

01173

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

DIVISIÓN DE ESTUDIOS DE POSGRADO

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE MECÁNICA

TESIS:

"CONTROL DE ROBOTS MANIPULADORES"

PARA OBTENER EL GRADO DE MAESTRO EN INGENIERÍA

PRESENTA:

JUAN ALEJANDRO FLORES CAMPOS

DIRECTOR DE TESIS:

DR. LUIS REYES ÁVILA

COASESOR DE TESIS:

M. en I. EUSEBIO JIMÉNEZ LÓPEZ

267153

CD. UNIVERSITARIA, SEPTIEMBRE DE 1998.

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



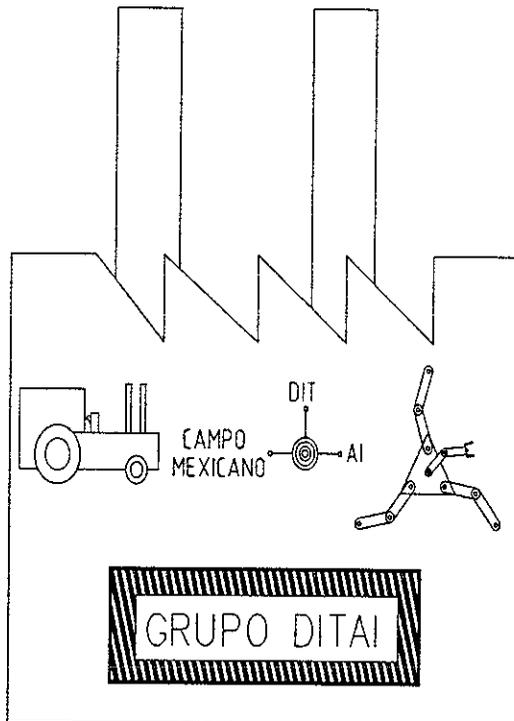
UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

EL GRUPO DITAI



AGRADECIMIENTOS

A todos mis profesores que fueron ejes en mi formación académica:

Dr. Luis Reyes Ávila.
Dr. Luis Ferrer Argote.
M.I. Mario Márquez.
M.I. Mauricio Mendez.
M.I. Irma Salinas.
M.I. Francisco Cuenca.

Quienes compartieron conmigo sus conocimientos, su entusiasmo y dedicación,
Gracias...

Agradezco también a Eusebio Jimenez por su valiosa participación en el desarrollo de este trabajo.

A mis compañeros a Alfredo Rubio, al carnal, y a todos mis compañeros que estuvieron conmigo.

DEDICATORIA.

A mamá

*quién me enseñó a andar por
el camino largo,
por el terreno brusco
por la cuesta ardua y el paisaje mustio.*

ÍNDICE

Tema	Nombre	Pág
	Introducción.....	1
CAPÍTULO I.		
1.1	El Marco Teórico.....	2
1.2	Definiciones Generales.....	11
1.3	Espacios Vectoriales y Transformaciones Lineales..	16
1.4	El Álgebra de Cuaterniones.....	21
CAPÍTULO II.		
2.1	Modelación Cinemática de un Manipulador de 2 GDL.....	29
CAPÍTULO III.		
3.1	Diseño Conceptual y de Forma de un Manipulador de 2 GDL.....	45
3.2	Diseño de Forma.....	53
CAPÍTULO IV.		
	Control.....	79
4.1	Aspectos Electromecánicos de los Motores PAP.....	80
4.2	Manejadores (Drives) para Motores PAP.....	96
4.3	Creación del Hardware de Control.....	103
4.4	Programa de Control del Manipulador de 2 GDL.....	124
	CONCLUSIONES	133
		134
	APÉNDICE A.....	
	APÉNDICE B.....	136
	APÉNDICE C.....	139
	BIBLIOGRAFÍA.....	149

INTRODUCCIÓN

El presente trabajo forma parte de la investigación en el área de Robótica en el departamento de Mecánica de la DEPMI UNAM (División de Estudios de Posgrado de la Facultad de Ingeniería). Donde **el principal objetivo** es modelar una cadena cinemática de dos grados de libertad desde la teoría cinemática hasta la comprobación experimental, utilizando para esto, marcos teóricos rigurosos en especial el álgebra de Cuaterniones, sin considerar la etapa del aspecto dinámico. Se construirá un prototipo que simule la cadena cinemática modelada de tal forma que se proponga y desarrolle la etapa de control (cinemático) con sus respectivas interfases.

En la etapa de control sólo se simularán los parámetros obtenidos de la cinemática inversa y se crearán las interfases de control electrónicas y programas necesarios para tal fin.

El capítulo I contiene las definiciones generales y el marco teórico en el cual nos apoyaremos.

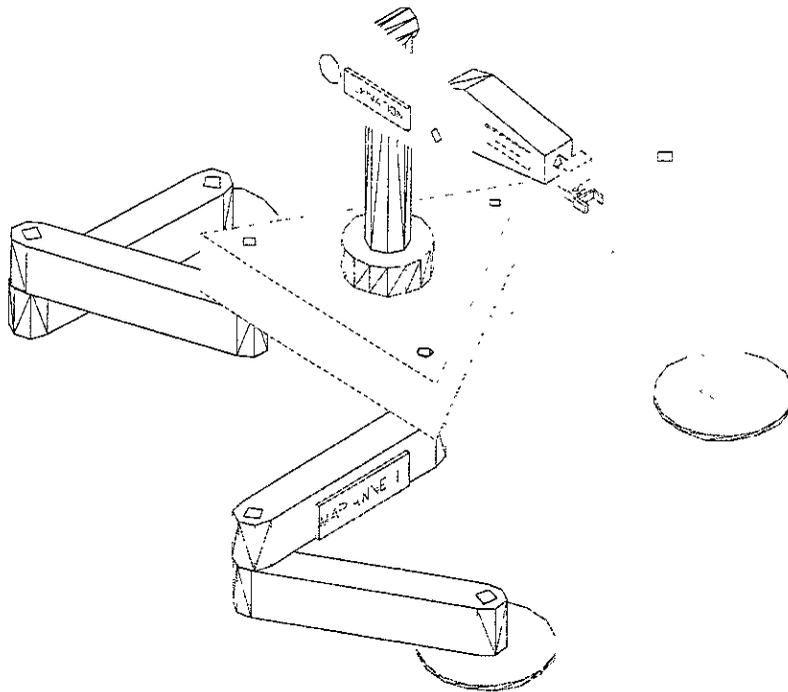
En el capítulo II desarrollaremos el modelo cinemático de un prototipo de robot de dos grados de libertad.

El capítulo III enmarca el diseño de el prototipo a controlar (Robot-Manipulador de dos grados de libertad).

En el capítulo IV se desarrolla la etapa de control utilizando para los movimientos del Robot-Manipulador motores paso a paso y se construirán las interfases de control.

CAPÍTULO I

1.1 EL MARCO TEÓRICO



ROBOT COMPUESTO MARIANE III

I.1 DEFINICIONES GENERALES

INTRODUCCIÓN

En el presente trabajo se presenta la aplicación del álgebra de Cuaterniones para generar las ecuaciones que gobiernan la cinemática de cuerpos rígidos como cuerpos acoplados comúnmente llamados robots, se plantea identificar las secuencias de movimientos de los manipuladores con el propósito de establecer las bases para simular un robot de dos grados de libertad. el principal objetivo es mostrar la validez del álgebra de Cuaterniones así como sus aplicaciones a los sistemas mecánicos que se encuentran en las máquinas en general. Para iniciar este trabajo es necesario realizar una breve introducción al campo de la robótica y realizar las definiciones apropiadas en cuanto a los diversos tipos de robots industriales que se utilizan en la actualidad. El siguiente resumen de las definiciones fué consultado en [1].

Definición I.1.1 La palabra "robot" proviene de la palabra Checa "ROBOTA" que significa trabajo. Para el "Robot Institute of America", un robot es un manipulador reprogramable multifuncional diseñado para mover materiales, piezas y dispositivos especializados a través de movimientos variables programados para la realización de una diversidad de tareas.

Los robots son importantes porque pueden realizar tareas manipuladoras sencillas tanto o mas eficientemente que el hombre y son capaces de realizarlas en forma repetitiva y de manera incansable, pero lo que verdaderamente los hace útiles e importantes es el hecho de que pueden trabajar en medios ambientes en los que un ser humano puede poner en riesgo su integridad física y aún su vida, tal es el caso por ejemplo: en operaciones de manejo de materiales radioactivos; tareas con sustancias químicas; entre otras. Actualmente los robots son utilizados también en la industria espacial, en simuladores de vuelo, en la industria farmacéutica y en general en la industria de la transformación.

Definición I.1.2 Los robots son dispositivos mecánicos y electrónicos de alta precisión que se dañan fácilmente si no se manejan adecuadamente, además de ser herramientas de trabajo muy caras, por estos motivos es necesario proporcionar a las personas que van a trabajar con ellos un medio ambiente confiable que les permita familiarizarse con éstos sin arriesgar la integridad física del robot y menos aún de las personas. De esta manera es necesario capacitar al personal en simuladores computacionales que tienen la ventaja de ser baratos y seguros, ya que lo peor que le puede suceder a un simulador es dejar de interactuar con el usuario, pero este problema se puede resolver fácilmente reiniciando el simulador.

Las metodologías matemáticas para interpretar los movimientos particulares y globales de los robots tuvieron sus principios en la década de los años 40, por esas fechas apareció el primer robot industrial fabricado por la empresa americana UNIMATION, sin embargo, desde que los investigadores Denavit- Hartenberg publicaron los llamados parámetros de Denavit- Hartenberg los cuales forman los componentes de una matriz de rotación y traslación, no se han desarrollado metodologías matemáticas que nos permitan interpretar los movimientos geométricos de los sistemas mecánicos dadas las exigencias de un desarrollo tecnológico que necesita de parámetros óptimos en la creación de prototipos más sofisticados. A la fecha se sigue empleando esta metodología, otra disciplina que interactúa con la robótica es la Inteligencia Artificial, se necesita en esta área seleccionar las secuencias de movimientos más elementales de los sistemas mecánicos para obtener criterios más acertados en la solución de un problema específico y así evitar programas muy complejos y extensos que en lugar de ser óptimos pierden la esencia física del problema.

Con el objetivo de desarrollar metodologías matemáticas más eficientes, se realizará un análisis riguroso y sistemático de los movimientos geométricos de los sistemas mecánicos, con el fin de evitar desarrollar prototipos con exagerada arquitectura mecánica y con un control muy caro, esto es con el fin de que los parámetros de respuesta sean lo suficientemente óptimos, para poder competir en el mercado, y no incrementar el costo de los robots haciéndolos inaccesibles para las empresas.

Un robot posee dos características importantes:

- 1) Versatilidad, la cual puede definirse como la potencialidad estructural de ejecutar tareas diversas y/o ejecutar una misma tarea de forma diversificada.
- 2) La adaptabilidad al entorno. Esto es, el robot debe, por sí solo, alcanzar su objetivo (la ejecución de una tarea) a pesar de las perturbaciones imprevistas (pero limitadas) de su entorno, a lo largo de la ejecución de una tarea.

Un robot manipulador, en principio, es un dispositivo de propósito general. En la práctica los manipuladores son usualmente diseñados pensando en que cubran al menos una clase amplia de aplicaciones, como podría ser, soldar; manejar materiales; ensamblar; entre otras tareas. Estas aplicaciones en gran parte dictan la elección de varios parámetros de diseño del manipulador, incluyendo su estructura cinemática. Por ejemplo, el ensamblado de tarjetas de circuitos es realizada por un manipulador tipo SCARA¹.

Definición I.1.3 Los manipuladores pueden ser clasificados por muchos criterios, tales como su geometría , estructura cinemática, el tipo de aplicación para la cual fueron diseñados, la manera en que son controlados.

Muchos manipuladores industriales actuales tienen 6 ó menos grados de libertad, son usualmente clasificados cinemáticamente sobre la base del primer brazo o las primeras 3 juntas, describiendo por separado el órgano terminal o muñeca. La mayoría de estos manipuladores se clasifican en uno de los siguientes tipos de geometría.

- 1 Articulados.
- 2 Esféricos.
- 3 Tipo SCARA.
- 4 Cilíndricos.
- 5 Cartesianos.

¹ Selective Compliance Assembly Robot Arm

Algunos de los robots industriales más usados son los tipo PUMA² y ABB System de seis grados de libertad para transporte de piezas y ensamblado de partes, el tipo SCARA de cuatro grados de libertad para operaciones de inspección, robot tipo MotoMan de cinco grados de libertad para operaciones de soldadura y el tipo AS-RS para trabajos de almacenaje de dos grados de libertad. Existen en la actualidad cadenas cinemáticas especiales llamadas plataformas o robots paralelos que tiene ventajas sobre los manipuladores convencionales dada su alta precisión, sin embargo, su uso no esta bien difundido por lo complicado de sus modelos matemáticos y altas redundancias, por lo que se encuentran en laboratorios de las universidades y su uso esta restringido en la industria manufacturera, sin embargo, en la industria aeronáutica y espacial son usados para construir simuladores complejos. De esta manera es necesario definir los robot paralelos de una forma correcta y estudiarlos para una aplicación mas general. Con el fin de un buen desarrollo de este trabajo se definirán los siguientes conceptos:

Definición I.1.4 Un simulador de robot es un sistema que imita el funcionamiento o comportamiento del robot. Tiene el objetivo de enseñar a los futuros operadores las características y restricciones de robot que se va a manejar.

Definición I.1.5 Las cadenas cinemáticas están constituidas por elementos llamados eslabones, el acoplamiento entre dos eslabones se realiza por medio de pares, los cuales pueden ser de dos clases: pares inferiores y pares superiores. Un par inferior existe cuando un elemento es acoplado a otro por medio de una acción envolvente y el contacto se realiza entre superficies. Si el contacto tiene lugar a lo largo de una línea o punto (un par de engranes), el acoplamiento se conoce como un par superior. En robótica se utilizan frecuentemente los pares inferiores.

Definición I.1.6 Los pares inferiores pueden clasificarse en seis tipos:

² Programmable Universal Machine for Assembly

- a) Par de rotación (R), el cual permite un giro alrededor de un eje y evita traslaciones. Impone cinco restricciones, tres de traslación y dos de rotación.
- b) Par prismático (P), permite únicamente traslación a lo largo de una dirección. También impone cinco restricciones, evita traslación sobre dos direcciones y rotaciones con respecto a tres ejes.
- c) Par de tornillo (H), permite traslación a lo largo de un eje y rotación alrededor del mismo eje, existiendo una relación entre ellos. Por lo tanto, también impone cinco restricciones.
- d) Par cilíndrico (C), permite dos movimientos independientes, traslación sobre un eje y rotación alrededor del mismo. Impone cuatro restricciones.
- e) Par esférico (S), permite rotación alrededor de tres ejes no coplanares e impide movimientos de traslación. Impone tres restricciones.
- f) Par plano (E), permite traslación a lo largo de dos direcciones independientes y rotación alrededor de un eje perpendicular al plano de esas direcciones. Impone tres restricciones.

Definición I.1.7 El grado de libertad de un sistema mecánico se define como un número entero correspondiente al número mínimo de coordenadas requeridas para especificar una configuración geométrica de un sistema dado.

Para ubicar un cuerpo rígido en el espacio se necesitan seis grados de libertad, es decir, una traslación a lo largo de cada una de las tres direcciones del espacio y una rotación alrededor de un eje tridimensional.

Definición I.1.8 Un multicuerpo es un conjunto de cuerpos rígidos en el cual los cuerpos pueden tener movimiento relativo unos con respecto a otros. La conexión de eslabones mediante pares cinemáticos constituye una cadena cinemática.

Definición I.1.9 Los sistemas mecánicos se pueden clasificar de tres formas:

- 1) De cadena abierta. Es un conjunto de eslabones rígidos unidos por pares cinemáticos, donde uno de los extremos es fijado a una base, y el otro está libre, el cual es equipado con alguna herramienta que realiza una tarea específica, se les conoce comúnmente como manipuladores.
- 2) De cadena cerrada. Los eslabones (≥ 3) forman lazos cerrados y el órgano terminal está conectado a una base fija, también se les llama mecanismos.
- 3) Plataformas (Robots paralelos). Son mecanismos de cadena cinemática cerrada, constituido por un órgano terminal de n grados de libertad y de una base fija unida entre sí por cadenas cinemáticas independientes, cada una de ellas cuenta a lo sumo de dos segmentos articulados, la articulación entre estos dos segmentos deberá tener un solo grado de libertad. El movimiento de este mecanismo se efectúa por n actuadores simples, uno por cada cadena.

Con las definiciones anteriores se puede clasificar a los robots por medio de sus pares inferiores, por ejemplo:

El robot PUMA posee seis articulaciones del tipo revolución entonces es del tipo RRRRRR, el tipo SCARA se clasifica como RRRP (P es un par prismático), el MotoMan del tipo RRRRR, en tanto el AR-SR del tipo PP. Una plataforma de tres grados de libertad en el plano puede tener una configuración tipo RRR si posee articulaciones de revolución, (ver fig 1.1).

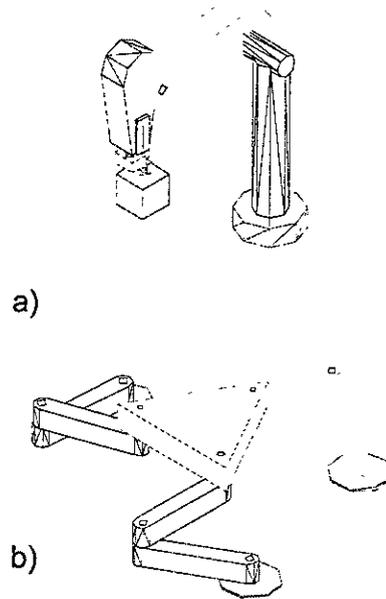


Figura 1.1 a) Robot PUMA, b) Robot paralelo tipo RRR.

Definición 1.1.10 Los robots se clasifican también en clases, por ejemplo los primeros robots que se construyeron son de la

- a) Clase I, sus movimientos se restringen solo a moverse en trayectorias puntuales, específicas y su interacción con el personal es directa, y no puede trabajar en medios ambientes desconocidos, es decir, no reconocen su entorno.
- b) Clase II, se refiere a una generación de robots dotados de sensores capaces de reconocer el entorno y pueden evitar obstáculos, sus lenguajes de programación son más sofisticados y pueden interactuar con medios ambientes hostiles.
- c) Clase III, son robots dotados de inteligencia artificial y con sistemas de visión capaces de reconocer objetos variados y tomar decisiones por ellos mismos, su comunicación es a través de lenguaje de hombre o por sistemas de realidad virtual.

De esta manera se a clasificado a los robots tanto por sus pares cinemáticos como por las clases de generaciones a través del tiempo. Para el propósito de este trabajo sólo se trabajará con cadenas cinemáticas que poseen pares de revolución y se realizará una simulación computacional de un robot de la clase II, es decir, de un robot tipo RR.

En este contexto utilizaremos el álgebra de Cuaterniones como una alternativa para describir los movimientos geométricos de las cadenas cinemáticas, se observará como los cambios de base representados a través de operaciones binarias son mas eficientes que las representadas con matrices trigonométricas. Los parámetros de Denavit - Hartenberg no son necesarios utilizarlos, además, esta metodología no permite su uso, modificando con esto las estructuras matemáticas tradicionales que se emplean comúnmente en la modelación cinemática de robots.

I.2 MARCO TEÓRICO

I.2.1 TEORÍA DE CONJUNTOS[2]

Un robot podemos analizarlo como un conjunto de cuerpos rígidos, en tanto que se compone de cables, tornillos cadenas, eslabones, juntas, etc. El robot lo podemos particionar en conjuntos y por ende en clases, al caracterizar sus elementos bajo alguna relación. Es en este sentido que conviene utilizar la teoría de conjuntos, mediante la cual se analizarán los elementos del robot.

El concepto de conjunto (de una colección de elementos) desempeña un papel de extraordinaria importancia en las ciencias modernas no sólo porque la propia teoría de conjuntos ha pasado a ser en la actualidad una disciplina sumamente vasta, sino, principalmente, en virtud de la influencia que la teoría de conjuntos ha ejercido y ejerce sobre las matemáticas que a su vez son el fundamento de las ciencias aplicadas, entre ellas la robótica.

En este escrito sólo trataremos los conceptos más importantes sobre conjuntos relacionados con la robótica. Designaremos los conjuntos con letras mayúsculas A, B, C, etc. La afirmación de que "un elemento a pertenece al conjunto A" se denotará por $a \in A$; $a \notin A$ significará que "el elemento a no pertenece al conjunto A". Si todos los elementos que componen al conjunto A pertenece también al conjunto B (con la particularidad de que $A = B$ no está excluido), se dirá que "A es subconjunto de B" lo que se escribe $A \subset B$.

Para denotar las operaciones sobre conjuntos emplearemos los símbolos comúnmente usados en los libros de matemáticas. Esos símbolos son: \cup , \cap , \setminus , Δ , y denotan la unión, intersección, diferencia y diferencia simétrica de conjuntos, respectivamente.

Definición I.2.1.1 Dados dos conjuntos M y N de naturaleza arbitraria, se dice que está definida sobre M una función f (que se llamará en este escrito función de

caracterización) con valores en N , si a cada elemento $x \in M$ se le pone en correspondencia un elemento y solo uno y de N .

Definición 1.2.1.2 Sea $a \in M$, el elemento $b = f(a)$ de N que le corresponde se denomina imagen del elemento a (para la aplicación f). La colección de elementos de M que tienen por imagen el elemento $b \in N$ se denomina imagen recíproca del elemento b y se denota mediante $f^{-1}(b)$.

Sea f la función de caracterización $f: I \rightarrow P(\Omega)$ con las siguientes propiedades:

- 1 La función es uno a uno;
- 2 La $\text{Im}(f) = C \subset P(\Omega)$ [1.1]
- 3 Para toda $i \in I$, $f(i) = C_i$ (I es el conjunto de índices)

Bajo esas indicaciones se tiene que los elementos de un robot se pueden caracterizar mediante:

$$f(i) = C_i \Leftrightarrow \text{Im}(f) = \{ C_i \mid i \in I \} = C \subset P(\Omega). \quad [1.2]$$

Donde C_i son los subconjuntos en los que se puede subdividir a un robot. Luego todo conjunto C se puede escribir como :

$$C = \{ C_i \}_{i \in I} = C_1 + C_2 = \text{Im}(f) \quad [1.3]$$

es decir los elementos de los robots pueden ser clasificados en familias C_i

Definición 1.2.1.3 Sea Ω un conjunto cualesquiera, $C \subset P(\Omega)$ y sea I otro conjunto arbitrario, entonces la familia $C = \{ C_i \}_{i \in I}$ indicada por I es una función

$$f: I \rightarrow P(\Omega) \text{ tal que:}$$

- 1 la función f es uno a uno ; [1.4]

2 La $\text{Im}(f) = C = \{C_i\}_{i \in I}$

Teorema I.2.1.1 Sea $C = \{C_i\}_{i \in I}$ una familia de subconjuntos de un conjunto Ω ; entonces las siguientes proposiciones 1 y 2 son verdaderas:

1 Existe un conjunto, y un sólo conjunto que se denotará

$$\bigcup_{i \in I} X_i \quad [1.5]$$

que tiene las siguientes propiedades:

a) $\bigcup_{i \in I} X_i \subset \Omega$

b) $X_i \subset \bigcup_{i \in I} X_i \quad \forall i \in I$

a) si $Y \subset \Omega$ es tal que $X_i \subset Y \quad \forall i \in I$, entonces $\bigcup_{i \in I} X_i \subset Y$ [1.6]

En otras palabras como la relación de inclusión \subset es una relación de orden en $P(\Omega)$, entonces lo que se está afirmando es que

$$\bigcup_{i \in I} X_i = \sup\{X_i\}_{i \in I} \quad [1.7]$$

o sea, que el conjunto $\bigcup_{i \in I} X_i$, que se llamará la unión de la familia $X = \{X_i\}_{i \in I}$, es la menor de las cotas superiores de la familia C ; es decir su supremo.

2) Existe un conjunto, y un sólo conjunto que se denotará

$$\bigcap_{i \in I} X_i \quad [1.8]$$

y se llamará la intersección de la familia $X = \{X_i\}_{i \in I}$, que se caracteriza por tener las siguientes propiedades:

- a) $\bigcap_{i \in I} X_i \subset \Omega$
- b) $\bigcap_{i \in I} X_i \subset X_i \quad \forall i \in I$ [1.9]
- c) si $Y \subset X_i \quad \forall i \in I$; entonces es tal que $X_i \subset Y \quad \forall i \in I$, entonces $Y \subset \bigcap_{i \in I} X_i$

Esto es, la intersección de la familia es el ínfimo de ella bajo la relación de orden inducida por la relación de inclusión \subset en $P(\Omega)$:

$$\bigcap_{i \in I} X_i = \inf\{X_i\}_{i \in I} \quad [1.10]$$

ya que $\bigcap_{i \in I} X_i$ es una cota inferior de la familia C y es la mayor de las cotas inferiores de la familia.

Definición I.2.1.4 Sea $C = \{C_i\}_{i \in I}$ una familia de subconjuntos de C_i de un conjunto Ω , indicada por el conjunto I . Se define entonces

- 1 $\bigcup_{i \in I} C_i = \{x \in \Omega \mid \text{existe un } i \in I \text{ tal que } x \in C_i\}$
- 2 $\bigcap_{i \in I} C_i = \{x \in \Omega \mid x \in C_i \quad \forall i \in I\}$ [1.11]

Cuando el conjunto de índices I sea el conjunto de los números naturales N , esto es,

cuando $C = \{C_i\}_{i \in I} = \{C_n\}_{n=1}^{\infty}$ sea una sucesión con

$$I = N = \{1, 2, 3, \dots, n, \dots, \dots\}$$

la unión y la intersección de esta sucesión se denotará por

1 $\bigcup_{i \in I} C_i = \bigcup_{n=1}^{\infty} C_n$ [1.12]

$$2 \quad \bigcap_{i \in I} C_i = \bigcap_{n=1}^{\infty} C_n$$

Sea $I = \{1, 2\}$ y que $C = \{C_i\}_{i \in I} = \{C_1, C_2\}$, entonces

$$\begin{aligned} \bigcup_{i \in I} C_i &= \{x \mid \text{existe } i \in C_i \text{ tal que } x \in C_i\} = \\ &= \{x \mid x \in C_1 \text{ o } x \in C_2\} = C_1 \cup C_2 \end{aligned} \quad [1.13]$$

Así mismo,

$$\begin{aligned} \bigcap_{i \in I} C_i &= \{x \mid x \in C_i, \forall i \in I\} = \\ &= \{x \mid x \in C_1 \text{ y } x \in C_2\} = C_1 \cap C_2 \end{aligned} \quad [1.14]$$

Teorema I.2.1.2 Sea $\{A_i\}_{i \in I}$ una familia de conjuntos; entonces:

$$\begin{aligned} 1 \quad (\bigcup_{i \in I} A_i)^c &= \bigcap_{i \in I} A_i^c \\ 2 \quad (\bigcap_{i \in I} A_i)^c &= \bigcup_{i \in I} A_i^c \end{aligned} \quad [1.15]$$

Teorema I.2.1.3 Sea $\{A_i\}_{i \in I}$ una familia de conjuntos, y supóngase que algunos de los conjuntos es vacío, esto es:

$$\exists j \in I \text{ tal que } A_j = \emptyset$$

entonces:

$$\bigcap_{i \in I} A_i = \emptyset \quad [1.16]$$

Definición I.2.1.5 Sea $I = \{1, 2, 3\}$ y que $C = \{C_i\}_{i \in I} = \{C_1, C_2, C_3\}$, entonces La diferencia entre los conjuntos C_1, C_2 esta dada por el conjunto C_3

$$C_3 = C_1 - C_2$$

El marco teórico aquí definido es de fundamental importancia cuando se caracteriza el área de trabajo de los robots.

1.3 ESPACIOS VECTORIALES Y TRANSFORMACIONES LINEALES[4]

Se presenta en esta parte las bases conceptuales de los espacios vectoriales y las transformaciones lineales.

1.3.1 Estructuras Algebraicas

Definición I.3.1.1 Sea X un conjunto y $*$: $X \times X \rightarrow X$ una operación binaria. La pareja $(X, *)$ es un grupo si:

- 1 Ley asociativa: $x *(y *z) = (x *y) *z \quad \forall x,y,z \in X,$
- 2 Existencia de la identidad: $\forall x \in X,$ existe $l \in X,$ tal que:

$$x * l = x,$$

- 3 $\forall x \in X$ existe un $x^{-1} \in X$ tal que: [1.17]

$$x * x^{-1} = l$$

Definición I.3.1.2 El grupo $(X, *)$ es un grupo conmutativo o abeliano si:

$$x * y = y * x, \quad \forall x, y \in X, \quad [1.18]$$

Definición I.3.1.3 Sea X un conjunto y \oplus : $X \times X \rightarrow X,$ $*$: $X \times X \rightarrow X$ dos operaciones binarias. La terna $(X, \oplus, *)$ es un campo si:

- i) La pareja (X, \oplus) es un grupo conmutativo,
- ii) La pareja $(X, *)$ es un grupo conmutativo, excepto por la existencia del inverso con respecto a la operación $*$: $X \times X \rightarrow X$ del elemento nulo de la operación \oplus : $X \times X \rightarrow X,$ además, las siguientes propiedades distributivas deben ser satisfechas:

$$\begin{aligned}x * (y \oplus z) &= (x * y) \oplus (x * z), \\(x \oplus y) * z &= x * z + y * z.\end{aligned}\tag{1.19}$$

1.3.2 Espacios Vectoriales

Definición 1.3.2.1 Sea (X, \oplus) un grupo conmutativo y sea $(\kappa, \oplus, *)$ un campo. la pareja (X, \oplus) es un espacio vectorial si existe una operación

$\bullet : \kappa \times X \rightarrow X$, llamada multiplicación escalar tal que:

$$\begin{aligned}1) \quad x \oplus (-1 \bullet x) &\equiv 0, & \forall x, 0 \in X, \\2) \quad \alpha \bullet (x \oplus y) &\equiv \alpha \bullet x \oplus \alpha \bullet y, & \forall x, y \in X, \forall \alpha \in \kappa \\3) \quad \alpha \bullet (x * y) &\equiv (\alpha \bullet x) * y, & \forall x, y \in X, \forall \alpha \in \kappa, \\4) \quad \alpha \bullet (\beta \bullet x) &\equiv (\alpha \bullet \beta) \bullet x, & \forall x \in X, \forall \alpha, \beta \in \kappa, \\5) \quad 1 \bullet x &\equiv x, & \forall x \in X.\end{aligned}\tag{1.20}$$

Donde -1 es el inverso con respecto a $\oplus : X \times X \rightarrow X$ de la identidad con respecto a $*$: $X \times X \rightarrow X$

Definición 1.3.2.2 Se llama base de un espacio n -dimensional X a cualquier conjunto $B = \{e_k\} \subset X$ generador de vectores linealmente independientes.

Definición 1.3.2.3 Si el número de elementos de una base $B = \{e_k\}_{k=1}^m$ de un espacio vectorial tiene un número finito de elementos, entonces se dice que el espacio es de dimensión finita. El número de elementos de la base es llamado la dimensión del espacio vectorial.

1.3.3 Transformaciones Lineales .

Definición 1.3.3.1 Sean (X, \oplus) y (Y, \oplus) dos espacios vectoriales sobre el mismo campo κ .

La transformación $T: X \rightarrow Y$ es lineal si:

$$T(\alpha \cdot x \oplus \beta \cdot y) = \alpha \cdot Tx \oplus \beta \cdot Ty \quad \forall x, y \in X, \forall \alpha, \beta \in \kappa. [1.21]$$

Definición I.3.3.2 Sea $T: X \rightarrow Y$ lineal, entonces el conjunto,

$$N_T = \{u \in X : Tu = 0\} \subset X \text{ es llamado el núcleo de } T: X \rightarrow Y \quad [1.22]$$

El conjunto

$$R_T = \{v \in Y : \text{existe algún } u \in X, \text{ tal que } v = Tu\} \subset Y,$$

es llamado el rango de $T: X \rightarrow Y$.

Definición I.3.3.3 La transformación $T: X \rightarrow Y$ es uno a uno si:

$$Tx = Ty \Rightarrow x = y \quad \forall x, y \in X. \quad [1.23]$$

Definición I.3.3.4 La transformación $T: X \rightarrow Y$ es sobre si:

$$R_T = Y. \quad [1.24]$$

Definición I.3.3.5 La transformación $T: X \rightarrow Y$ es un isomorfismo si T es lineal, uno a uno y sobre.

Definición I.3.3.6 Sea X un espacio vectorial sobre κ . Un tensor de segundo orden es una transformación lineal $T: X \rightarrow X$.

Definición I.3.3.7 Sea X un espacio vectorial sobre el campo κ , el conjunto definido por

$$L(X, X) \equiv \{ T: X \rightarrow X, T \text{ lineal} \}$$

dicho conjunto con la estructura algebraica definida por

$$i) \quad (S + W) x = S x + W x, \quad \forall S, W \in L(X, X), x \in X \quad [1.25]$$

$$ii) \quad \alpha \bullet (S) x = \alpha \bullet (S x) \quad \forall \alpha \in \kappa, S \in L(X, X), x \in X$$

es un espacio vectorial real y es llamado el espacio de tensores de segundo orden.

Definición I.3.3.8 Sea X un espacio vectorial sobre κ . La transformación $\langle \bullet, \bullet \rangle: X \times X \rightarrow \mathfrak{R}$ es un producto interno si:

$$\begin{aligned} 1) \quad & \langle x, y \rangle \equiv \langle y, x \rangle, & \forall x, y \in X, \\ 2) \quad & \langle \alpha x + \beta y, z \rangle \equiv \alpha \langle x, z \rangle + \beta \langle y, z \rangle, & \forall x, y, z \in X, \forall \alpha, \beta \in \kappa, \\ 3) \quad & \langle x, x \rangle \geq 0, & \forall x \in X, \\ & \langle x, x \rangle = 0, & \Leftrightarrow x = 0 \end{aligned} \quad [1.26]$$

Definición I.3.3.9 Sean $x, y \in X$, entonces la transformación $x \otimes y: X \rightarrow X$ definida por:

$$(x \otimes y)z = \langle y, z \rangle x \quad [1.27]$$

Definición I.3.3.10 Sea X un espacio vectorial sobre κ . La función $n: X \rightarrow \mathfrak{R}$ es una norma en X si:

$$\begin{aligned} 1) \quad & n(x) \geq 0, & \forall x \in X & \text{ y } n(x) = 0 \text{ si y sólo si } x = 0 \\ 2) \quad & n(x \otimes y) \leq n(x) \otimes n(y) & \forall x, y \in X, & \\ 3) \quad & n(\alpha \bullet x) = \alpha \bullet n(x) & \forall x \in X, \forall \alpha \in \mathfrak{R}, & \end{aligned} \quad [1.28]$$

es llamado el producto tensorial de x, y .

Definición I.3.3.11 Las componentes cartesianas del tensor $S \in L(X, X)$ se definen por:

$$S_{ij} = e_i \bullet S e_j \quad [1.29]$$

Definición I.3.3.12 La traza de la transformación $S: X \rightarrow X$, se define por:

$$\text{tr } S \equiv S_{ii}, \quad [1.30]$$

Definición I.3.3.13 La transformación $Q: X \rightarrow X$ es ortogonal si:

$$\langle Qx, Qy \rangle = \langle x, y \rangle, \quad \forall x, y \in X \quad [1.31]$$

Definición I.3.3.14 Sean $S \in L(X,X)$ y $W \in L(X,X)$, la función $\bullet: L(X,X) \times L(X,X) \rightarrow \mathfrak{R}$ definida por

$$S \bullet T = \text{tr} (S^T W), \quad [1.32]$$

es un producto interno en $L(X,X)$.

Definición I.3.3.15 Sea $R: X \rightarrow X$ una transformación ortogonal. Se dice que R es una rotación si

$$\det R > 1 \quad [1.33]$$

Si el $\det R < 1$ se dice que R es una reflexión.

Definición I.3.3.16 Sea $R: X \rightarrow X$, una rotación, el conjunto

$$E_Q \equiv \{x \in X: x = Rx\}, \quad [1.34]$$

es llamado el eje de la rotación de R .

Definición I.3.3.17 Sean $u, v \in X$, el producto vectorial de los vectores u y v se define por:

$$w = u \times v = \varepsilon_{ijk} u_i v_j e_k, \quad [1.35]$$

donde

$$\varepsilon_{ijk} = \begin{cases} 0 & \text{si algún índice se repite,} \\ 1 & \text{si los índices siguen una permutación par de } i, j, k, \\ -1 & \text{si los índices siguen una permutación impar de } i, j, k. \end{cases}$$

Definición I.3.3.18 Un escalar w es un eigenvalor de un tensor $S : X \rightarrow X$ si existe un vector \underline{e} tal que:

$$S\underline{e} = w\underline{e}, \quad \forall \underline{e} \in X \quad [1.36]$$

I.4 EL ALGEBRA DE QUATERNIONES[4].

El siglo pasado el matemático inglés Sir William Hamilton desarrolló los llamados cuaterniones desde un punto de vista geométrico pero no fue sino hasta principios de los años noventa de este siglo cuando en Francia el Institut National de Recherche en Informatique et en Automatique publicó un artículo llamado Quaternion: Une Representation Parametrique Systematique De Rotations Finies Partie 1: Le Cadre Théorique.(Luis Reyes Avila), donde se presenta una forma paramétrica de representar las rotaciones. Así, los cambios de base son representados mediante dos operaciones binarias y se plantea una transformación lineal que permite representar de forma adecuada la física de los sistemas mecánicos.

El presente capítulo es un resumen de este reporte(INRIA) con el fin de mostrar la sistematización del álgebra de cuaterniones, en la primera parte se presentará la estructura algebraica en \mathbb{R}^4 que permitirá identificar la estructura de los cuaterniones (\mathbb{Q}) como un grupo multiplicativo no conmutativo, y se mostrará igualmente que \mathbb{Q} es un espacio vectorial normado. En la segunda parte se definirá una transformación lineal $\rho(p, \bullet): \mathbb{Q} \rightarrow \mathbb{Q}$, $p \in \mathbb{Q}$ fijo, que preserva el producto interno y en donde la representación matricial nos permite identificar un grupo de transformaciones ortogonales propias que representan en la mecánica las rotaciones finitas de cuerpos

rígidos. En la tercera parte se encontrarán algunos resultados importantes de la representación $\rho(p, \bullet) : Q \rightarrow Q$, $p \in Q$ fijo concerniente a la composición de rotaciones. En la cuarta parte se introducirán la forma paramétrica de los valores característicos asociados a la matriz de rotación, y finalmente se estudiará las soluciones del problema cinemático inverso de cuerpos rígidos.

Se definen dos operaciones binarias en \mathfrak{R}^4 , una aditiva $\oplus : \mathfrak{R}^4 \times \mathfrak{R}^4 \rightarrow \mathfrak{R}^4$ y una multiplicativa $*$: $\mathfrak{R}^4 \times \mathfrak{R}^4 \rightarrow \mathfrak{R}^4$, donde los conjuntos (\mathfrak{R}^4, \oplus) y $(\mathfrak{R}^4, *)$ forman un grupo aditivo conmutativo y un grupo multiplicativo no conmutativo. También se definirá una multiplicación escalar $\bullet : \mathfrak{R} \times Q \rightarrow Q$ y un producto interno $\langle \bullet, \bullet \rangle : Q \times Q \rightarrow \mathfrak{R}$, mediante tal estructura el conjunto $Q = (\mathfrak{R}^4, \oplus, *, \bullet, \langle \bullet, \bullet \rangle)$ es un espacio vectorial con producto interno llamado el espacio vectorial de cuaterniones

Sea el conjunto en \mathfrak{R}^4 el cual se definen dos operaciones binarias $\oplus : \mathfrak{R}^4 \times \mathfrak{R}^4 \rightarrow \mathfrak{R}^4$ y $*$: $\mathfrak{R}^4 \times \mathfrak{R}^4 \rightarrow \mathfrak{R}^4$. Esto es,

- i) $(a, b, c, d) \oplus (\alpha, \beta, \gamma, \delta) = (a + \alpha, b + \beta, c + \gamma, d + \delta)$
- ii) $(a, b, c, d) * (\alpha, \beta, \gamma, \delta) = (a\alpha - b\beta - c\gamma - d\delta, a\beta + b\alpha + c\delta - d\gamma, a\gamma - b\delta + c\alpha + d\beta, a\delta + b\gamma - c\beta + d\alpha), \quad \forall (a, b, c, d), (\alpha, \beta, \gamma, \delta) \in \mathfrak{R}^4.$ [1.37]

La operación $\oplus : \mathfrak{R}^4 \times \mathfrak{R}^4 \rightarrow \mathfrak{R}^4$ es la suma usual en \mathfrak{R}^4 y es conocido que el conjunto (\mathfrak{R}^4, \oplus) es un grupo aditivo conmutativo. Además las operaciones anteriores satisfacen lo siguiente:

Teorema I.4.1 El conjunto $Q \equiv (\mathfrak{R}^4, \oplus, *)$ es un cuerpo no conmutativo.

Demostración

- 1) La operación $*$: $\mathfrak{R}^4 \times \mathfrak{R}^4 \rightarrow \mathfrak{R}^4$ es asociativa, dado $p, q, s \in Q$, tal que $p = (a, b, c, d)$, $q = (\alpha, \beta, \gamma, \delta)$ y $s = (x, y, z, w)$, entonces $p * (q * s) = (p * q) * s$
En efecto,

$$\begin{aligned}
p*(q*s) &= ((a\alpha-b\beta-c\gamma-d\delta)x - (a\beta+b\alpha+c\delta-d\gamma)y - (a\gamma-b\delta-c\alpha+d\beta)z - (a\delta+b\gamma-c\beta+d\alpha)w, \\
&\quad (a\beta+b\alpha+c\delta-d\gamma)x + (a\alpha-b\beta-c\gamma-d\delta)y - (a\delta+b\gamma-c\beta+d\alpha)z - (a\gamma-b\delta-c\alpha+d\beta)w, \\
&\quad (a\gamma-b\delta+c\alpha-d\beta)x + (a\delta+b\gamma-c\beta+d\alpha)y + (a\alpha-b\beta-c\gamma-d\delta)z - (a\beta+b\alpha+c\delta-d\gamma)w, \\
&\quad (a\delta+b\gamma-c\beta+d\alpha)x + (a\gamma-b\delta+c\alpha+d\beta)y + (a\beta+b\alpha+c\delta-d\gamma)z - (a\alpha-b\beta-c\gamma-d\delta)w) = \\
&= (p*q)*s)
\end{aligned}$$

- 2) El elemento $I = (1,0,0,0) \in Q$ es tal que : $I*p=p*I=p, \forall p \in Q$, esto es I es el elemento neutro de la multiplicación en Q .
- 3) Para toda $p \in Q, p \neq (0,0,0,0)$, existe un $p' \in Q$ tal que $p * p' = I$, donde p' es el inverso multiplicativo, dado $p=(a,b,c,d), p'=(\alpha,\beta,\gamma,\delta)$, p y $q \in Q$, tal que $p * p' = I$. En efecto si $p*p^{-1} = I$, entonces

$$a\alpha-b\beta-c\gamma-d\delta=0, a\beta+b\alpha+c\delta-d\gamma=1, a\gamma-b\delta-c\alpha+d\beta=1, a\delta+b\gamma-c\beta+d\alpha=1,$$

resolviendo el sistema de ecuaciones se verifica que el elemento inverso es:

$$p' = \left(\frac{a}{a^2+b^2+c^2+d^2}, \frac{-b}{a^2+b^2+c^2+d^2}, \frac{-c}{a^2+b^2+c^2+d^2}, \frac{-d}{a^2+b^2+c^2+d^2} \right)$$

- 4) La operación $*$: $\mathfrak{R}^4 \times \mathfrak{R}^4 \rightarrow \mathfrak{R}^4$ no es conmutativa. Dado $p=(0,1,0,0) \in Q$ y $q=(0,0,1,0) \in Q$ entonces $p*q = -q*p = (0,0,0,1)$.
- 5) Dados $p,q,s \in Q$ entonces son satisfechas las siguientes propiedades distributivas:
- $$(p \oplus q) * s = p * s \oplus q * s,$$
- $$p*(q \oplus s) = p*q \oplus p*s.$$
- 6) La operación \bullet : $\mathfrak{R} \times \mathfrak{R}^4 \rightarrow \mathfrak{R}^4$ definida por:

$$\alpha \bullet (a,b,c,d) = (a\alpha, b\alpha, c\alpha, d\alpha), \forall (a,b,c,d) \in \mathfrak{R}^4 \quad [1.38]$$

es una multiplicación escalar en \mathbb{Q} . Por lo tanto la terna $(\mathfrak{R}^4, \oplus, \bullet)$ es un espacio vectorial real.

La transformación $\langle \bullet, \bullet \rangle : \mathfrak{R}^4 \times \mathfrak{R}^4 \rightarrow \mathfrak{R}$, dada por:

$$\langle p, q \rangle = \sum_{i=0}^3 p_i q_i \in \mathfrak{R} \quad [1.39]$$

es un producto vectorial interno en \mathfrak{R}^4 y la norma asociada es:

$$\|p\| = \langle p, p \rangle^{\frac{1}{2}} \quad [1.40]$$

Por lo tanto $Q = (\mathfrak{R}^4, \oplus, *, \bullet, \|\bullet\|)$, es un espacio vectorial normado el cual se llamará Espacio Vectorial de Cuaterniones y sus elementos Cuaterniones.

Ahora se considera los siguientes subespacios vectoriales de Q

$$\begin{aligned} Q_R &= ((a, 0, 0, 0) : a \in \mathfrak{R}) \subset Q, \\ Q_V &= ((0, b, c, d) : (b, c, d) \in \mathfrak{R}^3), \end{aligned} \quad [1.41]$$

donde Q_R y Q_V es un espacio vectorial real en \mathfrak{R} , luego entonces se puede representar el espacio Q como:

$$Q = Q_R \oplus Q_V \equiv \{ p \in Q : p = p_R \oplus p_V, p_R \in Q_R, p_V \in Q_V \}.$$

entonces, las transformaciones $T_R : Q_R \rightarrow \mathfrak{R}$, $T_V : Q_V \rightarrow \mathfrak{R}^3$, definidas por :

$$\begin{aligned} T_R(a, 0, 0, 0) &= a \\ T_V(0, b, c, d) &= (b, c, d) \end{aligned}$$

son isomorfismos, por lo tanto, si $q = (a, b, c, d) \in Q$, entonces se puede expresar a q como sigue:

$$q = T_R^{-1}(a) \oplus T_V^{-1}(b, c, d). \quad [1.42]$$

Se observa entonces que la estructura algebraica definida por el teorema **1.4.1** es una generalización a \mathfrak{R}^4 del álgebra de los números complejos, entonces se puede introducir el concepto siguiente:

Un quaternion conjugado $\bar{p} \in Q$ de $p = (a, b, c, d)$ está definido por:

$$\bar{p} = (a, -b, -c, -d). \quad [1.43]$$

Teorema 1.4.2 Dado $p = (p_0, p_1, p_2, p_3) \in Q$, $q = (q_0, q_1, q_2, q_3) \in Q$ entonces

- 1) $\overline{p \oplus q} = \bar{p} \oplus \bar{q},$
- 2) $\overline{p * q} = \bar{q} * \bar{p},$ [1.44]
- 3) $p * \bar{p} = \bar{p} * p \in Q_{\mathfrak{R}}.$

Demostración:

Utilizando las propiedades de la adición y la definición de quaternion conjugado se tiene:

- 1) $\overline{p \oplus q} = (p_0 + q_0, -p_1 - q_1, -p_2 - q_2, -p_3 - q_3) = \bar{p} \oplus \bar{q},$
- 2) $\overline{p * q} = (p_0 q_0 - p_1 q_1 - p_2 q_2 - p_3 q_3, p_0 q_1 - p_1 q_0 - p_2 q_3 + p_3 q_2,$
 $- p_0 q_2 + p_1 q_3 - p_2 q_0 - p_3 q_1, - p_0 q_3 - p_1 q_2 + p_2 q_1 - p_3 q_0) = \bar{q} * \bar{p},$

$$3) \quad p * \bar{p} = (p_0^2 + p_1^2 + p_2^2 + p_3^2, 0, 0, 0) = p * \bar{p}$$

La transformación $\langle \bullet, \bullet \rangle : \mathfrak{R}^4 \times \mathfrak{R}^4 \rightarrow \mathfrak{R}$ es definida por:

$$\langle p, q \rangle = p_0 q_0 + p_1 q_1 + p_2 q_2 + p_3 q_3 \quad \forall p, q \in \mathfrak{R}^4 \quad [1.45]$$

La norma $\| \bullet \| : Q \rightarrow \mathfrak{R}$ admite la siguiente representación:

$$\|p\| = \langle p, p \rangle^{1/2} = \{ \bar{p} * p \}^{1/2} \quad [1.46]$$

1.4.1 Representación paramétrica de rotaciones finitas.

El objetivo de esta parte es unificar y por consecuencia simplificar las diferentes formas utilizadas en la mecánica para representar paramétricamente las rotaciones finitas de cuerpos rígidos.

Para este objetivo se define la siguiente transformación lineal :

Sea $\rho(p, \bullet) : Q \rightarrow Q$, $p \in Q$, entonces:

$$\rho(p, q) = p * q * p^{-1} = \frac{1}{\|p\|^2} \bullet (p * q * \bar{p}), \quad \forall p, q \in Q, \quad [1.47]$$

Dicha transformación preserva el producto interno, la norma y el ángulo.

Teorema 1.4.1.1 La transformación $\rho(p, \bullet) : Q \rightarrow Q$ es lineal, ortogonal y $\rho(p, q) \in Q_v$, $\forall q \in Q_v$.

Demostración:

Utilizando las propiedades asociativas y distributivas de la estructura algebraica de Q , se tiene que:

$$\begin{aligned} \rho(p, q \oplus s) &= \frac{1}{\|p\|^2} \cdot \{p^*((q \oplus s) * \bar{p})\} = \frac{1}{\|p\|^2} \cdot \{p^*((q * \bar{p} \oplus s * \bar{p}))\} = \\ &= \frac{1}{\|p\|^2} \cdot \{p^*(q * \bar{p}) \oplus p^*(s * \bar{p})\} = \rho(p, q) \oplus \rho(p, s), \quad \forall p, q, s \in Q. \end{aligned}$$

$$\begin{aligned} \rho(p, \alpha \cdot q) &= \frac{1}{\|p\|^2} \cdot \{p^*(\alpha \cdot q) * \bar{p}\} = \frac{\alpha}{\|p\|^2} \{p^* q * \bar{p}\} \\ &= \alpha \cdot \rho(p, q), \quad \forall p, q, s \in Q. \quad \alpha \in \mathfrak{R} \end{aligned}$$

Para probar la ortogonalidad se debe demostrar que:

$$\langle \rho(p, q), \rho(p, s) \rangle = \rho \langle q, s \rangle \quad \forall q, s \in Q.$$

En efecto, utilizando los teoremas I.4.1, y I.4.2 se tiene que:

$$\begin{aligned} \langle \rho(p, q), \rho(p, s) \rangle &= \frac{1}{2\|p\|^4} \{ (p^* q * \bar{p}) * (p^* s * \bar{p}) \oplus (p^* s * \bar{p}) * (p^* q * \bar{p}) \} = \\ &= \frac{1}{2\|p\|^4} \{ (q * \bar{p}) * \bar{p} * (p^* s * \bar{p}) \oplus ((s * \bar{p}) p^*) * (p^* q * \bar{p}) \} = \\ &= \frac{1}{2\|p\|^2} \{ p^* (\bar{q} * s \oplus \bar{s} * q) * \bar{p} \} = \rho \langle q, s \rangle. \end{aligned}$$

Se observa que si $q = (0, q_1, q_2, q_3) \in Q_V$ se tiene que:

$$\begin{aligned} \rho(p, q) &= \frac{1}{\|p\|^2} (0, (p_0^2 + p_1^2 - p_2^2 - p_3^2)q_1 + 2(p_1 p_2 - p_0 p_3)q_2 + 2(p_0 p_2 + p_1 p_3)q_3, \\ &\quad 2(p_0 p_3 - p_1 p_2)q_1 + (p_0^2 - p_1^2 + p_2^2 - p_3^2)q_2 + 2(p_2 p_3 - p_0 p_1)q_3, \\ &\quad 2(p_1 p_3 - p_0 p_2)q_1 + 2(p_0 p_1 + p_2 p_3)q_2, (p_0^2 + p_1^2 - p_2^2 + p_3^2)q_3) \in Q_V. \end{aligned}$$

Dado lo anterior , se puede construir la matriz $M_{\rho(p,\bullet)} \in M_{4 \times 4}$ asociada a la transformación [1.47]

Teorema 1.4.1.2 La matriz $M_{\rho(p,\bullet)} \in M_{4 \times 4}$ asociada a la transformación $\rho(p, \bullet) \in L(Q, Q)$, dada la base $B = \{e_j\}_{j=1}^4$ es :

$$M_{\rho(p,\bullet)} = \frac{1}{\|p\|^2} \begin{pmatrix} \|p\|^2 & 0 & 0 & 0 \\ 0 & p_0^2 + p_1^2 - p_2^2 - p_3^2 & 2(p_1 p_2 - p_3 p_0) & 2(p_0 p_2 + p_1 p_3) \\ 0 & 2(p_0 p_3 + p_1 p_2) & p_0^2 - p_1^2 + p_2^2 - p_3^2 & 2(p_2 p_3 - p_0 p_1) \\ 0 & 2(p_1 p_3 - p_0 p_2) & 2(p_0 p_1 + p_2 p_3) & p_0^2 - p_1^2 - p_2^2 + p_3^2 \end{pmatrix} \quad [1.49]$$

$$p = (p_0, p_1, p_2, p_3) \in Q$$

Demostración en [4]

Entonces la transformación $\rho(p, \bullet) : Q_V \rightarrow Q_V$, utiliza el isomorfismo entre Q_V y \mathbb{R}^3 y se puede tener su representación en una matriz de orden 3×3 . Para nuestro caso se construirá la matriz de $\rho(p, \bullet) : Q \rightarrow Q$ de orden 4×4 respecto a la estructura algebraica de Q .

CAPÍTULO II

MODELACIÓN CINEMÁTICA DE UN MANIPULADOR DE 2 GRADOS DE LIBERTAD

2.1 Cadenas Cinemáticas Abiertas [1].

En este capítulo se desarrollará la metodología para modelar cadenas cinemáticas compuestas por cuerpos rígidos.

Los marcos teóricos tratados en el capítulo I se usarán para modelar de forma sistemática las cadenas cinemáticas que componen a los robots y mecanismos. El objetivo de este capítulo es el de unificar las diferentes metodologías que existen para representar las rotaciones en base al marco teórico del álgebra de Cuaterniones[4]. Un aspecto importante en esta parte es la generalización del método para la modelación de las cadenas cinemáticas abiertas y cerradas, así como para cadenas cinemáticas compuestas.

Normalmente los manipuladores son modelados con la metodología de Denavit-Hartenberg, en tanto los mecanismos son tratados con el álgebra compleja, en su representación exponencial. La modelación con el álgebra de Cuaterniones permite las caracterizaciones cinemáticas de robots más complejos, además de ser útil en los sistemas de giroscopios y en general a los sistemas mecánicos que produzcan movimiento.

Para mostrar el uso del álgebra de Cuaterniones en las rotaciones finitas de cuerpos rígidos acoplados, se modela una cadena cinemática de dos grados de libertad tipo RR, que utilizaremos en nuestra aplicación.

2.1.1 Modelación cinemática de un sistema mecánico de cadena abierta tipo RR.

El multicuerpo a analizar, en esta parte, es una cadena cinemática abierta formada por dos cuerpos rígidos acoplados por medio de juntas rotacionales y su campo de acción es el plano XY.

El problema consiste en construir las ecuaciones que gobiernan la cinemática de la cadena mostrada en la figura 2.1 , el extremo final del segundo eslabón, está obligado, en este caso, a seguir una trayectoria específica descrita en el plano XY, con el fin de colocar una herramienta de trabajo que simule una operación de un proceso determinado.

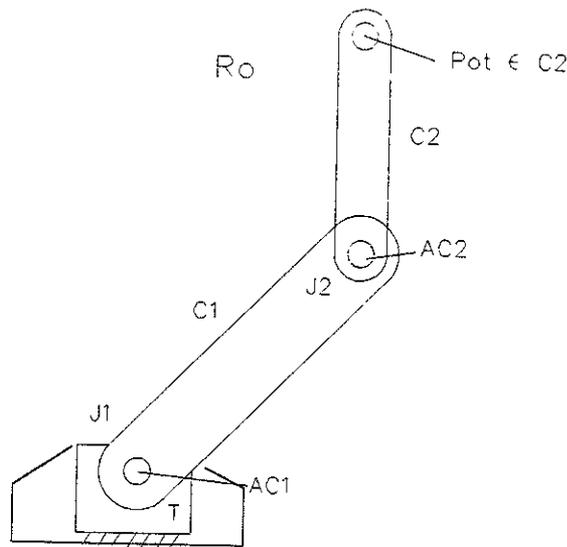


Figura 2.1	Caracterización de robot R _o .
------------	-------------------------------------------

Consideremos el multicuerpo (R_o), mostrado en 2.1 , observemos que R_o puede ser representado por:

$$R_0 = C_1 \cup C_2 \quad [2.1]$$

El multicuerpo está montado en el cuerpo T, mostrado en la figura anterior, al cual llamaremos la tierra del sistema. Llamaremos la junta del sistema denotada por J_1 , al conjunto de puntos dados por la intersección siguiente:

$$J_1 = T \cap C_1 \quad [2.2]$$

En la tierra se montará un actuador, al cual llamaremos AC_1 el cual se acoplará al cuerpo C_1 , a través de la junta J_1 y del eje del actuador.

La junta que conecta al cuerpo C_1 con el cuerpo C_2 se define como:

$$J_2 = C_1 \cap C_2 \quad [2.3]$$

En esta parte del cuerpo se localiza un segundo actuador al cual llamaremos AC_2 y está acoplado a los cuerpos C_1 y C_2 a través de la junta J_2 .

En este trabajo el multicuerpo R_0 se considerará formado de la forma siguiente:

$$R_0 = C_2 \cup J_2 \cup AC_2 \cup C_1 \cup AC_1 \cup J_1 \cap T \quad [2.4]$$

Sobre la junta J_1 se define un sistema cartesiano inercial, también sobre el eje de C_1 se define un sistema local y sobre la junta J_2 se define el sistema de referencia del cuerpo C_2 como se muestra en la figura 2.2. Los sistemas de referencia asociados a los cuerpos C_1 y C_2 serán llamados en este trabajo las bases móviles y están fijas en los cuerpos C_1 y C_2 . Las longitudes de los cuerpos C_1 y C_2 serán denotadas por L_1 y L_2 según se muestra en la figura 2.2. Por tanto, los vectores de posición que unen el centro de la junta J_1 con la junta J_2 y de la junta J_2 al punto $p_{OT} \in C_2$ mostrado en la figura 2.2, al cual llamaremos

llamaremos punto final de R_o , están definidos con respecto a las bases locales. Esto es:

$$\underline{L}_1 = l_1 \underline{e}'_1, \quad \underline{L}_2 = l_2 \underline{e}'_2. \quad [2.5]$$

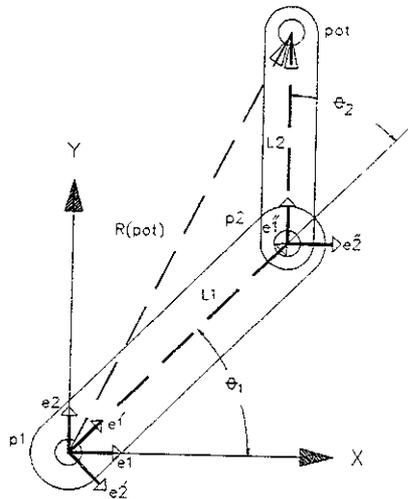


Figura 1.2 Configuración no deformada de R_o .

observemos que dichos vectores, \underline{L}_1 , \underline{L}_2 , al fijar los puntos $p_1 \in J_1$, $p_2 \in J_2$, $p_{ot} \in C_2$ pueden ser también expresados mediante la siguiente diferencia de puntos:

$$\underline{L}_1 = p_2 - p_1, \quad \underline{L}_2 = p_{ot} - p_2 \quad [2.6]$$

También el vector de posición de $p_{ot} \in C_2$ puede ser expresado como:

$$\underline{R}(p_{ot}) = p_{ot} - p_1 \quad [2.7]$$

El vector $\underline{R}(p_{ot}) \in \mathcal{R}^2$ es representado a través de la siguiente ecuación:

$$\underline{R}(p_{ot}) = \underline{L}_1 + \underline{L}_2 \quad [2.8]$$

2.1.1.1 Análisis en la configuración no deformada de R_o .

Consideremos que el multicuerpo R_o ocupa la posición mostrada en la figura 2.2. Entonces, el vector de posición de $p_{ot} \in C_2$, está dado por:

$$R(p_{ot}) = l_1 \underline{e}'_1 + l_2 \underline{e}''_1. \quad [2.9]$$

En este trabajo consideremos que las bases móviles son rotaciones rígidas de la base inercial fija, de acuerdo a [4], tales rotaciones pueden expresarse, en el caso de Cuaterniones unitarios por:

$$\begin{aligned} \underline{e}'_1 &= \rho(p, \underline{e}_1) \equiv p * \underline{e}_1 * \bar{p} \\ \underline{e}''_1 &= \rho(q, \rho(p, \underline{e}_1)) \equiv (q * p) * \underline{e}_1 * (\overline{q * p}) \end{aligned} \quad [2.10]$$

donde $\rho(p, \bullet) : Q \rightarrow Q$, $p \in Q$, está fijo. En base a [2.10], [2.9] puede ser escrita de la forma siguiente:

$$R(p_{ot}) = l_1 \{ p * \underline{e}_1 * \bar{p} \} + l_2 \{ (q * p) * \underline{e}_1 * (\overline{q * p}) \} \quad [2.11]$$

También considerando las normas unitarias de las Cuaterniones. Esto es,

$$\| p \| = \| q \| = 1. \quad [2.12]$$

obtenemos las ecuaciones que gobiernan la configuración no deformada de R_o .

Los componentes de los Cuaterniones asociados $p, q \in \mathfrak{R}^4$ son definidos por:

$$\begin{array}{llll}
 p \in \mathbb{R}^4 & p = (p_o, \underline{p}_v) & p_o \in \mathbb{R} & p_o = \pm \text{Cos} \frac{\theta_1}{2} \\
 & & \underline{p}_v \in \mathbb{R}^3 & \underline{p}_v = \pm \text{Sen} \frac{\theta_1}{2} \underline{e}_3, \\
 q \in \mathbb{R}^4 & q = (q_o, \underline{q}_v) & q_o \in \mathbb{R} & q_o = \pm \text{Cos} \frac{\theta_2}{2} \\
 & & \underline{q}_v \in \mathbb{R}^3 & \underline{q}_v = \pm \text{Sen} \frac{\theta_2}{2} \underline{e}_3.
 \end{array}
 \tag{2.13}$$

2.1.1.2 Planteamiento de los problemas cinemáticos directo e inverso.

En esta parte, definiremos dos problemas asociados al multicuerpo en estudio. El primero de ellos, será el problema de la cinemática directa y el segundo corresponde a la cinemática inversa. Tales problemas están asociados a la posición mostrada en la figura 2.2, la cual, será para nosotros la configuración no deformada. Consideremos entonces el siguiente problema:

2.1.1.2.1 Problema cinemático directo.

“Dados $p = (p_o, 0, 0, p_3)$, $q = (q_o, 0, 0, q_3)$ con $\|p\| = \|q\| = 1$ y $l_1 \in \mathbb{R}^+$, $l_2 \in \mathbb{R}^+$, encuentre $R(p_{ot}) \in \mathbb{R}^2$ tal que [2.11] sea satisfecha.”

2.1.1.2.2 Problema cinemático inverso.

Presentaremos ahora en esta parte el problema inverso de la cinemática de cuerpos rígidos para el cuerpo en estudio. Esto es,

“Dado $R(p_{0i}) \in \mathbb{R}^2$ y $l_1 \in \mathbb{R}^+$, $l_2 \in \mathbb{R}^+$, encuentre $p=(p_0, 0, 0, p_3)$, $q=(q_0, 0, 0, q_3)$ tal que [2.11] sea satisfecha” y

$$p_0^2 + p_3^2 = 1 \quad [2.14]$$

$$q_0^2 + q_3^2 = 1$$

El problema de la cinemática inversa genera 4 ecuaciones algebraicas no lineales con 4 incógnitas del tipo polinomial.

2.1.1.3 Análisis de la configuración deformada de R_0 .

En esta parte se supone que el robot (R_0) ha sido movido de su posición inicial (configuración no deformada) a través de accionamientos de los actuadores AC_1 y AC_2 (ver figura 2.3). El problema ahora consiste en determinar el lugar geométrico que ocupa el extremo final del multicuerpo. La configuración así obtenida se denominará configuración deformada.

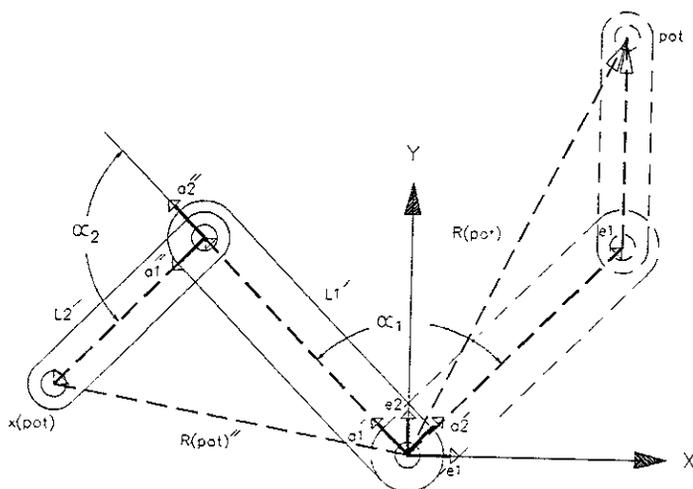


Figura 2.3 Configuración deformada de R_0 .

Observemos que el vector de posición $R(p_{ot})''$ correspondiente a la configuración deformada es expresado de la forma siguiente:

$$R(p_{ot})'' = L_1' + L_2' \quad [2.15]$$

Los vectores de posición L_1', L_2' serán para nosotros los vectores deformados, los cuales también pueden ser escritos en términos de las bases locales. Esto es,

$L_1' = l_1 \underline{a}_1', L_2' = l_2 \underline{a}_1''$. Las bases $\underline{a}_1', \underline{a}_1''$ serán para nosotros las bases móviles deformadas. En términos de Cuaterniones las bases deformadas se expresan de la forma siguiente:

$$\begin{aligned} \underline{a}_1' &= \rho(P, \underline{e}_1') \equiv P * p * \underline{e}_1 * \overline{P * p} \\ \underline{a}_1'' &= \rho(Q, \rho(P, \underline{e}_1')) \equiv Q * P * q * p * \underline{e}_1 * (\overline{Q * P * q * p}) \end{aligned} \quad [2.16]$$

En base a [2.11] el vector de posición $R(p_{ot})'' \in \mathbb{R}^2$ puede ser representado de la forma siguiente:

$$R(p_{ot})'' = l_1 \{ P * p * \underline{e}_1 * \overline{P * p} \} + l_2 \{ Q * P * q * p * \underline{e}_1 * (\overline{Q * P * q * p}) \} \quad [2.17]$$

También considerando las normas unitarias de las Cuaterniones $P, Q \in \mathbb{R}^4$. Esto es,

$$\| P \| = \| Q \| = 1. \quad [2.18]$$

de esta obtenemos las ecuaciones que gobiernan la configuración deformada del multicuerpo en estudio.

Al igual que en [1.13] los componentes de los Cuaterniones $P, Q \in \mathbb{R}^4$ son:

$$\begin{array}{llll}
 P \in \mathcal{R}^4 & P = (P_o, \underline{P}_V) & P_o \in \mathcal{R} & [2.19] \\
 & & \underline{P}_V \in \mathcal{R}^3 & \\
 Q \in \mathcal{R}^4 & Q = (Q_o, \underline{Q}_V) & Q_o \in \mathcal{R} & \\
 & & \underline{Q}_V \in \mathcal{R}^3 &
 \end{array}$$

Ahora procederemos a plantear el problema cinemático inverso en la configuración deformada.

2.1.1.4 Problema cinemático inverso en la configuración deformada.

“Dado $\mathcal{R}(p_{ot}) \in \mathcal{R}^2$ y $l_1 \in \mathcal{R}^+$, $l_2 \in \mathcal{R}^+$, encuentre $p = (p_o, 0, 0, p_3)$, $q = (q_o, 0, 0, q_3)$ con $\|p\| = \|q\| = 1$ tal que [2.17] sea satisfecha y

$$\begin{array}{ll}
 P_o^2 + P_3^2 = 1 & [2.20] \\
 Q_o^2 + Q_3^2 = 1 &
 \end{array}$$

El problema de la cinemática inversa en la posición deformada genera 4 ecuaciones algebraicas no lineales con 4 incógnitas del tipo polinomial.

2.1.1.5 Análisis de los movimientos de R_o .

En esta parte analizaremos 2 posibles formas de movimiento del multicuerpo R_o para alcanzar la configuración deformada mostrada en la figura 2.3.

Secuencia 1. La primer secuencia de movimientos mostrada en la figura 2.4 corresponde a mover primero al cuerpo C_1 a través del actuador AC_1 y después al cuerpo C_2 por medio del actuador AC_2 .

Observemos entonces que la configuración b) mostrada en 2.4 es para nosotros la configuración intermedia del multicuerpo en estudio. La ecuación que gobierna el primer movimiento mostrado en 2.4 es la siguiente:

$$R(p_{ot})'_{s1} = l_1 \{ P * p * \underline{e}_1 * \overline{P * p} \} + l_2 \{ P * q * p * \underline{e}_1 * (\overline{P * q * p}) \} \quad [2.21]$$

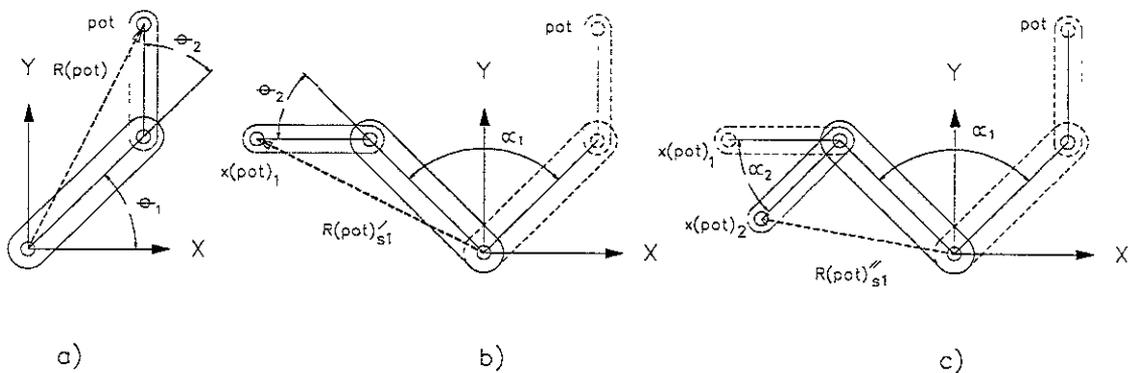


Figura 2.4	Secuencia 1.
------------	--------------

La configuración c) es alcanzada mediante el movimiento del cuerpo C_2 desde la posición intermedia $R(p_{ot})'_{s1}$. Observemos que la ecuación que gobierna la posición deformada para la secuencia 1 es [2.17].

Secuencia 2. Una secuencia de especial interés es la mostrada en la figura 2.5. La descripción de este movimiento es: mover al cuerpo C_1 y al mismo tiempo mover también

al cuerpo C_2 con la misma cantidad de rotación que C_1 , finalmente desde la posición intermedia $R(p_{ot})'_{s3}$ mover al cuerpo C_2 hasta la configuración deformada. La posición intermedia de esta secuencia es dada por la ecuación siguiente:

$$R(p_{ot})'_{s3} = I_1 \{ P * p * \underline{e}_1 * \overline{P * p} \} + I_2 \{ r * \underline{e}_1 * \bar{r} \} \quad [2.22]$$

siendo $r = P * q * p$.

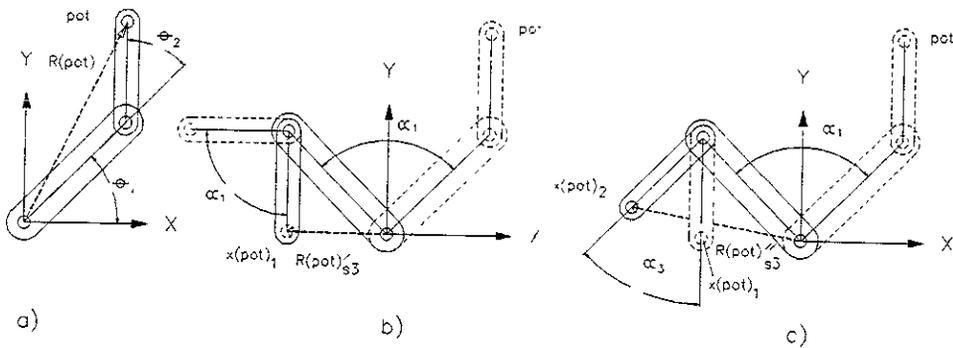


Figura 2.5	Secuencia 2.
------------	--------------

La configuración deformada es alcanzada por el movimiento C_2 desde la posición intermedia $R(p_{ot})'_{s3}$ mostrada en configuración b) de 2.5 . La ecuación asociada a la configuración deformada es la siguiente:

$$R(p_{ot})''_{s3} = I_1 \{ P * p * \underline{e}_1 * \overline{P * p} \} + I_2 \{ s * \underline{e}_1 * s \} \quad [2.23]$$

siendo $s = Q * r$.

Notemos también que las diferencias entre las dos secuencias se pueden caracterizar por dos formas, las cuales son:

- 1.- Por las configuraciones intermedias.
- 2.- Por el número de accionamientos de los actuadores AC_1 y AC_2 .

Observemos que la secuencia 2 necesita al menos tres accionamientos de los actuadores para alcanzar la configuración deformada, en tanto la secuencia 1 utiliza 2 accionamientos.

2.1.1.6 Análisis de las secuencias de rotación para interpretación de los ángulos.

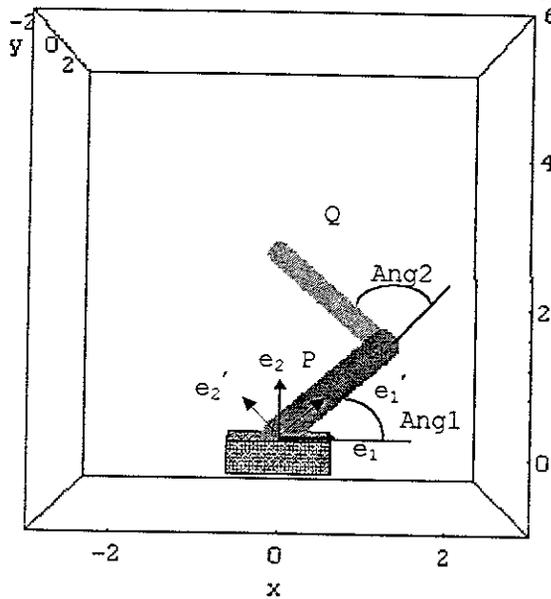
La Interpretación física de los ángulos de rotación es la siguiente. Los ángulos de la configuraciones deformadas 1,2,3 y 4 (ver tabla 2.1) son tomados desde la configuración no deformada, sin embargo, para enviar esta información a los motores del robot es necesario tomar la configuración deformada 1 como la configuración no deformada para la configuración deformada 2, y después de haber realizado las rotaciones respectivas, tomar la configuración deformada 2, como la configuración no deformada hasta llegar a la configuración deformada 3, y así sucesivamente. Es decir, los ángulos se miden desde la posición inmediata anterior para llegar a la siguiente posición, según los puntos seleccionados de la recta. (Ver tabla 2.1).

Nota: Los ángulos están dados en grados.

Tabla 2.1
Secuencias para el seguimiento de los puntos de una recta

$$px = -2 + x$$

$$pz = 1.1 + (x * 6), \quad (0 = x < 4)$$

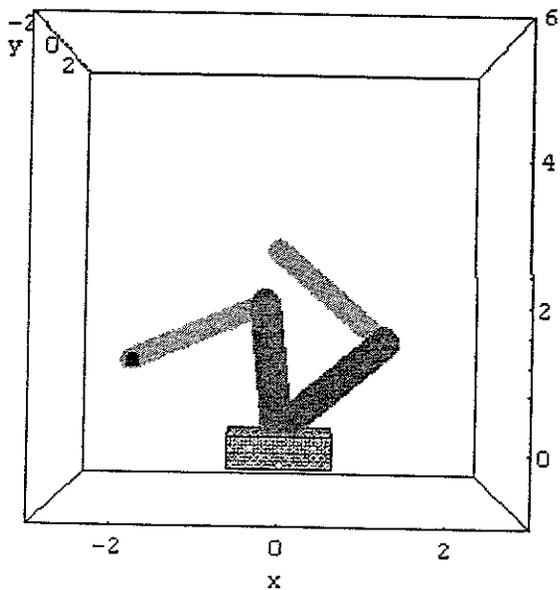


Configuración no deformada (Home)

Ang1= 45.
Ang2= 90.
(Desde la base inercial fija)

$p = \{0.92388, 0, -0.382683, 0\}$
 $q = \{0.707107, 0, -0.707107, 0\}$

z Vector posición =
-16
 $\{0, 4.44089, 10, 0, 2.82843\}$

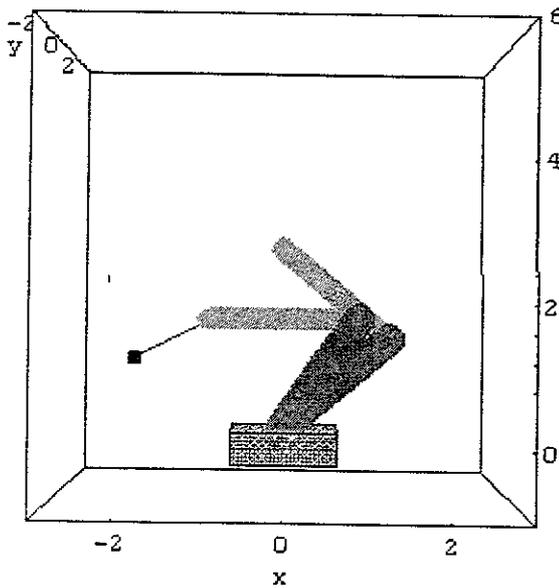


Configuración deformada (1)

Ang1= 50.9838
Ang2= 71.3946
(Desde la posición de Home)

$P = \{0.902646, 0, -0.430383, 0\}$
 $Q = \{0.812111, 0, -0.583503, 0\}$

z Vector posición = $\{0, -2., 0, 1.1\}$

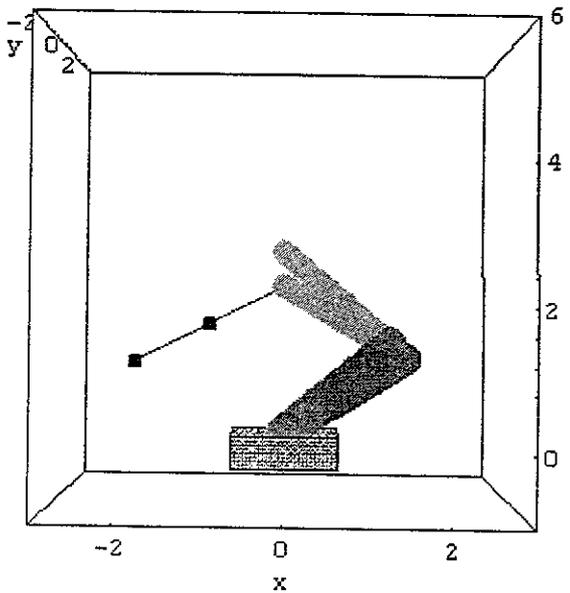


Configuración deformada (2)

Ang1= -15.0086
 Ang2= -45.9225
 (Desde la posición de Home)

P= {0.991435, 0, -0.1306, 0}
 Q= {0.920769, 0, -0.390109, 0}

z Vector posición = {0, -1., 0, 1.7}

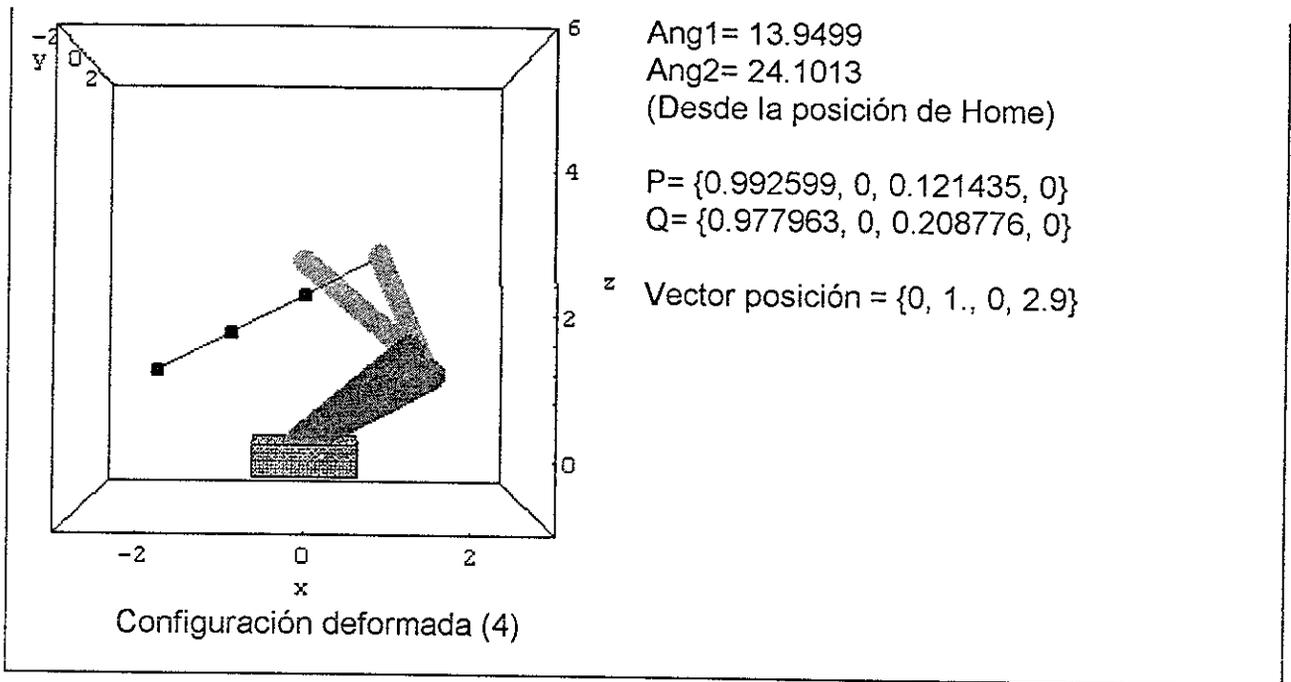


Configuración deformada (3)

Ang1= 9.90037
 Ang2= -9.90037
 (Desde la posición de Home)

P= {0.99627, 0, 0.0862896, 0}
 Q= {0.99627, 0, -0.0862896, 0}

z Vector posición = {0, 0., 0, 2.3}



Las cantidades de rotación de los eslabones 1 y 2, están compuestas, es decir, cuando el eslabón 1 rota un ángulo θ_1 , el eslabón 2 rota un ángulo $\theta_1 + \alpha_1$. Donde α_1 es la cantidad de rotación correspondiente al eslabón 2. Por ejemplo, considerando la configuración deformada (1) de la tabla 2.2. Tenemos que el eslabón 1 y 2 rotan un ángulo de Ang1= -50.9838 y el eslabón 2 rota un ángulo de Ang2= -71.3946 - (-50.9838) = -20.4108 grados. El signo (-) indica que rotarán en sentido contrario a las manecillas del reloj.

Tabla 2.2.
 Valores de los ángulos de rotación para el seguimiento de 4 puntos de una línea recta

Configuración	Compuestos		Individuales		Relativos c/Redondeo (Al programa de control)	
	Ang1	Ang2	Ang1	Ang2 -Ang1	Ang1	Ang2
No deformada	45	135	45	90	45	90
Deformada (1)	-50.9838	-71.3946	-50.9838	-20.4108	51.0	20.4
Deformada (2)	-15.0086	-45.9225	-15.0086	-30.9139	-36.0	10.5
Deformada (3)	9.90037	-9.90037	9.90037	-19.80074	-24.9	-11.1
Deformada (4)	13.9499	24.1013	13.9499	10.1514	-4.0	-30.0

Los valores mostrados en la tabla 2.2 representan un análisis de las configuraciones deformadas de el robot siguiendo 4 puntos de una recta, se puede observar que desde la configuración 1 los valores de los ángulos se miden desde la configuración inmediata anterior con el fin de seguir una secuencia lógica, esto es debido a que en la modelación matemática todos los ángulos de las deformaciones se obtienen con respecto a la configuración no deformada, y es necesario medir los ángulos de rotación desde las configuraciones deformadas ya que en la realidad el robot va tomando físicamente las configuraciones deformadas y no se puede regresar a la configuración no deformada cada vez que realice una deformación.

El Robot también intentará seguir los puntos mostrados en la fig. 2.6, se procede de igual forma que lo anteriormente expuesto.

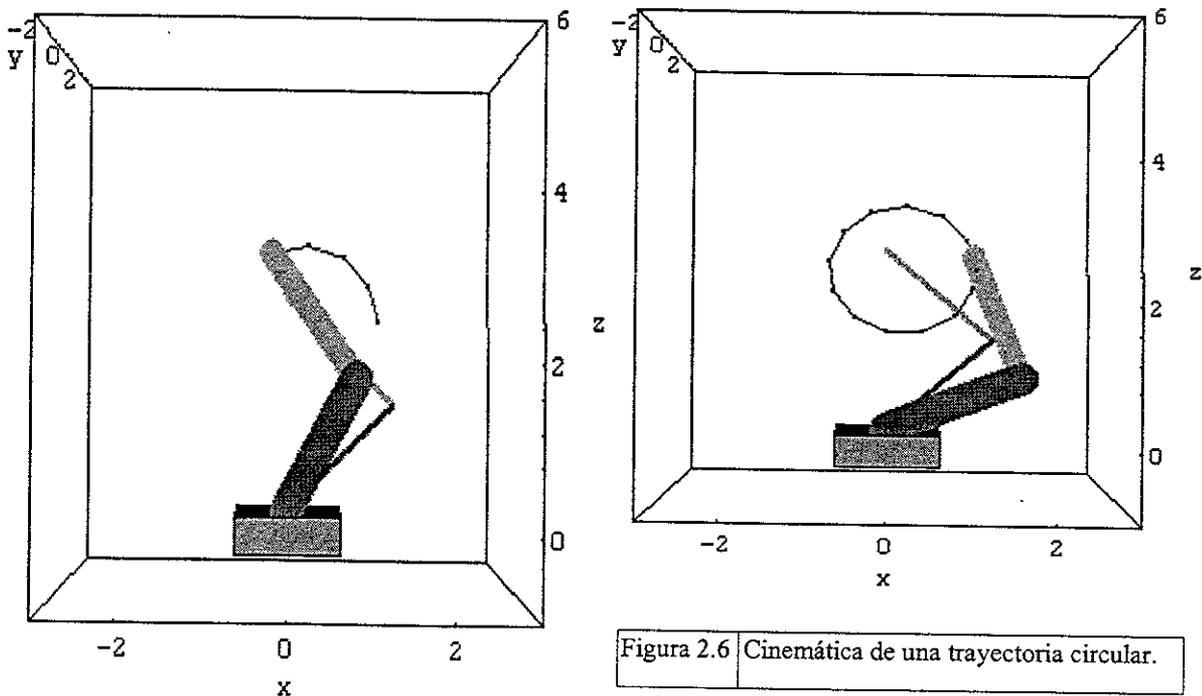


Figura 2.6 Cinemática de una trayectoria circular.

La ecuación paramétrica es:

$$px=0.2+\text{Cos}[x], \quad py=0, \quad pz=2.5+\text{Sin}[x]; \quad \text{para} \quad (0 < x < 14) \quad [2.24]$$

CAPÍTULO III

DISEÑO CONCEPTUAL Y DE FORMA DE UN MANIPULADOR DE DOS GRADOS DE LIBERTAD.

INTRODUCCIÓN:

En este capítulo, se justifica desde un punto de vista general, la arquitectura del robot de 2 GDL motivo de nuestro estudio.

El objetivo en este capítulo es el de utilizar la teoría de conjuntos para fundamentar al diseño conceptual y de forma del robot. Las ecuaciones de forma serán utilizadas para caracterizar las diferentes etapas de construcción conceptual de algunas piezas o elementos que componen la estructura de nuestro manipulador. La fundamentación de este capítulo, forma parte del proyecto automatización global de una fábrica futurista de ensamble y soldadura (Global Automation Assembly And Welding Futuristic Factory) Sistema Global Marianne XI desarrollado por el grupo DITAI¹. La parte del proyecto a la cual está referida este capítulo es al módulo de Ensamble y corresponde al reporte interno de investigación #20, llamado "DISEÑO GEOMÉTRICO DEL MÓDULO DE ENSAMBLE" [5].

3. 1 Diseño de un manipulador de dos grados de libertad.

Una parte importante en la creación de prototipos es el proceso de Diseño Geométrico. El diseño permite objetivizar una idea, crear técnicas de integración de partes, caracterizar los elementos importantes de un prototipo y finalmente dimensionarlo para su manufactura. El proceso de Diseño Geométrico será dividido en tres partes, estas son:

¹Desarrollo e Integración Tecnológica y asesoría en Investigación.

- 1).- Diseño conceptual
- 2).- Diseño de forma
- 3).- Diseño de detalle y dimensionamiento.

En la primera parte conceptualizaremos el problema a resolver, esto otorga una primera solución de la posible geometría del prototipo. La segunda parte otorga información en cuanto a como generar las partes y los procesos de manufactura asociados a ellos. Finalmente la tercera parte prepara el llamado "plano de fabricación " para la manufactura del prototipo (En este trabajo no se realizarán consideraciones en cuanto a los planos de fabricación).

La generación de tecnología deberá acompañarse de una variable importante como lo es el mercado, esta variable será introducida en este trabajo. Bajo restricciones de producción, el diseño de los prototipos debe realizarse de manera eficiente. Para iniciar el problema de diseñar al robot iniciaremos definiendo el problema a resolver junto con las restricciones generales.

3.1 Definición del problema

Necesitamos utilizar un Robot de dos grados de libertad tipo RR para la comprobación experimental de la modelación cinématica del Robot. Las operaciones a realizar por este robot, son la de seguir trayectorias a través de puntos cuyas coordenadas son dadas por la solución de las ecuaciones que gobiernan la cinemática inversa de posicionamiento del Robot. El robot a diseñar al cual llamaremos R_0 deberá ser conceptualizado, formado y dimensionado siguiendo las restricciones de planta y de proceso.

3.1.1 Restricciones (R_0) del problema

La fabricación del robot deberá cumplir con las siguientes restricciones generales:

- 1).- El diseño de partes deberá permitir la fabricación en serie del Robot.
- 2).- El volumen que ocupará el robot no deberá exceder de 40 cm².
- 3).- El máximo radio de alcance (Reach) debe ser de 200 mm.
- 4).- La velocidad máxima del órgano terminal debe ser de 0.5 m/s.
- 5).- La temperatura de operación es de 30°C.
- 6).- Las geometrías asociadas a los eslabones no deberán tener procesos de manufactura complicados.
- 7).- Las partes importantes del robot deberán cumplir con restricciones de diseño por ensamble.
- 8) .- La norma utilizada será la Mexicana.
- 9).-Las piezas normalizadas deberán encontrarse en el mercado de forma rápida.
- 10).- Los materiales asignados deberán cumplir con la restricción (9).

Definido el problema a resolver y sus restricciones, ahora procederemos a caracterizar el Diseño Conceptual del prototipo.

3.1.2 Definición del Problema de Diseño Conceptual.

En base a un proceso de discusión en cuanto a propuesta de la posible forma del robot, se ha llegado a la conclusión que la configuración adecuada del prototipo es la mostrada en la figura 3.1.

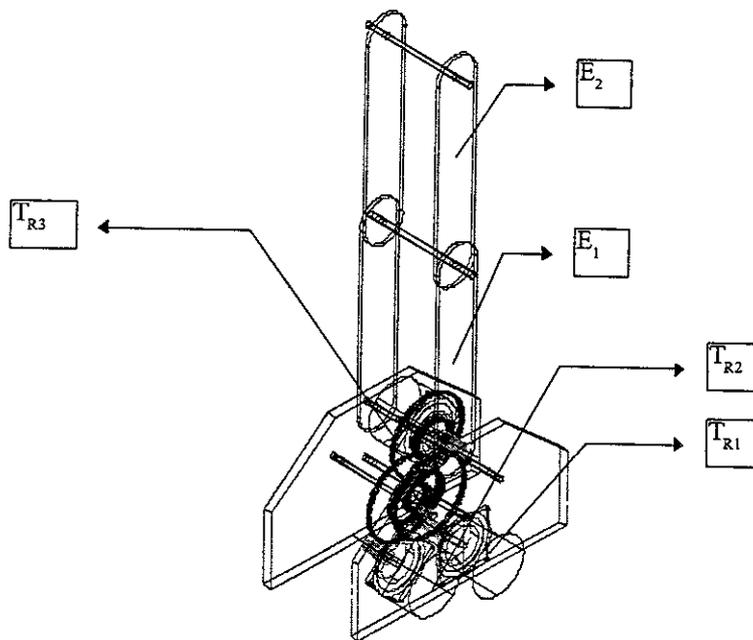


Figura 3.1 El robot de dos grados de libertad.

El prototipo consta de 2 eslabones articulados y una base que soporta la estructura del robot. En el proceso del Diseño Conceptual no se requiere analizar al prototipo siguiendo rigidamente las restricciones anteriores, sin embargo, algunas de ellas deberán ser incluidas, por ejemplo las restricciones (6) y (9).

Las consideraciones para realizar el Diseño Conceptual son abiertas, y se requiere de al menos 2 propuestas para discutir la posible configuración del robot. También debe de conocerse la posición de "home" o la configuración no deformada del robot, la cual se muestra en la figura 3.2. En este trabajo las operaciones del robot serán en el plano euclideo xz mostrado en la figura 3.2.

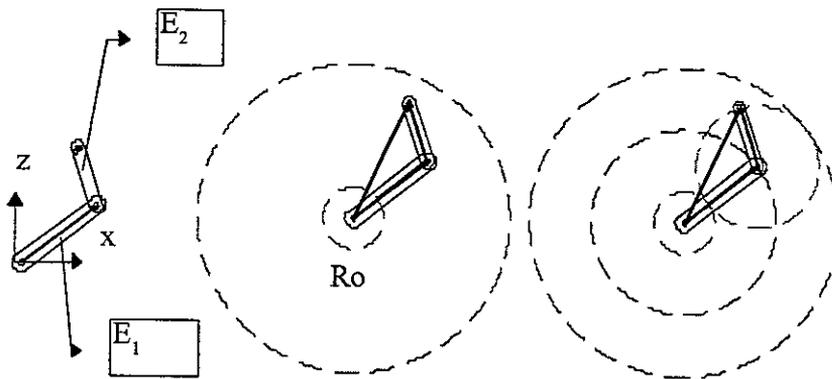


Figura 3.2	Configuración no deformada del robot.
------------	---------------------------------------

Al caracterizar una forma determinada del prototipo (R_0), el siguiente paso es la generación de los elementos que componen al robot, esto es en el sentido general, es decir, la forma de los eslabones, y las bases que son los elementos principales, además de los elementos de acoplamiento y transmisión de potencia.

3.1.3 Descripción del Diseño Conceptual del Robot de 2 gdl.

“El robot a diseñar está compuesto de 2 eslabones, cada eslabón está integrado por dos cuerpos unidos por acopladores y sistemas de sujeción por tornillo y tuerca, el movimiento de los eslabones se realiza por dos motores de pasos, uno para cada eslabón, la transmisión de movimiento es mediante engranes y sistemas polea-banda dentada. La ubicación de los motores estará en las bases 1 y 2 mostradas en las figuras 3.1 y 3.3. Los sistemas de transmisión cuentan con 5 elementos, un piñón colocado en la flecha de cada motor y una corona con una polea acoplada mediante tornillos además de otra polea colocada en los acopladores y unida al eslabón 1, si es la transmisión 1”. Como se observa en 3.1.

La transmisiones 1 y 2 están asociadas a los eslabones 1 y 2 mediante poleas y bandas dentadas, colocadas en cada eje donde rotarán los eslabones. Utilizando la teoría de conjuntos, procederemos a definir a las transmisiones de potencia. El conjunto T_{R1} mostrado en la figura 3.3, será para nosotros la transmisión 1 y contiene: la base 1, el motor 1 y al sistema de transmisión 1 por engranes y poleas, el conjunto T_{R1} contiene también al eslabón 1, al eje acoplador 1 y a los sistemas de sujeción. Una forma de representar al robot R_0 motivo de nuestro estudio en términos de sus componentes utilizando para esto la teoría de conjuntos es vista en [1].

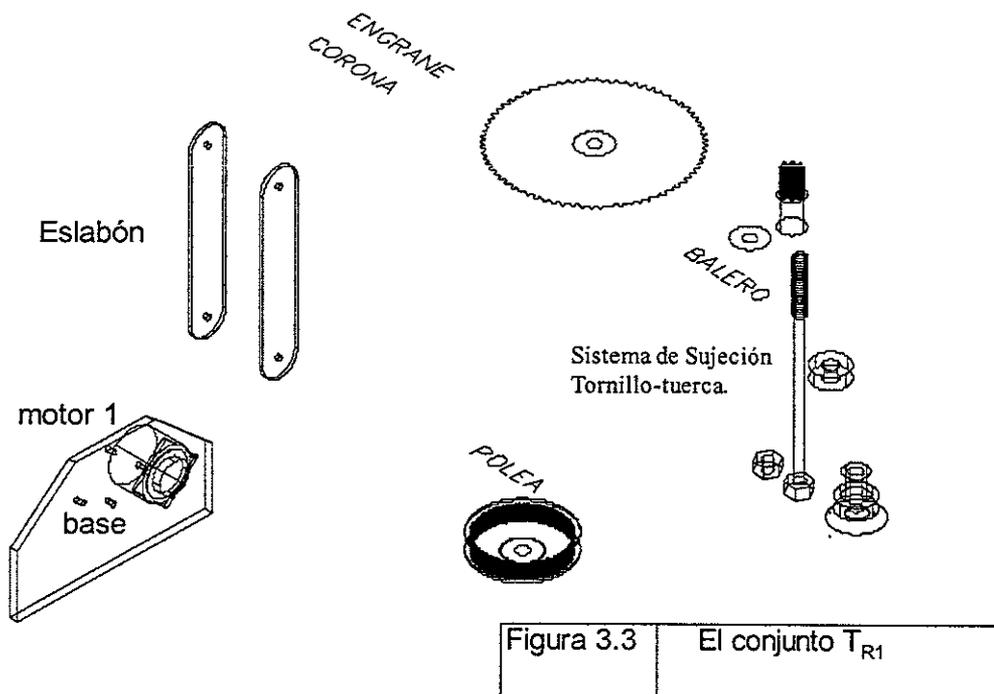


Figura 3.3 El conjunto T_{R1}

Por lo tanto definiremos al robot en este trabajo por medio de sus conjuntos de elementos a través de las transmisiones (ver figura 3.1). Esto es,

$$R_0 = T_{R1} \cup T_{R2} \quad [3.1]$$

siendo T_{R1} el conjunto asociado a la transmisión 1 y T_{R2} el conjunto de la transmisión 2. Observemos que dentro de los dos subconjuntos $T_{R1}, T_{R2} \subset R$, existen relaciones, por

ejemplo:

$$T_{R1} \cap T_{R2} = \beta_2 \quad [3.2]$$

siendo β_2 la base 2 a la cual están asociados 2 motores como se muestra en la figura 3.1 la intersección anterior indica que la base β_2 es común a ambos sistemas de transmisión. Otra relación de importancia es la siguiente:

$$T_{R1} \cap T_{R2} = A_{C1} \quad [3.3]$$

con A_{C1} el acoplador 1.

Igualmente el conjunto $T_{R1} \subset R_0$ en este trabajo será escrito de la forma siguiente:

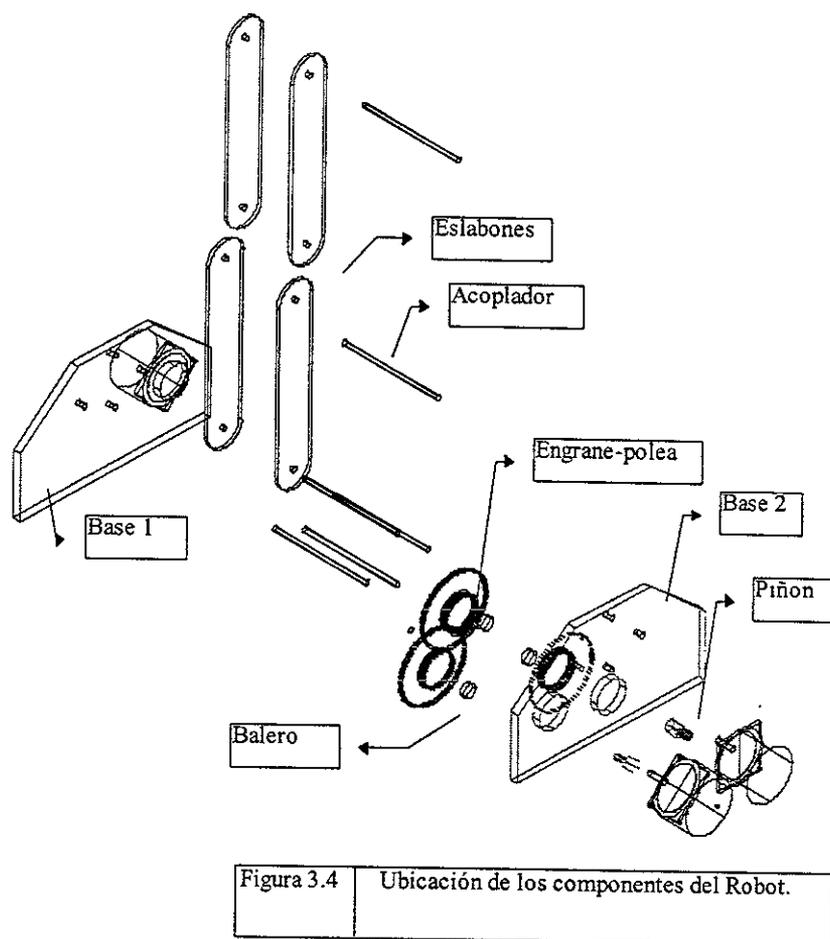
$$T_{R1} = \beta_1 \cup M_1 \cup T_E \cup T_{SP\beta} \cup A_{C1} \cup E_1 \quad [3.4]$$

siendo M_1 el motor 1, T_E el sistema de transmisión por engrane, $T_{SP\beta}$ el sistema de transmisión polea-banda, A_{C1} el acoplador y E_1 el eslabón 1.

La forma de representar al Robot mediante operadores booleanos, toma un sentido importante el Diseño Conceptual mediante el marco teórico de la teoría de conjuntos. Los operadores booleanos representan las operaciones básicas de generación de las geometrías asociadas a los elementos que componen al Robot. También los operadores booleanos son utilizados por los sistemas de diseño computarizado como un sistema CAD[6]. La parte de modelado de sólidos incluida en este software, es un herramienta de mucha utilidad sobre todo en la caracterización del Diseño Conceptual. La tabla 3.1 muestra los elementos importantes que componen al Robot(R_0).

Tabla 3.1		
Elementos que componen al (R_0).		
Elemento de trabajo.	Operación.	#
Base.	Soporte de estructura del robot.	2
Eslabones.	Cuerpos móviles.	2 (dos por cada eslabón)
Motores de pasos.	Movimiento de eslabones.	2
Sistemas de engranajes.	Transmisión de potencia.	2
Poleas dentadas.	Transmisión de potencia.	9
Acopladores.	Unión de bases.	4
Sujetadores.	Unión de eslabones.	2
Bandas dentadas.	Transmisión de potencia.	4
Baleros.	Ajuste y libre rotación.	10

La ubicación de estos elementos es mostrada en la figura 3.4.



Una vez caracterizados los elementos que componen al Robot, esto a través del Diseño Conceptual, ahora procederemos a formar las partes que componen al robot, mediante el Diseño de Forma.

3.2 Diseño de Forma

El Diseño de forma involucra las geometrías primitivas base de las cuales se formarán las geometrías finales de los elementos que componen al Robot, el Diseño de Forma caracteriza los procesos de manufactura en un sentido general, sin llegar al detalle de los mismos.

En el Diseño de Forma los materiales asociados a las partes del Robot no serán caracterizados, tampoco las dimensiones. Sin embargo las restricciones generales (2), (8) y (9) son incluidas.

Para iniciar el Diseño de Forma definiremos los siguientes conceptos los cuales serán de utilidad en el desarrollo de este trabajo.

Definición 3.2.1 Llamaremos cuerpos primitivos base aquellos cuerpos donde se realizará la forma de los elementos que componen al robot.

Definición 3.2.2 Llamaremos cuerpos primitivos de forma fija al cubo unitario, cilindro unitario, etc.

Definición 3.2.3 Los cuerpos ya formados los llamaremos en este trabajo los elementos base de forma.

3.2.1 Definición de los cuerpos primitivos base

Normalmente los cuerpos donde realizaremos la forma de los elementos que componen al robot, ya tienen una forma específica. Por ejemplo para realizar una flecha, se necesita disponer de una barra sólida, esto es, de un cilindro macizo, igualmente para formar las bases de una estructura se necesita una placa rectangular. Entonces caracterizaremos los cuerpos primitivos base por dos formas, normalizadas estandar o básicas, esto es, rectángulos y círculos si es en el plano, prismas rectangulares y cilindros si es en el espacio, ver figura 3.5.

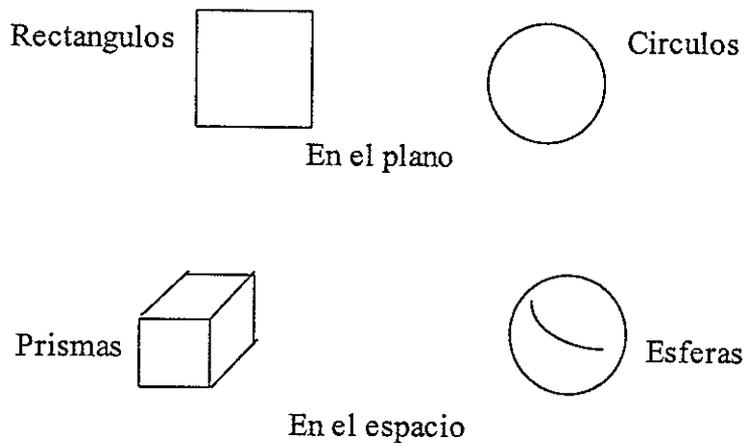


Figura 3.5.	Forma de los cuerpos primitivos base
-------------	--------------------------------------

3.2.2 Geometrías básicas de cuerpos primitivos base

Los cuerpos primitivos base se pueden caracterizar en dos grandes grupos:

- 1) De forma prisma
- 2) De forma cilíndrica.

Sea C_{PB} el conjunto de cuerpos primitivos base y $C_{PB1} \subset C_{PB}$ el subconjunto de los cuerpos de forma de paralelepípedo regular, también $C_{PB2} \subset C_{PB}$ el subconjunto de los cuerpos de forma cilíndrica mostrados en la figura 3.5, entonces los grupos C_{PB1} y C_{PB2} serán para nosotros las dos formas básicas de la materia prima en la construcción de los elementos que componen al Robot.

Los cuerpos primitivos base deberán tener una restricción importante, esto es, el volumen asociado a ellos, deberá ser mayor que los cuerpos primitivos de la forma estándar, los cuales serán definidos más adelante. Por otro lado, los cuerpos primitivos base deben cumplir con la restricción (9), en el sentido de que dichas formas deben ser encontradas

de forma rápida en el mercado.

La cardinalidad de los conjuntos C_{PB1} y C_{PB2} es caracterizada del Diseño Conceptual, esto es, se debe conocer cuantos eslabones se necesitan, cuantas bases, ejes, etc.,

Para este robot, el Diseño Conceptual indica: 2 piezas por cada eslabón, 2 bases unidas por 3 barras de acoplamiento, también se necesitan 2 barras sólidas en las articulaciones. Para un seguimiento adecuado de este trabajo en el Diseño de Forma, se caracterizarán las piezas asociadas a las bases, barras de conexión y eslabones, para posteriormente caracterizar los elementos normalizados como lo son bandas, poleas, motores, tornillos etc.

De lo anterior la cardinalidad del conjunto C_{PB1} es de 8, en tanto el conjunto C_{PB2} tiene cardinalidad de 6.

Entonces se necesitan 2 placas rectangulares de dimensión $A \times B \times C$ y 4 placas de dimensión $D \times E \times F$ ($A \times B \times C$ son especificaciones de mercado) para formar los eslabones, en tanto para formar las barras se necesitan 4 cuerpos de forma cilíndrica. (Se observa que no se incluyen las medidas).

La cardinalidad de los conjuntos C_{PB1} y C_{PB2} difiere del diseñador, por ejemplo si se tiene una placa rectangular que permitirá la formación de las bases y otra placa que permita la generación de las 6 partes que componen a los eslabones, la cardinalidad del C_{PB1} es 2, en tanto si se dispone de una sola barra para generar los acoplamientos, entonces la cardinalidad del conjunto C_{PB2} es de 1.

Sin embargo, la primera caracterización de la cardinalidad los conjuntos C_{PB1} y C_{PB2} cumple con las restricciones (2), (8) y (9), esta consideración será explicada más adelante. Ahora se procederá a caracterizar los cuerpos primitivos estandar.

3.2.3 Definición de los cuerpos primitivos unitarios estandar.

Los cuerpos primitivos estandar son aquellos que tienen una geometría también ya definida como lo son: cubos; rectángulos prismales; cilindros; cuñas; esferas; etc.,. Estos cuerpos difieren de los primitivos base por que son cuerpos unitarios² (ver figura 3.6) utilizados para formar las piezas y están normalmente asociados a los procesos de manufactura. Al igual que los cuerpos primitivos base, estos cuerpos unitarios serán caracterizados por conjuntos.

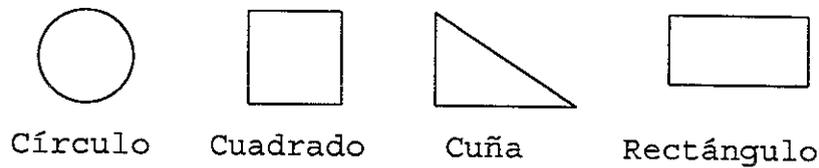


Figura 3.6	Cuerpos Primitivos.
------------	---------------------

Sea C_p el conjunto de cuerpos primitivos y $C_{p_1} \subset C_p$ el subconjunto de los cubos unitarios, también $C_{p_2} \subset C_p$ es el subconjunto de los cilindros en tanto $C_{p_3} \subset C_p$ el subconjunto de las cuñas y C_{p_4} el subconjunto de los rectángulos (o prismas rectangulares). La cardinalidad asociada a estos subconjuntos es 1 , esto es, los cuerpos primitivos en realidad no son cuerpos sino procesos de manufactura, sin embargo la cardinalidad de todo el conjunto C_p es de 4 para este caso.

Así entonces para formar una barra de unión tomaremos al elemento unitario $C_{p_{11}} \subset C_{p_1}$ y en caso de que se requieran 2 primitivas, entonces se tomarán $2C_{p_{11}} \subset C_{p_1}$, aquí el 2 no es referido a la cardinalidad del subconjunto C_{p_1} si no al número de primitivas que se

² Tomaremos cuerpos unitarios en este trabajo, sin embargo en términos reales no lo son.

necesitan.

Los cuerpos primitivos unitarios son muy importantes cuando se trata de generar geometría complicadas.

3.2.4 Operadores Boleanos asociados a los Procesos de Manufactura

Para realizar la formación de una pieza necesitamos asociar procesos de manufactura, esto es, las operaciones básicas de forma, por ejemplo para realizar una perforación se usa una máquina herramienta llamada taladro, en procesos de desvaste se usa un torno o fresadora y para procesos de corte seguetas eléctricas, entre otras máquinas.

Las operaciones de forma pueden ser caracterizadas a través de operadores booleanos, esto es, la unión, la intersección y la diferencia pueden asociarse a un proceso de manufactura [7]. Por ejemplo la unión de uno a varios cuerpos realizada por un proceso de soldadura, puede caracterizarse con el operador booleano "Unión (U) en tanto los procesos de corte con el operador diferencia (-) y con el operador intersección (\cap).

Definiremos en esta parte 2 grandes grupos de procesos de manufactura los cuales son:

Grupo 1.- Aquellos procesos que necesitan operaciones de soldadura, unión de cuerpos a presión o ensambles, etc.

Grupo 2.- Los procesos asociados a las operaciones de corte, desvaste, rectificado, fresado, taladrado, etc.

Al grupo 1 le asociaremos el operador booleano unión (U), en tanto al grupo 2 le asignaremos los operadores booleanos diferencia (-) e intersección (\cap).

Las estructuras booleanas, forman la base de los softwares de diseño de los sistemas CAD en lo que se refiere a modelado de sólidos.

Nota: “Los grupos de maquinados se refieren a caracterizar formas de cuerpos, por lo tanto no es necesario incluir el material. En base a las caracterizaciones anteriores definiremos el problema asociado al Diseño de Forma esto es:

3.2.5 Problema del Diseño de Forma [7].

“Dado un Diseño Conceptual del prototipo, el conjunto de primitivas base, el conjunto de primitivas estandar y los operadores booleanos asociados a los procesos de manufactura, encuentre las geometrías asociadas a los elementos que componen al robot. Las ecuaciones de forma y subforma tal que las restricciones generales (2),(8) y (9) sean satisfechas”.

3.2.5 Diseño de Forma de los elementos que componen al Robot “(Eslabones, barras de unión y bases)”.

En esta parte del diseño conviene agrupar los elementos ó piezas que componen al robot. Entonces sea C_E el conjunto de los elementos o cuerpos formados, también $C_{E1} \subset C_E$ es el subconjunto de las bases , $C_{E2} \subset C_E$ el subconjunto de los eslabones y $C_{E3} \subset C_E$ el subconjunto de las barras unión. La cardinalidad de los subconjuntos es caracterizada del Diseño Conceptual.

Entonces la cardinalidad del conjunto C_E es de 10 o sea 2 bases, 4 eslabones y 4 barras. Esto es:

$$C_{E1} \rightarrow 2$$

$$C_{E2} \rightarrow 4$$

$$C_{E3} \rightarrow 4$$

[3.5]

3.2.6 Diseño de Forma de las Bases.

La forma general de las bases caracterizadas en el diseño conceptual se observa en la figura 3.7.

3.2.6.1 Formación de la base 1

La forma de la base 1 es mostrada en la figura 3.7 generada de la forma siguiente:

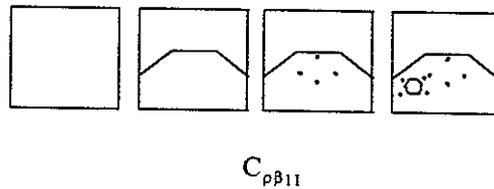


Figura 3.7.	Forma de la base 1.
-------------	---------------------

Sea $C_{PB11} \subset C_{P1}$, $C_2 \subset C_P$ y $C_{P3} \subset C_P$, entonces:

$$C_{E11} = C_{PB11} - C_{P4} - 2C_{P3} - 9C_{P2}$$

[3.6]

C_{E11} será para nosotros la forma de la base 1, se observa entonces que C_{PB11}

corresponde al cuerpo primitivo base 1 y C_{P3} a la primitiva unitaria cuña, en tanto C_{P2} es la primitiva unitaria del círculo y C_{P4} la primitiva rectángulo.

La forma de la geometría del elemento $C_{E''}$, es mostrada en la figura 3.8.

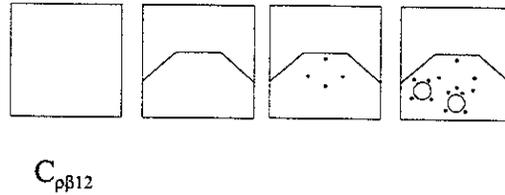


Figura 3.8 Geometría de forma del elemento C_{E11} .

Definiremos cuerpos $C_{E11}^I, C_{E11}^{II}, C_{E11}^{III}, C_{E11}^{IV}$ y C_{E11}^V corresponden a una secuencia de operaciones de forma de la placa 1 y serán caracterizados por las siguientes ecuaciones:

$$C_{P\beta1} - C_{P4} = C_{E11}^I \quad [3.7]$$

$$C_{E11}^I - 2C_{P3} = C_{E11}^{II} \quad [3.8]$$

$$C_{E11}^{II} = C_{E11}^I - 5C_{P2} \quad [3.9]$$

$$C_{E11}^{III} = C_{E11}^{II} - C_{P2}^* \quad [3.10]$$

$$C_{E11}^{IV} = C_{E11}^{III} - 4C_{P2}^{**} \quad [3.11]$$

Aquí, C_{P2}^* se refiere al agrandamiento del barreno 5 mostrado en la figura 3.9 y C_{P2}^{**} son los barrenos de sujeción del motor 1 en la forma geométrica. Al tomar las consideraciones anteriores, la forma de la ecuación 3.6 es la siguiente:

$$C_{E1} = C_{P\beta1} - C_{P4} - 2C_{P3} - 5C_{P2} - C_{P2}^* - 4C_{P2}^{**} \quad [3.12]$$

Las subformas del cuerpo $C_{E11} \subset C_E$ son mostradas en la figura 3.9.

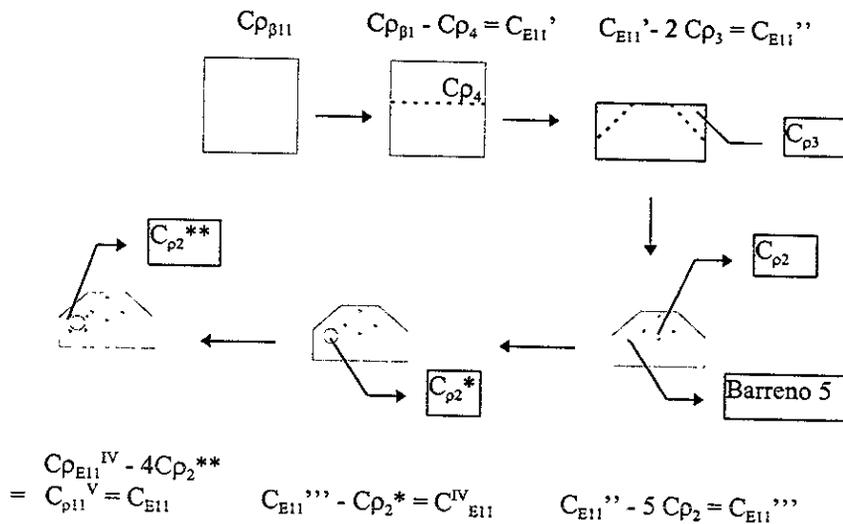


Figura 3.9 Forma geométrica.

La subforma C_{E11}^{II} es generada por un proceso de manufactura realizado por segueta electrica y caracterizado por el operador boleano diferencia (-).

También el cuerpo C_{E11}^{II} es formado por segueta, en tanto el cuerpo C_{E11}^{III} , C_{E11}^{IV} y C_{E11}^V son formados por otros maquinados que pueden ser por taladro o por fresadora, sin embargo, dichos procesos están contenidos en el grupo 2 al cual el operador boleano diferencia le es asociado. Las C_{E11}^{III} , C_{E11}^{IV} y C_{E11}^V son generadas con el conjunto de primitivas C_{PB2} , observemos que las primitivas estandar C_{P2}^* y C_{P2}^{**} corresponde a primitivas de mayor o menor tamaño que C_{P2} .

Las partes C_{E11}^I , C_{E11}^{II} , C_{E11}^{III} , C_{E11}^{IV} y C_{E11}^V se forman por una secuencia lógica de procesos de manufactura donde la restricciones (2) y (9) estan implícitas.

3.2.6.2 Formación de la base 2

En esta parte procederemos a encontrar la ecuación de forma de la base 2., mostrada en la figura 3.1, siguiendo el mismo procedimiento de la base 1.

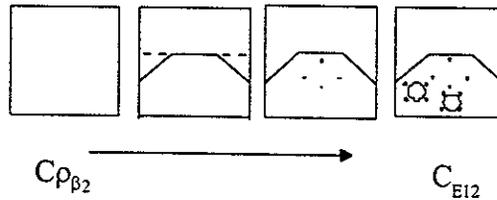


Figura 3.10 Forma del cuerpo primitivos base 2.

observemos que para formar la base 2, se necesitan los procesos de corte por segueta, taladrado y desvaste. La ecuación de forma asociada al cuerpo C_{E12} que será para nosotros la base 2 es la siguiente:

$$C_{E12} = C_{PB2} - C_{P4} - 2C_{P3} - 16C_{P2} \tag{3.13}$$

En este caso, $C_{PB2} \subset C_{PB}$, $C_{P4}, C_{P3}, C_{P2} \subset C_{CP}$.

El cuerpo C_{PB2} corresponde al cuerpo primitivo base del cual se forma el cuerpo C_{E12} . Las subformas asociadas al cuerpo C_{E12} son las siguientes:

$$C_{E12}^I = C_{PB2} - C_{P4} \tag{3.14}$$

$$C_{E12}^{II} = C_{E12}^I - 2C_{P3} \tag{3.15}$$

$$C_{E12}^{III} = C_{E12}^{II} - 6C_{P2} \tag{3.16}$$

$$C_{E12}^{IV} = C_{E12}^{III} - C_{P2}^* \tag{3.17}$$

$$C_{E12}^V = C_{E12}^{IV} - 8C_{P2}^{**} \tag{3.18}$$

Observe que la ecuación de forma se expresa de la forma siguiente:

$$C_{E12} = C_{PB2} - C_{P4} - 2C_{P3} - 6C_{P2} - 2C_{P2}^* - 8C_{P2}^{**} \quad [3.19]$$

$$\forall C_{E12} \in C_E, C_{PB12} \in C_{PBI}, C_{P2}, C_{P3}, C_{P4} \in C_P$$

La secuencia de forma se describe a continuación:

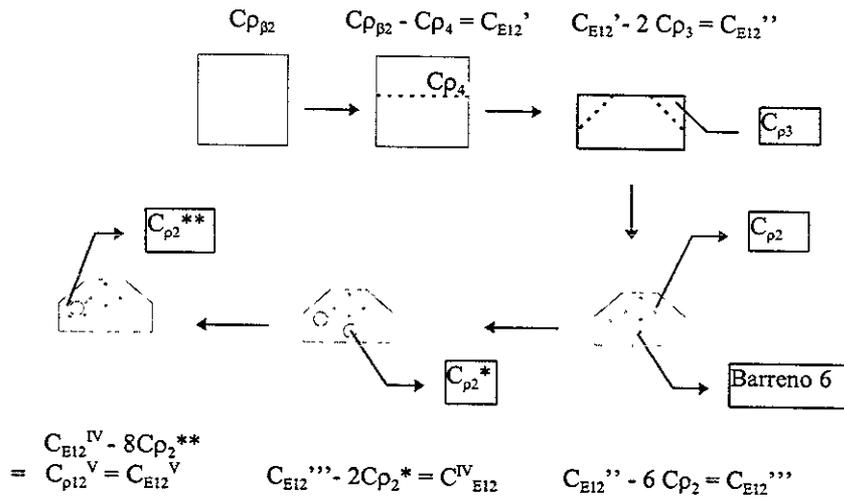


Figura 3.11 Forma geométrica.

Notemos que la placa 2 contiene un motor más, esto es debido a que se requiere montar otro eslabón más en trabajos futuros.

Hasta este momento se han caracterizado las formas de los cuerpos asociados a las bases, también con la información anterior es posible caracterizar las secuencias de operaciones para formación de las piezas, esto es:

3.2.7 Diseño de Forma de los Eslabones que componen al Robot.

Del diseño conceptual se observa que existe una forma básica de los eslabones, ya que son diseñados de manera que las dimensiones del eslabón 1 sean igual a las dimensiones del eslabón 2

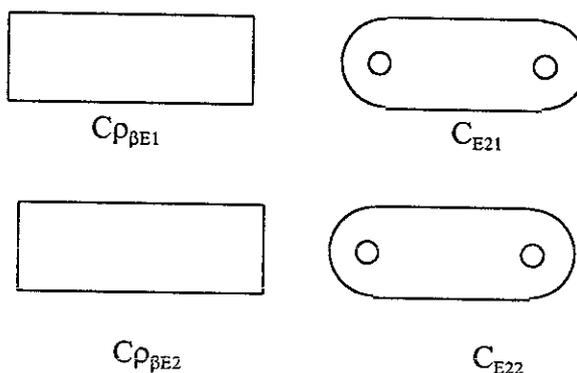


Figura 3.12 Formas básicas de los eslabones.

Antes de iniciar a construir las ecuaciones de forma se procederá a integrar un cuerpo primitivo al conjunto C_p , esto es, sea $C_{P5} \subset C_p$ un cuerpo primitivo que tiene la forma siguiente:

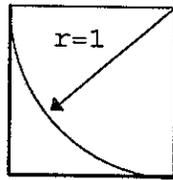


Figura 3.13	Cuerpo primitivo en forma de cuña.
-------------	------------------------------------

Esto es, un cuerpo se forma de cuña, cuya curva asociada a los puntos extremos es una circunferencia de radio unitario:

“ Observación: Los cuerpos primitivos tienen la propiedad de escalamiento” .

Bajo esta premisa se procederá a construir las ecuaciones de forma de los eslabones, entonces para el eslabón 1 (C_{E21}) se tiene que:

$$C_{E21} = C_{P\beta E1} - C_{P4} - 2C_{P2} - 4C_{P3}^* \quad [3.20]$$

Se tiene que las ecuaciones de subforma son:

$$C_{E21}' = C_{P\beta E1} - C_{P4} \quad [3.21]$$

$$C_{E21}'' = C_{E21}' - 2C_{P2} \quad [3.22]$$

$$C_{E21}''' = C_{E21}'' - 4C_{P3}^* \quad [3.23]$$

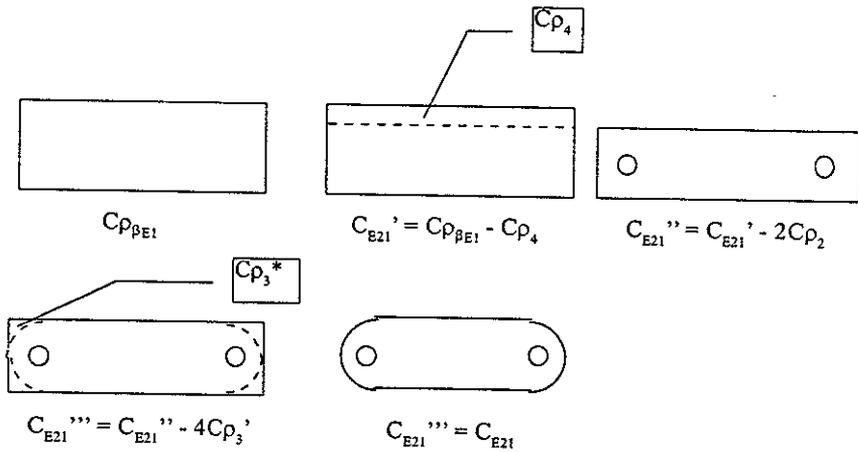


Figura 3.14 Formas básicas de los eslabones.

3.2.8 Diseño de forma de los Ejes de Acoplamiento.

Para diseñar la forma de las barras de acoplamiento se definirá lo siguiente:

Condición 1.- Los primitivos base son de volúmen mayor que los primitivos, en términos de cilindros se dirá que el diámetro de los cilindros primitivos base son mayores que los primitivos.

Condición 2.- El diámetro asociado tanto a los primitivos base como a los primitivos es el mismo.

También la longitud de los cilindros primitivos base es tal que existan 3 y que sean suficientes para generar las 5 barras de acoplamiento. Entonces la cardinalidad del subconjunto $C_{PB3} \subset C_{PB}$ es de 3. La ecuación de forma para la condición 1 es la siguiente:

$$2C_{E3} = C_{PB31} - C_{P6}^* - C_{P2} \quad [3.24]$$

y para la condición 2 la ecuación es:

$$2C_{E3} = C_{PB31} - C_{P2} \quad [3.25]$$

La primitiva $C_{P6}^* \subset C_p$ es un cilindro unitario hueco al cual se le asocia un proceso de desvaste por torno y la primitiva $C_{P2} \subset C_p$ está asociada a un proceso de corte en donde el cilindro asociado es tan pequeño como el espesor de una sierra, ver figura 3.15

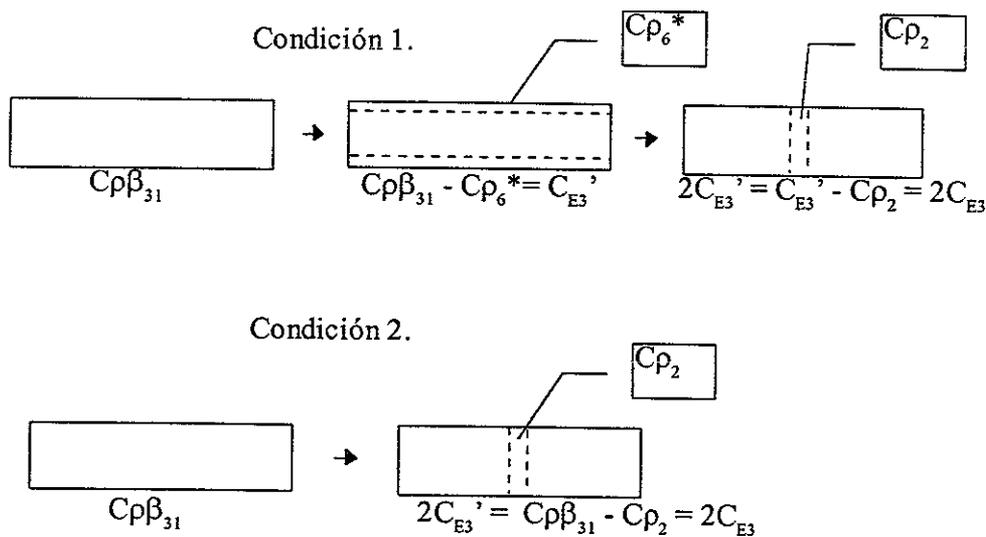


Figura 3.15 | Secuencia de forma.

Hasta este momento se ha realizado el diseño de forma de las geometrías de las bases, eslabones y ejes de acoplamiento. En base al diseño conceptual la tabla Tabla 3.2. Muestra el número total de partes formadas.

Descripción	Cuerpo formado	N_0	N_0 de secuencias de operaciones
Base 1	C_{E11}	1	6
Base 2	C_{E12}	1	6
Eslabones del tipo 1	C_{E21}	4	2
Eslabones del tipo 1	C_{E22}	2	2
Ejes (condición 2)	C_{E31}	6	1

En el diseño de forma ya es posible obtener el volumen del cuerpo primitivo base y su forma de tal manera que en base al diseño conceptual las posibles piezas que se requieren en el robot otorgan información sobre la forma de las piezas primitivas base.

3.2.9 Consideraciones de forma de los elementos normalizados.

Un diseño conceptual debe marcar si no de una forma rigurosa si de alguna manera funcional, los elementos que servirán de sujeción, de transmisión, de ajuste, etc. Por ejemplo un elemento de sujeción como lo es un tornillo con tuerca ya han sido formados mediante procesos de manufactura, estos elementos fueron procesados siguiendo alguna ecuación de forma. Si se caracterizarán los procesos de manufactura asociados a los elementos normalizados se tendría que enunciar el problema inverso relacionado con el diseño de forma.

3.2.10 El problema inverso del diseño de forma de los elementos normalizados [7].

“Dada la geometría de un cuerpo normalizado o terminado(ingeniería de reversa), las

primitivas base, las primitivas y los operadores booleanos, encuentre las ecuaciones de forma asociadas a la geometría inicial, las secuencias de operaciones, ó de procesos de manufactura y las máquinas herramientas asociadas tal que se cumplan las restricciones siguientes:

1. *Las máquinas herramientas deberán contar con cabezal sencillo de operación.*
2. *Utilizar el operador “unión” si la pieza normalizada(ó terminada) incluye otras piezas de operación”.*

En este caso la pieza normalizada es el sistema de sujeción tornillo – tuerca. Para realizar el problema inverso del diseño de forma, es conveniente primero dividir en dos conjuntos las operaciones tanto de la tuerca como del tornillo. Esto es,

Sea T_N la pieza normalizada tornillo y R_N la tuerca, entonces

$$T_N \cap R_N = S \quad [3.26]$$

siendo S el conjunto completo asociado al sistema de sujeción. En la figura 3.16 se muestra a la pieza normalizada de sujeción.



Figura 3.16	Pieza normalizada de sujeción.
-------------	--------------------------------

3.2.11 Descripción del conjunto S.

El sujetador está compuesto de un tornillo de cabeza exagonal con rosca parcial en el vástago, y es completado por una tuerca exagonal con orificio roscado de las mismas características de la rosca del tornillo. Así entonces empezaremos por caracterizar primero a la tuerca.

El cuerpo primitivo base puede ser de dos formas siguientes:

- 1) Una barra cuadrada ó cilíndrica.
- 2) Una barra exagonal con dimensiones normalizadas. (ver figura 3.16).

Observemos entonces que si se elige la primera opción, entonces debemos formar la cabeza exagonal, esto involucra un maquinado y por lo tanto desde el punto de vista de operaciones las barras cuadradas o cilíndricas no son la mejor de opción (siempre y cuando exista una barra exagonal normalizada con las dimensiones requeridas, si esto no es posible, entonces automáticamente se debe optar por la primera opción).

Consideremos que en el mercado existen barras exagonales que permitan no tomar la primera opción, entonces procederemos a encontrar las ecuaciones de forma del sistema de sujeción.

Con las consideraciones anteriores se ha caracterizado la geometría del cuerpo primitivo base. Entonces sea P_{BN} el conjunto de primitivas bases con $P_{BN1} \subset P_{BN}$ el cuerpo exagonal, en este caso la cardinalidad del conjunto P_{BN} es de 2, esto es, se utilizará la forma del cuerpo primitivo base para caracterizar el tornillo.

Para continuar con el diseño de forma se caracterizará un conjunto C_{EE} llamado cuerpos especiales, en el están incluidos las roscas, las formas de los engranes, de las levas, etc.

Entonces los maquinados asociados a la forma del cuerpo R_N serán:

- 1) Corte por barra de torno.
- 2) Taladrado.
- 3) Machueleado o roscado por torno.

De lo anterior es posible interpretar la forma geométrica de cada maquinado sobre el cuerpo primitivo base $C_{PBN1} \subset C_{PBN}$. Sin embargo, antes de describir la forma de los maquinados asociados al cuerpo C_{PBN} tendremos que asociar las primitivas y los cuerpos especiales. Esto es,

1	Corte por barra	→	C_{P2}^*
2	Taladrado	→	C_{P2}^{**}
3	Machueleado	→	$C_{EE1} \in C_{EE}$

También las operaciones que se realizarán en el cuerpo primitivo base pertenecen al grupo 2 anteriormente el cual tiene asociado el operador "diferencia"(-).

Supongamos una secuencia de operaciones $1 \rightarrow 2 \rightarrow 3$ como la anteriormente descrita, entonces la ecuación de forma sobre el cuerpo C_{PBN1} asociada a la secuencia 1 es dada por:

$$R_N = C_{PBN1} - C_{P2}^* - C_{P2}^{**} - C_{EE1} \quad [3.27]$$

$$\forall C_{PBN1} \in C_{PB}, C_{P2}^* \in C_P, C_{P2}^{**} \in C_P, C_{EE1} \in C_{EE}$$

Aquí C_{P2}^{\cdot} es una primitiva de corte asociada a la dimensión de una barra de corte, en tanto la primitiva C_{P2}^{**} esta asociada a la operación de taladrado. También C_{P2}^{\cdot} existe siempre que:

$$V_{CBN1} > V_{CP2}^{\cdot} \quad [3.28]$$

siendo V_{PBN1} el volumen del cuerpo primitivo base.

La ecuación [3.30] en términos de los procesos asociados toma la forma siguiente:

$$R_N = R_N^{///} = R_N^{//} - C_{EE1} \quad [3.29]$$

donde

$$R_N^{//} = R_N^{\cdot} - C_{P2}^{**} \quad [3.30]$$

$$R_N^{\cdot} = C_{BN1} - C_{P2}^{\cdot} \quad [3.31]$$

De lo anterior se deduce la ecuación general de forma de la secuencia 1. Esto es,

$$R_N = \{ [C_{PBN1} - C_{P2}^{\cdot}] - C_{P2}^{**} \} - C_{EE1} \quad [3.32]$$

Esto es, los paréntesis de la ecuación determinan cual es la secuencia de proceso. Por ejemplo, primero se corta la barra y se le asocia la letra A, después se barrena → B y finalmente se hace la rosca → C, entonces [3.31] toma la forma siguiente:

$$R_N = (A-B) - C \quad [3.33]$$

y es llamada la ecuación general de proceso para la secuencia 1. La forma geométrica del diseño de forma de la tuerca se muestra en la fig. 3.17.

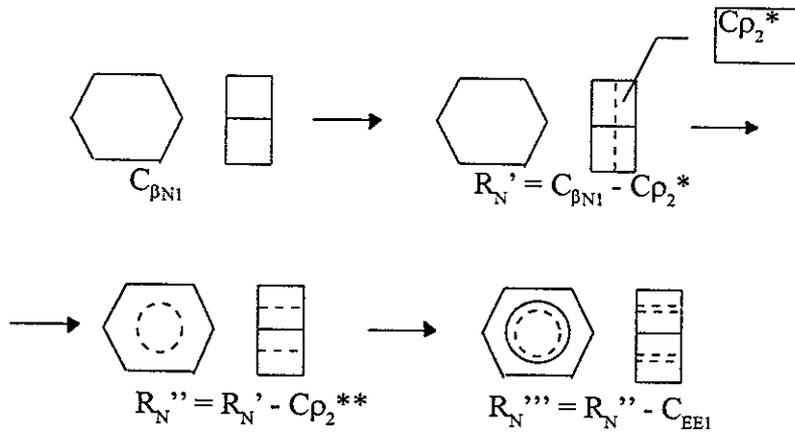


Figura 3.17 | Diseño de forma de la tuerca.

Igualmente se procederá a diseñar la forma del tornillo ahora sobre el cuerpo primitivo base C_{PBN2} .

De la configuración del tornillo vista en la fig. 3.11. Se observa que existe los procesos siguientes:

- 1) Corte por barra de corte.
- 2) Desvaste por torno
- 3) Rosca por terraja ó roscada por torno.

Este grupo de operaciones pertenece al grupo 2 el cual tiene asociado el operador boleano (-), también se tomará la secuencia 1, 2, 3.

Los cuerpos primitivos asociados a los procesos de manufactura sobre el cuerpo C_{PBN2} son:

- 1) Corte $\rightarrow C_{P2}^{**}$

2) Desvaste $\rightarrow C_{P2}^{**}$

3) Roscado $\rightarrow C_{EE2}$

En base a lo anterior, la ecuación de forma para la secuencia 1 es la siguiente:

$$T_N = C_{PBN2} - C_{P2}^* - C_{P2}^{**} - C_{EE2} \quad [3.35]$$

$$\forall C_{PBN2} \in C_{PB}, C_{P2}^* \in C_P, C_{P2}^{**} \in C_P, C_{EE2} \in C_{EE}$$

La ecuación [3.35] en términos de procesos es :

$$T_N = T_N^{III} = T_N^{II} - C_{EE2} \quad [3.36]$$

Aquí :

$$T_N^{II} = T_N^I - C_{P2}^{**} \quad [3.37]$$

$$T_N^I = C_{PBN2} - C_{P2}^* \quad [3.38]$$

La ecuación general de forma para la secuencia 1 es la siguiente:

$$T_N = \{ [C_{PBN2} - C_{P2}^*] - C_{P2}^{**} \} - C_{EE2} \quad [3.39]$$

Si:

$$C_{PBN2} - C_{P2}^* \rightarrow D$$

$$C_{P2}^{**} \rightarrow E$$

$$C_{EE2} \rightarrow F$$

Entonces [3.39] toma la forma siguiente:

$$T_N = (D - E) - F. \quad [3.40]$$

La forma geométrica de las ecuaciones de forma es mostrada en la fig. 3.18.

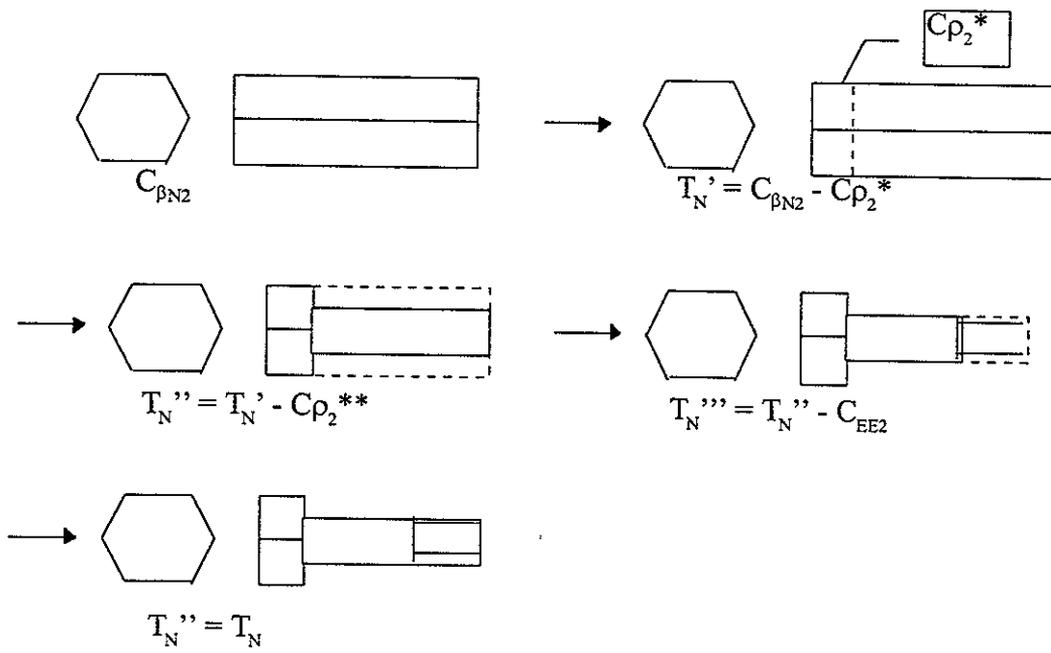


Figura 3.18 Forma geométrica de la ecuación de forma.

Por otro lado el conjunto S es dado por:

$$S = T_N \cup R_N = ((A-B)-C) \cup ((D-E)-F) \quad [3.41]$$

y representa la ecuación general de la sujeción completa en términos de maquinados y ensamble.

Las conclusiones de los anteriormente se descrito pueden conducir a la postulación de un axioma. Esto es;

3.2.12 Axioma 1 del Diseño de Forma[7].

“Sea n un número finito de operaciones asociadas a procesos de manufactura, sea también C_{PB} un cuerpo primitivo base, entonces existirán $n!$ ecuaciones de forma sobre el

cuerpo primitivo base de forma única, siendo A una pieza formada". También el número de secuencias es tal que :

$$N_s = E_c = n! \quad [3.42]$$

Aquí E_c el número de ecuaciones de forma y N_s el número de secuencia de operaciones.

3.2.13 Colorario 1.

" Existirán n subformas de A siempre que cada subforma le sea asociado uno y solo un maquinado con la condición de que la máquina herramienta asociada a las operaciones ejecute una forma y solo una".

Entonces :

$$N_{TS} = n - 1 \quad [3.43]$$

Siendo N_{TS} el número total de subformas de A.

3.2.14 Consideraciones sobre las ecuaciones de forma.

En esta parte, caracterizaremos algunas consideraciones importantes sobre las ecuaciones de forma.

Notemos que en base a los procesos de manufactura asociados a las ecuaciones de forma existirán en el caso de que $n=3$ (Axioma 1) 6 diferentes ecuaciones para cualquier proceso (con tres operaciones básicas), sin embargo existirá una y solo una la cual deberá ser implementada tomando consideraciones de maquinaria, herramental y

distribución de planta. Por otro lado las ecuaciones de forma no son asociativas ni conmutativas debido a que dependen de la secuencia de operaciones con la cual determinada pieza es manufacturada.

En base a las consideraciones anteriores reescribiremos las ecuaciones generales de forma de la base 1 y 2, así como la del eslabón 1. Esto es;

$$C_{E1} = [[[[C_{\rho_{\beta 1}} - C_{\rho_4}] - 2 C_{\rho_3}] - 5 C_{\rho_2}] - C_{\rho_2}] - 4 C_{\rho_2}^{**} \quad [3.44]$$

$$C_{E2} = [[[[C_{\rho_{\beta 2}} - C_{\rho_4}] - 2 C_{\rho_3}] - 6 C_{\rho_2}] - 2 C_{\rho_2}^*] - 8 C_{\rho_2}^{**} \quad [3.45]$$

$$C_{E21} = [[C_{\rho_{\beta E1}} - C_{\rho_4}] - 2 C_{\rho_2}] - 4 C_{\rho_3}^* \quad [3.46]$$

Finalmente consideraremos que las demás piezas que conforman al robot son manufacturadas siguiendo las ecuaciones de forma propias de cada pieza.

CAPÍTULO IV

4.- CONTROL

Hasta ahora se han descrito en los capítulos I y II los principios que rigen la cinemática de los robots-manipuladores en este caso uno de dos grados de libertad.

En el presente capítulo se pretende caracterizar el funcionamiento de un dispositivo electromecánico llamado Motor de paso a paso (PAP) * que es el actuador que inducirá el movimiento a los eslabones del robot-manipulador modelado en el capítulo II.

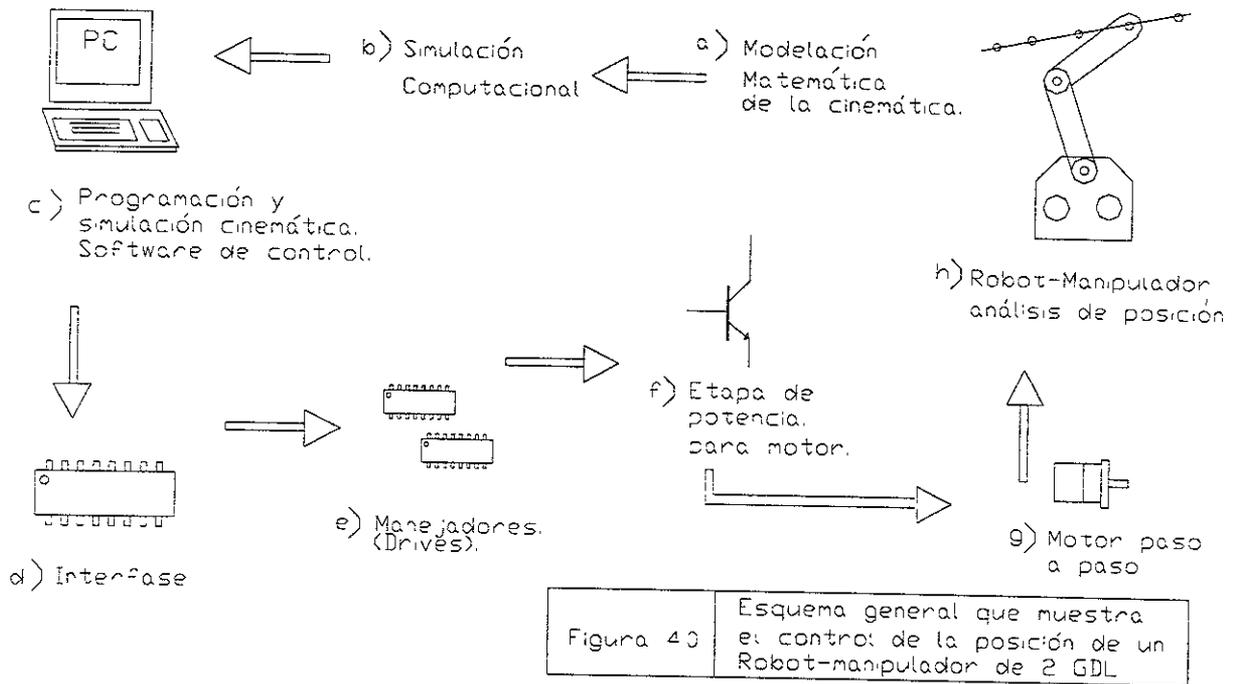
Se efectuará un análisis conceptual del sistema electromecánico a través de un diagrama de bloques general del prototipo mostrado en la figura 4.0 con el propósito de integrar en forma clara y ordenada la manera de como se realizará el control de la operación del prototipo esto en lo que se refiere a la parte de posicionamiento expuesto como objetivo en este trabajo.

En la figura 4.0 se muestra un diagrama de bloques el cual representa las etapas desde el modelo matemático hasta el control cada uno con un inciso, se realizará un análisis detallado de cada etapa, esto es; desde el principio de operación y funcionamiento hasta la aplicación integrada.

Cabe aclarar que las partes que corresponden a las etapas a) y b) de la figura 4.0 ya han sido desarrolladas en los capítulos anteriores.

ESTA TESIS NO DEBE

* De aquí en adelante se referirá al Motor de paso a paso (PAP) como motor.



4.1 - Aspectos electromecánicos de los motores paso a paso.

El inciso g) de la figura 4.0 muestra esquemáticamente un motor que convierte la información digital en un giro proporcional del eje del motor. Este dispositivo permite controlar el ángulo de giro del motor en forma discreta por medio de pulsos eléctricos, estos pulsos son enviados desde una PC por medio de un software de control.

Desde su aparición [10] los motores han aportado mejoras en los aspectos de precisión y rápido posicionamiento en máquinas, tales como; controladores de discos magnéticos; impresoras; máquinas de escribir; graficadores; robótica; control de flujo de combustible; tornos y fresadoras de control numérico entre otras aplicaciones.

Además en los sistemas mecánicos ya sea de cadena abierta o cerrada el motor facilita un mejor posicionamiento, es confiable y preciso.

El eje del motor gira un ángulo determinado llamado ángulo de paso por cada pulso de control que recibe, el ángulo de paso puede ser desde 1.8° hasta 15° grados

dependiendo del tipo de motor. Los pulsos pueden ser generados por un oscilador de frecuencia ajustable o de algún otro tipo de fuente que en este caso será el puerto paralelo de 8 bits de la computadora.

Las ventajas de los motores con respecto a los motores de corriente directa se observan en la tabla 4.1.

Tabla 4.1	
Características	
Motores de paso a paso.	Motores de corriente continua
<p>MECÁNICAS</p> <ol style="list-style-type: none"> 1.- Buen Torque. 2.- Poca velocidad. 3.- Mucha precisión. 4.- Robustos. <p>ELÉCTRICAS</p> <ol style="list-style-type: none"> 1.- Mucha corriente de operación. <p>CONTROL</p> <ol style="list-style-type: none"> 1.- Complejo. <p>COSTO</p> <ol style="list-style-type: none"> 1.- Alto. 	<p>MECÁNICAS</p> <ol style="list-style-type: none"> 1.- Bajo Torque. 2.- Mucha velocidad. 3.- Poca precisión. 4.- Ligeros. <p>ELÉCTRICAS</p> <ol style="list-style-type: none"> 1.- Poca corriente de operación. <p>CONTROL</p> <ol style="list-style-type: none"> 1.- Poco complejo. <p>COSTO</p> <ol style="list-style-type: none"> 1.- Bajo.

4.1.1 Principios de operación de los motores paso a paso.

El correcto funcionamiento de los motores se logra mediante el cambio sucesivo de la dirección del flujo de la corriente en las bobinas del estator (ver figura 4.1), y se debe realizar en una secuencia adecuada de tal manera que se obtenga un campo giratorio aparente.

El campo fijo del rotor sigue al campo giratorio aparente según la ley de atracción y repulsión de los polos magnéticos, el sentido de giro del rotor está determinado por la secuencia de conmutación (switcheo) en las bobinas del motor. La velocidad de la secuencia de conmutación de las bobinas del estator determina la velocidad del campo giratorio aparente y éste a su vez determina la velocidad del rotor.

En la figura 4.1a se muestra la conexión interna de un motor sin derivación central, esto es; se tiene un núcleo y una bobina con sus dos polos (sur y norte), en cambio en la figura 4.1b se muestra la conexión interna de un motor con derivación central se representa por la terminal b del estator y permite tener como consecuencia dos bobinas en un núcleo.

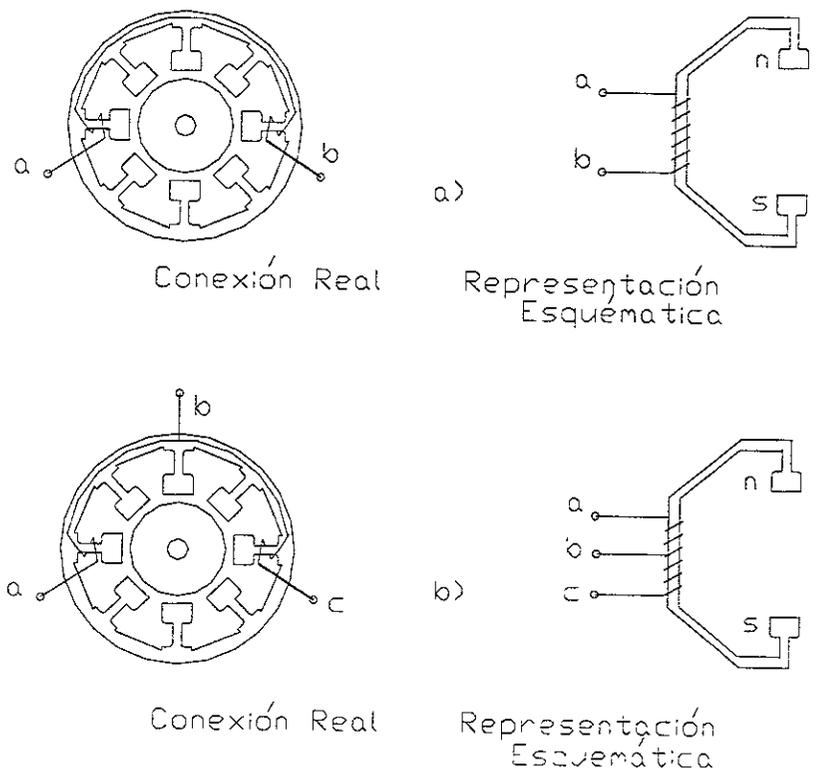


Figura 41	Conexión interna de las bobinas de motores paso a paso (PAP)
-----------	--------------------------------------------------------------

Existen dos métodos para cambiar la polaridad de los campos del estator para generar el campo giratorio aparente, estos son:

1.- Forma Unipolar.- *Para los motores que cuentan con dos bobinas en cada núcleo magnético.* Esto es, cada una de las bobinas producirá una polaridad del campo, (motor Unipolar), por lo que bastará conectar una y desconectar la otra para producir el cambio de polaridad del campo del estator.

2.- Forma Bipolar.- *Para los motores que cuentan con una bobina por núcleo magnético.* Cada una de las bobinas producirá ambas polaridades del núcleo magnético, por lo que es necesario conmutar las terminales de alimentación cada vez que se requiera cambiar la polaridad del campo del estator.

4.1.2 Control para motores paso a paso en forma unipolar.

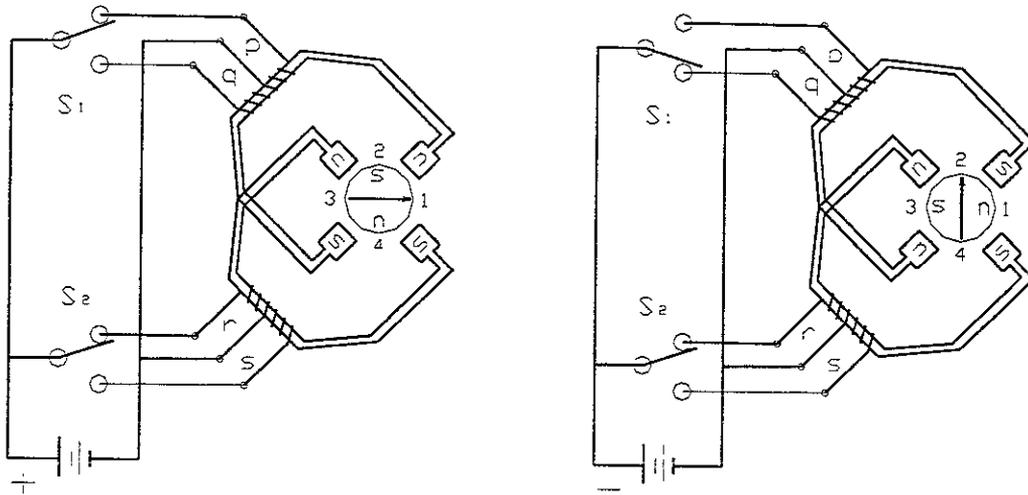
El estator es el dispositivo en el cual se genera el campo giratorio aparente, cuenta con una derivación central permitiendo tener dos bobinas con un punto común a la alimentación. Con esta derivación se puede controlar la dirección de la corriente a través de interruptores eléctricos (conmutadores).

Con la conexión y desconexión adecuada de la mitades de las bobinas de los núcleos magnéticos del estator, se logrará cambiar la polaridad de los campos generandose así el campo giratorio aparente.

4.1.3 Motor paso a paso de 2 núcleos magnéticos y 4 fases.

La figura 4.2 muestra un motor de 4 fases p, q, r, s, donde p y r se encuentran energizadas y el rotor toma la posición indicada (Paso "0"). Sí el interruptor s_2 conmuta, entonces el rotor habrá girado 90° grados dando origen al (Paso "1"). De

aquí que los interruptores controlados adecuadamente generarán el campo giratorio aparente.



a) Paso "0".

b) Paso "1".

Figura 4.2	Motor paso a paso (PAP) 2 Núcleos, 4 Fases.
------------	------------------------------------------------

4.1.4 Motor paso a paso de 4 núcleos magnéticos y 8 fases.

Los motores de 8 fases como el que se representa en la figura 4.3 funciona similarmente como el motor de 4 fases. En este motor el ángulo de giro por cada paso es la mitad del anterior debido a que cuenta con el doble de polos magnéticos. Con esta observación se puede decir que el ángulo de giro depende de el número de polos y fases con que cuenta el motor (ver ecuación 4.1).

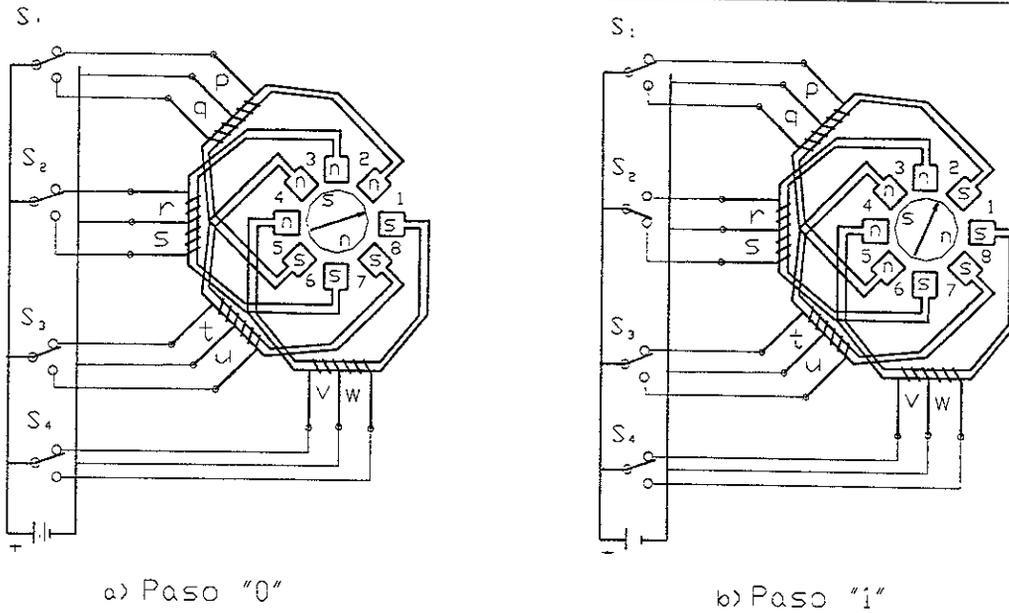
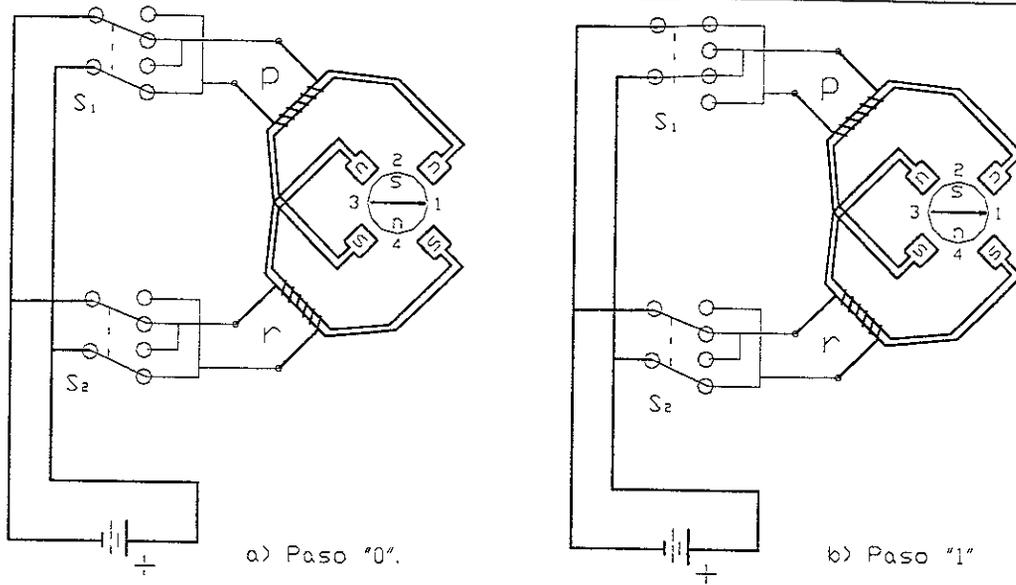


Figura 4.3	Motor paso a paso de 4 Núcleos 8 Fases.
------------	--------------------------------------------

4.1.5 Control para motores paso a paso en forma bipolar.

Las bobinas del estator de un motor bipolar no cuentan con derivación central, entonces para cambiar la polaridad del campo es necesario cambiar la dirección de la corriente que fluye por cada bobina. Para lograr esto, es necesario conmutar ambas terminales de alimentación.

En la figura 4.4a se observa que en el (Paso "0") los conmutadores s1, s2, s3, s4 se encuentran en un estado definido inicial, y para llegar a la posición de (Paso "1") el conmutador s2 cambio su posición (ver figura 4.4b) permitiendo crear el principio de funcionamiento en que se basa el campo giratorio aparente, el cual es aplicado tanto al control unipolar como al control bipolar. Se observa en cambio que lo único que varia en el control bipolar con respecto al control unipolar es que en el bipolar no existe la derivación en las bobinas del motor.

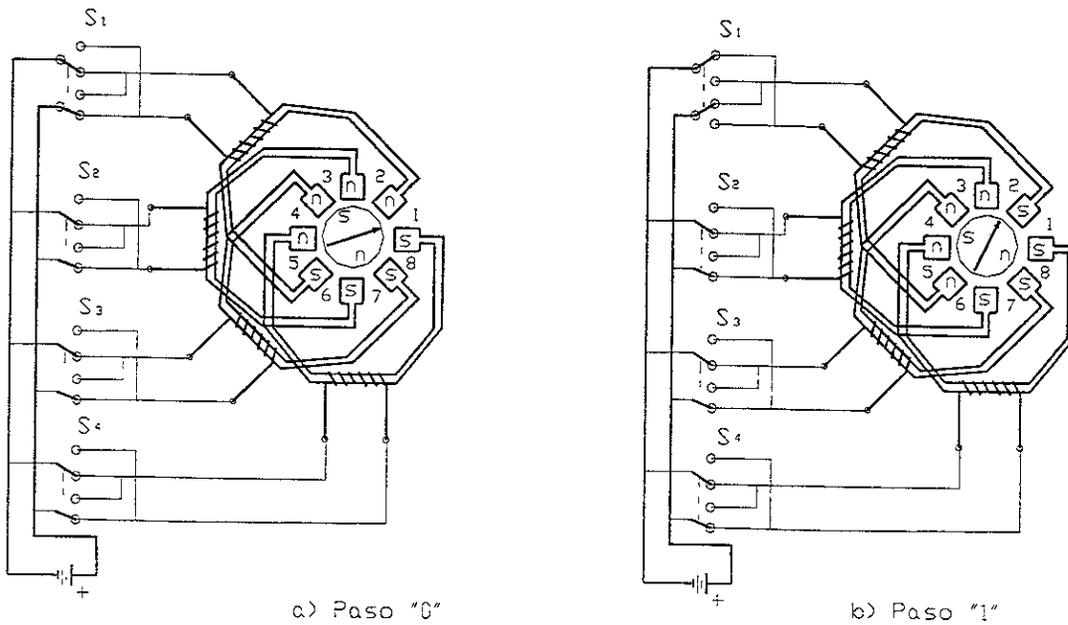


a) Paso "0".

b) Paso "1".

Figura 44 Motor paso a paso (PAP) Bipolar 2 núcleos y 4 fases

En la figura 4.5 se puede observar un motor de 4 núcleos y 8 fases con control bipolar en donde las bobinas no cuentan con derivación central.



a) Paso "0"

b) Paso "1"

Figura 45 Motor paso a paso Bipolar 4 Núcleos 8 Fases

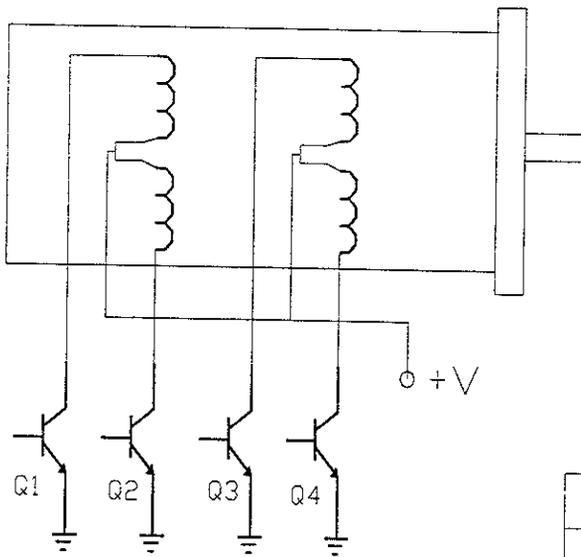
4.1.5 Configuraciones prácticas y secuencias de operación para los motores paso a paso.

Existe una infinidad de configuraciones para mover los motores, pero sólo cinco son básicas y las demás son combinaciones o derivaciones de algunas de ellas.

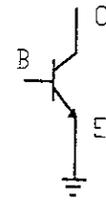
Las conexiones dependen en gran medida del tipo de motor con el cuál se quiere trabajar. Por razones de ilustración se usarán símbolos de transistores como interruptores electrónicos, igualmente se ilustrará con tablas las secuencias de interrupción de los transistores para generar el campo giratorio aparente.

Los transistores en estado de **Corte** o no conducción serán definidos como **OFF**, y los que estén en estado de **Saturación** o conducción serán definidos como **ON**. Por otro lado, la secuencia de operación se debe realizar a partir del (Paso "1") hasta el (Paso "4") en forma secuencial. Si se quiere hacer girar el eje del motor en sentido contrario entonces se empezará por realizar la secuencia desde el (Paso "4") hasta el (Paso "1").

En la figura 4.6a se puede observar la conexión unipolar para motores de 4 fases, también la forma de como se conectan a la alimentación positiva las derivaciones de las 2 bobinas, mientras que las terminales de salida pasan a través de un juego de transistores que funcionarán como interruptores electrónicos, ya sea en estado de corte o saturación. Es importante hacer notar también que en la tabla de la figura 4.6c se enlista la secuencia correcta de operación de los transistores para generar el campo giratorio aparente.



a) Diagrama de conexiones.



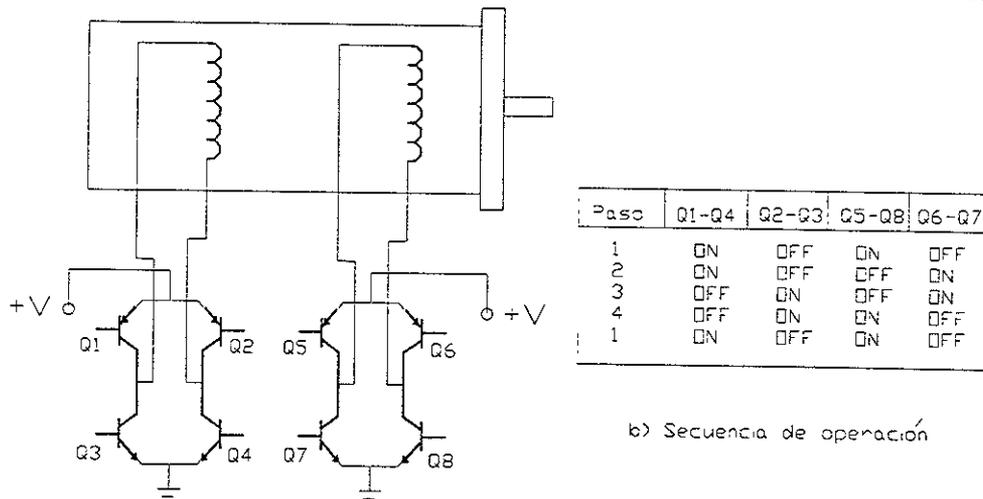
b) Símbolo de transistor.

Paso	Q1	Q2	Q3	Q4
1	ON	OFF	ON	OFF
2	ON	OFF	OFF	ON
3	OFF	ON	OFF	ON
4	OFF	ON	ON	OFF
:	ON	OFF	ON	OFF

c) Tabla de secuencia de operación.

Figura 4.6	Conexión unipolar para Motor paso a paso (PAP) 4 Fases.
------------	---------------------------------------------------------

En la figura 4.7 se muestra una conexión bipolar para un motor de 2 fases. En esta configuración dos de los interruptores electrónicos funcionan al mismo tiempo con el fin de cambiar el sentido de flujo de la corriente que fluye por las bobinas. También se puede observar la secuencia de operación de interruptores la cuál se realiza en pares de dos transistores.



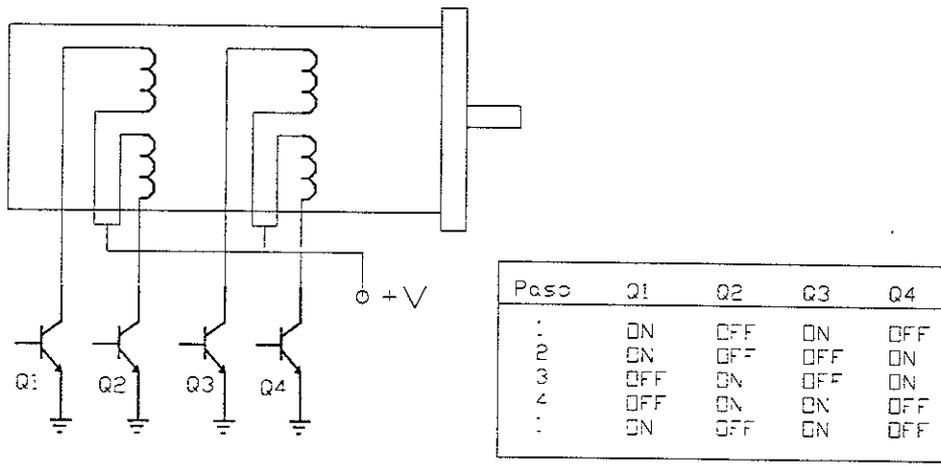
Paso	Q1-Q4	Q2-Q3	Q5-Q8	Q6-Q7
1	ON	OFF	ON	OFF
2	ON	OFF	OFF	ON
3	OFF	ON	OFF	ON
4	OFF	ON	ON	OFF
1	ON	OFF	ON	OFF

a) Diagrama de conexiones

b) Secuencia de operación

Figura 47 Conexión Bipolar para Motores paso a paso (PAP) 2 Fases

En la figura 4.8 se observa la conexión unipolar para un motor de 4 fases, y representa la forma más simple de conectar un motor, ya que sólo se requieren 4 interruptores electrónicos, y los otros cuatro extremos van conectados directamente a la alimentación positiva. La secuencia de operación de los interruptores es la misma para la conexión unipolar antes vista.



Paso	Q1	Q2	Q3	Q4
1	ON	OFF	ON	OFF
2	ON	OFF	OFF	ON
3	OFF	ON	OFF	ON
4	OFF	ON	ON	OFF
1	ON	OFF	ON	OFF

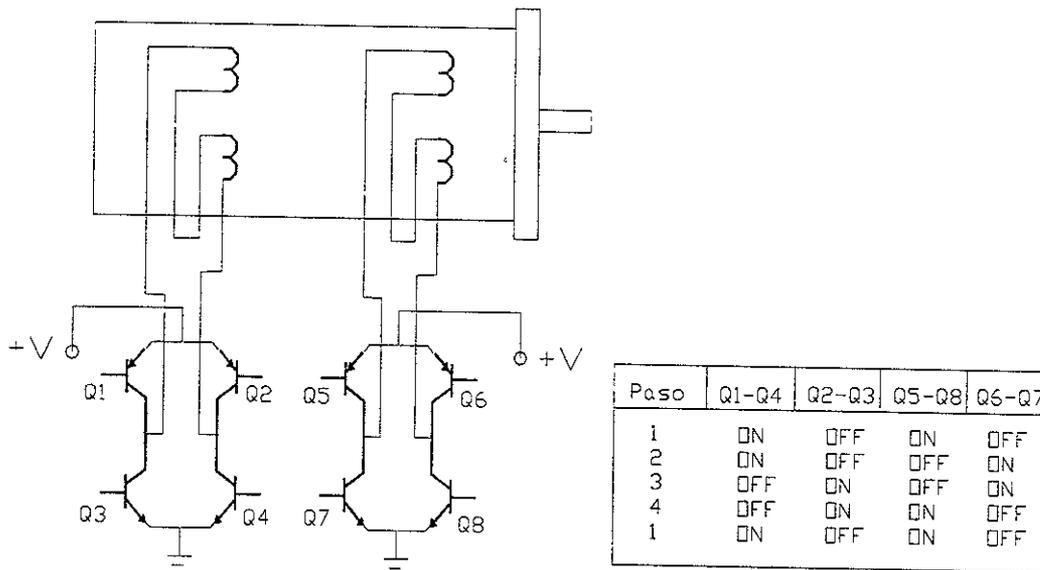
a) Diagrama de conexiones

b) Secuencia de operación

Figura 48 Conexión Unipolar para Motor paso a paso (PAP) 4 Fases

La conexión bipolar serie para un motor de 4 fases se puede observar en la figura 4.9, a pesar de que hay un número mayor de fases en el motor el número de interruptores electrónicos requeridos es el mismo para el que se requiere en un motor de 2 fases conectado en forma bipolar.

En esta configuración las bobinas se encuentran conectadas en serie, lo que permite que la corriente fluya en grupos de dos bobinas fluyendo por cada una de ellas en el momento de estar energizadas.

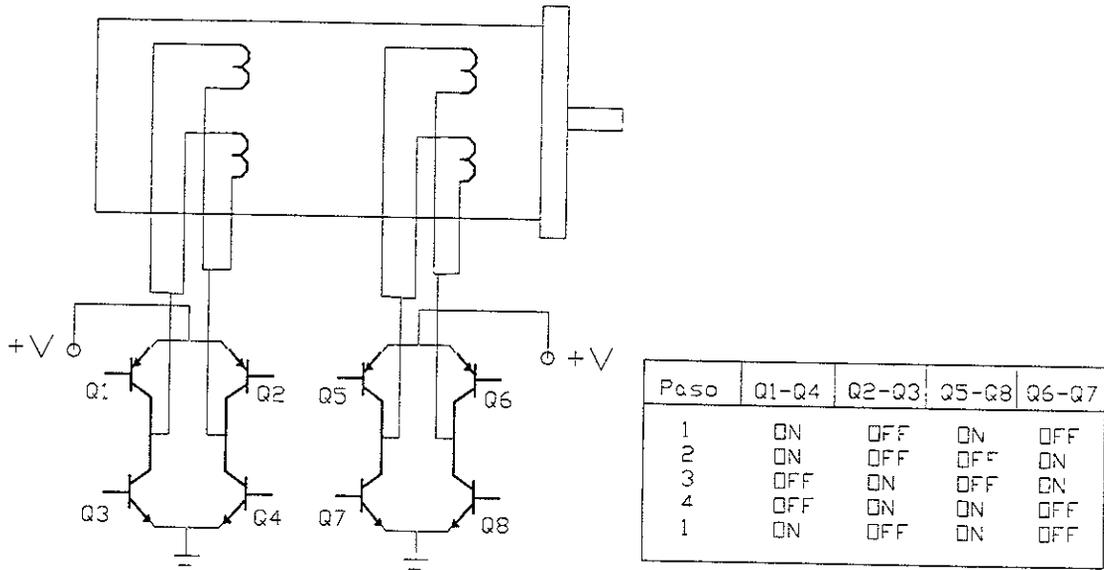


a) Diagrama de conexiones

b) Secuencia de operación

Figura 49 Conexión Bipolar Serie para Motor paso a paso (PAP) Bipolar 4 Fases

En la figura 4.10 se ilustra una conexión bipolar paralelo para un motor de 4 fases. La diferencia básica que existe entre esta configuración y la anterior es que las bobinas se encuentran conectadas en paralelo y la corriente que circula por los interruptores electrónicos es la suma de la corriente que requiere cada una de las bobinas energizadas.



a) Diagrama de conexiones.

b) Secuencia de operación

Figura 4.10	Conexión Bipolar Paralelo para Motor paso a paso (PAP) Bipolar de 4 Fases
-------------	---------------------------------------------------------------------------

4.1.6 Selección del motor paso a paso a utilizar.

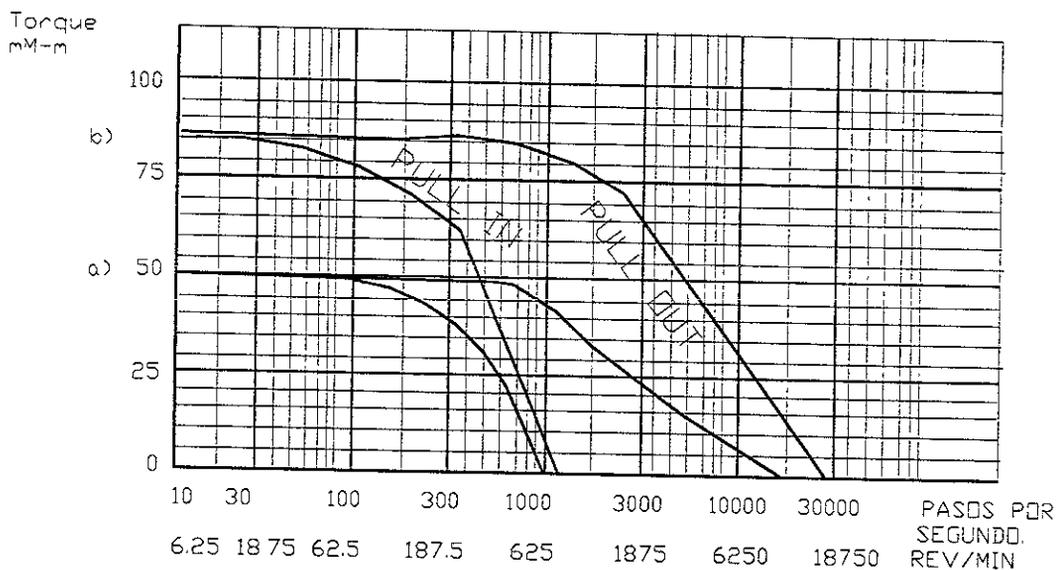
Es necesario conocer el ángulo de giro de los motores entre otras características para hacer la selección adecuada de un motor. A continuación se presenta la ecuación para conocer el ángulo de giro de un motor.

$$\alpha = \frac{360}{f \cdot n} \quad [4.1]$$

donde:

$\alpha \in \mathfrak{R}$ es el ángulo de giro por paso (en grados),
 f es el número de fases del motor paso a paso y
 n es el número de dientes del rotor.

Una forma de seleccionar los motores es apoyarse en la curva característica llamada *Torque - velocidad (PPS)*¹ que se utiliza para seleccionar adecuadamente un motor. Las curvas que se muestran en la gráfica 4.1 representan el comportamiento Torque - velocidad de un motor controlado en forma a) *unipolar* b) *bipolar*. Se puede observar un considerable incremento en el torque permisible en la forma de control bipolar de un motor, sin embargo, el control es más complejo.



a) Forma unipolar.
b) Forma bipolar.

Gráfica 4.1

Curvas de comportamiento de forma unipolar y bipolar.

La curva de PULL IN (Momento torsional de ajuste a sincronismo); muestra cual es la carga a la cual el motor puede ser puesto en marcha o detenerse sin perder algún paso a velocidad constante PPS (Pasos por segundo), es decir; la velocidad del motor es constante.

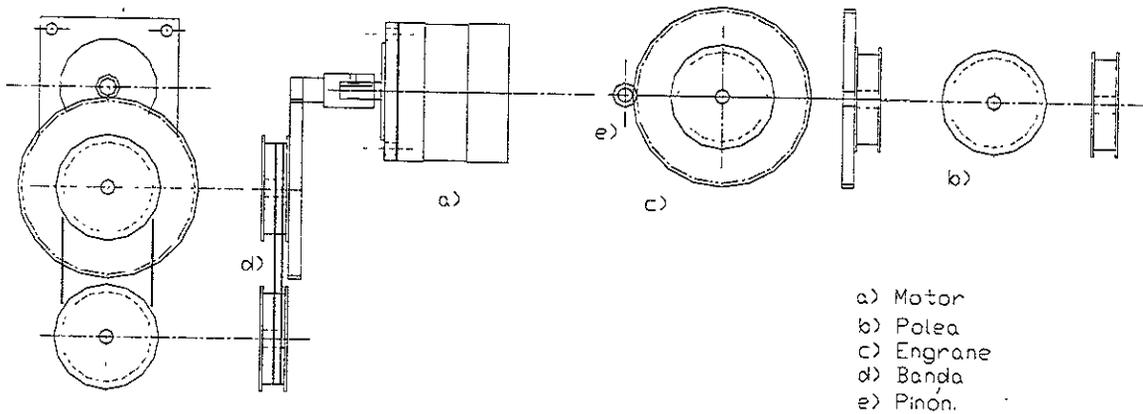
La curva de PULL OUT (Par motor crítico o límite); es el torque permisible donde el motor es lentamente acelerado para trabajar a cierta velocidad PPS. La diferencia

¹ PPS se refiere a pasos por segundo (Pasos/s).

entre las curvas PULL IN y PULL OUT es el torque que se utiliza para acelerar la masa de inercia acoplada al rotor del motor incluyendo la masa inercial del rotor mismo.

El uso de la curva característica de *Torque - Velocidad PPS* permite seleccionar correctamente un motor según la aplicación a desarrollar, así como también la forma de control. Para seleccionar un motor que cumpla con los requerimientos del prototipo a controlar es necesario aplicar la siguiente metodología;

a) Calcular la relación de engranaje necesaria para conseguir que con el torque que el motor suministre logre mover la carga prevista, en este caso será el peso del hombro y brazo del robot-manipulador, incluyendo el peso de las poleas, tornillos, separadores, etc. y fuerzas de fricción ejercidas sobre los acoplamientos polea-eje.



Acot. en
Esc. 1:1

Figura 4.11 Configuración de motor, banda y polea de la aplicación

La figura 4.11 muestra la configuración y distribución de los elementos que transmitirán movimiento a los eslabones del manipulador, donde el inciso c) representa a un piñón acoplado al eje del motor, el piñón transmitirá movimiento al engrane el cuál tiene

acoplado una polea, inciso b). La banda y la polea son dentadas, esto es, con el fin de evitar deslizamiento entre banda y polea.

La siguiente ecuación determina la relación de engranaje.

$$\frac{N_e}{N_p} = Re \quad [4.2]$$

donde: N_e .- Número de dientes del engrane.
 N_p .- Número de dientes del piñon.

Entonces la relación Re ; con el piñon de $N_p = 8$ dientes y el engrane de $N_e = 72$ dientes es de la ecuación 4.2 tenemos;

$$Re = \frac{72}{8} = 9. \quad [4.3]$$

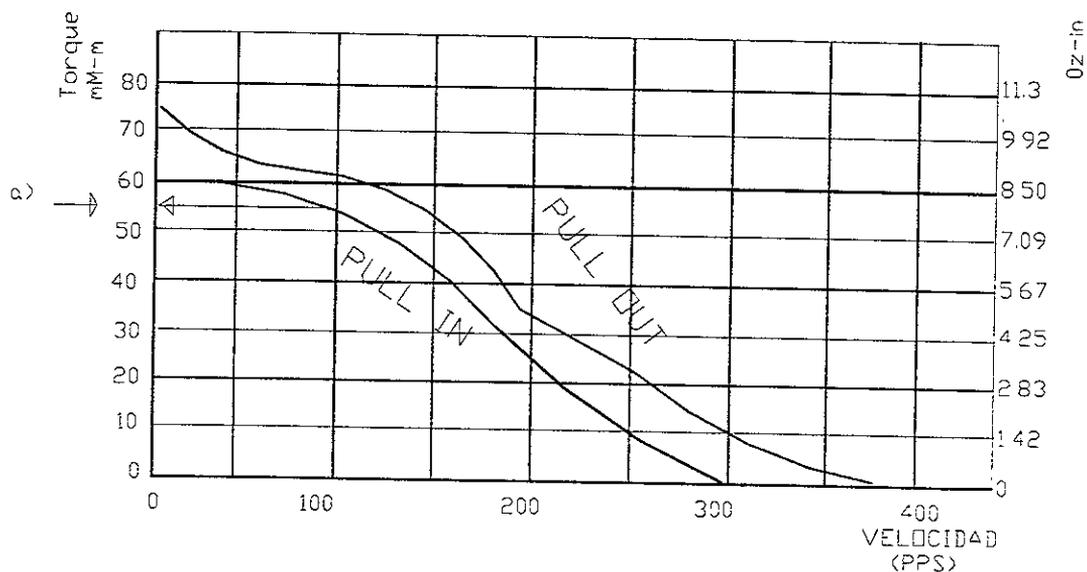
La relación $Re = 9$ permite que por cada pulso que se envíe al motor el engrane girará 0.1° grados de ángulo de giro, recordando que el eje del motor girará 1.8° por pulso recibido, esto permite que la resolución obtenida dé mayor exactitud en el control de posicionamiento del Robot.

b) Proponer el tipo de motor a utilizar haciendo referencia a sus datos técnicos y a su curva característica Torque-velocidad.

Se propone la utilización del motor de cuatro fases tipo 103-771-1242 Step-Syn[11], cuya curva característica de *Torque - Velocidad PPS* se muestra en la gráfica 4.2 y los datos técnicos en la tabla 4.1.

Tabla 4.1 Datos técnicos del motor tipo 103-771-1242 Step-Syn	
Tensión de operación.	12 v. cd.
Resistencia por bobina.	6.3 ohms.
Inductancia por bobina.	4.5 mH.
Holding Torque*.	55 mNm
Momento de inercia del rotor.	4×10^4 gr.m
Detent Torque*.	55 mNm.
Angulo de Paso.	1.8° grados.
Tolerancia en el ángulo de Paso.	$\pm 1^\circ$.
Pasos por una revolución del eje.	200 Pasos.
Temperatura máxima de operación.	100° C.
<i>Para mayor información referirse a [11].</i>	
<i>*Para ver terminología Apéndice "A".</i>	

Según la gráfica 4.2 a una velocidad menor a 100 PPS inciso a) se tiene un torque de 55 mN-m en la curva de PULL IN. Se seleccionó esta curva ya que representa la máxima tasa de conmutación de los interruptores electrónicos (velocidad) en la cuál un motor con carga puede ponerse en marcha sin perdr pasos. Es decir, el máximo torque que puede ser aplicado al eje del motor cuando es puesto en marcha.



Grafica 4.2 Curvas de comportamiento del motor tipo 103-771-1242 step-syn

El torque que se tiene disponible es aumentado 9 veces debido a la relación de engranaje $R_e = 9$, permitiendo un torque de 55 mN-m según a) de la gráfica 4.2, se tiene entonces $55 \times 9 = 495$ mN-m.

El torque es suficiente para poder otorgar movimiento a la carga (500 gr. aproximadamente) ejercida por los eslabones del prototipo pequeño de Robot-manipulador de 2 grados de libertad con una velocidad de operación de los motores menor a 100 PPS. Además, el tipo de control que se utilizará será en la forma Bipolar. Con esta información se podrá llegar al objetivo de controlar el posicionamiento del robot.

4.2 Manejadores (Drives) para motores paso a paso.

Del inciso e) de la figura 4.0 se puede observar que los manejadores (Drives) son dispositivos que reciben información en forma de pulsos eléctricos y éstos generan una secuencia lógica creando un campo giratorio aparente que obliga a girar al eje del motor.

Aunque este campo puede generarse por medios mecánicos de rotación o conmutadores, ahora son usados para este propósito ciertos circuitos integrados, ya sea directamente o para grandes motores a través de transistores de potencia.

Estos circuitos integrados realmente simplifican la tarea aplicando automáticamente la secuencia lógica de corriente a las bobinas del motor el cuál responde a un pulso por cada Paso de salida. Algunos de estos manejadores (Drives) se muestran en la tabla 4.2.

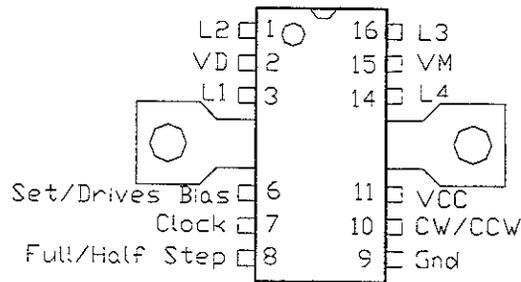
Tabla 4.2 Tipos de manejadores y características.		
TIPO	FABRICANTE	OBSERVACIONES
SAAA 1027	Signetics/Mullard	Generador de paso a 350 mA, para motores de 4 fases.
SAA 1042	Motorola	Generador de paso a 500 mA, para motores de 4 fases con modo de paso completo y medio paso.
UCN-4204 B	Sprague	1.25 Amp para motores de 4 fases, con modo de paso completo y medio de paso.
L 293 D	SGS	Generador de paso, para motores de 2 fases.
75 TEA 1012	Signetics/Millard	Generador de paso, con modo de paso completo y medio paso, y control de corriente 50mA para motores de 4 fases.
L 297	SGS-ATES	Generador de paso, con modo de paso completo y medio paso, y control de corriente para motores de 2 y 4 fases.
L 298	SGS-ATES	Puente de potencia H para el manejador (Driver) L297.
UDN-28778/9	Sprague	Driver cuadrupolar de 4 Amp de potencia para motores de 4 fases.
555	Motorola/National Signetics/Texas	Generador estándar industrial tiempo/pulso. Reloj de pulsos.

Esto permite darse una idea general de la gran gama de manejadores que existene en el mercado y la posibilidad de adecuarse al más conveniente para alguna aplicación en especial.

4.2.1 Principios de operación de los manejadores (Drives) para motores paso a paso.

De los manejadores (Drives) mencionados en la tabla 4.2 el SAA 1042 será utilizado para el desarrollo de este trabajo ya que se utiliza para controlar motores de dos fases en forma bipolar.

En la figura 4.12 se muestra la numeración de los pines los cuales tienen diferentes funciones L1, L2, L3, L4 son las salidas a las terminales del motor, VD, VM, VCC y Gnd los voltajes de alimentación, se tiene también los pines 6, 7, 8,10 que reciben información para mover el motor, más adelante se dará una descripción más detallada de las funciones de los pines



Vista superior.

Figura 4.12 Asignación de los pines del Driver SAA 1042

Las características del manejador del driver SAA1042; fué diseñado para controlar motores de 6 y 12 v. con corrientes de hasta 500 mA por fase, contiene también un juego de diodos internos para evitar sobrevoltajes también acepta comandos para controlar la dirección de giro CW/CCW (Sentido horario/Sentido antihorario) y operaciones de Half/Full Step (medio Paso/Paso completo).

Con una entrada se puede definir el estado lógico de las salidas, además contiene un disipador de calor integrado para evitar sobrecalentamiento debido a las operaciones de carga imprevistas dando una alta eficiencia.

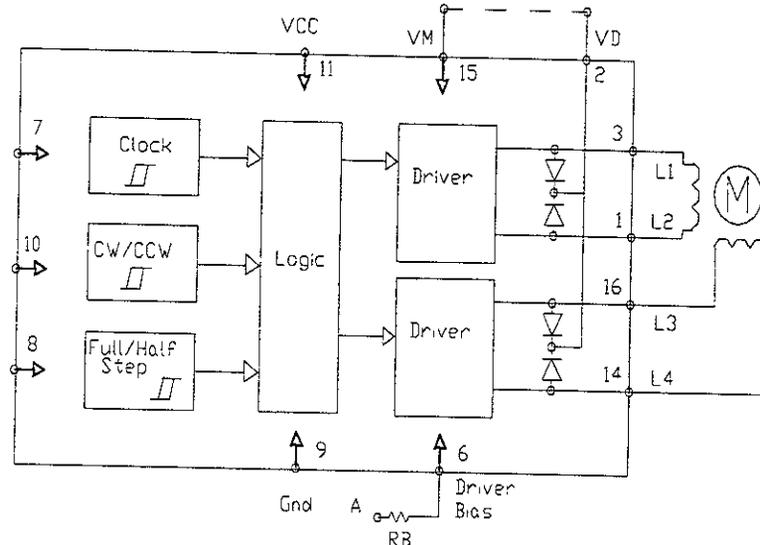


Figura 4.13 Diagrama de Bloques del Driver SAA 1042

En la figura 4.13 se presenta el diagrama de bloques del driver SAA 1042 en forma simplificada, se observa el funcionamiento interno y cómo es que se procesan las señales de entrada y salida. A continuación se presenta la descripción de los pines del driver y las características de sus bloques internos.

CLOCK (Reloj).- Pin** 7.- Esta entrada se activa con el borde positivo del pulso del reloj y acepta un "1" lógico como nivel de entrada dependiendo del voltaje e incluyendo la histéresis para la inmunidad de ruido.

CW ClockWise / CCW Counter Clockwise (Sentido horario/Sentido antihorario).- Pin 10. Esta entrada determina la dirección rotacional del motor. Es decir, cuando el nivel de entrada es "0" la dirección del motor paso a paso es CW ClockWise, cuando la entrada está en estado lógico "1" en dirección CCW Counter Clockwise, esto depende de las conexiones del motor.

Full/Half Step (medio Paso/Paso completo)- Pin 8.- Esta entrada determina el ángulo de rotación de el motor por cada pulso del reloj. En el estado bajo rotará un paso completo por cada pulso aplicado mientras que en el estado alto rotará medio paso.

** Pin es una clavija o terminal de un circuito integrado que se inserta o ensamblada a algún circuito impreso.

VD (voltaje diodo) - Pin 2.- Este pin es usado para para proteger las salidas (1,3,14,16). donde el brinco positivo más largo ocurre debido a los cambios de polaridad de la corriente sobre las fases. El máximo voltaje permitido sobre éstos pins es el voltaje V_{clam} , y éste se obtiene de la siguiente ecuación;

$$V_{clam} = V_M + 6 \text{ v.} \quad [4.4]$$

donde:

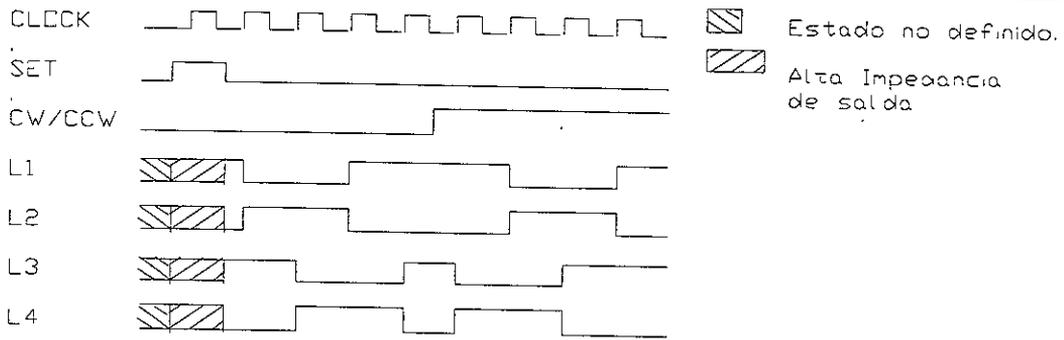
V_{clam} .- Voltaje máximo permitido para las bobinas de un motor.

V_M .- es el voltaje de operación del motor paso a paso.

Set/Bias Input (Fijar/entrada) Pin 6.- Esta entrada tiene dos funciones; en la primera la resistencia R_B adapta el Driver SAA 1042 a la corriente del motor, y la segunda es que con un pulso a través de la resistencia R_B las salidas (1,3,14,16) se fijan a un estado definido.

El valor de la resistencia R_B se puede determinar con la gráfica 4.6 de acuerdo a la corriente y al voltaje del motor. Pequeños valores de R_B incrementarán el poder de disipación del circuito y para valores más altos de R_B puede incrementarse el voltaje de saturación de los transistores del Driver SAA 1042. Cuando la función "set" (fijar) no es usada, la terminal A de la resistencia debe ser conectada a tierra. Cuando la función "set" es usada, la terminal A debe ser conectada a un circuito de colector abierto. La figura 4.14 muestra esta configuración. El circuito off-state (estado de corte) tiene que sostener el voltaje V_M del motor cuando un pulso es aplicado a través de la resistencia R_B .

Durante el pulso los transistores del Driver SAA 1042 están en estado de corte después del cambio del pulso las salidas (1,3,14,16) se fijan a un estado definido, este estado se debe dar después de la transición negativa del pulso de la función "set", Pin 6, y hasta la transición positiva del pulso, las salidas serán $L1 = L3 = "1"$ y $L2 = L4 = "0"$.



Manejador(Driver) en modo de Paso completo en sentido horario y antihorario

Gráfica 4.3	Diagrama de tiempo del Driver SAA 1042
-------------	----------------------------------------

La gráfica 4.3 muestra el diagrama de tiempos que para cada pulso del Clock (reloj) las salidas L1, L2, L3, L4 se encuentran en un estado definido. La figura 4.14 ilustra una aplicación típica en la cuál el Driver SAA 1042 controla un motor de 12 v. con un consumo de corriente de 200 mA por fase.

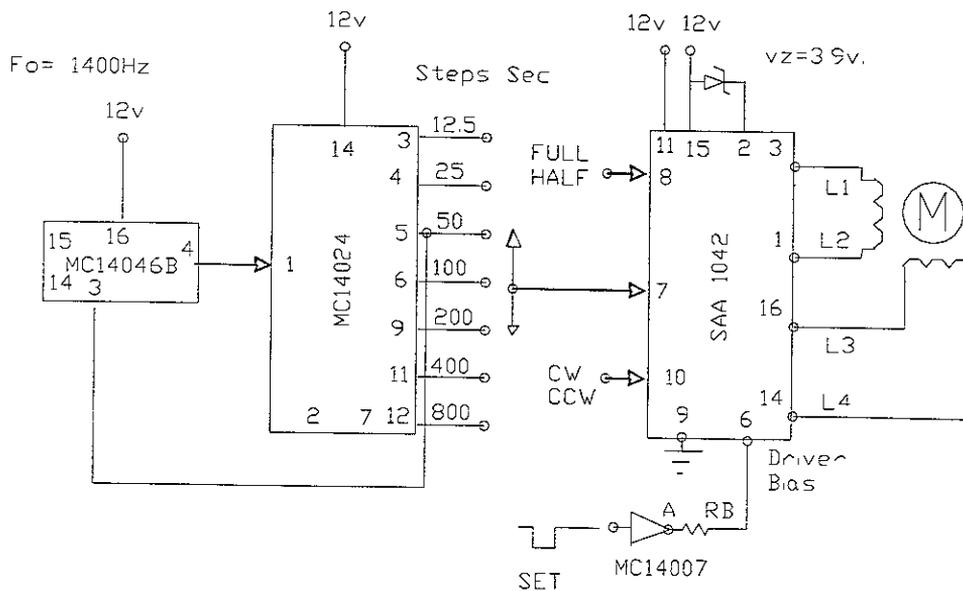
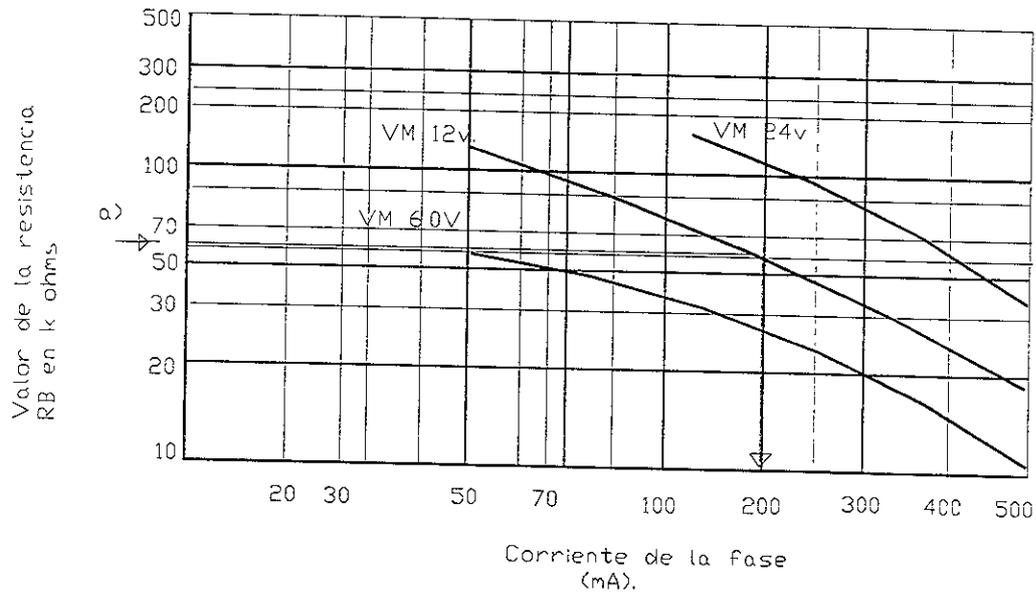


Figura 4.14	Una aplicación típica del Driver SAA 1042
-------------	-------------------------------------------

Una resistencia R_B de $56\text{ k}\Omega$ se seleccionó de la gráfica 4.5 el máximo voltaje permitido a las salidas (1,3,14,16) es otorgado a través de aplicar la ecuación 4.4; en esta aplicación $V_M = 12\text{ v}$ por lo tanto, el máximo voltaje permitido es de 18 v . Las salidas están protegidas por diodos internos y también por un diodo zener que se conecta a los pines (2,15).



Gráfica 4.4 Resistencia VS Corriente del Motor paso a paso (PAP)

y su valor se calcula de la siguiente manera; de la gráfica 4.5a. se obtiene el valor de V_f (voltaje directo) de caída en los diodos internos es de 1.7 v . a 200 mA . esto da como resultado el voltaje que debe de existir entre los pines (2,15). Para calcular el valor del voltaje del diodo zener (regula el votaje de entrada) se utiliza la siguiente ecuación

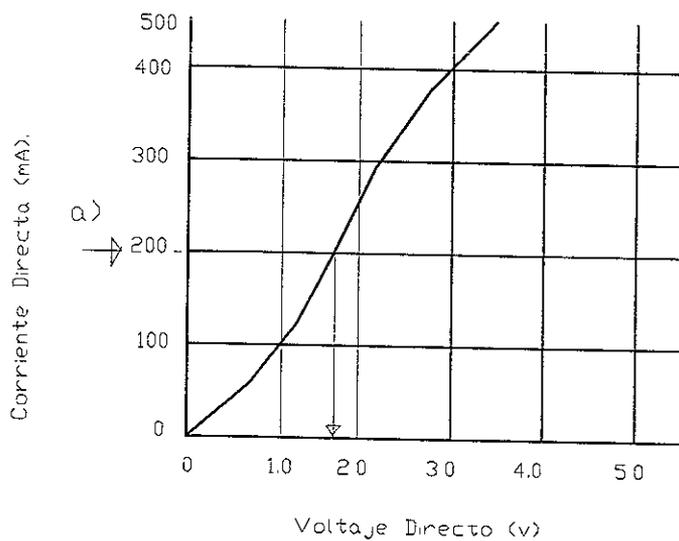
$$V_z = 6\text{v} - V_f \quad [4.5]$$

Donde:

- V_z .- Es el valor del diodo zener.
- V_f .- Voltaje directo.

De la ecuación 4.5 con un valor $V_f = 1.7$, $V_z = 4.3$ v tomando en cuenta el margen de tolerancia y de productos comerciales en el mercado se tomó un valor de 3.9 v para el diodo zener en este ejemplo. El reloj se deriva de la línea de frecuencias el cuál se controla por un MC14046B y el MC14024. El voltaje de la entrada del reloj, está normalmente en estado lógico bajo "0" el motor paso a paso avanza un paso cuando pasa al estado lógico alto "1" del pulso del reloj.

Un "0" lógico aplicado a la entrada Full/Half, Pin 8, opera el motor en el modo de Full Step y un "1" lo hará en el modo de Half Step sí el Pin 10 recibe un "1", entonces el motor girará en sentido (CW) y si tiene "0" girará en sentido contrario (CCW) dependiendo de la configuración del motor.



Gráfica 4.5	Corriente de diodos VS Voltaje directo
-------------	----------------------------------------

4.3 Creacion del hardware de control.

Haciendo referencia a la figura 4.0 se empezará con una introducción breve de la computadora personal PC como herramienta de trabajo para el control del robot. El

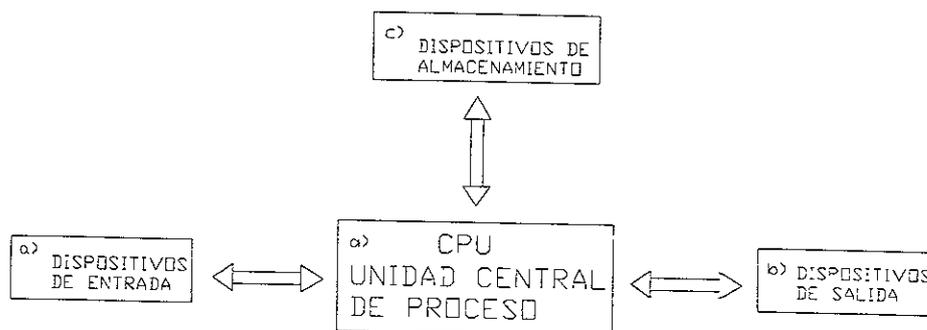
hardware de control abarca la electrónica del sistema que recibe la información desde la PC, además de los motores, la tarjeta de interfase de control se incluye.

4.3.1 La computadora personal PC.

Una computadora es un "Sistema de procesamiento de datos de propósito general"; es decir, permite la entrada de datos, éstos pueden ser gestionados por medio de programas, de forma tal, que los resultados obtenidos sean legibles y entendibles para los usuarios.

La potencia de una computadora esta en función de la velocidad, formalidad y facilidad de programación de las tareas a realizar. Contiene un sistema de reloj ("clock") para sincronizar todas sus acciones, realizando una acción en varios pulsos de reloj. Los componentes básicos de la electrónica de una computadora son los que se muestran en la grafica 4.6 las cuales son;

- a).- Los dispositivos de entrada.
- b).- Los dispositivos de salida.
- c).- Los dispositivos de almacenamiento.
- d).- La unidad central de proceso (CPU).



Gráfica 4.7 Diagrama a bloques de una PC en su forma simplificada

a).- *Los dispositivos de entrada* son: el teclado; él cual permite una entrada manual de datos, y los convertidores analógico-digital (A/D), los cuales transforman una señal analógica en una serie de valores digitales que pueden ser manipulados por la computadora (éstos son usados en sistemas de medición, control, etc.).

b).- *Los dispositivos de salida*; como el monitor, que a pesar de ser muy rápido, (porque la información que presenta es volátil), es también un dispositivo ideal para la comunicación con el usuario, la impresora para salida de textos impresos en papel, los graficadores ("plotters") los cuales ofrecen una alta calidad en gráficas de multicolores, y los convertidores digital-analógico (D/A) que transforman los valores digitales en señales analógicas que pueden ser usadas en sistemas de control.

c).- *Los dispositivos de almacenamiento*; son las memorias y sus respaldos de memoria, tales como: los discos flexibles con una capacidad media de almacenamiento, y los discos duros con una alta capacidad de almacenamiento, ambos de acceso aleatorio. Otros dispositivos son las cintas magnéticas de gran capacidad pero de acceso secuencial, lo cual, reduce la velocidad de procesamiento siendo usadas sólo para respaldo de discos duros.

d).- *La unidad central de proceso (CPU)*; que es la encargada de controlar la operación de la computadora. Esta unidad contiene un procesador, la memoria y la circuitería para comunicarse con los dispositivos de entrada/salida y de almacenamiento.

La mayoría de las computadoras están construidas en forma modular, es decir, en tarjetas electrónicas que implementan bloques funcionales y que están montadas en una tarjeta principal ("Motherboard"), la cual contiene las ranuras de expansión, unidas todas ellas por un bus (canales de comunicación desde el CPU hasta los periféricos de entrada/salida).

En una computadora la tarjeta principal controla el bus del sistema y manipula los datos, muchas funciones y componentes están contenidos en un chip (circuito integrado), el microprocesador es el corazón de la tarjeta principal.

La tarjeta principal contiene al microprocesador y toda la circuitería que el microprocesador necesita para operar (como el sistema de reloj que impulsa al microprocesador para que realice la próxima instrucción). El microprocesador actúa colocando instrucciones, diciendo exactamente como manipular los datos. Típicamente un microprocesador reconoce de 50 a 250 diferentes instrucciones, por tal razón, el microprocesador es el componente más importante dentro de la tarjeta principal.

Los programas consisten en listas de instrucciones, de decenas a miles de éstas, y son introducidas a la computadora mediante dispositivos de entrada y registrados por medio de los dispositivos de almacenamiento. Para realizar las tareas en forma eficiente el CPU está equipado con memoria volátil, la cual se perderá cuando la computadora se apague. Por esta razón antes de la ejecución de los programas estos deben ser cargados por la computadora desde los dispositivos de almacenamiento a la memoria, entonces el microprocesador leerá las instrucciones una por una y actuará de acuerdo a ellas.

Estas instrucciones pueden ser para leer datos de un dispositivo de entrada (por ejemplo: la retroalimentación de la tarjeta de interés) y almacenarlos en memoria, otras manipulan los datos (comparan, suman, restan, etc.) y envían los resultados a un dispositivo de salida (por ejemplo; a la interfase que conecta al motor).

4.2.2 El bus de la computadora personal (PC).

Cada una de las PC's trae internamente una serie de ranuras en las cuales se pueden insertar tarjetas de propósitos específicos. A éstas ranuras se les conoce como ("slots") y contiene 62 líneas paralelas. Estas líneas proporcionan todas las señales que son necesarias para algún equipo que pueda ser unido a la PC.

El bus de una computadora consiste de un gran número de pistas hechas de material conductor al se conectan todas las tarjetas para establecer comunicación entre ellas; para esto se utilizan algunas señales de control del flujo de la información. Sólo a una tarjeta le es permitido enviar datos al bus en un instante dado, mientras que al menos una de las demás debe estar recibiendo la información.

El bus de la computadora, está formado por líneas de datos con direcciones de control y de alimentación. Normalmente el bus de direcciones consiste de 16, 20, 24 o más líneas paralelas que llevan las señales de las direcciones, a través de estas líneas el CPU (Unidad Central de Proceso) envía en forma de señales las direcciones de las localidades de memoria a ser escritas o leídas. El número de localidades que el CPU puede direccionar está determinado por el número de direcciones. De acuerdo con reglas establecidas, si el CPU tiene n líneas entonces puede direccionar directamente 2^n direcciones, por ejemplo: un CPU como el 6800 de 16 líneas de direcciones puede direccionar 2^{16} direcciones diferentes o sea 65536 o 64k localidades de memoria (1 k = 1024 localidades de memoria y una localidad de memoria es un Byte, además, 1 byte = 8 bits), ya sea para la lectura o escritura. Así pues, un CPU de 20 líneas como los 8086 y 8088 pueden direccionar hasta 2^{32} o sea 4096M de localidades (4 Gigabytes), ejemplos de éstos son los 80386 y el 68020, uno manufacturado por Intel y el otro por Motorola respectivamente.

El bus de datos consiste de 8, 16 o 32 líneas paralelas. Como puede verse en cualquier esquema o diagrama a bloques de un sistema computador, las líneas del bus de datos son bidireccionales; esto, debido a la necesidad del manejo de la información de y hacia el exterior por parte del sistema y el usuario. El bus de datos es el encargado de transportar la información dentro del sistema, dependiendo del tamaño del bus es el tamaño de la unidad de información que acarrea en un tiempo. Así, un bus de 8 pistas transporta sólo unidades de información de 8 bits, con lo cual alcanza a manejar hasta 256 caracteres codificados o sea 2^8 , por lo que entre mayor sea el bus de datos, mayor será el tamaño de la unidad de información a manejar en un lapso de tiempo.

La unidad básica de información es el Bit (Binary digit) un grupo de n bits genera 2^n valores diferentes como el que se mencionó con 8 bits. La unidad básica de datos en una computadora es llamada palabra, ésta también define la longitud mínima de un código de instrucción.

Muchos dispositivos en una computadora tienen sus entradas y salidas conectadas a este bus de datos. Debido a que sólo un dispositivo puede acceder o ser accesado en un instante determinado por la computadora, los otros deben ser desconectados. La interacción de los dispositivos con este bus debe ser en forma de tres estados (parlante, escucha o desconectado).

El bus de alimentación es el encargado de hacer llegar la corriente de la fuente de alimentación a los distintos componentes del sistema y varía de acuerdo a las necesidades del sistema. En tiempos pasados los buses de alimentación tenían que suministrar hasta tres distintas tensiones de alimentación. Hoy en día estos buses se han reducido gracias a los avances en los desarrollos de los dispositivos con necesidades de suministro de tensiones de un sólo nivel y tierra, normalmente 5 volts. Hay otro nivel más alto de 12 volts el que puede ser utilizado para distintos propósitos.

El bus de control consiste de cuatro a diez líneas paralelas por las que circulan señales de control de la computadora, el bus de control lleva información sobre la temporización del sistema, o sea, las señales de reloj, información de órdenes como son accesos de memoria o puertos, también información acerca del sentido de los datos mediante las señales de lectura y escritura, por este bus el CPU envía las señales de control para evitar conflictos entre todos los dispositivos unidos al sistema.

Entonces para comunicarse desde la computadora al medio exterior (Dispositivos E/S a controlar) se requiere de una Interfase de control que permita tal fin. La interfase Paralela de una PC ofrece al usuario un método sencillo y de alta velocidad para comunicarse con el CPU. Operar con datos en paralelo en lugar de en serie ofrece varias ventajas, tales como:

- a).- Eliminación del Codificador y Decodificador.
- b).- No requiere de un Reloj para la velocidad de transmisión.
- c).- Sincronización inherente.

Los datos técnicos del puerto paralelo son:

Tipo de Trasmisión: Paralelo (Bloques de 8 Bits).
 Niveles de Señales: **Bajo** 0.0 a + 0.4 v.
Alto +2.4 a + 5.0 v.

Se presenta un diagrama de señales de tiempo en la figura 4.15a. y la descripción de las terminales(clavijas) del conector macho DB25 figura 4.15b se detallan en la tabla 4.2.

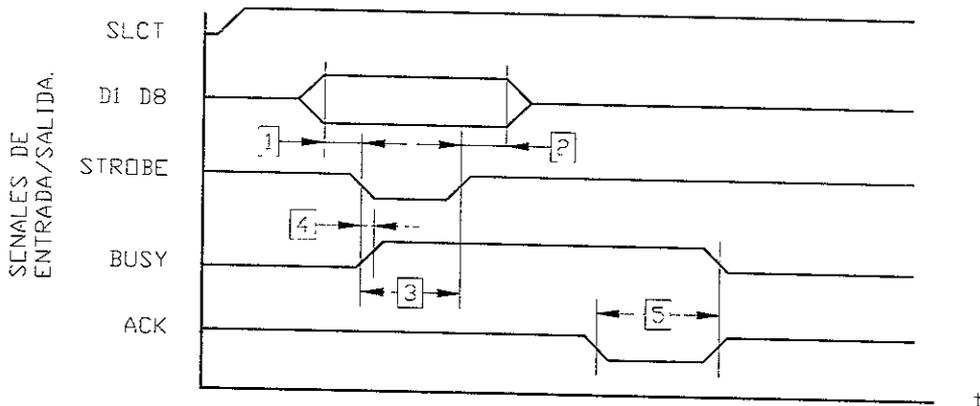


Figura a.

1	$\geq 0.5 \mu s$
2	$\geq 0.5 \mu s$
3	$\geq 0.5 \mu s$
4	$= 400 ns$
5	$= 8 \mu s.$

μs microsegundos

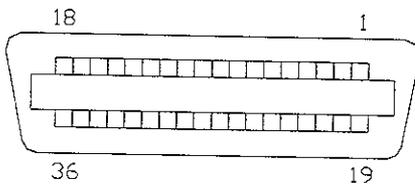


Figura b.

Figura 4.15	Protocolo de comunicación del Puerto Paralelo de un CPU
-------------	---------------------------------------------------------

Tabla 4.2

Descripción de las señales del puerto paralelo de la PC.

No. de CLAVIJA	SEÑAL	E	S	DESCRIPCIÓN
1	STROBE		x	Es la señal que valora los datos, sincroniza la toma de las señales de datos D1...D8.
2	D1		x	Dato 1.
3	D2		x	Dato 2.
4	D3		x	Dato 3.
5	D4		x	Dato 4.
6	D5		x	Dato 5.
7	D6		x	Dato 6.
8	D7		x	Dato 7.
9	D8		x	Dato 8.
10	ACKNLG	x		Esta señal es enviada por la impresora para la confirmación de recepción de datos, y señala al ordenador la disposición de la impresora para la próxima recepción de datos.
11	BUSY	x		Esta señal indica si la impresora está lista para la recepción de datos, y no está lista cuando: ERROR =0, SLTCT=0. La impresora no está On Line.
12	PE	x		Es la señal de final de papel.
13	SLT	x		SLT =1, cuando la impresora está en On Line.
16	GND		x	Tierra. 0 v.
17	CHASIS GND		x	Tierra de chasis 0 v.
18	+ 5 v.		x	Salida con +5v. cd.
19-30	GND		x	Tierra
31	INIT		x	Esta señal se utiliza para que inicie la impresora.
32	ERROR	x		Esta señal se activa en caso de avería, es decir, cuando no es posible la impresión debido a que el cabezal de impresión está obstaculizado.
33	GND		x	Tierra

Se puede observar que el número máximo de Bits que se puede usar con el puerto paralelo de la PC es de 8 Bits, y esto provoca muchas limitaciones a la hora de dar

aplicaciones ya que por ejemplo; los motores necesitan 4 Bits para ser controlados directamente desde la PC.

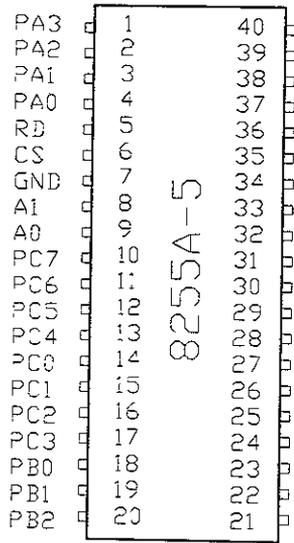
Esta limitación obligó a incrementar el número de Bits, ya que la aplicación así lo requería. Por lo tanto, se ha utilizado la interfase de control de 24 Bits con el circuito integrado 8255A-5 de Intel para poder solventar ésta limitación.

4.3.3 Interfase para el puerto paralelo de 24 bits para PC.

La Interfase 8255A-5. es una Interfase periférica programable, ésta fué hecha para trabajar con el microprocesador XT 8088, sin embargo, también se puede usar para las familias AT. El 8255A-5 cuenta con tres puertos de entrada y salida cada uno compatible con dispositivos de tecnología TTL (Lógica Transistor Transistor), y un puerto de control interno para programar el puerto para cualquiera de los dos modos diferentes de operación. Dependiendo del modo de trabajo en que se encuentre el 8255A-5 las líneas de cada puerto trabajan de manera diferente.

En el modo "0" el puerto A y el puerto B actúan como entradas de datos, y el puerto C es dividido en dos grupos de 4 bits (Nibble), en cambio en el modo "1" de operación, los puertos A y B actúan como salida de datos, pero los dos grupos de 4 Bits del puerto C son usados para propósitos de enlace y control en conjunto con los puertos A y B. Las líneas del puerto C son usadas para las señales de Strobe y Busy de la impresora entre otras aplicaciones.

A continuación en la figura 4.16 se muestra la configuración de terminales de la interfase 8255A-5.



a) Configuración de terminales

TERMINAL	DESCRIPCIÓN
D7-D0	BUS DATOS BIDIRECCIONAL,
RESET	ENTRADA DE INICIALIZACION
CS	SELECCIÓN DE CHIP
RD	ENTRADA DE LECTURA
WR	ENTRADA DE ESCRITURA
A0-A1	DIRECCION DE PUERTO
PA7-PA0	BITS DEL PUERTO A
PB7-PB0	BITS DEL PUERTO B
PC7-PC0	BITS DEL PUERTO C
Vcc	+ 5v DC
FND	0 V

b) Descripción de terminales

Figura 4.16 El 8255A-5 configuración

En la figura 4.17 se presenta el diagrama de bloques de la Interfase 8255A-5 de Intel, se puede observar la distribución y comunicación interna, además en la tabla 4.3 se da una descripción detallada de las señales de entrada y salida que genera la interfase.

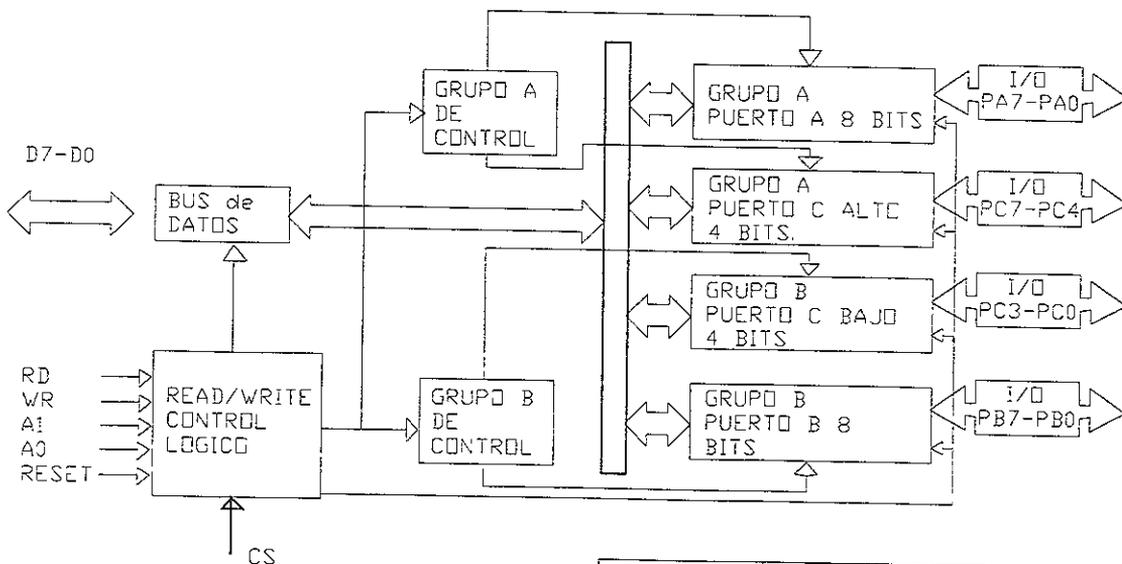


Figura 4.17 Diagrama de Bloques 8255A-5

Tabla 4.3

Descripción de las señales del 8255A-5.

Bus de Datos.	El Bus de Datos bidireccional de tres estados de 8 Bits es usado para comunicar el circuito con el Bus de datos de la computadora. Los datos son transmitidos o recibidos por el Bus de Datos dependiendo de las instrucciones del CPU., ya sean entradas o salidas . Las palabras de control y el nivel de información son transmitidos a través del Bus de Datos.
Read/Write y control lógico.	La función de este bloque es manipular todas las transferencias internas o externas de palabras de control o de datos que entran y salen del dispositivo. Este acepta entradas del CPU del bus de direcciones y del bus de control, y en su turno transmite comandos a ambos grupos de control.
cs	Chip Select. Una señal baja en esta terminal de entrada habilita la comunicación entre el 8255A-5 y el CPU.
RD	Read. Una nivel bajo en esta terminal de entrada habilita al 8255A-5. para mandar datos o comandos de control al CPU a través del Bus de Datos. En esencia esta señal permite al CPU leer del 8255A-5.
WR	Write. Una señal baja en esta terminal de entrada habilita al CPU a escribir una palabra de control o de datos en el 8255A-5.
A0 y A1	Selección de puerto "0" y selección de puerto "1". Estas señales de entrada junto con las señales de RD y WR (Entradas), controlan la selección de uno de los tres puertos o de el puerto de control (o registro de las palabras de control). Estos se encuentran por lo general conectado en los Bits mas significativos del Bus de Direcciones.
RESET	Reset. Un nivel alto en esta entrada limpia los registros de control y todos los puertos A,B y C son puestos en la forma de entrada de datos.
GRUPO A y GRUPO B de CONTROL	<p>La configuración funcional de cada puerto es definida a través del programa de la computadora que se está usando, esto es, que el CPU manda palabras de control al 8255A-5. Estas palabras de control contienen información a cerca de el modo en que se utilizará el puerto.</p> <p>Los bloques de control (Grupo A y Grupo B) aceptan las señales de Read, Write y Control lógico, y reciben las palabras de control del Bus interno de datos y emiten comandos a cada uno de sus puertos correspondientes.</p> <p>El grupo de control A, contiene a el puerto A y la parte alta de el puerto C. (PC7-PC4).</p> <p>El grupo de control B, contiene a el puerto B y a la parte baja de el puerto C. (PC3-PC0). La palabra de control solo puede ser escrita en el interior del 8255A-5.</p>

4.2.4 Tarjeta de Interfase para PC de 24 Bits.

Para comunicarse con dispositivos periféricos el 8088 utiliza puertos. Los puertos son áreas de memoria que el 8088 puede acceder utilizando instrucciones especiales de ensamblador. El 8088 tiene la capacidad de direccionar 65 356 puertos de hardware utilizando 1k de direcciones de I/O las cuales van de 00h a 3ffh, estas direcciones o localidades de memoria son accesibles tanto al microprocesador como a los dispositivos de I/O.

Las instrucciones de ensamblador IN y OUT manejan la transferencia de información entre el 8088 y los puertos. Generalmente un dispositivo utiliza más de un puerto, uno para entrada, uno para salida y uno para describir su estado (Status).

Ahora lo que se necesita saber es que direcciones de I/O se pueden utilizar, ya que el PC tiene definidas algunas direcciones para los puertos con funciones o usos específicos. Los primeros 256 puertos (0-Ffh) están reservados para uso del System Board, como son los Timmers(Temporizadores), controladores de interrupciones , coprocesadores , etc.

El resto de los puertos son usados para propósitos generales, como controlador de video, controlador de teclado, etc. Para poder realizar la construcción de la tarjeta es necesario saber como están contruídos los Slots o Ranuras de Expansión de la PC.

El slot Input/Output (Canal Entrada/Salida) es una ranura en el tablero principal de la pc, consiste en un bus o canal bidireccional de datos de 8 Bits, 20 líneas de dirección , 6 niveles de interrupción, líneas de control para lectura y escritura, líneas de reloj, 3 canales de DMA, líneas de poder (+5 v. y tierra). La figura 4.18. muestra un esquema del canal Entrada/Salida.

La comunicación entre el CPU y los puertos de I/O se realiza a través de direcciones, ya que cada puerto y dispositivo conectado al CPU cuenta con una dirección. Por ejemplo; el puerto serial Com1 está localizado en la dirección 03FH. El manual técnico

de referencia de la IBM especifica las direcciones de cada uno de los dispositivos conectados al CPU.

GND	B1	A1	IO CH CK
RESET	B2	A2	SD7
+ 5V.	B3	A3	SD6
IRQ2/	B4	A4	SD5
- 5V.	B5	A5	SD4
DRQ2	B6	A6	SD3
- 12 V.	B7	A7	SD2
OWS	B8	A8	SD1
+ 12 V.	B9	A9	SD0
GND	B10	A10	- IO CH RDY
- SMEMW	B11	A11	AEN
- SMEMR	B12	A12	SA19
-IOW	B13	A13	SA18
-IOR	B14	A14	SA17
-DACK3	B15	A15	SA16
DRQ3	B16	A16	SA15
-DACK1	B17	A17	SA14
DRQ1	B18	A18	SA13
-REFRESH	B19	A19	SA12
CLOCK	B20	A20	SA11
IRQ7	B21	A21	SA10
IRQ6	B22	A22	SA9
IRQ5	B23	A23	SA8
IRQ4	B24	A24	SA7
IRQ3	B25	A25	SA6
-DACK2	B26	A26	SA5
T/C	B27	A27	SA4
BALE	B28	A28	SA3
+ 5 V.	B29	A29	SA2
OSC.	B30	A30	SA1
GND	B31	A31	SA0

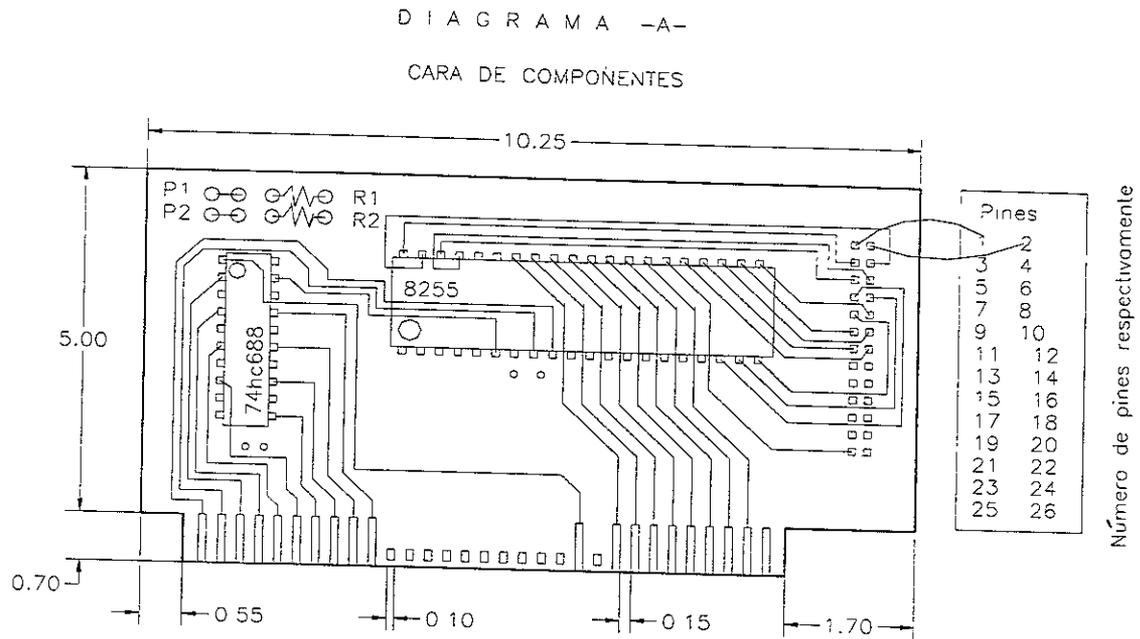
figura 4.18 Canal I/O Slot de ISA de una PC-XT.

Se tiene a continuación la una descripción de las líneas del slot de expansión ISA (International Standard Arquitectude) mostrado en la figura 4.18.

- SDO...SD7** Líneas de datos bidireccionales usadas para transmitir datos entre el microprocesador, memoria, puertos, discos, etc. La dirección de los datos depende de las señales de la micro -IOR (leer) y -IOW (escribir).
- SA0...SA19** Son salidas del BUS de direcciones del micro provistas para direccionar diferentes partes de las tarjetas conectadas al BUS.
- OSC** Es una salida de 14.31818 Mhz usada internamente para el reloj de tiempo real, interfases RS232, etc. No tiene relación con la señal de CLOCK.
- CLOCK** Es una salida de la frecuencia que usa el micro.
- RESET** Es una salida provista por el mecanismo interno de reset para resetear a las tarjetas conectadas al BUS.
- BALE** Es una salida del controlador del BUS (8088) usada para indicar cuando las señales A0..A19 son correctas. Esta señal se usa para sincronizar las direcciones con los datos.
- I/O CH CK** Es una entrada para reportar, al micro, una condición de error sobre el BUS que atañe a la interface conectada al BUS. Produce una interrupción del micro del más alto nivel (NMI).
- I/O CH RDY** Es una salida usada para extender el ciclo de espera del micro sobre una tarjeta conectada al BUS (Wait State 1,2,3,...).
- IRQ2...IRQ7** Entradas de interrupción al micro. Como el micro tiene una sola entrada de interrupción se le adiciona un controlador que posee 8 entradas, de las cuales 2 ya están usadas en la tarjeta madre (IRQ0 Y IRQ1). Se utilizan para informar al micro que requerimos su atención para pedirle o mandarle datos ejecutando un subprograma específico a cada IQR.

-
- IOR** Salida sincronizada con los datos SD0...SD7 para su ingreso al micro.
Solo para puertos.
- IOW** Salida sincronizada con los datos SD0...SD7 para su egreso del micro.
Sólo para puertos.
- SMEMW -SMEMR** Idem anteriores, pero para direccionar memoria.
- DRQ1...DRQ3** Entradas para pedir un ciclo de DMA (Acceso directo a memoria), el método más rápido de acceso a memoria.
- DACK0...DACK3** Salida del 8253 (Controlador DMA) para el reconocimiento de un ciclo DMA.
- AEN** Señal de salida que indica, en nivel bajo (0 volts), la dirección válida de acceso a memoria o port a través del BUS. En nivel alto (5 volts), indica que se está realizando un ciclo DMA.
- TC** Salida que indica la terminación de un ciclo DMA; ya sea de un bloque o de un carácter.

La figura 4.19. representa la distribución de los componentes en la Tarjeta de Interfase.



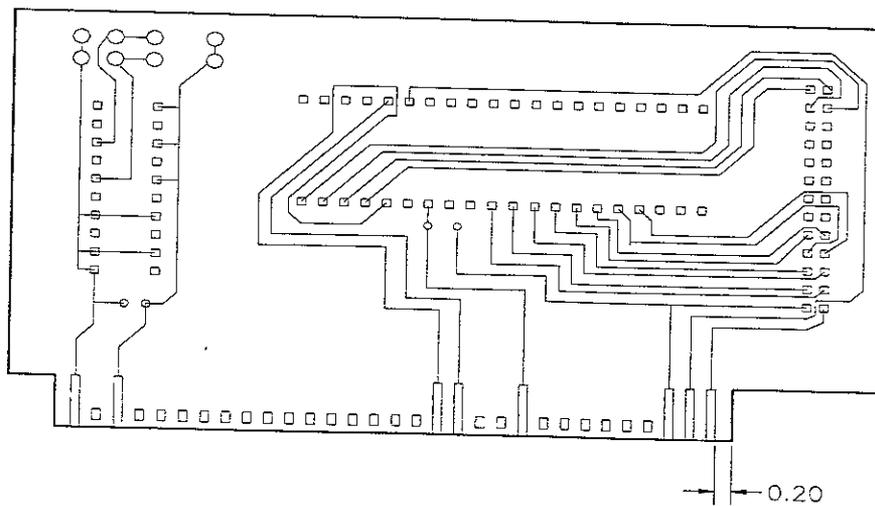
Acot. mm
Esc: 1:1

Figura 4.19	Tarjeta de Interfase
-------------	----------------------

D I A G R A M A -B-

CARA DE SOLDADURAS

VISTA INVERTIDA EN ESPEJO



Acot. mm
Esc. 1:1

Figura 4.20	Tarjeta de Interfase.
-------------	-----------------------

La figura 4.20. Muestra la cara posterior de la Tarjeta de Interfase de 24 bits de información, capaz de soportar los requerimientos del proyecto. p1 y p2 son puentes que conjugados con el código de programación permiten programar la tarjeta de interfase, r1 y r2 las resistencias con un valor de 220 ohms cada una.

4.2.5 Descripción de la tarjeta de interfase.

La tarjeta de Interfase en su parte material esta conformada por:

- a) Una Tarjeta con un circuito impreso de doble cara.
- b) Dos circuitos integrados una Interfase programable 8255A-5 Intel y un multiplexor 74HC688.
- c) Un conector de peine doble hilera 26 pines.
- d) Una tira doble de 26 pines.
- e) Un cable plano de 26 vías con dos conectores.

El circuito integrado 74HC688 permite el direccionamiento y la activación de la interfase. El circuito 8255 consta de 24 Bits (líneas de entrada/salida). En su parte lógica, la tarjeta de Interfase se maneja a través de comandos de control enviado por medio de instrucciones en diversos lenguajes como : basic, pascal, c, prolog, etc. Estos comandos sirven para programar a los puertos en sus distintas modalidades (ver tabla 4.4), así como para la recepción y el envío de datos desde la computadora hacia los dispositivos externos.

La tarjeta de Interfase debe ser insertada en una ranura de expansión o slot dentro del gabinete de la computadora para lo cual deben seguirse los siguientes pasos:

- 1.- Apagar y desconectar de la toma de corriente a la PC.
- 2.- Abrir el gabinete de la PC.
- 3.- Localizar una ranura o slot desocupado.

- 4.- Retirar la tira metálica que protege a la ranura del exterior.
- 5.- Conectar el cable plano de 26 vías en la tira de pines de la tarjeta de interfase.
- 6.- Insertar la tarjeta de interase en la ranura seleccionada, con el lado de los componentes situado de modo similar a las demás tarjetas existentes.
- 7.- Introducir el extremo libre del cable plano hacia la parte posterior de la PC a través del hueco de la tira metálica anteriormente retirada
- 8.- Colocar nuevamente la tira metálica citada, si es posible hacerlo sin maltratar el cable plano; de otro modo dejar el hueco.
- 9.- Cerrar el gabinete y probar la interfase.

TABLA 4.4

Descripción de los hilos de la tira de doble de pines.

PUERTO A.	Pines 1 a 8	Líneas	A0 a A7
PUERTO B.	Pines 9 a 16	Líneas	B0 a B7
PUERTO C.	Pines 17 a 24	Líneas	C0 a C7
+ 5 V.	Pin 25	(corriente de 100/200 mA)	
Tierra	Pin 26	(limitar corriente 100 mA)	

Para el manejo de circuitos de potencia es conveniente emplear optoacopladores para prevenir daños a la computadora o introducción de ruidos.

Ahora bien, desde Quick Basic (lenguaje de programación) se puede hacer uso de los puertos con las instrucciones OUT e INP (ver tabla 4.4). En el caso de la interfase estas instrucciones se utilizan para programar su funcionamiento y además para enviar o recibir señales.

Para aclarar más este punto se dan a continuación algunos ejemplos de programación de entrada y salida de datos.

Ejemplo 1.- Definir los puertos A y B como salidas; el puerto C como entrada.

OUT 771,137

En donde 771 es la dirección del registro de control, con Pte 1 y Pte 2 abiertos, y 137 es el valor que define las funciones de los puertos A,B,C.

Ejemplo 2.- Definir al puerto A como salida; al puerto B como entrada; finalmente, al puerto C en su parte superior (bits 4 a 7) como entrada y en su parte inferior (bits 0 a 3) como salida.

OUT 775,138

En donde 775 es la dirección del registro del control con Pte 1 cerrado y Pte 2 abierto.

Ejemplo 3.- Enviar un dato a un puerto definido como salida.

OUT 768,63

En donde 768 es la dirección del puerto A y 63 un valor que saldrá por las líneas A0 a A7 en forma de código binario. Es decir 00111111.

Ejemplo 4.- Leer un dato de un puerto definido como entrada.

LET Z = INP(770)

En donde INP es la instrucción de lectura de un puerto, 770 es la dirección del puerto c y LET Z es la instrucción que asigna a una variables el resultado de la lectura.

Después de que se trataron los principios de funcionamiento de los motores, drives, etc. ahora se presenta la aplicación general en donde se logra integrar y describir la aplicación para controlar un motor, las partes que forman parte de la aplicación son: una fuente de poder, una tarjeta de interfase, dos manejadores(driver), dos motores, ocho transistores de potencia, se empezará por describir la fuente de poder;

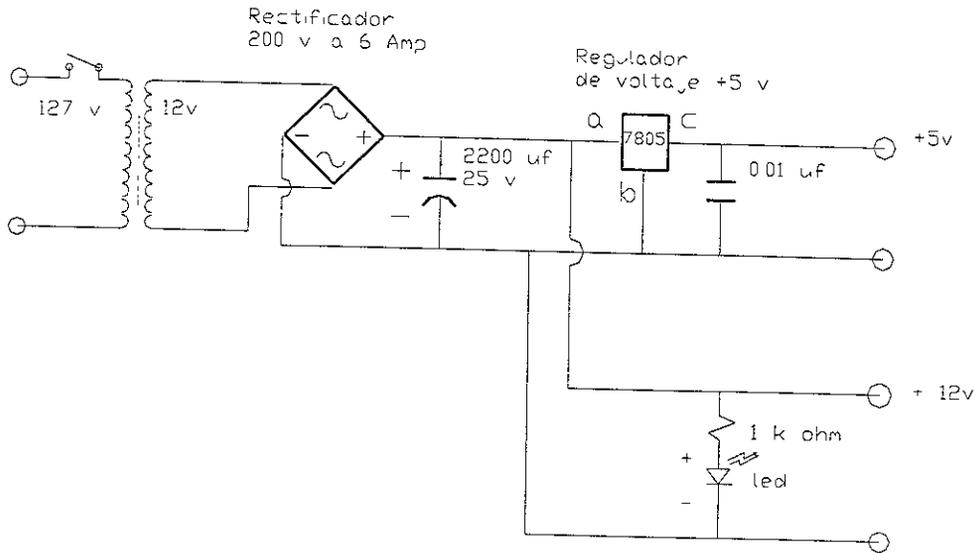


Figura 4.21 Fuente de poder para alimentar los motores del robot-manipulador

La figura 4.21 muestra esquemáticamente la fuente de poder que alimenta a los motores y etapas de potencia con un transformador de 5 Amp con tap central. Debido a que cada motor consume 1.5 Amp. A 12 v. Son dos motores, por lo tanto se requieren de $2 \times 1,5 \text{ Amp.} = 3.0 \text{ Amp}$ y se tiene un excedente de amperaje para la alimentación de las etapas de potencia. Se puede observar que la fuente suministra +5 v regulados que alimentará a los manejadores (Drives).

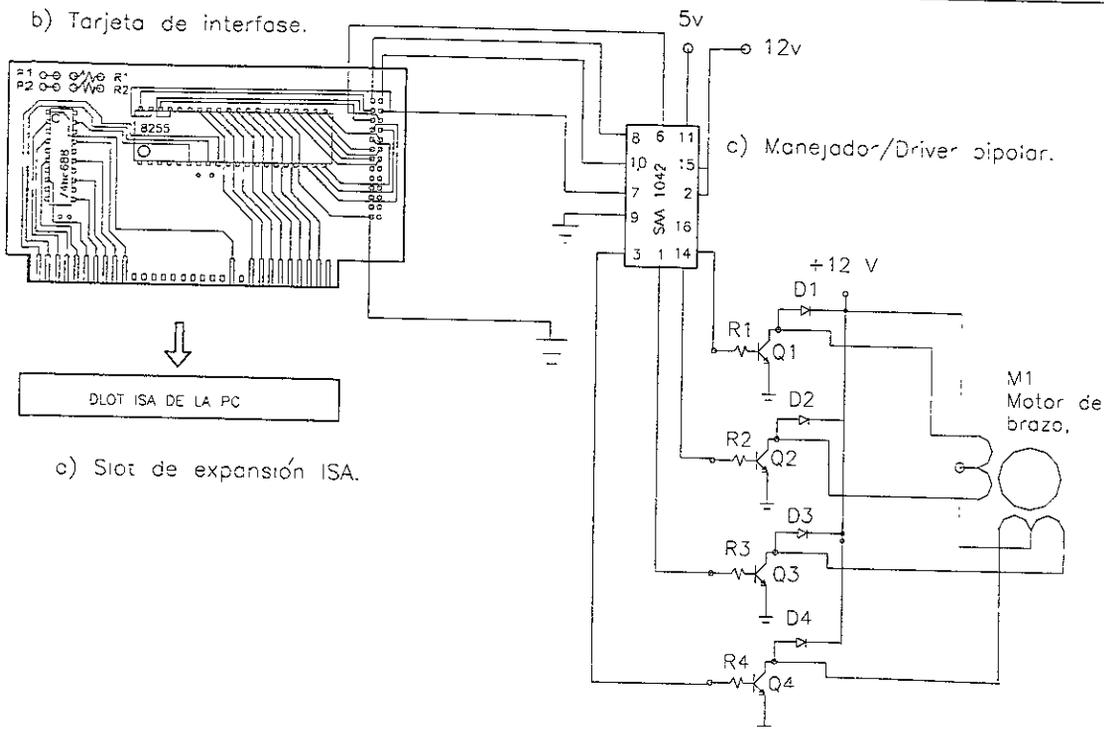


Figura 4.22 Diagrama que representa la aplicación completa para el motor M1

En la figura 4.22 se presenta la aplicación completa que permite el control de el motor M1 que es un motor Step-Syn. Type 103-771-1242. 12 v. A 1.5 Amp. Con un ángulo de paso de 1.8° grados/paso. Los transistores Q1,Q,2,Q3,Q4 alimentan las bobinas del motor con un amperaje de hasta 3 Amp estos son del tipo TIP 121. Las resistencias R1,R2,R3,R4 son de 4k7 ohms.

El manejador(driver) envia hasta 500 mA a sus salidas para excitar a los transistores y hacer que conduzcan. El otro manejador que controla el motor M2 debe tener una configuración similar al del motor M1. Con la finalidad de evitar sobrecalentamiento en los transistores TIP 121, éstos deben ir montados cada uno en un disipador de calor.

La tarjeta es comandada desde un programa de control que será el que le envíe la información ya procesada en forma de pulsos a los manejadores y éstos generarán el campo giratorio aparente necesario para dar movimiento a los motores.

4.3 Programa de control del robot-manipulador de dos grados de libertad.

En este capítulo se desarrolló el programa que permitirá coordinar y enviar la información ya procesada en forma de pulsos a los manejadores, el programa fue hecho con el lenguaje de programación lenguaje C++[®], este lenguaje ofrece mucha flexibilidad para acceder a los puertos de la PC, enviar y recibir información a través de sus líneas de comunicación (bus), así como también, la facilidad de ser modificado para futuras mejoras.

De la figura 4.0c el software (programa) que coordina el número y la duración de los pulsos enviados a los manejadores que controlan los motores M1 y M2 se presenta en la figura 4.24. Las características del programa son;

- a) El cuadro de diálogo (ver figura 4.24) permite enviar pulsos a los elementos manejadores y estos pulsos van registrándose en archivos.
- b) Otra característica importante es la capacidad de leer un archivo de texto en donde se han introducido manualmente los valores resultados de la solución de la ecuación que gobierna la cinemática del robot con Mathematica[®], esto con el fin de alcanzar un punto de alguna trayectoria específica.
- c) También permite cambiar el ángulo de paso de 1.8° a 0.9° grados, es decir, de modo de Full step/paso completo a Half step/medio paso.
- d) El programa da la flexibilidad de operar en tiempo real los movimientos de los motores, modificar o editar antes de registrar la información.
- e) Envía al robot a una posición de "home" que representa la configuración no deformada.
- f) Se puede variar la velocidad de los motores en forma independiente permitiendo observar los movimientos lenta o rápidamente.

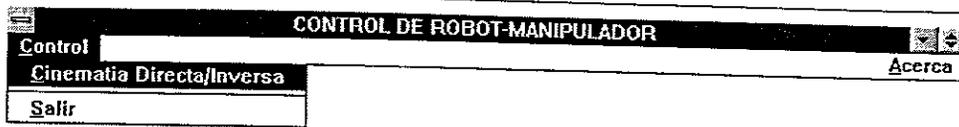


Figura 4.23	Menú del programa de control.
-------------	-------------------------------

En la figura 4.23 se muestra el menú desplegable "Control" que al ser oprimido permite seleccionar el submenú "cinemática Directa/Inversa." que despliega el cuadro de diálogo que se muestra en la figura 4.24. El submenú "salir" termina una sesión y sale del programa, además el submenú de "Acerca" muestra los datos personales del creador del programa. El programa fue desarrollado en el ambiente de trabajo de windows 3.11.

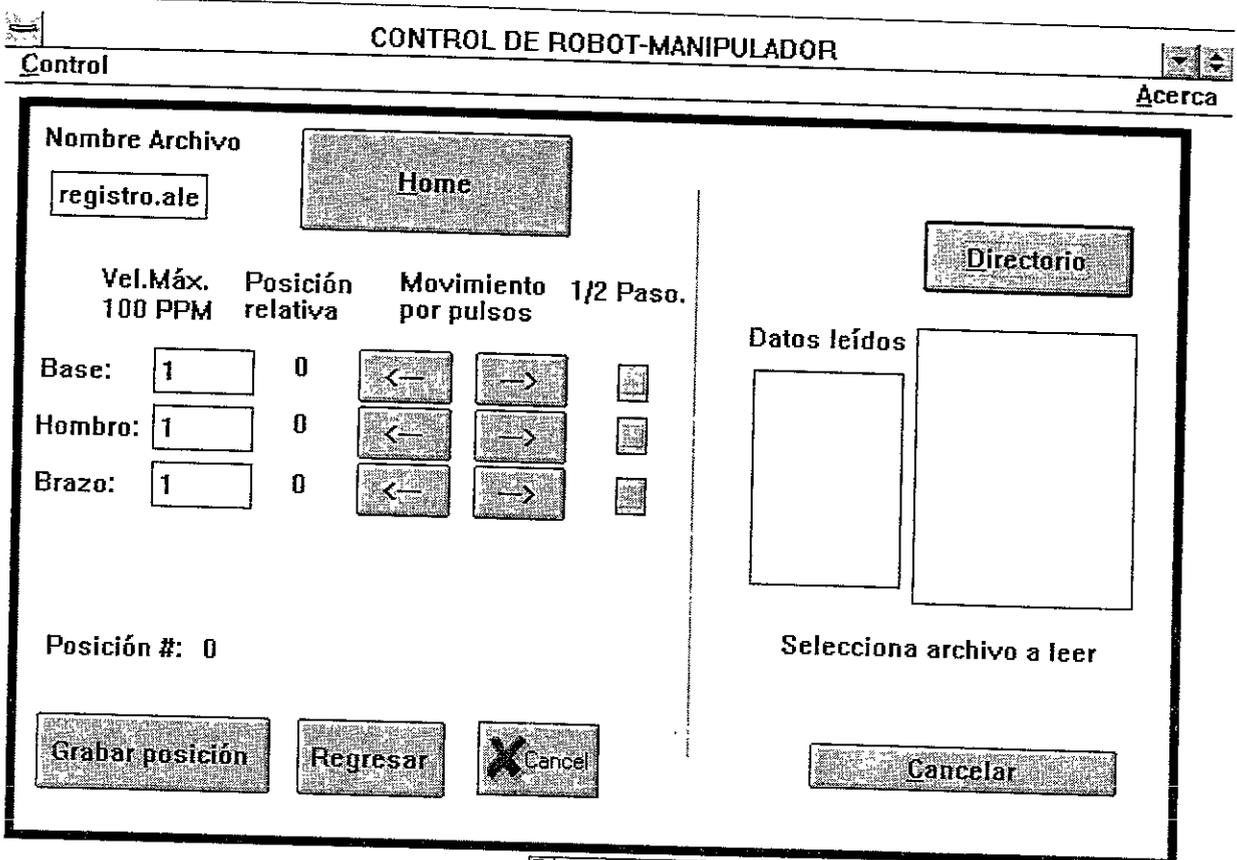


Figura 4.24 Cuadro de diálogo para la escritura y lectura de archivos.

El cuadro de diálogo mostrado en la figura 4.24 representa el módulo principal de operación del programa, este módulo se divide en dos partes; la primera parte está a la izquierda de la línea vertical, aquí se escriben los archivos con las secuencia de movimiento deseada. En la parte de la izquierda se leen los archivos y se ejecutan los movimientos del robot en forma secuencial.

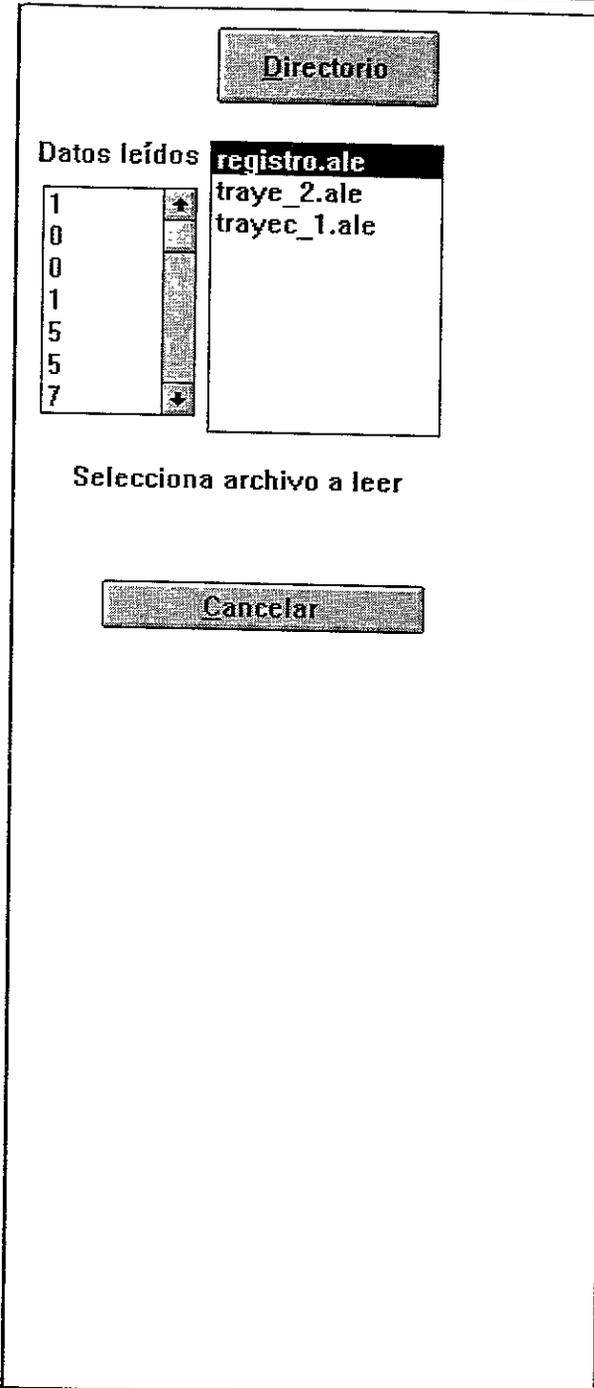
Para dar una descripción más detallada de los controles que contiene el programa se creó la tabla 4.5 la cual muestra la función y la forma de activar cada uno de los controles (botones, cuadro de edición, radio botón, etc.) del programa de control.

Tabla 4.5

Descripción de las funciones de los controles del programa de control.

Control	Descripción
<p>Nombre Archivo</p> <p><input type="text" value="registro.ale"/></p>	<p>Es importante teclear el nombre del archivo antes de iniciar una sesión de control de posición del robot.</p>
<p><input type="button" value="Home"/></p>	<p>El botón de "Home" hace que el robot tome una posición predefinida que será la configuración no deformada, con el fin de poder aplicar la cinemática inversa y llevar al robot a una posición o punto de alguna trayectoria en particular.</p>
<p>Vel.Máx. P_c 100 PPM re</p> <p>Base: <input type="text" value="100"/></p> <p>Hombro: <input type="text" value="150"/></p> <p>Brazo: <input type="text" value="200"/></p>	<p>Los cuadros de edición que se presentan en la figura de la izquierda permiten teclear la velocidad deseada de rotación de cada motor por separado. Es decir, se pueden hacer mover los eslabones a velocidades diferentes.</p>
<p>Posición relativa</p> <p>1</p> <p>11</p> <p>15</p>	<p>Estas leyendas van registrando el número de pulsos enviados a los motores paso a paso que imprimen movimiento los eslabones de la base, hombro y brazo respectivamente.</p>
<p>Movimiento por pulsos</p> <p><input type="button" value="←"/> <input type="button" value="→"/></p> <p><input type="button" value="←"/> <input type="button" value="→"/></p> <p><input type="button" value="←"/> <input type="button" value="→"/></p>	<p>Los pares de botones de arriba hacia abajo corresponden al control de pulsos de la base, hombro y brazo respectivamente, si se envía un pulso con algún botón de la columna derecha entonces el motor girará en sentido antihorario, y si se envía un pulso con</p>

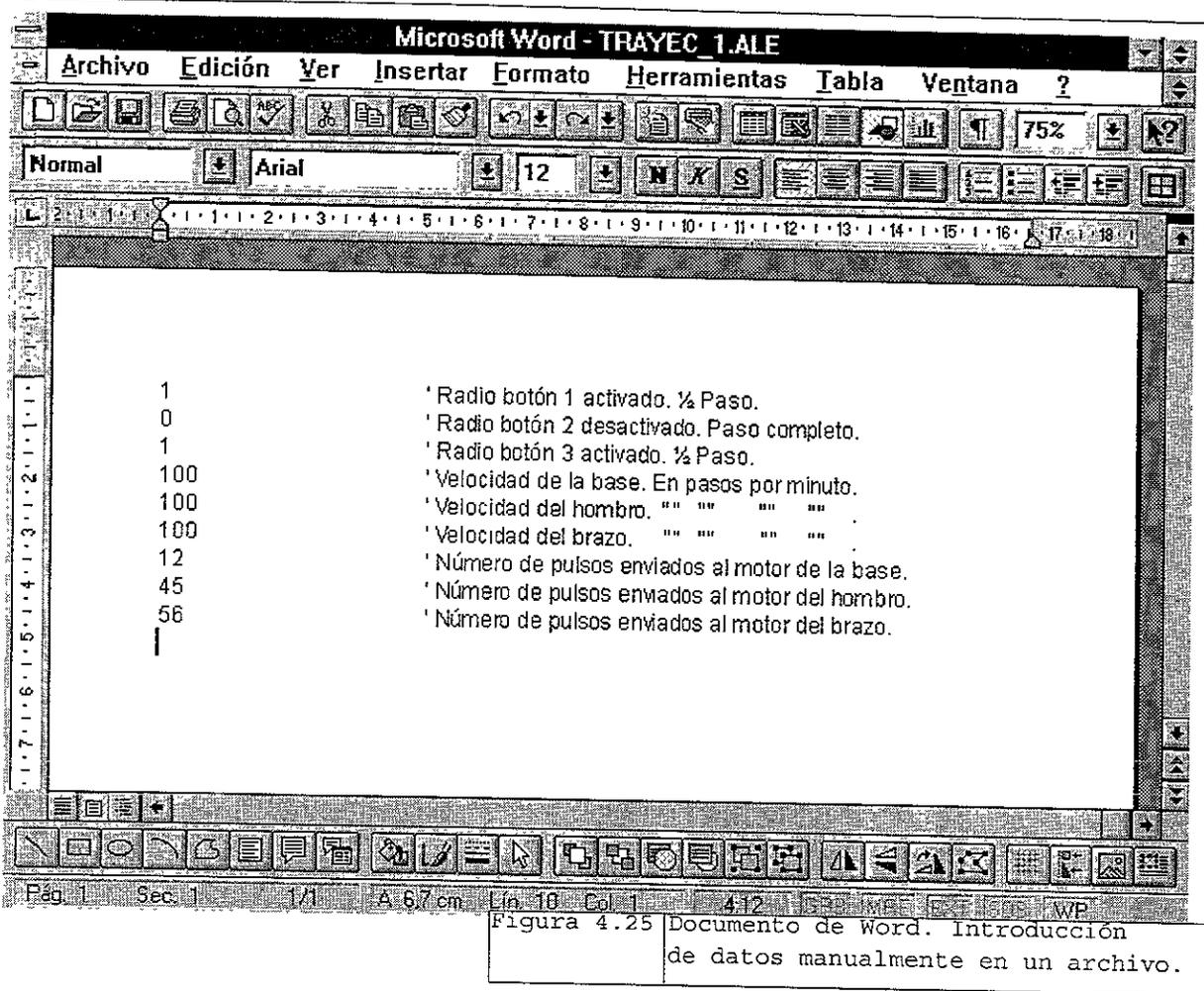
	<p>alguno de los botones de la columna izquierda el motor girará en sentido horario al de las manecillas del reloj. Con esta implementación se puede ir controlando las posiciones del robot.</p>
<p>1/2 Paso.</p> <p><input checked="" type="checkbox"/></p> <p><input type="checkbox"/></p> <p><input checked="" type="checkbox"/></p>	<p>Los radio botones mostrados de arriba hacia abajo corresponden a la base, hombro y brazo respectivamente, el primero está encendido indicando que la base va ir avanzando a medio paso, es decir el ángulo de paso pasará de paso completo a medio paso, que es de 0.9° grados por cada pulso enviado. El segundo esta desactivado indicando que avanzará a paso completo en este caso 1.8° grados de rotación del eje del motor por cada pulso. El tercer radio botón hará que el motor paso a paso gire a medio Paso.</p>
<p>Posición #: 0</p> <p>Grabar posición</p>	<p>El botón "Grabar posición" después de haber construido la secuencia deseada graba la secuencia en el archivo abierto, añadiendo cada secuencia posteriormente. La leyenda "Posición #:" va indicando el número de posiciones registradas.</p>
<p>Regresar</p> <p>X Cancel</p>	<p>El botón "cancelar" cancela la posición anterior y permite rehacer la secuencia. El botón "Cancel" ejecuta la salida del programa.</p>



El botón con la leyenda de "Directorio" al ser oprimido despliega los nombres de los archivos existentes en el cuadro de lista de abajo, esto da la opción de elegir con el "mouse" el archivo deseado, posteriormente se despliegan los datos del archivo en el cuadro de lista de la izquierda ejecutándose el movimiento del robot al mismo tiempo. Es importante aclarar que los archivos pudieron ser grabados manualmente como se muestra en la figura 4.23 desde algún procesador de textos, esto se hace con el fin de poder vaciar manualmente los datos provenientes de las soluciones arrojadas de la modelación matemática y simulación de la cinemática inversa realizada con el software de cálculo formal Mathematica®, que es en realidad el objetivo principal de este trabajo.

El botón "Cancelar " cancela la opción anterior y sale del programa sin registrar cambio alguno.

La parte de mayor importancia ahora es la de describir como se capturan los datos manualmente en los archivos



Se muestra en la figura 4.25 el orden en que se deben introducir los datos que vienen de la solución de las ecuaciones con la ayuda del software de cálculo formal Mathematica®, que serán los datos del número de pulsos que recibirá cada motor. Los datos de paso completo y medio paso son dados por el usuario y también las velocidades. Sin embargo, cabe aclarar que los valores de las soluciones dados por mathematica® están en forma de parámetros, y éstos deben ser convertidos a sus equivalentes en ángulos. Se logra la conversión utilizando la ecuación [4.9]. Cada grado representa 10 pulsos enviados a los motores correspondientes. Es decir, si tenemos $\theta = 10^\circ$ entonces el programa de control enviará al motor correspondiente 100 pulsos, ya que la relación de engranaje es $R_e = 9$, y la resolución por pulso es de 0.1° de rotación para el engrane que lleva acoplado el eslabón.

Sea $P = (P_0, P_1, P_2, P_3) \in Q$ está fijo, en [1] [4.6]

Donde Q es el espacio vectorial de Cuaterniones.

sus componentes tienen la siguiente interpretación física:

$$P_0 = \|P\| \cos(\theta/2), \quad [4.7]$$

$$P_v = \|P\| \sin(\theta/2) v, \quad [4.8]$$

donde θ es el ángulo de rotación, v es un vector unitario del eje de rotación y P_v $T_v(0, P_1, P_2, P_3)$ la parte vectorial del Cuaternion $P \in Q$. En las ecuaciones [4.7] [4.8] se observa que si $\|P\| = 1$, entonces se obtienen los parámetros de Euler. Por lo tanto, de la ecuación [4.7] se puede obtener el ángulo de rotación;

$$\theta = \frac{2 (\cos^{-1} P_0)}{F.C} \quad [4.9]$$

Donde $F.C = \frac{\pi}{180^\circ}$; es el factor de conversión de radianes a grados.

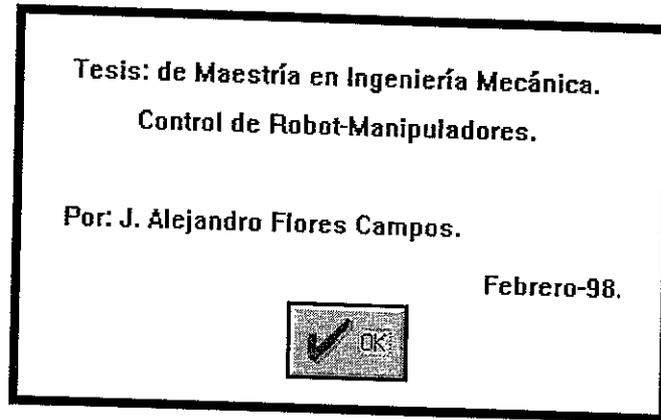


Figura 4.26 Cuadro de diálogo "Acerca".

El cuadro de diálogo de la figura 4.26 muestra los datos del programador y hace referencia a los motivos de la creación del programa.

CONCLUSIONES

La culminación de este trabajo arrojó resultados satisfactorios con respecto a la Modelación Cinemática, si se cumple en la parte experimental. Además se concluyó lo siguiente:

a) A lo referente del seguimiento de los puntos de trayectorias (Cinemática Inversa) como lo son las líneas rectas, arcos de circunferencia, etc. la trayectoria que realiza el órgano terminal del robot para llegar desde un punto a otro punto es poco predecible cuando se trata de distancias relativamente grandes. Es decir el órgano terminal del robot llega al punto deseado, sin embargo la trayectoria que sigue no precisamente es el de un segmento de línea recta. Cuando en realidad se trata de seguir los puntos de una recta, este órgano traza trayectorias semicirculares.

b) Entre más lejanos sean los puntos a alcanzar desde un punto de vista macroscópico serán las trayectorias para llegar, diversas. Es necesario entonces que los puntos sean suficientemente cercanos para conseguir un efecto deseado, y esta cercanía estará en función de las actividades que el robot realice.

Observación 1. Una observación importante de la Modelación Cinemática es que todas las configuraciones deformadas parten de la configuración no deformada a la cual le hemos llamado "home", y el robot al alcanzar la configuración deformada (1) desde la configuración no deformada (home) físicamente tiene ahora la configuración deformada (1) y debe partir de ahí para poder llegar a la siguiente configuración. Por lo tanto, debemos realizar cálculos extras para determinar los ángulos relativos de los eslabones con respecto a la siguiente configuración a alcanzar.

Observación 2. El control de el Robot se llevó a cabo a través de dispositivos electrónicos disponibles en el mercado. Los motores que se usaron son seminuevos y para la fabricación del robot algunas de sus piezas fueron partes de impresoras mecánicas. Se pueden seguir haciendo estudios experimentales que nos permitan llegar a un control más robusto, es decir, de mayor precisión y control.

CONCLUSIONES

Es importante mencionar las siguientes mejoras :

Para evitar el calentamiento excesivo de ,los motores se pueden agregar resistencias a las bobinas para reducir la corriente.

En el aspecto de transmisión los engranes pueden tener una relación mayor para minimizar la vibración del prototipo.

La tarjeta de interfase puede ser sustituida por un cable que vaya directo a la salida de la impresora vía puerto paralelo, esto es con la finalidad de evitar abrir la computadora para conectar al Robot.

Se pueden integrar sensores de posicionamiento como son : potenciómetros, optoacopladores para controlar la posición de home por ejemplo.

Y sobre el aspecto cinemático atacar el problema de el seguimiento de trayectorias rectas de un punto a otro a través de algoritmos de interpolación..

APENDICE "A"**TERMINOLOGIA**

DETENT TORQUE.- Es el máximo Torque que puede ser aplicado al eje del Motor paso a paso (PAP) cuando no se encuentra excitado, y sin causar un movimiento continuo de rotación. UNIDAD. mNm.

DESVIACION.- El cambio en la posición del eje de un Motor paso a paso (PAP) desde una posición sin carga hasta cuando un cierto Torque es aplicado al eje, cuando el Motor paso a paso (PAP) se encuentra excitado, UNIDAD grados.

HOLDING TORQUE.- El máximo Torque que puede ser aplicado al eje de un Motor paso a paso (PAP) cuando éste se encuentra excitado y sin causar un movimiento continuo de rotación. UNIDAD mNm.

MAXIMO PULL-IN RATE (Velocidad). La máxima tasa de conmutación del interruptor electrónico a la cual el Motor paso a paso (PAP) descargado puede ser puesto en marcha sin perder un Paso. UNIDAD. Pasos/seg. o Rev/min.

MAXIMO PULL-OUT RATE (Velocidad). La máxima tasa de conmutación del interruptor que un Motor paso a paso (PAP) sin carga puede seguir sin perder Pasos. UNIDAD. Pasos/seg o rpm.

MAXIMO TORQUE DE TRABAJO.- El máximo Torque que puede ser obtenido del Motor paso a paso (PAP) UNIDAD. mNm.

OVERSHOOT. La máxima amplitud de oscilación alrededor de la última posición del Rotor después del último pulso de conmutación de interruptores electrónicos. UNIDAD grados.

PERMANENTE OVERSHOOT. El número de Pasos que el Rotor se mueve después del último pulso de conmutación del interruptor electrónico. UNIDAD. grados.

PHASE.- Número de Fases del Motor paso a paso (PAP).

PULL-IN RATE. La máxima tasa de conmutación de interruptores en la cual un Motor paso a paso (PAP) con carga puede ponerse en marcha sin perder Pasos. UNIDAD. Pasos/seg. rpm.

PULL-IN TORQUE. El máximo Torque que puede ser aplicado al eje de un Motor paso a paso (PAP) cuando es puesto en marcha dentro de PULL-IN RATE. UNIDAD. mNm.

PULL-OUT TORQUE . El máximo Torque que puede ser aplicado al eje de un Motor paso a paso (PAP) cuando se mueve en dentro del PULL-OUT RATE. UNIDAD. mNm.

STAR RANGE.- Rango de tasas de conmutación de interruptores electrónicos entre las cuales el Motor paso a paso (PAP) puede ser puesto en marcha sin perder Pasos.

STEP ANGLE. El ángulo nominal que el eje del Motor paso a paso (PAP) da entre Pasos adyacentes. UNIDAD. grados.

SLEW RANGE. El rango de las tasas de conmutación de interruptores entre las cuales el Motor paso a paso (PAP) puede correr en una misma dirección y seguir esa tasa de conmutación de interruptores (Máxima cambio de velocidad), sin perder Pasos, pero no puede ser puesto en marcha, parar, o ir en sentido contrario.

Nota: Temperaturas máximas de un Motor paso a paso (PAP).

- *El incremento en la temperatura en un Motor paso a paso (PAP) depende de la potencia que éste consume. Los Motor paso a paso (PAP) Unipolares operan por lo general con una tensión de alimentación baja y tiene una baja entrada de potencia, lo que limita el incremento de la temperatura en el Motor paso a paso (PAP). si los Motores paso a paso (PAP) con controles Unipolares funcionan con baja temperatura ambiental y una alta tensión de alimentación, puede ser permisible demandarles un alto Torque.*

- *Los Motores paso a paso (PAP) que utilizan un controlador Bipolar de corriente constante, funcionan con una baja tensión de alimentación y tienen una elevada demanda de potencia que causa un gran incremento de temperatura en el Motor paso a paso (PAP). Por lo que es recomendable montarlos en una base que funcione como disipador de calor o instalar un sistema de enfriamiento forzado.*

A temperatura ambiente arriba de 25° C. el Torque del Motor paso a paso (PAP) Unipolar y Bipolar, decrece en un 2% por grado Kelvin, debido a cambios en la resistencia interna de las bobinas. También se dá con bajas temperaturas ambientales.

La inestabilidad de los Motores paso a paso (PAP) puede ocurrir bajo ciertas circunstancias. El momento de inercia del Rotor y su carga, junto con el magnetismo remanente pueden provocar los siguientes comportamientos.

a.- Resonancia a bajas velocidades de Pasos.

b.- Variaciones alrededor de la velocidad requerida.

Las áreas inestables están indicadas por las líneas punteadas en las curvas de comportamiento que aparece en esta sección. La resonancia puede ser minimizada aplicando una correcta fuerza de fricción en el eje del rotor.

APENDICE "B"

PROGRAMA DE MATHEMATICA. CINEMÁTICA INVERSA de un Manipulador de 2 GDL.	
Comandos	Descripción
<pre>QuaterM[P_,Q_]:= {P[[1]] Q[[1]]-P[[2]] Q[[2]]- P[[3]] Q[[3]]-P[[4]] Q[[4]], P[[2]] Q[[1]]+P[[1]] Q[[2]]-P[[4]] Q[[3]] +P[[3]] Q[[4]], P[[3]] Q[[1]]+P[[4]] Q[[2]]+P[[1]] Q[[3]] -P[[2]] Q[[4]], P[[4]] Q[[1]]-P[[3]] Q[[2]]+P[[2]] Q[[3]] +P[[1]] Q[[4]]}; QuaterCon[P_]:= {P[[1]],-P[[2]],-P[[3]],-P[[4]]} Rotar[P_,Q_]:=QuaterM[P,QuaterM[Q,QuaterCon[P]]];</pre>	<p>Ecuaciones paramétricas de rotaciones finitas. Operación multiplicación.</p> <p>El conjugado de $P \in Q$.</p> <p>La transformación de rotación.</p>
----- CONFIGURACION NO DEFORMADA -----	
<pre>ClearAll[P,Q,rpnd]; e1 = {0,1,0,0}; l1 = 2; l2 = 2; ang1=45; ang2=90; P0=Cos[ang1/2 Degree]; P2=-Sin[ang1/2 Degree]; Q0=Cos[ang2/2 Degree]; Q2=-Sin[ang2/2 Degree]; P = {P0,0,P2,0}; Q = {Q0,0,Q2,0}; e11 =Rotar[P,e1]; e21 =Rotar[Q,e11]; b1 = l1*e11; b2 = l2*e21; rpnd = b1+b2; gra=Graphics3D[{ {Cuboid[{-.7,-.7,-.5},{.7,.7,0]} }, {AbsoluteThickness[12.2],RGBColor[1,0,1], Line[{{0,0,0},{b1[[2]],b1[[3]],b1[[4]]}}] }, {AbsoluteThickness[8.2],RGBColor[0,1,1], Line[{{b1[[2]],b1[[3]],b1[[4]]},{rpnd[[2]],rpnd[[3]],r pnd[[4]]}]}]</pre>	<p>Limpiando variables. Definiendo la base global fija. Definición de las longitudes de los eslabones. Ángulos de la configuración no deformada. Definiendo los componentes de los parámetros P y Q.</p> <p>Rotando las bases locales con respecto a la base global fija.</p> <p>Generando la gráfica de la configuración no deformada. Línea del primer eslabón.</p> <p>Línea que representa al segundo eslabón.</p>

<pre> }); Show[{gra}, PlotRange->{{-3,3},{-3,3},{-1,6}}, Axes->True,AxesLabel- >{"x","y","z"},AspectRatio->1/1, ViewPoint->{0.008,-3.384,-0.041}]; Print["CONFIGURACION NO DEF ORMADA"]; Print[""]; Print[" P= ",N[P]," Ang1= ",N[(360*ArcSin[P[[3]]])/Pi]]; Print[" Q= ",N[Q]," Ang2= ",N[(360*ArcSin[Q[[3]]])/Pi]]; Print[" Vector posición = ",rpd//N]; </pre>	<p>Mostrando la gráfica del robot de 2 grados de libertad. Atributos de la imagen.</p> <p>Imprimiendo texto con los valores de los ángulos, vector posición y parámetros.</p>
<p>----- CONFIGURACION DEFORMADA -----</p>	
<pre> ClearAll[p0,p2,q0,q2]; p = {p0,0,p2,0}; q = {q0,0,q2,0}; a11 = Rotar[p,e11]; a21 = Rotar[q,e21]; d1=l1*a11; d2=l2*a21; rdd=Simplify[d1+d2]//N; </pre>	<p>Limpiando variables. Definiendo los parámetros p y q.</p> <p>Rotando las bases locales de las configuraciones deformadas. Construyendo la ecuación del vector posición.</p>
<p>----- GENERANDO TRAYECTORIA -----</p>	
<pre> ClearAll[sq1,sq2,raiz,x]; For[x=0,x<4,x++, px=-2+x; pz=1.1+(x*.6); px1[x]=-2+x; pz1[x]=1.1+(x*0.6); raiz[x]=FindRoot[{rdd[[2]]==px,rdd[[4]]==pz,p0^2+p2^2==1,q0^2+ q2^2==1}, {p0,1},{p2,0},{q0,1},{q2,0}, MaxIterations->100]; sq1[x]={p0,p2}/.raiz[x]; sq2[x]={q0,q2}/.raiz[x];]; </pre>	<p>Limpiando variables. Empezando ciclo for. Ecuación paramétrica de una recta.</p> <p>Comando para resolver los sistemas de ecuaciones no lineales del tipo polinomial.</p> <p>Cerrando el ciclo for.</p>
<p>----- RUTINA DE GRAFICACION -----</p>	

```

i=1;
For[x=0,x<4,x++,
  p0=sq1[x][[1]];
  p2=sq1[x][[2]];
  q0=sq2[x][[1]];
  q2=sq2[x][[2]];

  punto[i]=Graphics3D[{
    {PointSize[0.02],RGBColor[0,0,1],
    Point[{rdd[[2]],rdd[[3]],rdd[[4]]}],
  }

  If[x>0,{ AbsoluteThickness[1],RGBColor[0,0,1],
    Line[{{px1[x-1]/N,0,pz1[x-1]/N },{px1[x]/N
,0,pz1[x]/N }} ],
    {AbsoluteThickness[1],RGBColor[0,0,1],
    Line[{{px1[x]/N,0,pz1[x]/N },{px1[x]/N
,0,pz1[x]/N }} ] }
    ];
  gra2=Graphics3D[{
    {AbsoluteThickness[12.2],RGBColor[1,0,1],
    Line[{{0,0,0},{d1[[2]],d1[[3]],d1[[4]]}} ],
    {AbsoluteThickness[8.2],RGBColor[0,1,1],
    Line[{{d1[[2]],d1[[3]],d1[[4]]},{rdd[[2]],rdd[[
3]],rdd[[4]]}} ],
    {Cuboid[{-7,-7,-.5},{7,.7,0}]
    },
  PlotRange->{{-3,3},{-3,3},{-1,6}},AspectRatio-
>1/1,
  Axes->True,AxesLabel->{"x","y","z"},
  ViewPoint->{0.008,-3.384,-0.041}
  ];
  Show[{gra2,gra,Table[punto[t],{t,1,i}]}(*,Paramet
ricPlot3D[{-2+x,0,1.1+ x*0.6},{x, 0,4}]*)
  ];
  Print [" CONFIGURACION DEFOR
MADA --> ",i];
  Print["      "];
  Print["          p= ",N[p],"      Ang1= ",N[
(360*ArcSin[ p[[3]] ])/Pi ] ];
  Print["          q= ",N[q],"      Ang2= ",N[
(360*ArcSin[ q[[3]] ])/Pi ] ];
  Print["      Vector posición = ",rdd//N ];
  i++;};

```

inicializando i = 1.

Abriendo ciclo for.

Transfiriendo los valores de las soluciones a los componente de los parámetros de rotación.

Generando las gráficas de puntos.

Pintando un array de puntos.

Bloque de condición.

Generando las gráficas para la simulación de las configuraciones deformadas.

Pinta una línea que representa al primer eslabón.

Otra línea que representa al segundo eslabón.

Dibuja la base del robot.

Algunas opciones y argumentos para el despliegue de los gráficos.

Muestra las gráficas en forma secuencial.

Imprime texto con los valores de los ángulos rotados, parámetros y vectores posición de las configuraciones deformadas.

Se cierra ciclo de Graficación.

Termina el programa.

APENDICE "C"

PROGRAMA DE CONTROL.	
<pre>#include <dos.h> #include <owl.h> #include <dialog.h> #include <stdio.h> #include <string.h> #include <scrollba.h> #include <window.h> #include <windows.h> #include <edit.h> #include <static.h> #include <button.h> #include <iostream.h> #include <fstream.h> #include "dialtest.h" #include "stdlib.h" #include "ctype.h" #include <checkbox.h> #include "tiempo.cpp"</pre>	<p>El siguiente listado que se muestra a la izquierda son todas las librerías necesarias para que el programa de control opere correctamente. En ellas se encuentran los comandos, directivas e instrucciones que permiten construir el programa ejecutable.</p> <p>Es importante mencionar que la programación es orientada a objetos, esta técnica de programación permite aplicar, construir e implementar con objetos (botones, ventanas, cuadro de listas, etc) , y con ello la programación en ambiente windows fue desarrollada.</p>
<pre>const WORD ID_BUTTON1 = 109; const WORD MAX_TEXTLEN = 20;</pre>	<p>Se declara y define como WORD a las variables ID_BUTTON1=109 como constante; le da un número que identifica a el botón 1 y MAX_TEXLEN=20 que fija el número máximo de caracteres de 20.</p>
<pre>ofstream es; ifstream lec;</pre>	<p>Declaración de dos objetos para manipular archivos; "es" será el objeto de escritura de datos, y "lec" de lectura de datos desde un archivo.</p>
<pre>class TTestDialog : public Tdialog { public:</pre>	<p>Se declara una instancia llamada TTestDialog: y ésta nace heredando los atributos de la clase Tdialog que se encuentra en la librería de encabezado #include <dialog.h> para construir el objeto cuadro de diálogo, class es la instrucción para lograrlo.</p> <p>Public: permite declarar variables de carácter públicas.</p>

<pre> int i,ii,iii,p1,p2,p3,nr,paso,Buf1,Buf2,Buf3; char Buffer1[5],Buffer2[5],Buffer3[5],Buffer11[5],Buffer22[5],Buffer33[5]; char Buffer44[20]; TTestDialog(PTR WindowsObject AParent, LPSTR AName) :TDialog(AParent, AName) {j=0;ii=0;iii=0;p1=0;p2=0;p3=0;nr=0;paso=1;Buf1=0;Buf2=0;Buf3=0; outportb(771,128);}; virtual void HandleButtonMsg(RTMessage Msg) = [ID_FIRST + ID_BUTTON]; virtual void HandleListBoxMsg(RTMessage Msg) = [ID_FIRST + ID_LISTBOX]; virtual void HandleButtonMsg31(RTMessage Msg) = [ID_FIRST + ID_BASE_IZQ]; virtual void HandleButtonMsg32(RTMessage </pre>	<pre> Prepara a variables de tipo entero; i,ii,iii,p1,p2,p3,nr,paso,Buf1,Buf2,Buf3; Prepara variables del tipo carácter. Buffer1[5],Buffer2[5],Buffer3[5], Buffer11[5],Buffer22[5],Buffer33[5]; char Buffer44[20]; Declara el constructor TTestDialog, éste se correrá cada vez que se ejecute el programa. Declarando las siguientes variables; i=0;ii=0;iii=0;p1=0;p2=0;p3=0;nr=0;paso =1;Buf1=0;Buf2=0;Buf3=0; Esta instrucción programa la interfase para que sus tres puertos sean de salida de datos. Declara la función virtual HandleButtonMsg(RTMessage Msg) con argumento que envia y recibe mensajes de windows (Cuando se oprime el botón izquierdo del ratón, cuando se redimensiona una ventana, cuando se actualiza algún datos, etc.) y procesa los mensajes del botón de "Home". Declara la función virtual HandleListBoxMsg(RTMessage Msg) maneja los mensajes del cuadro de lista de archivos. Declara la función virtual HandleButtonMsg31(RTMessage Msg) que es la que se activa cuando es oprimido el botón de envio de señal al motor que mueve la base en sentido horario del manipulador. Nota: Es importante mencionar que el prototipo es de dos grados de libertad, y por lo tanto este botón quedará inhabilitado hasta aplicaciones futuras. </pre>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<pre> Msg) = [ID_FIRST + ID_HOMBRO_IZQ]; virtual void HandleButtonMsg33(RTMessage Msg) = [ID_FIRST + ID_BRAZO_IZQ]; virtual void HandleButtonMsg41(RTMessage Msg) = [ID_FIRST + ID_BASE_DER]; virtual void HandleButtonMsg42(RTMessage Msg) = [ID_FIRST + ID_HOMBRO_DER]; virtual void HandleButtonMsg43(RTMessage Msg) = [ID_FIRST + ID_BRAZO_DER]; virtual void HandleButtonGrabar(RTMessage Msg) = [ID_FIRST + ID_GRABAR]; /* virtual void Checar1(RTMessage Msg) = [ID_FIRST + 301]; */ }; </pre>	<p>Declara la función virtual HandleButtonMsg32(RTMessage Msg) que procesa las señales enviadas al motor que mueve el hombro del robot-manipulador en sentido horario.</p> <p>Declara la función virtual HandleButtonMsg33(RTMessage Msg) que controla el envío de pulsos al motor que mueve el brazo en sentido horario del manipulador.</p>
<pre> class Tabout : public TDialog { public: Tabout(PTWindowsObject AParent, LPSTR AName) :TDialog(AParent, AName) {}; }; class TTestWindow : public TWindow { public: // TEdit *Edit1, *Edit2; TTestWindow(PTWindowsObject AParent, LPSTR ATitle); virtual void CMTest(RTMessage Msg) = [CM_FIRST + CM_TEST]; virtual void CMTest2(RTMessage Msg) = [CM_FIRST + CM_TEST2]; virtual void CMTest3(RTMessage Msg) = [CM_FIRST + CM_TEST3] { CloseWindow(); } virtual BOOL CanClose(); virtual void WMLButtonDown(RTMessage Msg) </pre>	<p>Se define la clase Tabout como una clase derivada de Tdialog que permite construir los cuadros de diálogo necesarios para nuestra aplicación.</p> <p>Se declaran los procedimientos de los controles como son: los botones, los cuadros de lista, los radio botones, etc. Y estos controles se localizan en esta parte del programa.</p>

```

= [WM_FIRST + WM_LBUTTONDOWN];
virtual void WMRButtonDown(RTMessage Msg)
= [WM_FIRST + WM_RBUTTONDOWN];
virtual void HandleButtonMsg(RTMessage Msg)
= [ID_FIRST + ID_BUTTON1];
/* virtual void HandleButtonMsg3(RTMessage
Msg)
= [ID_FIRST + ID_BASE_IZQ]; */
virtual void Acerca(RTMessage Msg)
= [CM_FIRST + CM_ACERCA];
}; -

```

```

class TTestApp : public TApplication {
public:
TTestApp(LPSTR AName, HINSTANCE
hInstance, HINSTANCE hPrevInstance,
LPSTR lpCmdLine, int nCmdShow) :
TApplication(AName, hInstance,
hPrevInstance, lpCmdLine, nCmdShow) {};
virtual void InitMainWindow(); };

```

```

/*void TTestDialog::Checar1(RTMessage) {
if( IsDlgButtonChecked(HWindow, 301)==1){
MessageBox(HWindow, "Checado", "RADIO
BOTON 1.", MB_OK); } }*/

```

```

void
TTestDialog::HandleButtonMsg(RTMessage) {

DlgDirList(HWindow,
"*.*",ID_LISTBOX,NULL,NULL);

/* SendDlgItemMsg(ID_LISTBOX,
LB_ADDSTRING, 0, (LONG) nomm);
SendDlgItemMsg(ID_LISTBOX,
LB_ADDSTRING, 0, (LONG) d1 );
SendDlgItemMsg(ID_LISTBOX,
LB_ADDSTRING, 0, (LONG)"Item 7"); */
}

```

```

/* void
TTestDialog::HandleButtonMsg2(RTMessage) {
MessageBox(HWindow, "Salir", "SALIR...",
MB_OK); }
void
TTestDialog::HandleButtonScroll(RTMessage){
char texto[15];
GetDlgItemText(HWindow, ID_SCROLL, texto,

```

Se construye la aplicación con una clase deriva de TApplication.

Se define la tarea de la función HandleButtonMsg(RTMessage) que despliega en un cuadro de lista los nombres de los archivos existes con las secuencias de los movimientos registrados para el robot.

Sin duda la parte más importante del programa es donde se generan todos los pulsos enviados a los motores del robot.

<pre> 15); MessageBox(HWindow, texto, "scroll", MB_OK); } */ void TTestDialog::HandleButtonMsg31(RTMessage) { p1=--i; SetDlgItemInt(HWindow, ID_BASE_POS, p1, True); GetDlgItemText(HWindow, ID_BASE_VEL, Buffer1,5); timer a(Buffer1); if(IsDlgButtonChecked(HWindow, 301)==1){ outportb(768,7); outportb(768,0); } outportb(768,255); a.run(); outportb(768,0); MessageBeep(0); } void TTestDialog::HandleButtonMsg32(RTMessage) { p2=--ii; SetDlgItemInt(HWindow, ID_HOMBRO_POS, p2, True); GetDlgItemText(HWindow, ID_BASE_VEL, Buffer2,5); timer a(Buffer2); if(IsDlgButtonChecked(HWindow, 302)==1){outportb(888,1);} outportb(768,255); a.run(); outportb(768,0); MessageBeep(0); } void TTestDialog::HandleButtonMsg33(RTMessage) { p3=--iii; SetDlgItemInt(HWindow, ID_BRAZO_POS, p3, True); GetDlgItemText(HWindow, ID_BASE_VEL, Buffer3,5); timer a(Buffer3); if(IsDlgButtonChecked(HWindow, 303)==1){outportb(888,1);} outportb(768,255); a.run(); outportb(768,0); MessageBeep(0); } </pre>	<p>El código del control de botón dirige cada vez que se pulsa, un pulso hacia el motor que mueve a la base del robot en sentido horario.</p> <p>Envía pulsos al motor que mueve al hombro del robot.</p> <p>Envía información al motor que mueve el brazo del robot en sentido antihorario.</p>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<pre> void TTestDialog::HandleButtonMsg41(RTMessage) { p1=++j; SetDlgItemInt(HWindow, ID_BASE_POS, p1, True); GetDlgItemText(HWindow, ID_BASE_VEL, Buffer11,5); timer a(Buffer11); if(IsDlgButtonChecked(HWindow, 301)==1){outportb(888,1);} outportb(768,255); a.run(); outportb(768,0); MessageBeep(0); } void TTestDialog::HandleButtonMsg42(RTMessage) { p2=++ii; SetDlgItemInt(HWindow, ID_HOMBRO_POS, p2, True); GetDlgItemText(HWindow, ID_BASE_VEL, Buffer22,5); timer a(Buffer22); if(IsDlgButtonChecked(HWindow, 301)==1){outportb(888,1);} outportb(768,255); a.run(); outportb(768,0); MessageBeep(0); } void TTestDialog::HandleButtonMsg43(RTMessage) { p3=++iii; SetDlgItemInt(HWindow, ID_BRAZO_POS, p3, True); GetDlgItemText(HWindow, ID_BASE_VEL, Buffer33,5); timer a(Buffer33); if(IsDlgButtonChecked(HWindow, 301)==1){outportb(888,1);} outportb(768,255); a.run(); outportb(768,0); MessageBeep(0); } </pre>	<p>Envía información a la base del robot que girará en sentido horario al de las manecillas del reloj, cabe aclarar que por cada pulso enviado se actualizan los contadores que permiten tener un seguimiento de los pulsos dados. Además de un pitido por pulso, esto es con el fin de llamar la atención del usuario.</p> <p>Envía información al motor pero éste hace que gire en sentido horario al hombro del robot.</p> <p>Envía información al motor que imprime movimiento al brazo del robot, en sentido horario.</p>
<pre> void TTestDialog::HandleButtonGrabar(RTMessage) { char BufferR[300]; </pre>	<p>El código del botón grabar se muestra aquí, permite registrar en archivos las secuencias ejecutadas por el robot</p>

```

if( IsDlgButtonChecked(HWindow,
301)==1){Buf1=1;}
if( IsDlgButtonChecked(HWindow,
302)==1){Buf2=1;}
if( IsDlgButtonChecked(HWindow,
303)==1){Buf3=1;}

```

```

    GetDlgItemText(HWindow,
ID_BASE_VEL, Buffer1,5);
    GetDlgItemText(HWindow,
ID_HOMBRO_VEL,Buffer2,5);
    GetDlgItemText(HWindow,
ID_BRAZO_VEL, Buffer3,5);
    GetDlgItemText(HWindow,
ID_BASE_POS, Buffer11,5);
    GetDlgItemText(HWindow,
ID_HOMBRO_POS,Buffer22,5);
    GetDlgItemText(HWindow,
ID_BRAZO_POS, Buffer33,5);
    GetDlgItemText(HWindow,
ID_NOMBRE_ARC,Buffer44,20);

```

```

strcpy(BufferR,"          Velocidad
Posición");
strcat(BufferR,"\n");
strcat(BufferR,"BASE...   ");
strcat(BufferR,Buffer1);
strcat(BufferR,"          ");
strcat(BufferR,Buffer11);
strcat(BufferR,"\n");
strcat(BufferR,"HOMBRO.  ");
strcat(BufferR,Buffer2);
strcat(BufferR,"          ");
strcat(BufferR,Buffer22);
strcat(BufferR,"\n");
strcat(BufferR,"BRAZO..  ");
strcat(BufferR,Buffer3);
strcat(BufferR,"          ");
strcat(BufferR,Buffer33);
strcat(BufferR,"\n");
strcat(BufferR,Buffer44);

```

```

if(MessageBox(HWindow,BufferR,"Esta bien ? ",
MB_YESNO|MB_ICONQUESTION)==IDYES){

```

```

    if(paso==1){
        es.open(Buffer44,ios::out);
        es <<Buf1;
        es <<"\n";

```

permitiendo ejecutarlas cuantas veces se desee. Esta característica da un amplia gama de posibilidades de repetición para estudios posteriores de los movimientos del robot.

Almacena los datos a grabar en un cuadro de mensaje.

Este cuadro de mensaje aparece para la confirmación de que los datos a registrarse son correctos.

Los datos son grabados en el disco duro.

```

es <<Buf2;
es <<"\n";
es <<Buf3;
es <<"\n";
es <<Buffer1;
es <<"\n";
es <<Buffer2;
es <<"\n";
es <<Buffer3;
es <<"\n";
es <<Buffer11;
es <<"\n";
es <<Buffer22;
es <<"\n";
es <<Buffer33;
es <<"\n";
es.close();
paso=0;
}
else
es.open(Buffer44,ios::app);
es <<Buf1;
es <<"\n";
es <<Buf2;
es <<"\n";
es <<Buf3;
es <<"\n";
es <<Buffer1;
es <<"\n";
es <<Buffer2;
es <<"\n";
es <<Buffer3;
es <<"\n";
es <<Buffer11;
es <<"\n";
es <<Buffer22;
es <<"\n";
es <<Buffer33;
es <<"\n";
es.close();

Buf1=0;Buf2=0;Buf3=0;
int pnr =++nr;
SetDlgItemInt(HWindow,ID_POS_NUM, pnr,
True);
}

else
MessageBox(HWindow,"NO","Selecciono",
MB_OK);
//MB_ICONEXCLAMATION); IDNO

```

El archivo se abre para el registro secuencial de datos.

Se actualiza el número de la posición del robot.

<pre> } /*void TTestDialog::HandleButtonMsg51(RTMessage) { char Buffer1[10]; //strcpy(num1,num); GetDlgItemText (HWindow, ID_BASE_VEL, Buffer1, 80); // SetDlgItemText(HWindow,ID_BASE_POS,Buffer) ; MessageBox(HWindow,Buffer1,"Velocidad actual", MB_OK); // SetDlgItemInt(HWindow,ID_BASE_POS, --i, True); } */ void TTestDialog::HandleListBoxMsg(RTMessage Msg) { DWORD Idx; char SelectedText[10]; char d1[5],d2[5],d3[5],d4[5],d5[5],d6[5],d7[5],d8[5],d9[5]; if(Msg.LP.Hi == LBN_DBLCLK) { Idx = SendDlgItemMsg(ID_LISTBOX, LB_GETCURSEL, 0, 0L); SendDlgItemMsg(ID_LISTBOX, LB_GETTEXT, (WORD)Idx, (DWORD)SelectedText); MessageBox(HWindow, SelectedText, "Seleccionaste ", MB_OK); lec.open(SelectedText,ios::in); if (lec.bad()){ MessageBox(HWindow, "Error", "De lectura..", MB_OK); return; } while(!lec.eof()){ lec >>d1; lec >>d2; lec >>d3; lec >>d4; lec >>d5; lec >>d6; lec >>d7; </pre>	<p>Esta es la parte del programa para el botón leer. El archivo se encarga de dar lectura al archivo seleccionado, además ejecuta los movimientos del robot en tiempo real.</p> <p>Presenta también una actualización por display de los datos leídos al tiempo en que se van ejecutando los movimientos.</p> <p>Despliega los datos leídos en el cuadro de lista.</p> <p>Da lectura de los datos directa del disco.</p>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<pre> lec >>d8; lec >>d9; for(int r=1;r<=atoi(d7);r++){ timer a(d4),b(d4); outportb(768,255); a.run(); //for(long r=1;r<=4500;r++){ outportb(768,0); b.run(); }; //MessageBox(HWindow, d1, "Datos leídos", MB_OK); SendDlgItemMsg(135, LB_ADDSTRING, 0, (LONG) d1); SendDlgItemMsg(135, LB_ADDSTRING, 0, (LONG) d2); SendDlgItemMsg(135, LB_ADDSTRING, 0, (LONG) d3); SendDlgItemMsg(135, LB_ADDSTRING, 0, (LONG) d4); SendDlgItemMsg(135, LB_ADDSTRING, 0, (LONG) d5); SendDlgItemMsg(135, LB_ADDSTRING, 0, (LONG) d6); SendDlgItemMsg(135, LB_ADDSTRING, 0, (LONG) d7); SendDlgItemMsg(135, LB_ADDSTRING, 0, (LONG) d8); SendDlgItemMsg(135, LB_ADDSTRING, 0, (LONG) d9); }; lec.close(); } } </pre>	<p>Envía la información a los motores.</p> <p>Despliega los datos leídos en el cuadro de lista.</p> <p>Se cierra el archivo.</p>
<pre> BOOL TTestWindow::CanClose() { return MessageBox(HWindow, "Desea salir", "Desea salir ?", MB_YESNO MB_ICONQUESTION) == IDYES; } </pre>	<p>Función que procesa la salida del programa sin realizar cambios.</p>
<pre> void TTestWindow::CMTest(RTMessage) { GetApplication()->ExecDialog(new TTestDialog(this,"DIAL_1")); } void TTestWindow::CMTest2(RTMessage) { </pre>	<p>Estas funciones son las que permiten el llamado de los cuadros de diálogo con que cuenta el programa. Se conectan recibiendo y enviando información al y desde los cuadro de diálogo al programa.</p>

<pre> GetApplication()->ExecDialog(new TTestDialog(this,"TESTDIALOG")); } void TTestWindow::Acerca(RTMessage) { GetApplication()->ExecDialog(new TTestDialog(this,"DIAL_2")); } </pre>	
<pre> void TTestApp::InitMainWindow() { MainWindow = new TTestWindow(NULL, Name); } </pre>	Llama al constructor de la ventana principal y la despliega.
<pre> int PASCAL WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nCmdShow) { TTestApp TestApp("CONTROL DE ROBOT- MANIPULADOR", hInstance, hPrevInstance, lpCmdLine, nCmdShow); TestApp.nCmdShow = SW_SHOWMAXIMIZED; TestApp.Run(); return (TestApp.Status); } </pre>	El constructor de la aplicación total y que lleva el control global del recibo y despacho de señales a través de los canales de comunicación con el programa y windows.

BIBLIOGRAFIA.

- [1] Jiménez Eusebio, Simulación de un sistema de manufactura con obstáculo en línea de producción.
Tesis de Grado de Maestría. DEPMI-UNAM. 1998.
- [2] Jiménez Eusebio, Reyes Avila Luis, Roberto Martinez
El Sistema CHEVOTRONIC IV. Reporte de Investigación 5. DEPMI-UNAM.
Depto. De Mecánica. 1997.
- [3] Jiménez Eusebio, Reyes Avila Luis, Luciano Vela Martínez.
Síntesis Análítica de un Mecanismo de 4 Barras. Reporte de la Materia Análisis y Síntesis de Mecanismos. DEPMI-UNAM. Depto. De Mecánica. 1997.
- [4] Reyes Avila Luis. Quaternions: Une representation parametrique systematique des rotaions finies. Rapports de Recherche, INRIA. Octobre 1990.

<pre> GetApplication()->ExecDialog(new TTestDialog(this,"TESTDIALOG")); } void TTestWindow::Acerca(RTMessage) { GetApplication()->ExecDialog(new TTestDialog(this,"DIAL_2")); } </pre>	
<pre> void TTestApp::InitMainWindow() { MainWindow = new TTestWindow(NULL, Name); } </pre>	Llama al constructor de la ventana principal y la despliega.
<pre> int PASCAL WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nCmdShow) { TTestApp TestApp("CONTROL DE ROBOT- MANIPULADOR", hInstance, hPrevInstance, lpCmdLine, nCmdShow); TestApp.nCmdShow = SW_SHOWMAXIMIZED; TestApp.Run(); return (TestApp.Status); } </pre>	El constructor de la aplicación total y que lleva el control global del recibo y despacho de señales a través de los canales de comunicación con el programa y windows.

BIBLIOGRAFIA.

- [1] Jiménez Eusebio, Simulación de un sistema de manufactura con obstáculo en línea de producción.
Tesis de Grado de Maestría. DEPMI-UNAM. 1998.
- [2] Jiménez Eusebio, Reyes Avila Luis, Roberto Martinez
El Sistema CHEVOTRONIC IV. Reporte de Investigación 5. DEPMI-UNAM.
Depto. De Mecánica. 1997.
- [3] Jiménez Eusebio, Reyes Avila Luis, Luciano Vela Martínez.
Síntesis Análítica de un Mecanismo de 4 Barras. Reporte de la Materia Análisis y Síntesis de Mecanismos. DEPMI-UNAM. Depto. De Mecánica. 1997.
- [4] Reyes Avila Luis. Quaternions: Une representation parametrique systematique des rotaions finies. Rapports de Recherche, INRIA. Octobre 1990.

-
- [5] Jiménez Eusebio, Reyes Avila Luis, Flores Campos J. Alejandro.
Diseño Geométrico del Módulo de Ensamble. Reporte de Investigación # 20.
DEPFI-UNAM. Depto. De Mecánica. 1998.
- [6] AutoCad versión 12, D. Raker y H. Rice. PHH Prentice Hall.
- [7] Jiménez Eusebio, Reyes Avila Luis.
Las Ecuaciones de Forma. Reporte de Investigación # 18. DEPFI-UNAM. Depto.
De Mecánica. 1998.
- [8] Cameron Smith an Nacy Blachman. The Mathematica Graphics Guidebook.
Addison-Wesley Publishing Company, Inc. 1995.
- [9] Ilan Vardi. Computational Recreations in Mathematica. Addison-Wesley Publishing
Company, Inc. 1991.
- [10] Jean-Pierre Merlet. Les robots parallèles. Hermes, París, 1990.
- [11] Jack C. K. Chou. IEEE Transactions on Robotics an Automation, Vol 8, No. 1,
February 1992. "Qauterion kinamatic and Dinamic Differential Equations".
- [12] Stephen Wolfram. Mathematica a system for Doing Mathematics by Computer.
Addison-Weley Publishing Company, Inc. 1988.
- [13] Stephen Wolfram. Mathematica Reference Guide. Addison-Wesley Publishing
Company, Inc. 1992.
- [14] Fundamentals of contemporary set theory.
springer-verlag. 1979. keit j. devlin
- [15] Motores eléctricos biblioteca práctica del taller
GG/México Editorial Gustavo Gili, S.A. de C.V. 1995 . JIM COX
- [16] Técnicas de automatización Ingeniería en movimiento
Superior electric Bulletin MD4-295 Printed in U.S.A
- [17] Warner electric. linear motion and electronics division.
- [18] Revista saber electronica no.9,10,13 año 6.
editorial: america s.a. de c.v. Lucio Blanco No. 435. Azcapotzalco, MEXICO, D.F.
- [19] Revista: montajes electronicos para pc. no. 9
editorial: F&G Editores, S.A de C.V. Plaza República del Ecuador 2, 28016
Madrid, España.