1993]

Werler K. y Glavitsch H., “*Computation of Transients by Parallel Processing*”, *IEEE Transactions on Power Delivery*, Vol. 8, No. 3, Julio 1993.