



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
INGENIERÍA MECÁNICA – DISEÑO MECÁNICO

**MODELO CINEMÁTICO, SIMULADOR DE LA POSICIÓN Y PROTOTIPO DE
UN ROBOT PARALELO DE 5 BARRAS**

TESIS
QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN INGENIERÍA

PRESENTA:
ARTURO URBALEJO CONTRERAS

TUTOR
DR. FRANCISCO CUENCA JIMÉNEZ
FACULTAD DE INGENIERÍA

MÉXICO, D. F. MAYO 2013

JURADO ASIGNADO:

Presidente: Dr. Carrera Bolaños Jorge
Secretario: Dr. González Villela Víctor Javier
Vocal: Dr. Cuenca Jiménez Francisco
1^{er.} Suplente: M. I. Díaz Hernández Octavio
2^{d o.} Suplente: M.I. Martínez Zamudio Patricio

Lugar donde se realizó la tesis: Cd. Obregón Sonora

TUTOR DE TESIS:

Dr. Cuenca Jiménez Francisco

FIRMA

ÍNDICE

| | |
|--|-----|
| Dedicatoria | i |
| Agradecimientos | ii |
| Resumen | iii |
| Abstract | iv |
| LISTA DE FIGURAS | v |
| CAPITULO 1 INTRODUCCIÓN | 7 |
| 1.1 Antecedentes | 8 |
| 1.2 Planteamiento del Problema. | 13 |
| 1.3 Consideraciones para el análisis | 14 |
| 1.4 Objetivo | 14 |
| CAPÍTULO 2 MARCO TEÓRICO | 15 |
| 2.1 Descripción espacial de la posición, orientación y ejes coordenados | 15 |
| 2.2. Rotación de matrices. | 17 |
| 2.3. Método de las Matrices Homogéneas | 19 |
| 2.4 Cinemática Directa e Inversa. | 22 |
| CAPITULO 3. MODELO CINEMÁTICO | 23 |
| 3.1 Modelo Cinemático de Posición | 23 |
| 3.2 Modelo Cinemático Inverso | 34 |
| CAPÍTULO 4. SOFTWARE DE SIMULACIÓN | 50 |
| 4.1 Algoritmo de Programación | 50 |
| 4.2 Interfaz de Simulación | 51 |
| 4.3 Programación de las ecuaciones en Visual Basic 6.0 | 59 |
| 4.4 Librería OpenGL | 61 |
| 4.5 Requerimientos de Hardware y Software | 65 |
| CAPÍTULO 5 DESARROLLO DE UN PROTOTIPO | 66 |
| 5.1 Descripción del prototipo | 66 |
| 5.2 Aspectos importantes sobre el control del prototipo | 69 |
| 5.3 Aspectos importantes sobre la funcionalidad del robot | 71 |
| CONCLUSIONES | 73 |
| RECOMENDACIONES | 74 |
| BIBLIOGRAFÍA | 75 |

Dedicatoria

A Dios

Por enseñarme el camino del bien y de amar a mi semejante y hermano, y por todo lo que nos comparte día con día.

A mi Madre Maura Esthela Contreras González

Por ser un ejemplo en todo momento de amor, responsabilidad, justicia, trabajo, y por todos sus sacrificios para salir adelante.

A mi padre Arturo Urbalejo Leyva

Por ser un padre ejemplar y mi mejor amigo y por todo su amor y dedicación a su familia.

A mi esposa Lilian Andrea Terrazas González

Por su amor y comprensión, por su apoyo y por ser mi guía en el camino espiritual.

Agradecimientos

A Dios por iluminarme en la vida, darme fe, sabiduría e inteligencia.

A mi esposa Lilian Andrea por apoyarme siempre y estar en los momentos que más la necesito y por darme siempre aliento para seguir adelante.

A mis hijas Nicole Andrea y Heisy Lilian por darme su amor incondicional y por hacerme tan feliz.

A mis padres Arturo y Maura Esthela por educarme con humildad y sencillez y por ser el mejor ejemplo de vida familiar.

A mis hermanos Gabriela, Luís Antonio, Carlos de Jesús Y Xochitl María por creer en mí.

A mi asesor Francisco Cuenca por asesorarme en el desarrollo de este proyecto.

A mis amigos Javier Jacobo y Eusebio Jiménez por apoyarme en el desarrollo de este proyecto.

Resumen

En este trabajo de tesis se presenta la modelación cinemática, la simulación, el diseño y la fabricación de un prototipo robótico. El robot en estudio es tipo paralelo espacial de 6 GDL. El modelo cinemático de posicionamiento fue generado usando matrices homogéneas. Para realizar la programación de los modelos y visualizar los movimientos del robot se utilizó la plataforma de programación Visual Basic 6 y bibliotecas de OpenGL. Se desarrolló un prototipo del robot para mostrar la representatividad del modelo cinemático. El movimiento del robot se ejecuta por motores paso a paso con transmisiones de engranes rectos. Finalmente, el robot es utilizado para capacitar y entrenar alumnos y profesores de la Universidad tecnológica del Sur de Sonora

Abstract

In this thesis work it is presented the kinematic modeling, the simulation, the design and manufacturing of a robotic prototype. The robot under study is a special parallel type with 6 degrees of freedom. The kinematic model for the positioning was generated using homogeneous matrices. The Visual Basic 6 software and OpenGL libraries were used for the programming of the models and visualize the robot motion. A robot prototype was developed to show the representativity of the kinematic model. The robot motion is executed by stepper motors with spur gear transmissions. Finally, the robot is used for training students and teachers of Technological University of Southern Sonora.

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1.1 Configuración hipotética del primer robot paralelo | 9 |
| Figura 1.2 El primer robot paralelo industrial espacial, patentado en 1942 (US patente No. 2,286,571). | 9 |
| Figura 1.3 Robots paralelos tipo plataforma..... | 10 |
| Figura 1.4 Otro tipo de robot paralelo..... | 10 |
| Figura 1.5 Plataforma de Stewart..... | 11 |
| Figura 1.6 Configuración robot paralelo de Cappel..... | 11 |
| Figura 1.7 Robot paralelo hecho para un simulador | 12 |
| Figura 1.8 Configuración del robot paralelo a estudiar | 13 |
| Figura 2.1 a) Sistema cartesiano plano, b) Sistema cartesiano espacial | 15 |
| Figura 2.2 Orientación del gripper | 16 |
| Figura 2.3 Descripción de un cuerpo rígido en un sistema de coordenadas cartesianas..... | 17 |
| Figura 2.4 Sistemas coordenados locales..... | 19 |
| Figura 3.1. Base móvil 1 | 24 |
| Figura 3.2. Base móvil 2 | 25 |
| Figura 3.3. Base móvil 3 | 26 |
| Figura 3.4. Base móvil 4..... | 27 |
| Figura 3.5 Base móvil 5 | 28 |
| Figura 3.6. Base móvil 6 | 29 |
| Figura 3.7 Bases móviles sobre el gripper | 30 |
| Figura 3.8. Definición del vector de posición | 31 |
| Figura 4.1 Algoritmo de funcionamiento general | 50 |
| Figura 4.2 Interfaz del simulador..... | 51 |
| Figura 4.3 Punto de vista rotado | 52 |
| Figura 4.4. Configuración inicial del robot..... | 53 |
| Figura 4.5 Configuración de posición (20,50,75) orientación(0,0,0) | 54 |
| Figura 4.6 Configuración de posición (20,50,50) orientación(0,0,0) | 54 |
| Figura 4.7 Configuración de posición (20,70,50) orientación(0,0,0) | 55 |
| Figura 4.8 Configuración de posición (0,50,75) orientación(30,0,0) | 55 |
| Figura 4.9 Configuración de posición (0,50,75) orientación(30,30,0) | 56 |
| Figura 4.10 Configuración de posición (0,50,75) orientación(30,30,30) | 57 |

| | |
|---|----|
| Figura 4.11 Configuración de posición (0,50,75) orientación(0,-90,-80)..... | 57 |
| Figura 4.12 Matriz de articulaciones..... | 58 |
| Figura 4.13 Trayectoria lineal..... | 58 |
| Figura 5.1. Prototipo desarrollado..... | 66 |
| Figura 5.2. Dimensiones principales del prototipo | 67 |
| Figura 5.3. Tren de engranes..... | 68 |
| Figura 5.4. Motor paso a paso y sistema de engranaje | 69 |
| Figura 5.5. Circuitos integrados | 70 |
| Figura 5.6. Circuito impreso. | 71 |
| Figura 5.7. Algoritmo de control | 72 |
| Figura 5.8 Simulador en (0,50,55) | 72 |
| Figura 5.9 Prototipo en (0,50,55)..... | 72 |
| Figura 5.10 Simulador en (-20,60,25)..... | 72 |
| Figura 5.11 Prototipo en (-20,60,25)..... | 72 |

CAPITULO 1 INTRODUCCIÓN

Los procesos de producción que están a la vanguardia en calidad, eficiencia y productividad, son aquellos que están automatizados. Los productos se elaboran a través de celdas de manufactura integradas por computadora o sistemas flexibles de manufactura, en los cuales intervienen diferentes dispositivos; pero, el dispositivo que los caracteriza para lograr dicha automatización, es el Robot Industrial. Existe una gran variedad de tipos de robots. Sin embargo, en la actualidad se han incrementado las investigaciones de los robots paralelos debido a las grandes ventajas que presentan sobre los robots seriales, una de las más importantes es que este tipo de robot tiene más precisión por que soportan la carga con dos o más eslabones en paralelo.

El presente trabajo de investigación analiza los movimientos en el espacio de un manipulador paralelo de 5 eslabones desarrollando un modelo matemático y un software que simula los movimientos de dicho robot.

Este trabajo de tesis está estructurado de la manera siguiente:

En el capítulo 1 se hace una breve reseña de la historia de este tipo de manipuladores paralelos, posteriormente se plantea un caso específico de un manipulador paralelo de 5 eslabones con 6 GDL y juntas rotacionales.

En el capítulo 2 se presenta el marco teórico utilizado para la representación de un objeto en el espacio, traslación y rotación de matrices, transformaciones homogéneas y cinemática inversa.

En el capítulo 3, se desarrolla el modelo matemático de posición y se presenta el cálculo de la cinemática inversa a través del método de ecuaciones de lazo matricial usando matrices homogéneas.

En el capítulo 4 se presenta el algoritmo y el programa en Visual Basic 6.0 de un simulador relacionado con el robot motivo de estudio, obtenido a partir de un modelo matemático.

En el capítulo 5 se presenta el desarrollo de prototipo del robot motivo de estudio y finalmente se dan las conclusiones y recomendaciones.

1.1 Antecedentes

En la actualidad, los procesos de producción tienden hacia la automatización por diferentes razones, por ejemplo para aumentar la competitividad de un producto, lo cual implica cuestiones de calidad, innovación, costo y servicio post-venta, entre otras cosas. Para el logro de estos objetivos, uno de los principales elementos que intervienen en dicha automatización es el robot industrial.

Así como las necesidades de alimentación y comercialización dieron origen a la agricultura y a la revolución industrial, la necesidad de aumentar la productividad, mejorar la calidad y tener procesos automatizados flexibles dio origen a la creación de lo que hoy en día se conoce como robot industrial.

Los robots se clasifican de manera general en dos tipos: de cadena abierta y de cadena cerrada. Los robots de cadena abierta (llamados robot seriales) son eslabones unidos en serie con un extremo de la unión fijo en una base llamada tierra y el otro extremo queda libre para posicionar el órgano efector. Los robots de cadena cerrada (llamados robot paralelos) están compuestos por lo menos de dos cadenas abiertas fijas a una base y unidas en una articulación entre estas dos cadenas y dicha unión debe tener un solo grado de libertad. Merlet Jean [3] lo define de la siguiente manera: “Los robots paralelos son mecanismos de cadena cerrada, constituidos 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 por lo menos de dos segmentos articulados, la articulación entre estos dos segmentos deberá tener un solo grado de libertad”.

La presente investigación estudia un caso específico de un robot paralelo. Los motivos que implican el desarrollo de este proyecto son las ventajas que tienen los robot paralelos sobre los robot seriales de las cuales se puede mencionar la capacidad de carga elevada, posibilidad de movimientos de alta dinámica, elevada rigidez mecánica, poco error de repetibilidad (debido a su precisión por soportar el peso con dos o más eslabones en paralelo), etc.. Otro de los motivos es la generación de conocimiento en el área de los robots paralelos y la obtención de simuladores de movimiento de estos robots mediante la aplicación de la teoría desarrollada.

Por otro lado, en 1928 James E. Gwinnett solicitó una patente del primer mecanismo paralelo esférico orientado hacia la industria del entretenimiento. La figura 1.1 muestra la configuración de dicho robot. Posiblemente fue el primer mecanismo paralelo espacial patentado en 1931 (US patente No. 1,789,680). No se sabe si este era el primer sistema cinemático paralelo multi-grado-delibertad (GDL) que fué construido. Pero una cosa es segura James Gwinnett tenía pensamientos demasiado anticipados a su tiempo y la industria no estaba lista para la complejidad de su invención.

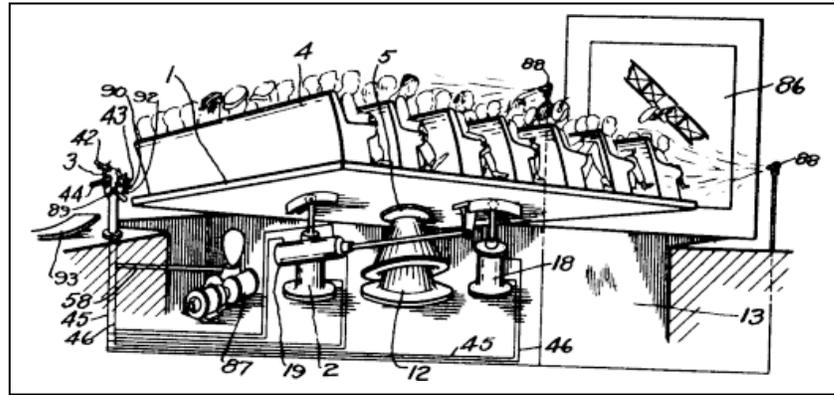


Figura 1.1 Configuración hipotética del primer robot paralelo

Una década más adelante y 17 años después de que se acuñara el término “robot”, un nuevo robot paralelo fue inventado para pintado en aerosol automático por Willard L. Pollard. En la comunidad paralela cinemática, el robot paralelo de Pollard es bien conocido como el primer diseño de un robot paralelo industrial el cual nunca llegó. La figura 1.2 muestra el primer robot paralelo patentado.

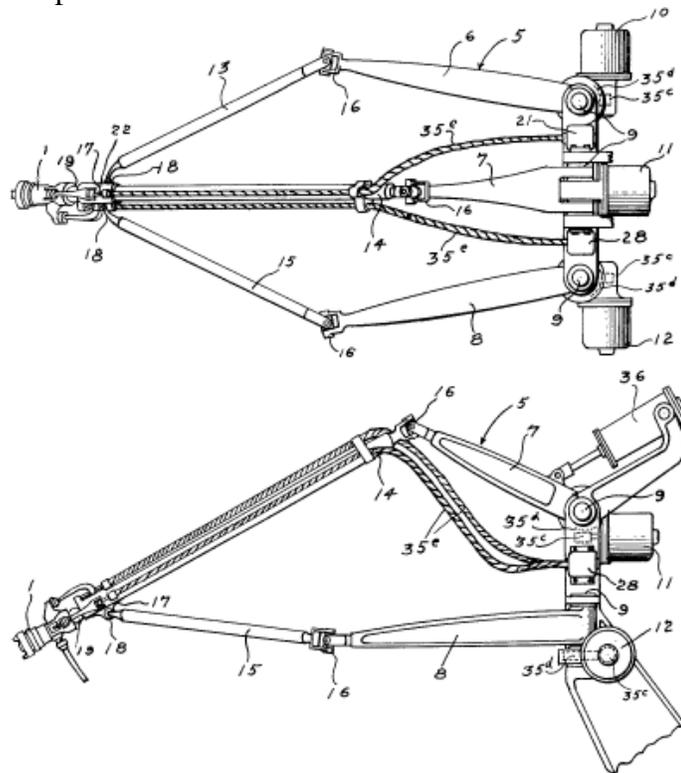


Figura 1.2 El primer robot paralelo industrial espacial, patentado en 1942 (US patente No. 2,286,571).

El primer robot industrial que fué construido no era el antes mencionado y no se puede acreditar al mismo Willard Pollard. El ingeniero que co-diseño el primer robot industrial fue el hijo de Willard L. V. Pollard, Willard L. G. Pollard Jr.

En 1947, en el otro lado del Atlántico, un nuevo mecanismo paralelo fue inventado, llegó a ser el más popular y cambió a la industria, y del cual se han hecho más de mil réplicas el Hexapod (seis patas) octaédrico con puntal de longitud variable. La figura 1.3 muestra algunos ejemplos de robots paralelos.



Figura 1.3 Robots paralelos tipo plataforma.

Por otro lado, el primer Hexapod octaédrico o la plataforma de Gough fué concebido en 1954 (Proc. ImechE, 1965-1966). Estos sistemas han sido muy populares, por las pequeñas variaciones “los ajustes del receptáculo son simples e interpretables”. Este tipo de sistemas son conocidos con las siglas MAST (Multi-Axis Simulation (or Shake) Table), y aun son manufacturados por numerosas compañías. La figura 1.4 muestra la configuración de otro robot paralelo.



Figura 1.4 Otro tipo de robot paralelo

En 1965, Stewart describe una plataforma móvil 6-GDL para usarla como simulador de vuelo. La figura 1.5 muestra el diagrama esquemático de la única “Plataforma de Stewart” (Proc. ImechE, 1965-1966).

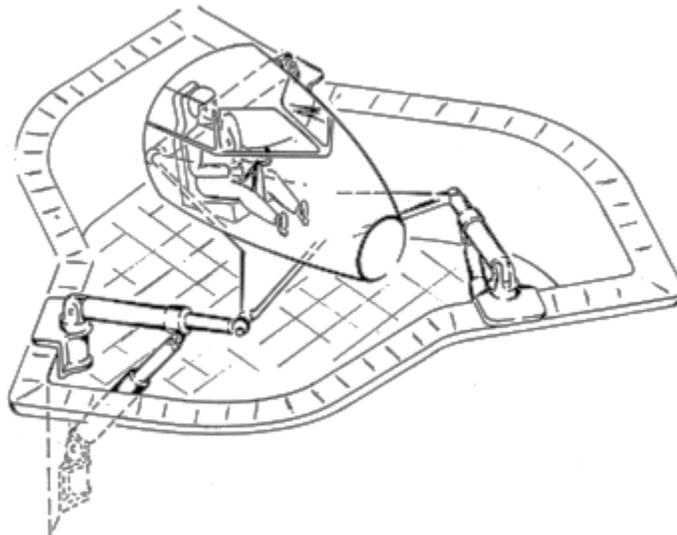


Figura 1.5 Plataforma de Stewart

En 1971, la US Patent concedió a Klaus Cappel una patente para ser usado como simulador de vuelo. La figura 1.6 muestra la configuración del robot de Cappel.

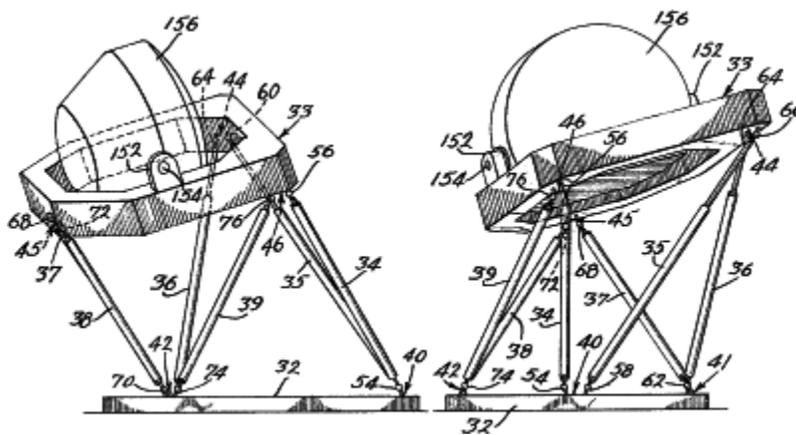


Figura 1.6 Configuración Robot paralelo de Cappel

Por otro lado, el primer simulador de vuelo basado en el Hexapod octaédrico se construyó como a mediados de los 60's (imagen cortesía de Klaus Cappel).

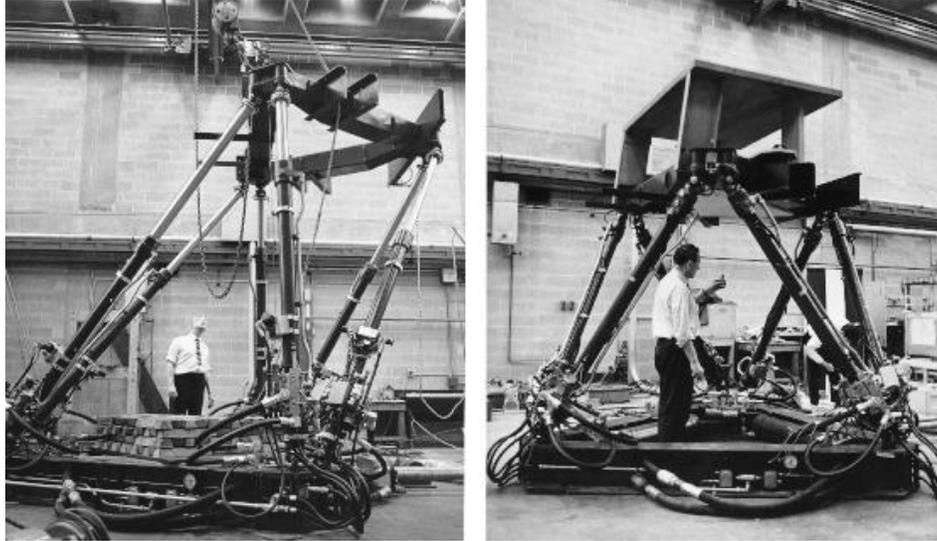


Figura 1.7 Robot paralelo hecho para un simulador

La información sobre la historia de los robots paralelos anteriormente descrita, fue extraída del artículo “The True Origins of Parallel Robots” de Ilian Bonev [6]

1.2 Planteamiento del Problema.

En esta sección se plantea el problema por solucionar en este trabajo de tesis. Esto es:

“Para motivar el estudio de la robótica paralela, se requiere generar simuladores que imiten los movimientos de un robot paralelo, como el mostrado en la figura 1.8. Tales simuladores implican generar los modelos cinemáticos de los robots, y los modelos requieren del uso de herramientas matemáticas. La necesidad es que, tanto alumnos, como profesores, usen para la enseñanza simuladores de robots y sobre todo, que aprendan todo el proceso de modelado y la programación de las ecuaciones resultantes.”

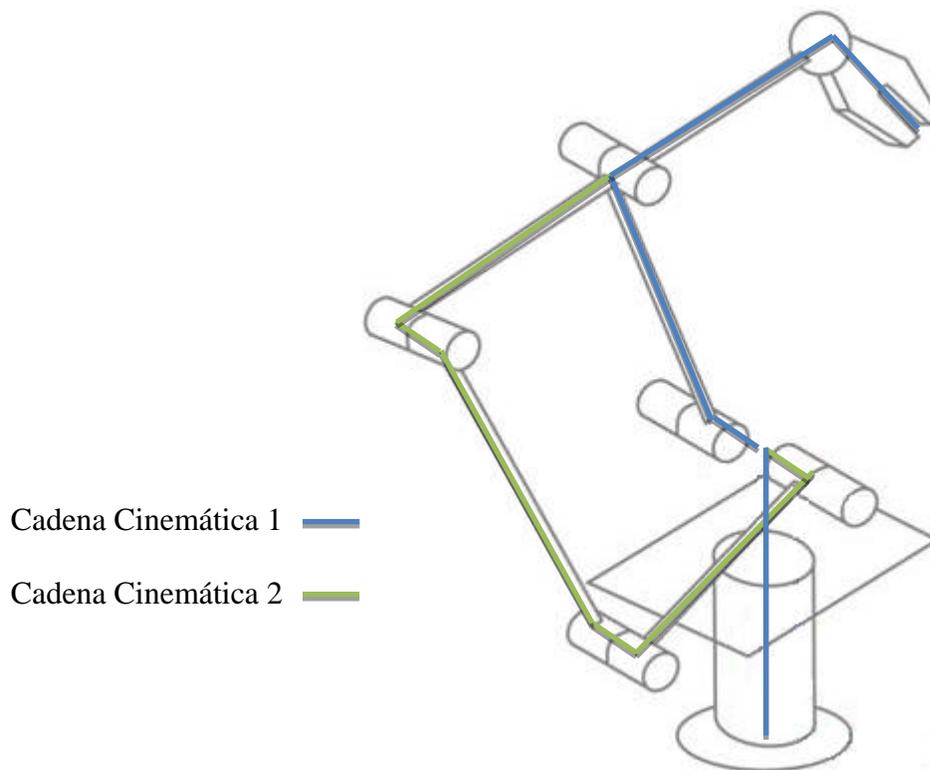


Figura 1.8 Configuración del robot paralelo a estudiar

1.3 Consideraciones para el análisis

- El análisis es cinemático de posición y de velocidad
- Solo se programarán y simularán los modelos de posición
- El robot es tipo paralelo
- Tiene 6 GDL
- El movimiento del robot es en el espacio
- Tiene juntas rotacionales

1.4 Objetivo

“Construir un simulador y un prototipo de robot paralelo como el mostrado en la figura 1.8, usando matrices homogéneas y la plataforma de programación Visual Basic, para motivar a la enseñanza de la robótica paralela.”

CAPÍTULO 2 MARCO TEÓRICO

En este capítulo se presenta el marco teórico que será utilizado en este trabajo de tesis. Se presentan los conceptos y las definiciones de interés relacionadas con la cinemática y la matemática necesaria para el modelado. Cabe mencionar que el objetivo principal de los robots, es manipular objetos en el espacio. Esto implica que el robot debe conocer cuando menos la posición y orientación de dichos objetos. Todos estos conceptos de interés serán explicados en las secciones siguientes.

2.1 Descripción espacial de la posición, orientación y ejes coordenados

Para representar la posición y orientación de cualquier objeto en el plano o en el espacio, se necesita un sistema de coordenadas de referencia. En el plano, la posición tiene dos grados de libertad, por lo tanto, cada punto en el plano estará representado por dos componentes independientes. En un espacio tridimensional, será necesario utilizar tres componentes [1].

Ejes coordenados (Sistema cartesiano de referencia)

Normalmente los sistemas de referencia se definen mediante ejes perpendiculares entre sí con un origen definido. Estos se denominan **sistemas cartesianos**, y en el caso de trabajar en el plano (2 dimensiones), el sistema de referencia OXY correspondiente queda definido por dos vectores coordenados OX y OY perpendiculares entre sí con un punto de intersección común O, como se muestra en la figura 2.1.a).

Si se trabaja en el espacio (tridimensional), el sistema cartesiano OXYZ está compuesto por una terna ortonormal de vectores coordenados OX, OY y OZ tal como observa en la figura 2.1.b

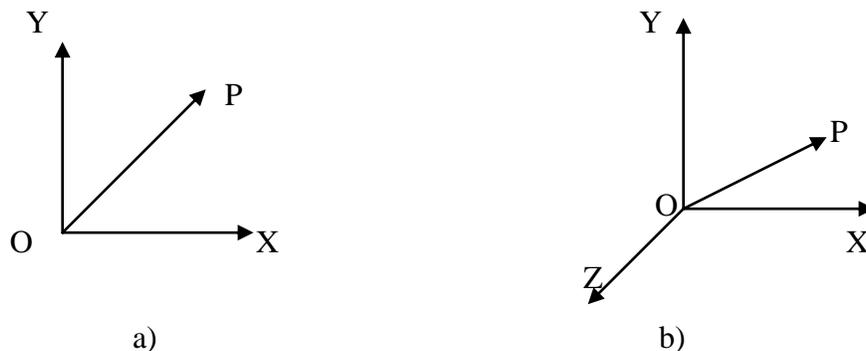


Figura 2.1 a) Sistema cartesiano plano, b) Sistema cartesiano espacial

Descripción de la posición (coordenadas cartesianas)

En el plano, el sistema coordenado OXY de referencia, representa cada punto a través de las componentes (x,y) correspondientes a los ejes coordenados OX para la componente x y OY para la componente y . Así, si se expresa el punto a en términos de coordenadas (x,y) , este punto tiene asociado un vector p que va desde el origen O del sistema coordenado OXY hasta el punto a (ver figura 2.1.a). Por lo tanto, la posición del vector p está caracterizado por las dos componentes (x,y) , denominadas coordenadas cartesianas del vector y que son las proyecciones del vector p sobre los ejes OX y OY.

En el caso de trabajar en un espacio tridimensional, el vector (3×1) viene definido con respecto a el sistema de referencia OXYZ mediante las coordenadas correspondientes a cada uno de los ejes coordenados (x,y,z) . (Ver figura 2.2).

Descripción de la orientación

Para localizar un punto en el espacio son necesarios los datos de su posición con respecto a su sistema de referencia (vector de posición de 3×1). Sin embargo, para localizar un objeto, además de los datos de posición, es necesario definir su orientación. Por ejemplo, el vector ${}^A P$ mostrado en la figura 2.2 localiza el PCH (punto central de la herramienta) del gripper con respecto a un sistema de referencia $\{0\}$. Faltaría por describir la orientación del gripper con respecto al mismo sistema de referencia. Para describir la orientación, se unirá un sistema de coordenadas $\{1\}$ al gripper y se describirá este sistema de coordenadas relativo al sistema de referencia, escribiendo en vectores unitarios de los tres principales ejes de $\{1\}$ en términos de $\{0\}$ [2].

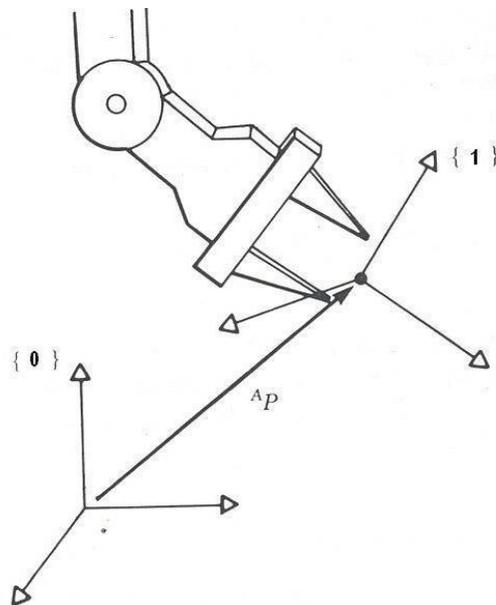


Figura 2.2 Orientación del gripper

Por otro lado, se describirán los vectores unitarios de los principales ejes del sistema de coordenadas {1} como \mathbf{X}_1 , \mathbf{Y}_1 y \mathbf{Z}_1 . Cuando se describa dicho sistema de vectores unitarios en términos del sistema de referencia {0}, se escribirá de la siguiente manera ${}^0\mathbf{X}_1$, ${}^0\mathbf{Y}_1$ y ${}^0\mathbf{Z}_1$. Además, si se ordenan estos tres vectores en forma de columna, se forma una matriz llamada **matriz de rotación**. Dicha matriz describe la rotación del sistema coordenado {1} relativo al sistema {0}.

Cabe señalar que, la orientación en el espacio tridimensional viene definida por tres grados de libertad o tres componentes linealmente independientes formando una matriz de 3 x 3. Por lo tanto, mientras que la posición de un punto es representada con un vector, la orientación de un objeto es representada con una matriz.

2.2. Rotación de matrices.

Las matrices de rotación representan el método más extendido para la descripción de la orientación, debido principalmente a la comodidad que proporciona el uso del álgebra matricial. Supóngase ahora que se tiene en el plano dos sistemas de referencia {0} y {1} con un mismo origen O, siendo {0} el sistema fijo y {1} el sistema móvil unido al objeto[1].

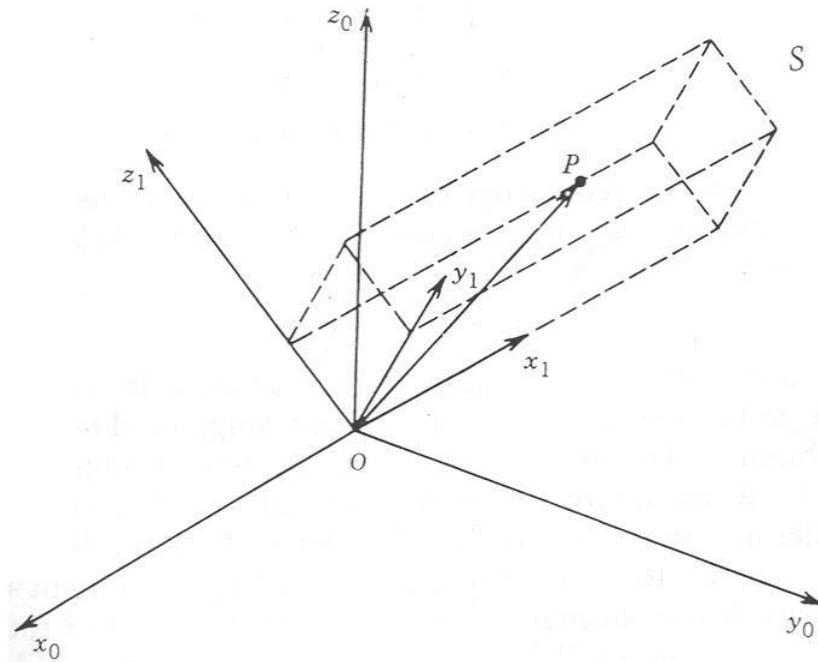


Figura 2.3 Descripción de un cuerpo rígido en un sistema de coordenadas cartesianas

La figura. 2.3. muestra un objeto rígido \mathbf{S} con un sistema de coordenadas $Ox_1y_1z_1$ unido a él. Nótese que si se quiere describir las coordenadas del punto \mathbf{p} en \mathbf{S} del sistema de coordenadas $Ox_1y_1z_1$ de \mathbf{p} en el sistema de referencia no rotado $Ox_0y_0z_0$, siendo $\{i_0, j_0, k_0\}$ las bases ortonormales del sistema $Ox_0y_0z_0$; y i_0, j_0, k_0 son los vectores unitarios a lo largo de los ejes x_0, y_0, z_0 , respectivamente, entonces \mathbf{p} es tal que:

$$\mathbf{p}_0 = p_{0x}\mathbf{i}_0 + p_{0y}\mathbf{j}_0 + p_{0z}\mathbf{k}_0$$

El punto P se puede describir con respecto a $ox_1y_1z_1$ de la manera siguiente:

$$\mathbf{p}_1 = p_{1x}\mathbf{i}_1 + p_{1y}\mathbf{j}_1 + p_{1z}\mathbf{k}_1$$

Como \mathbf{p}_0 y \mathbf{p}_1 son representaciones del mismo vector \mathbf{p} , las relaciones entre los componentes de \mathbf{p} en los dos sistemas de coordenadas se obtienen como sigue:

$$p_{0x} = \mathbf{p}_0 \cdot \mathbf{i}_0 = \mathbf{p}_1 \cdot \mathbf{i}_0$$

y

$$p_{0x} = p_{1x}\mathbf{i}_1 \cdot \mathbf{i}_0 + p_{1y}\mathbf{j}_1 \cdot \mathbf{i}_0 + p_{1z}\mathbf{k}_1 \cdot \mathbf{i}_0$$

De manera similar p_{0y} y p_{0z} se expresan:

$$p_{0y} = p_{1x}\mathbf{i}_1 \cdot \mathbf{j}_0 + p_{1y}\mathbf{j}_1 \cdot \mathbf{j}_0 + p_{1z}\mathbf{k}_1 \cdot \mathbf{j}_0$$

$$p_{0z} = p_{1x}\mathbf{i}_1 \cdot \mathbf{k}_0 + p_{1y}\mathbf{j}_1 \cdot \mathbf{k}_0 + p_{1z}\mathbf{k}_1 \cdot \mathbf{k}_0$$

$$\mathbf{p}_0 = \mathbf{R}_0^1 \mathbf{p}_1$$

Nosotros podemos escribir las tres ecuaciones juntas como:

$$\mathbf{R}_0^1 = \begin{bmatrix} \mathbf{i}_1 \cdot \mathbf{i}_0 & \mathbf{j}_1 \cdot \mathbf{i}_0 & \mathbf{k}_1 \cdot \mathbf{i}_0 \\ \mathbf{i}_1 \cdot \mathbf{j}_0 & \mathbf{j}_1 \cdot \mathbf{j}_0 & \mathbf{k}_1 \cdot \mathbf{j}_0 \\ \mathbf{i}_1 \cdot \mathbf{k}_0 & \mathbf{j}_1 \cdot \mathbf{k}_0 & \mathbf{k}_1 \cdot \mathbf{k}_0 \end{bmatrix}$$

La matriz \mathbf{R}_0^1 de 3x3 representa la matriz de transformación del punto \mathbf{p} del sistema de referencia $Ox_1y_1z_1$ con respecto al sistema de referencia $Ox_0y_0z_0$ [1].

2.3. Método de las Matrices Homogéneas

Considérese ahora el sistema de coordenadas $Ox_1y_1z_1$ obtenido de $Ox_0y_0z_0$ por una traslación paralela una distancia “d” como se observa en la figura 2.4. Así, i_0, j_0, k_0 es paralelo a i_1, j_1, k_1 , respectivamente. El vector d^1_0 es el vector del origen O_0 al origen O_1 expresado en las coordenadas del sistema $Ox_0y_0z_0$ [2].

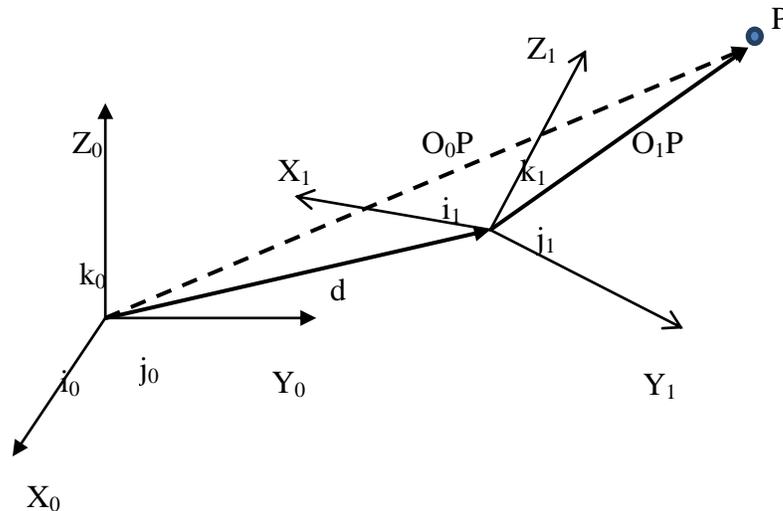


Figura 2.4 Sistemas coordenados locales

El punto p tiene representación p_0 y p_1 . Como los ejes coordenados respectivos de los dos sistemas son paralelos los vectores p_0 y p_1 se relacionan por:

$$p_0 = p_1 + d^1_0$$

En forma explícita se tiene:

$$p_{0x} = p_{1x} + d^1_{0x}$$

$$p_{0y} = p_{1y} + d^1_{0y}$$

$$p_{0z} = p_{1z} + d^1_{0z}$$

La relación más general entre los dos sistemas de coordenadas $Ox_0y_0z_0$ y $Ox_1y_1z_1$ puede ser expresada como la rotación pura y traslación pura hablando de movimientos rígidos.

Por otro lado, una transformación $p_0 = R p_1 + d$ se define como movimiento rígido si R es ortogonal. Sí se tienen dos movimientos rígidos, entonces:

$$\mathbf{p}_0 = \mathbf{R}^1_0 \mathbf{p}_1 + \mathbf{d}^1_0$$

y

$$\mathbf{p}_1 = \mathbf{R}^2_1 \mathbf{p}_2 + \mathbf{d}^2_1$$

Además, se puede definir un tercer movimiento rígido sustituyendo \mathbf{p}_1 en \mathbf{p}_0 . Esto es:

$$\mathbf{p}_0 = \mathbf{R}^1_0 \mathbf{R}^2_1 \mathbf{p}_2 + \mathbf{R}^1_0 \mathbf{d}^2_1 + \mathbf{d}^1_0$$

Así, la relación entre \mathbf{p}_0 y \mathbf{p}_2 es también un movimiento rígido que se puede describir como:

$$\mathbf{p}_0 = \mathbf{R}^2_0 \mathbf{p}_2 + \mathbf{d}^2_0$$

Comparando ecuaciones anteriores se tiene la siguiente identidad:

$$\mathbf{R}^2_0 = \mathbf{R}^1_0 \mathbf{R}^2_1$$

La ecuación anterior muestra que la transformación en la orientación puede ser una simple multiplicación de orientaciones. Esto es:

$$\mathbf{d}^2_0 = \mathbf{d}^1_0 + \mathbf{R}^1_0 \mathbf{d}^2_1$$

La ecuación anterior muestra que, desde el vector origen O0 hasta el vector origen O2, es la suma del vector \mathbf{d}^1_0 de O0 a O1 y el vector $\mathbf{R}^1_0 \mathbf{d}^2_1$ de O1 a O2 expresado en la orientación del sistema coordenado $Ox_0y_0z_0$.

Lo anterior puede ser escrito de manera matricial como:

$$\begin{bmatrix} \mathbf{R}^1_0 & \mathbf{d}^1_0 \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}^2_1 & \mathbf{d}^2_1 \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}^1_0 \mathbf{R}^2_1 & \mathbf{R}^1_0 \mathbf{d}^2_1 + \mathbf{d}^1_0 \\ \mathbf{0} & 1 \end{bmatrix}$$

donde “ $\mathbf{0}$ ” denota el vector (0,0,0), mostrando que el movimiento rígido puede ser representado por matrices de la forma:

$$\mathbf{H} = \begin{bmatrix} \mathbf{R} & \mathbf{d} \\ \mathbf{0} & 1 \end{bmatrix}$$

Además, la transformación inversa de H está dada por:

$$\mathbf{H}^{-1} = \begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T \mathbf{d} \\ \mathbf{0} & 1 \end{bmatrix}$$

Las matrices de transformación con esta forma, son llamadas **transformaciones homogéneas**. Para representar las transformaciones como una multiplicación matrices, es necesario agregar a los vectores \mathbf{p}_0 y \mathbf{p}_1 un cuarto componente como sigue:

$$\mathbf{P}_0 = \begin{bmatrix} p_0 \\ 1 \end{bmatrix}$$

$$\mathbf{P}_1 = \begin{bmatrix} p_1 \\ 1 \end{bmatrix}$$

Los vectores \mathbf{P}_0 y \mathbf{P}_1 son conocidos como **representaciones homogéneas** de los vectores \mathbf{p}_0 y \mathbf{p}_1 , respectivamente. Ahora se puede ver que, las transformaciones son equivalentes a las ecuaciones matriciales (homogéneas). Esto es:

$$\mathbf{P}_0 = \mathbf{H}\mathbf{P}_1 = \mathbf{H}^1_0\mathbf{P}_1$$

Las **transformaciones homogéneas básicas** son:

1) Para traslaciones se tiene que:

$$\mathbf{Trans}_{x,a} = \mathbf{T}_{z1}(a) = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \mathbf{Trans}_{y,b} = \mathbf{T}_{z2}(b) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$\mathbf{Trans}_{z,c} = \mathbf{T}_{z3}(c) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2) Para rotaciones se tiene que:

$$\mathbf{Rot}_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha & -s\alpha & 0 \\ 0 & s\alpha & c\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \mathbf{Rot}_{y,\phi} = \begin{bmatrix} c\phi & 0 & s\phi & 0 \\ 0 & 1 & 0 & 0 \\ -s\phi & 0 & c\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \mathbf{Rot}_{z,\theta} = \begin{bmatrix} c\theta & -s\theta & 0 & 0 \\ s\theta & c\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

La forma general de las transformaciones homogéneas es la siguiente:

$$\mathbf{T} = \begin{bmatrix} n_x & s_x & a_x & d_x \\ n_y & s_y & a_y & d_y \\ n_z & s_z & a_z & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{n} & \mathbf{s} & \mathbf{a} & \mathbf{d} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.4 Cinemática Directa e Inversa.

La cinemática del robot estudia el movimiento del mismo con respecto a un sistema de referencia. Así, la cinemática se interesa por la descripción analítica del movimiento espacial del robot como una función del tiempo, y en particular por las relaciones entre la posición y la orientación del extremo final del robot con los valores que toman sus ejes de rotación [4].

Existen dos problemas fundamentales a resolver en la cinemática de un robot; el primero de ellos se conoce como el **problema cinemático directo**, y consiste en determinar cual es la posición y orientación del extremo final del robot, con respecto a un sistema de coordenadas que se toma como referencia, siempre que sean conocidos los valores de las articulaciones y los parámetros geométricos de los elementos del robot; el segundo, denominado **problema cinemático inverso**, resuelve la configuración que debe adoptar el robot para una posición y orientación del extremo conocidas. En esta tesis se resolverá el problema cinemático inverso de un robot paralelo de cinco barras.

CAPITULO 3. MODELO CINEMÁTICO

3.1 Modelo Cinemático de Posición

En este capítulo, se modelan las rotaciones de robot motivo de estudio en esta tesis. Para tal propósito, se usarán matrices homogéneas. Cabe mencionar, que solo se modelan las posiciones angulares y lineales de robot.

Para poder modelar las rotaciones del robot, es necesario definir sistemas o bases locales sobre los eslabones que componen el robot. Una vez definidas dichas bases, se generan o construyen las ecuaciones de la lazo.

Definición de sistemas o bases móviles

En la base del robot mostrado en la figura 3.1 se define el sistema de referencia cero SR_0 ($Ox_0y_0z_0$), rotando SR_0 un ángulo θ_1 en el eje Z se obtiene el sistema de referencia uno SR_1 y trasladando una distancia d_1 el SR_1 sobre el eje de las Z's se obtiene el SR_2 .

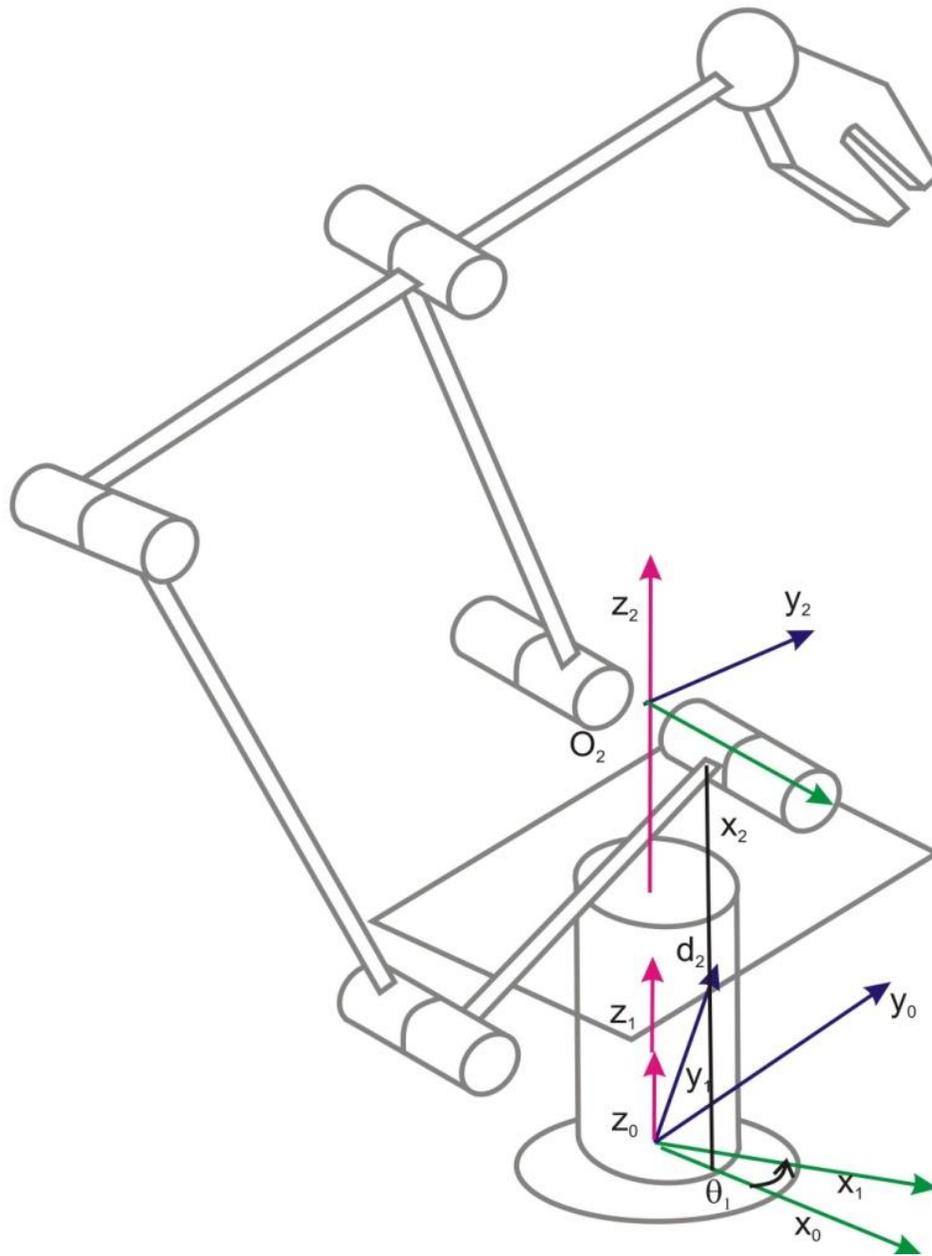


Figura 3.1. Base móvil 1

El sistema de referencia tres SR_3 se obtiene de trasladar SR_2 una distancia d_3 en el eje de las X 's, SR_4 se obtiene de rotar SR_3 un ángulo θ_4 en el eje de las X 's y SR_5 se obtiene de trasladar SR_4 una distancia d_5 en el eje de las Z . (Ver figura 3.2)

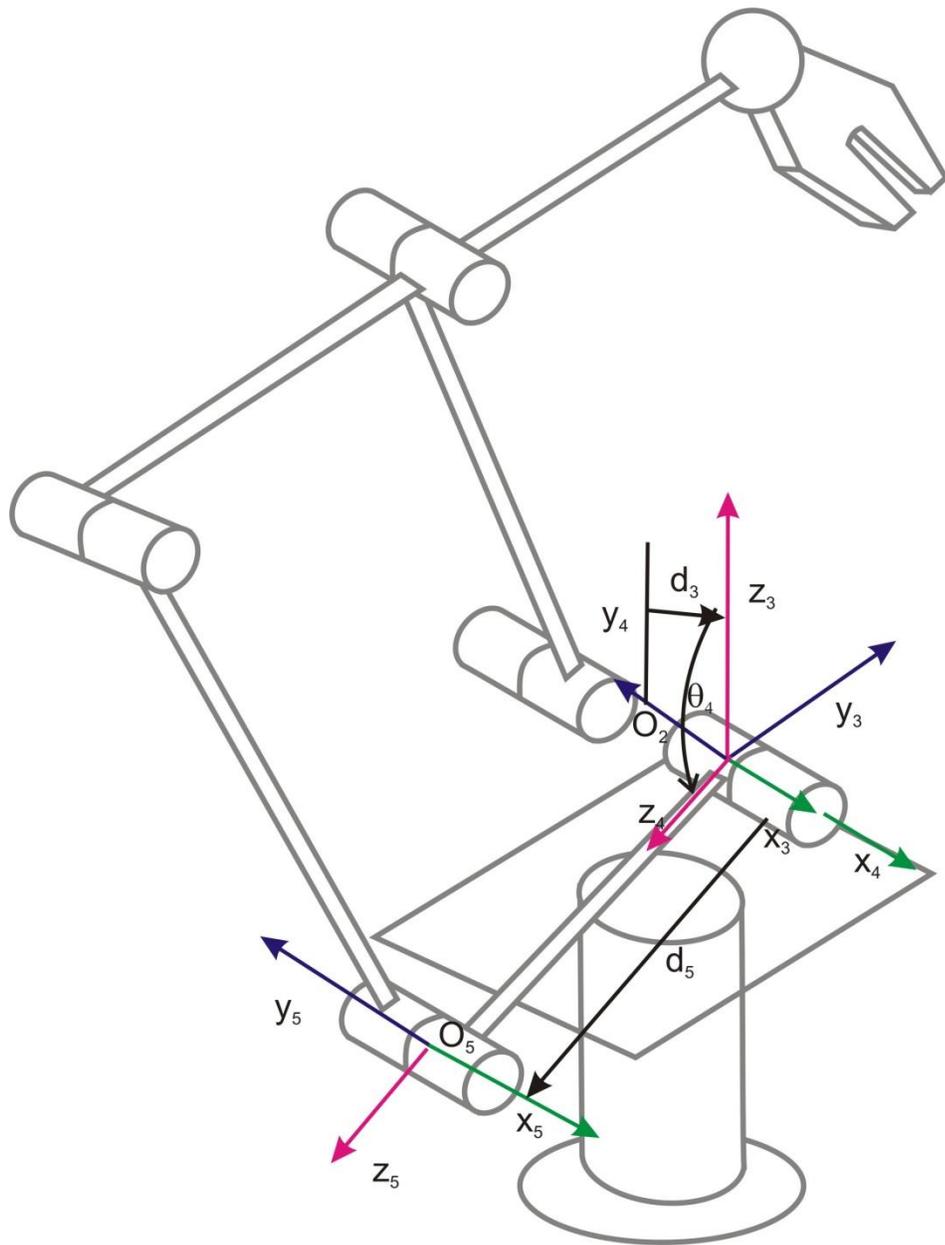


Figura 3.2. Base móvil 2

Por otro lado, el sistema de referencia seis SR_6 se obtiene de trasladar SR_5 una distancia $-d_6$ en el eje de las X 's, SR_7 se obtiene de rotar SR_6 un ángulo θ_7 en el eje de las X 's, SR_8 se obtiene trasladando SR_7 una distancia d_8 en el eje de las Z 's y SR_9 se obtiene de trasladar SR_8 una distancia $-d_9$ en el eje de las X 's. (Ver figura 3.3)

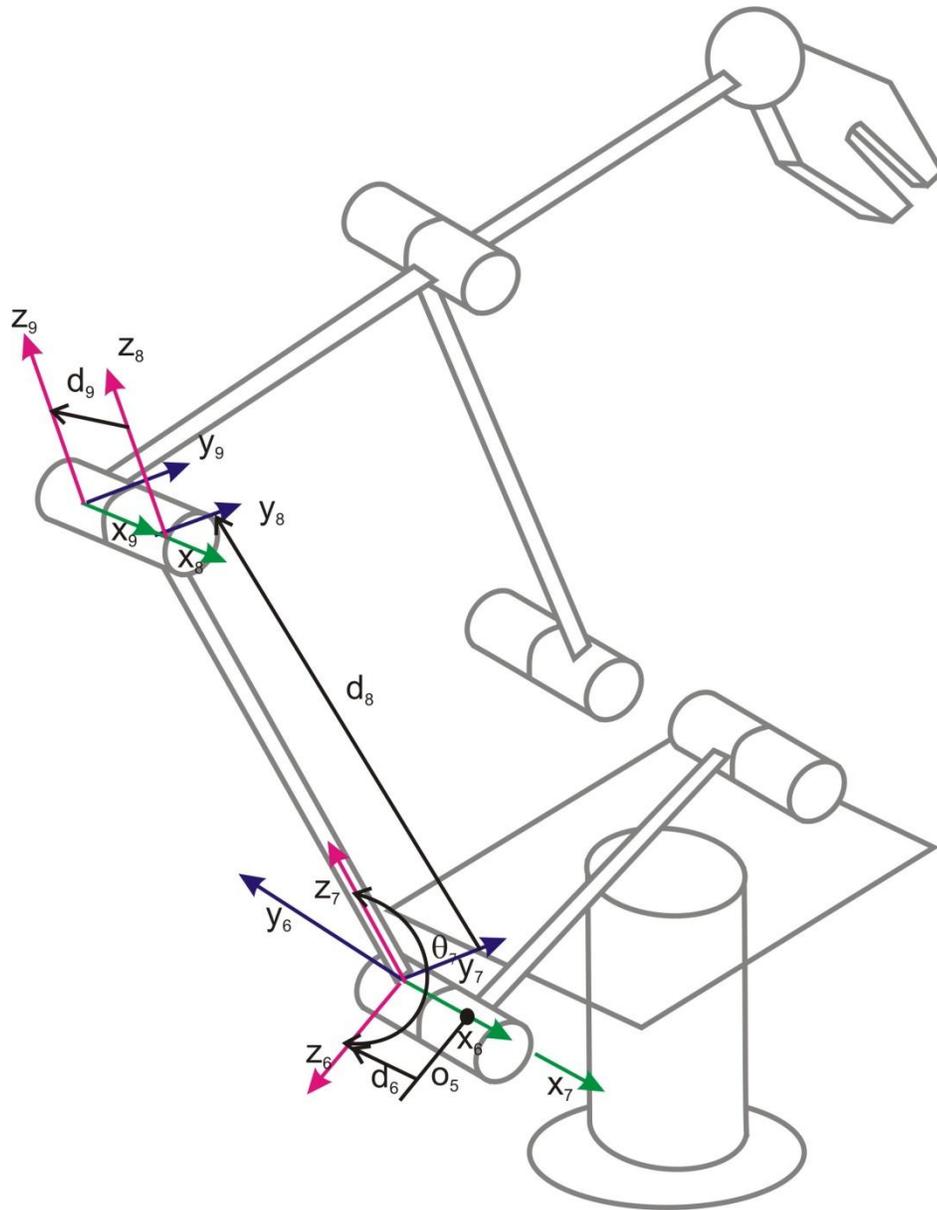


Figura 3.3. Base móvil 3

El sistema de referencia diez SR_{10} se obtiene de rotar SR_9 un ángulo θ_{10} en el eje de las X 's, SR_{11} se obtiene de trasladar SR_{10} una distancia d_{11} en el eje de las Z 's y SR_{12} se obtiene de trasladar SR_2 una distancia d_{12} en el eje de las X 's. (Ver figura 3.4).

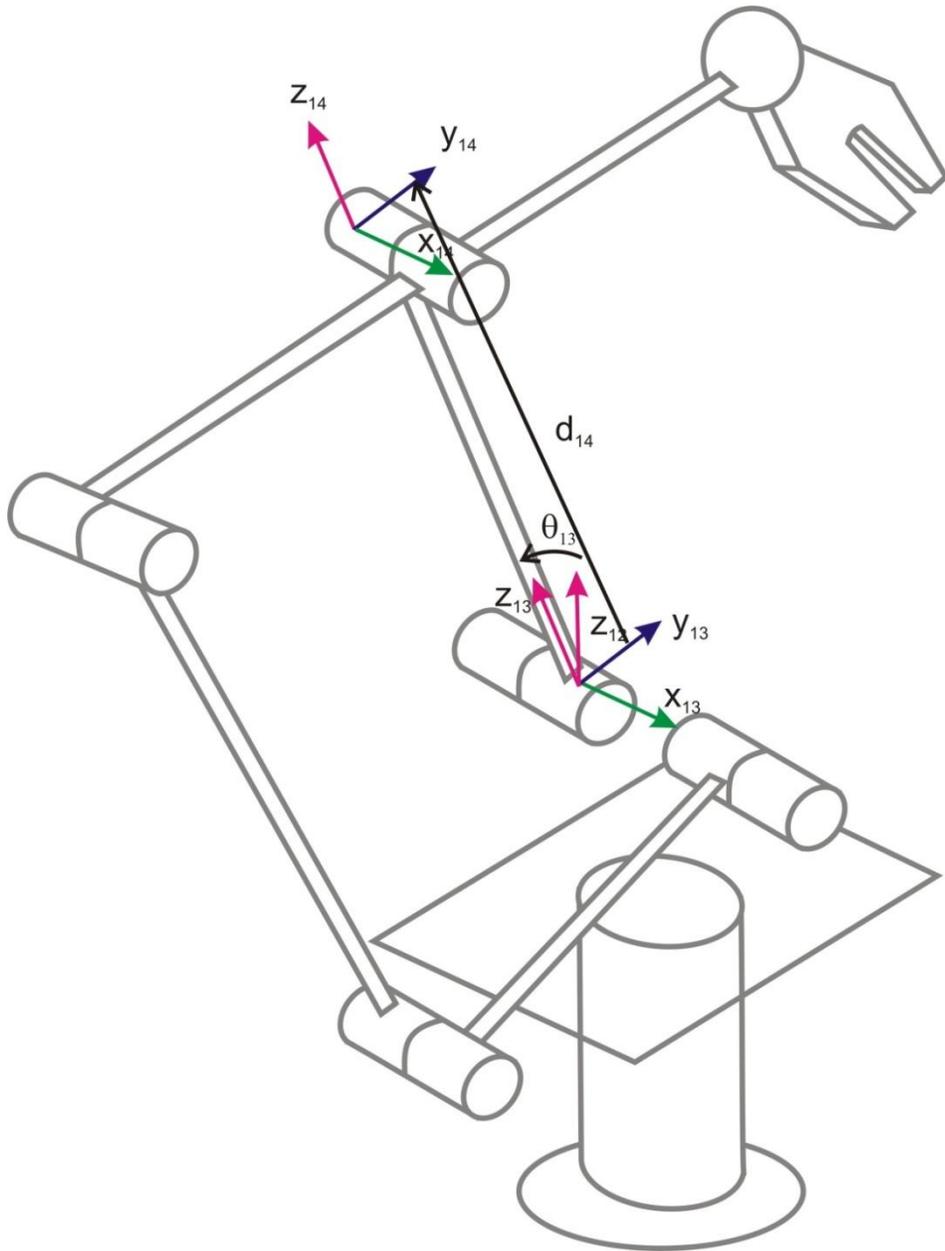


Figura 3.5 Base móvil 5

Por otro lado, el sistema de referencia quince SR_{15} se obtiene de trasladar SR_{14} una distancia d_{15} en el eje de las X's, SR_{16} se obtiene de rotar SR_{15} un ángulo θ_{16} en el eje de las X's y SR_{17} se obtiene de trasladar SR_{16} una distancia d_{17} en el eje de las Z's. (Ver figura 3.6)

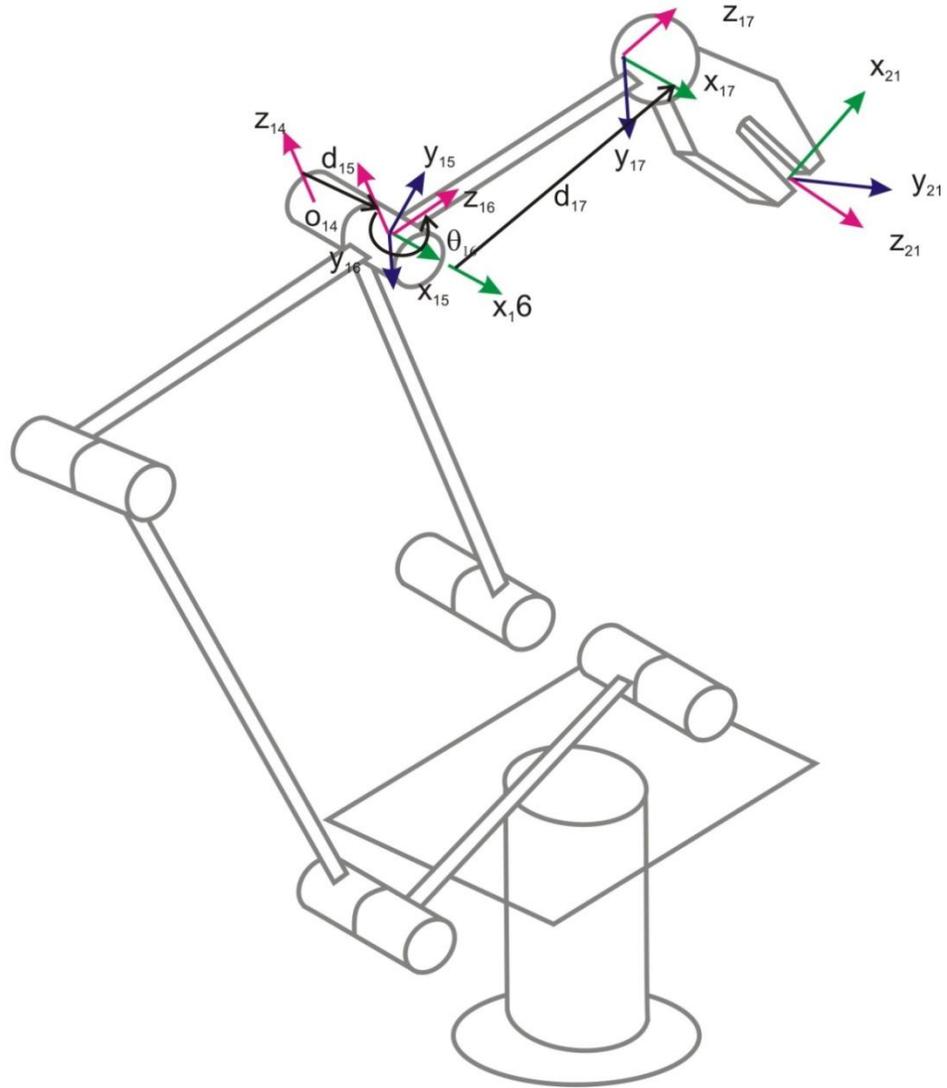


Figura 3.6. Base móvil 6

Además, el sistema de referencia dieciocho SR_{18} , se obtiene al rotar SR_{17} un ángulo θ_{18} en el eje de las Z, SR_{19} se obtiene de rotar SR_{18} un ángulo θ_{19} en el eje de las Y's, SR_{20} se obtiene al rotar SR_{19} un ángulo θ_{20} en el eje de las X's y SR_{21} se obtiene de trasladar SR_{20} una distancia d_{21} en el eje de las Z. (Ver figura 3.7)

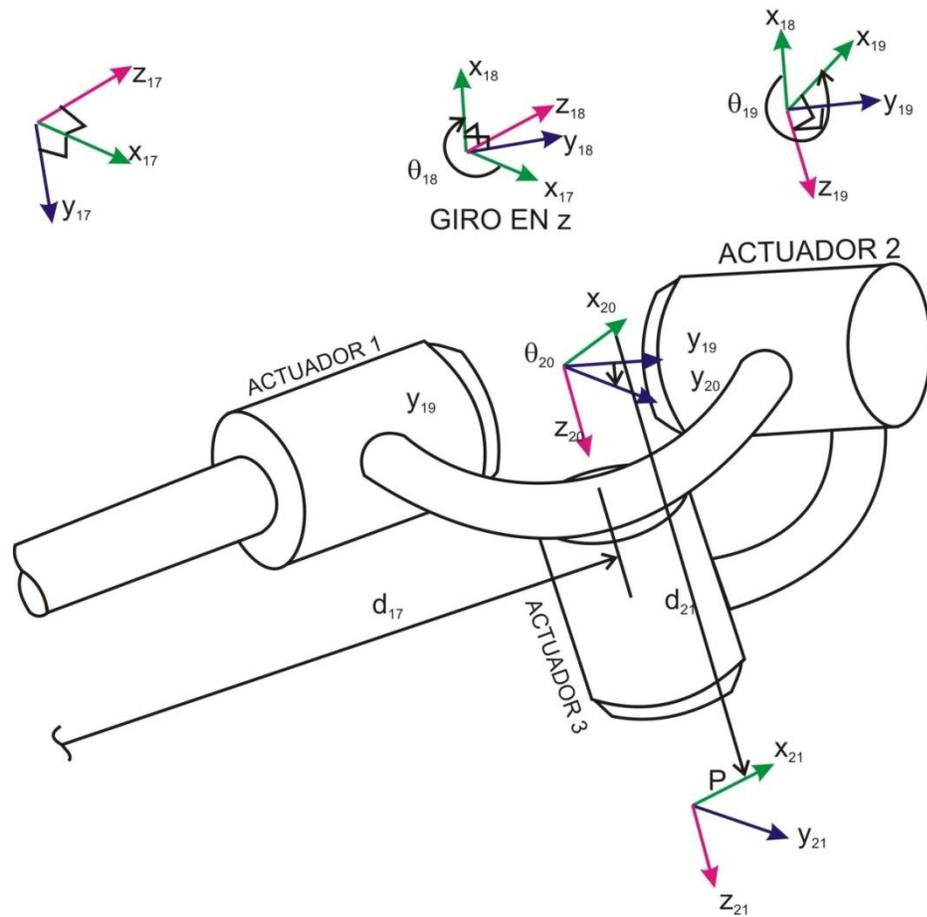


Figura 3.7 Bases móviles sobre el gripper

Por último, en la figura 3.8 definimos el vector de posición del órgano terminal mediante el vector $\mathbf{R}_p=(x,y,z)$. Además el sistema de referencia $(O_xp y_p z_p)$, está orientado respecto al sistema $(O_xo_yo_zo)$, mediante los tres ángulos de Euler (ψ,θ,ϕ)

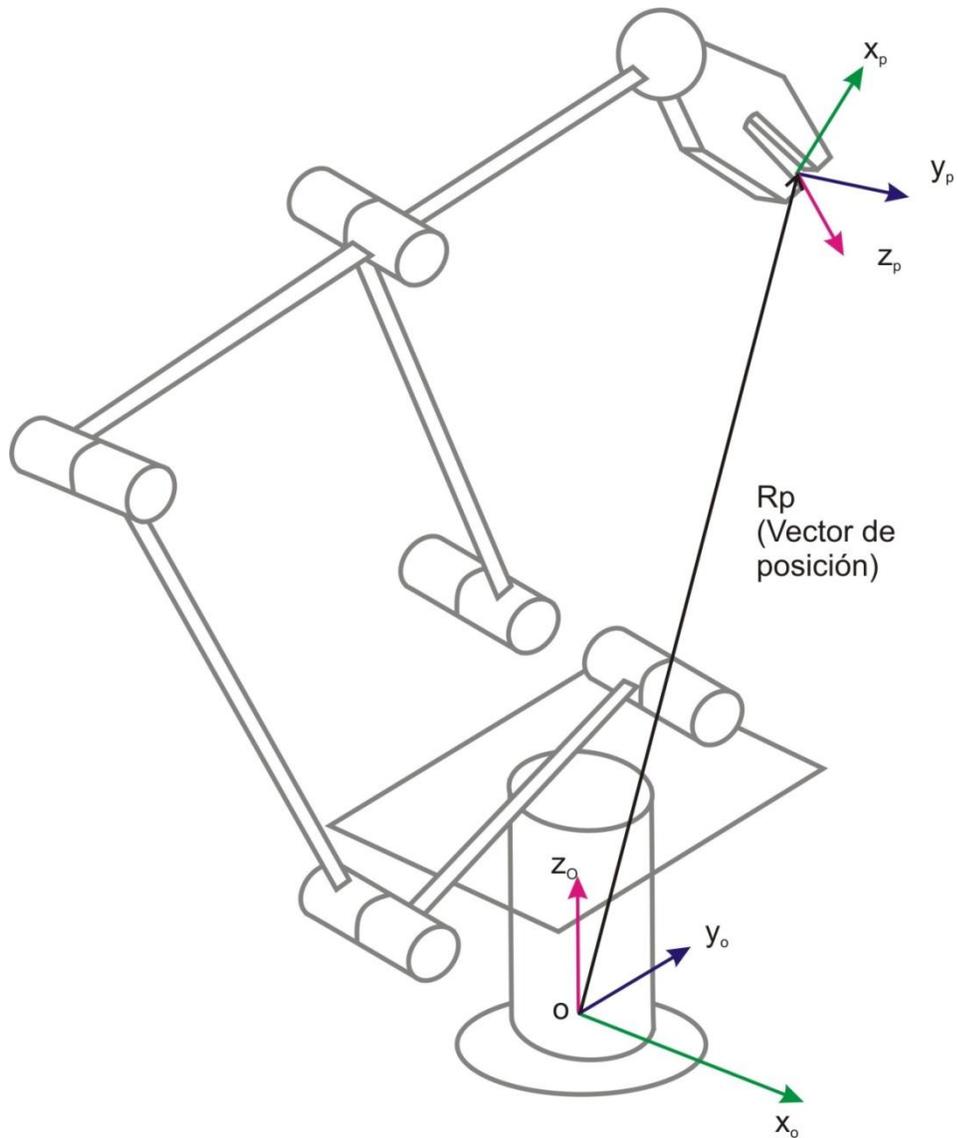


Figura 3.8. Definición del vector de posición

Ecuaciones de lazo

Una vez definidas los sistemas locales sobre los eslabones que componen el robot motivo de estudio en esta tesis, el paso siguiente es determinar las ecuaciones de lazo, las cuales serán usadas más adelante, para formular los problemas cinemáticos. Dichas ecuaciones se obtienen de la manera siguiente:

Utilizando las bases locales de cada eje de rotación y traslación, mostradas en las figuras de la sección anterior, se plantean las siguientes ecuaciones matriciales de lazo:

(1)

(2)

(3)

(4)

(5)

(6)

Definiendo:

(7)

(8)

(9)

(10)

(11)

(12)

(13)

(14)

(15)

(16)

Donde:

(17)

Cabe mencionar, que el robot motivo de estudio está compuesto por dos cadenas cinemáticas independientes, por lo tanto, existen dos ecuaciones de lazo, una por cada cadena. Tales ecuaciones son las siguientes:

1) LAZO MATRICIAL 1

(18)

2) LAZO MATRICIAL 2

(19)

3.2 Modelo Cinemático Inverso

Una vez obtenidas las ecuaciones de lazo que determinan el movimiento rotacional y de desplazamiento lineal de los eslabones del robot estudiado, a continuación se procederá a obtener y explicitar el modelo que permita formular el problema inverso. La modelación se hará tomando en cuenta cada rotación independiente. Dado el vector de posición $R_p=(x,y,z)$ y los ángulos (ψ,θ,ϕ) , que definen respectivamente la posición del origen del sistema de referencia del órgano terminal y la orientación del sistema de referencia del órgano terminal, determinar los ángulos $\theta_1, \theta_4, \theta_7, \theta_{10}, \theta_{13}, \theta_{16}, \theta_{18}, \theta_{19}, \theta_{20}$ esto es:

Calculando Ángulo

De la ecuación (18) se tiene:

(18)

Para poder despejar el ángulo θ_1 , se aplica \cos y después \arccos en ambos lados izquierdos de la igualdad, es decir:

Así se tiene que:

(20)

Donde:

Además, si se elijen las ecuaciones más simples, por ejemplo:

Se tienen las siguientes ecuaciones:

Por otro lado, si se sustituye el valor de x de la primera ecuación, en la segunda se obtiene:

Al factorizar en $x^2 - 1$ se tiene:

Renombrando:

(21)

Donde:

La ecuación (21) tiene la solución final:

$$\boxed{\quad - \quad = \quad \underline{\underline{\quad}} \quad}$$

(22)

Para poder elegir el cuadrante correcto se usa la función ArcTan2. El signo \pm indica dos posibilidades de orientar el cuerpo con el ángulo . Se tomó el signo +, ya que dio la configuración deseada para el simulador del robot.

Calculando Ángulo

De la ecuación (18) se tiene:

(18)

Para poder despejar el ángulo , se aplica la transformación y después en ambos lados izquierdos de la igualdad y se aplica en ambos lados derechos de la igualdad, es decir:

Así se tiene:

(23)

Donde:

Al igual que el proceso hecho en la sección anterior, se eligen las ecuaciones más simples, igualando por ejemplo:

y se tienen las siguientes ecuaciones:

La primera ecuación no considera el ángulo α , por lo tanto, no será útil para el cálculo. De la segunda ecuación se despeja $\sin \alpha$ y de la tercera ecuación se despeja $\cos \alpha$, entonces se tiene:

$$\sin \alpha = \frac{a}{c} \tag{24}$$

$$\cos \alpha = \frac{b}{c} \tag{25}$$

Estos ángulos se relacionan mediante la siguiente identidad trigonométrica:

$$\sin^2 \alpha + \cos^2 \alpha = 1 \tag{26}$$

Sustituyendo las ecuaciones (24) y (25) en (26) y agrupando en $\sin^2 \alpha$ se tiene que:

$$\sin^2 \alpha + \left(\frac{b}{c}\right)^2 = 1 \tag{27}$$

Donde:

$$\sin \alpha = \frac{a}{c}$$

$$\cos \alpha = \frac{b}{c}$$

$$\boxed{\quad\quad\quad} \quad (29)$$

Nota: Cuando se graficó la función (29), el arctan2 calculó el valor del ángulo como -90° , siendo que el valor esperado debería ser 270° según la figura 3.6, para obtener la postura deseada para el simulador del robot se le sumó 360° al resultado.

Calculando Ángulo

En esta etapa del análisis ya se conocen los ángulos θ_1 y θ_2 . Para despejar el ángulo θ_3 , se parte de la ecuación (20):

$$(20)$$

Donde:

Eligiendo las ecuaciones más simples, se iguala por ejemplo:

y se tienen las siguientes ecuaciones:

$$(29)$$

$$(30)$$

$$(31)$$

Al despejar θ_3 de ecuación (29):

$$\text{---} \tag{32}$$

Y al sustituir la ecuación (32) en las ecuaciones (30) y (31) , se tiene que:

$$\tag{33}$$

$$\tag{34}$$

Por otro lado, se despeja de ecuación (33), y

$$\text{---}$$

Al sustituir la ecuación (35) en la ecuación (34) y simplificando:

Al despejar de la expresión:

$$\text{---}$$

Se obtiene:

$$\boxed{\text{---}} \tag{36}$$

Donde:

Calculando Ángulo

En esta etapa del análisis ya se conocen los ángulos ángulo , se parte de las ecuaciones (30) y (31) y se tiene que:

Para despejar el

(30)

(31)

Resolviendo para _____ y simplificando la expresión:

Renombrando:

(37)

(38)

Donde:

Por otro, se puede despejar el ángulo _____ de:

Y, por lo tanto,

(39)

Calculando Ángulo

En esta etapa del análisis ya se conocen los ángulos _____, se parte de la ecuación (20):

Para despejar el

(20)

Donde:

Al elegir las ecuaciones más simples, se tiene que:

Por lo tanto,

(30)

(31)

Al resolver para se tiene que:

$$\frac{\quad}{\quad}$$

Además, siendo:

$$\frac{\quad}{\quad}$$

(40)

$$\frac{\quad}{\quad}$$

(41)

Donde:

Se puede despejar el ángulo de la manera siguiente:

$$\frac{\quad}{\quad}$$

$$\boxed{\quad\quad\quad} \quad (42)$$

Calculando Ángulo

En esta etapa del análisis ya se conocen los ángulos α y β . Para despejar el ángulo γ , se parte de la ecuación (19). Esto es:

$$(19)$$

Para poder despejar el ángulo γ , se aplica la transformación $\cos(\pi - \gamma) = -\cos(\gamma)$ en ambos lados izquierdos de la igualdad y después $\sin(\pi - \gamma) = \sin(\gamma)$ en ambos lados derechos de la igualdad, es decir:

Así se tiene que:

$$(43)$$

Donde:

Al elegir las ecuaciones más simples, se iguala por ejemplo:

Al resolver para α y β :

$$\text{---} \tag{44}$$

$$\text{---} \tag{45}$$

Estos ángulos se relacionan mediante la siguiente identidad trigonométrica:

$$\tag{46}$$

El sustituir las ecuaciones (44) y (45) en (46) y agrupando en y se tiene que:

$$\tag{47}$$

Donde:

$$\text{---}$$

$$\text{---}$$

$$\text{---}$$

La ecuación (47) es equivalente a:

$$\boxed{\text{---} \quad \text{====}} \tag{48}$$

El signo \pm indica dos posibilidades de orientar el cuerpo con el ángulo α . Se tomó el signo $-$ y se le sumaron 360° , ya que dio la postura deseada para el simulador del robot.

Calculando Ángulo

En esta etapa del análisis ya se conocen los ángulos α y β . Para despejar el ángulo γ , se parte de la ecuación (43):

$$(43)$$

Para poder despejar el ángulo γ , se aplica la transformación en ambos lados izquierdos de la igualdad, es decir:

Así se tiene que:

$$(49)$$

Donde:

Al elegir las ecuaciones más simples. Esto es:

Resolviendo para α y β :

$$— (50)$$

$$— (51)$$

Renombrando variables se tiene que:

$$— (52)$$

Así se tiene:

(55)

Donde:

Al elegir las ecuaciones más simples, se tiene que:

Al resolver para x y y :

—

(56)

—

(57)

Renombrando:

—

(58)

—

(59)

Donde:

Por otro lado, de la ecuación

CAPÍTULO 4. SOFTWARE DE SIMULACIÓN

En este capítulo se presenta el diseño del software del robot paralelo motivo de estudio en esta tesis. La plataforma de programación y visualización es, para este caso, Visual Basic 6.0. Se tomó como base el software de simulación “Desarrollo de un programa de simulación de un modelo cinemático de un robot articulado de 6 grados de libertad CRS A465 del ITESCA” desarrollado por el Mtro. Javier Jacobo Peña [5] en el cual utiliza la librería de OpenGL para los gráficos en 3D.

4.1 Algoritmo de Programación

En el capítulo anterior se desarrolló el modelo matemático de cinemática inversa del cual se obtuvieron las ecuaciones para todos los ángulos de posición y orientación del robot, a continuación se presenta un algoritmo con el que procesará la información el software de simulación denominado RP5E (robot paralelo 5 eslabones)

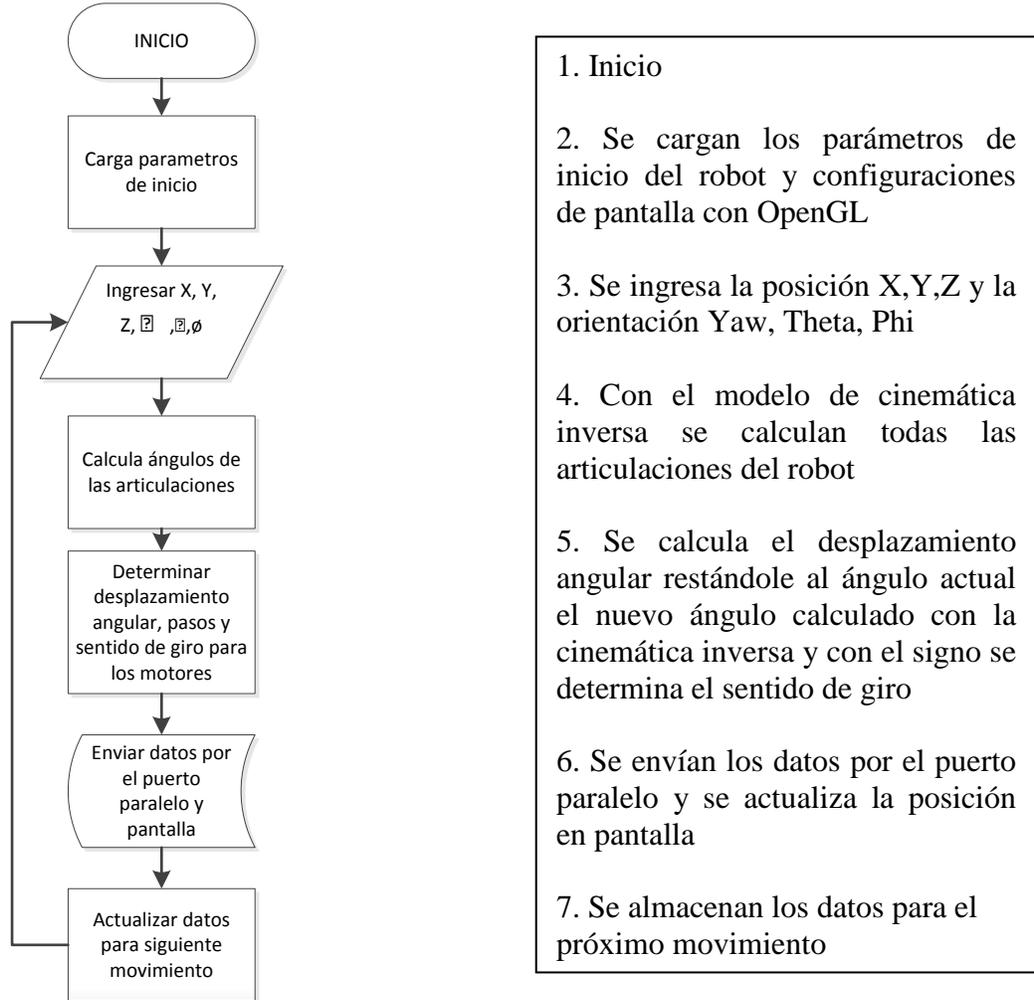


Fig. 4.1 Algoritmo de funcionamiento general

4.2 Interfaz de Simulación

Una vez elaborado el algoritmo que utilizará el software de simulación, se procedió a desarrollar la interfaz gráfica para demostrar la validez del modelo matemático de cinemática inversa. En la figura 4.2 podemos ver las secciones que contiene el software de simulación

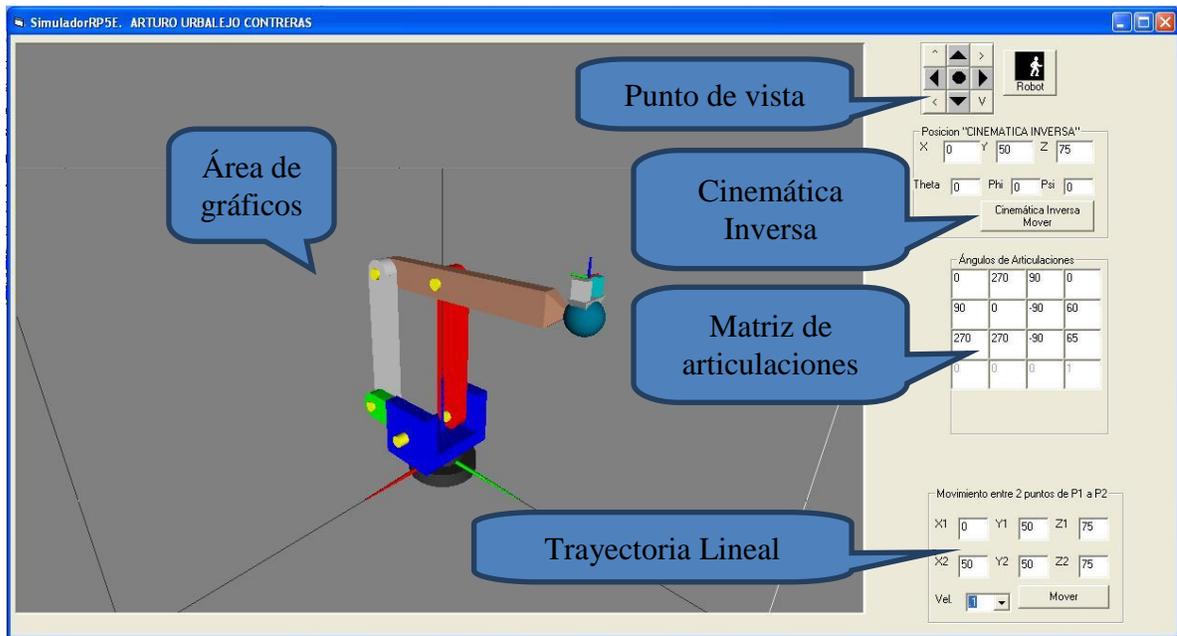


Figura 4.2 Interfaz del simulador

Área de gráficos



En esta sección del programa se muestran todos los gráficos en 3D que se hacen en OpenGL. De acuerdo con la figura anterior, al presionar el acceso “Robot” aparece el robot paralelo de 5 barras, donde los actuadores que controlan la posición en el espacio están relacionados con los ángulos θ_1 (base), θ_4 (eslabón verde) y θ_{13} (eslabón rojo).

Punto de Vista

En la sección de Punto de vista se puede rotar la perspectiva que se tiene del robot, se puede rotar hacia la derecha, izquierda, arriba y hacia abajo. Con esta opción se puede apreciar de mejor manera los movimientos que realizará el robot. En la figura 4.3 podemos ver como se cambió el punto de vista rotado hacia la izquierda y hacia arriba.

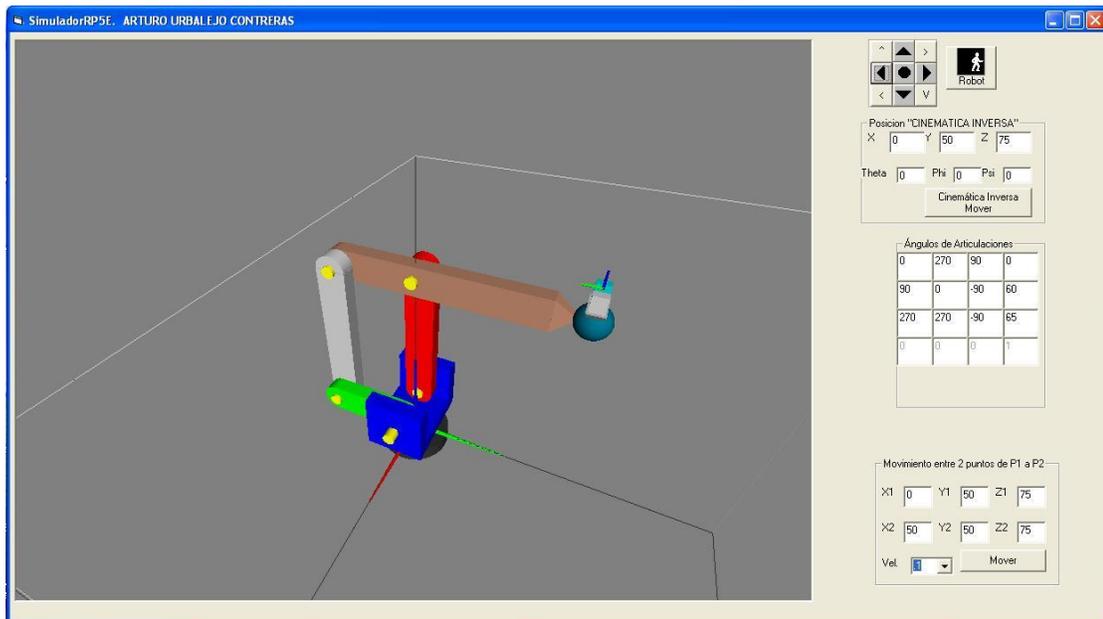


Figura 4.3 Punto de vista rotado

Cinemática Inversa

En esta sección del programa se ingresa el punto en el espacio X, Y, Z y la orientación ψ, θ, ϕ y se le da click en el botón “Cinemática Inversa Mover” y el simulador ejecuta el algoritmo mostrado en la figura 4.1 donde calcula los ángulos, con el modelo cinemático inverso, para llegar a ese punto y orientación espacial, los muestra en el área de gráficos y calcula en base a el desplazamiento angular la cantidad de pasos para cada motor para enviarlos por el puerto paralelo de la computadora. A continuación se muestra la posición inicial del simulador y diferentes posiciones y orientaciones en el espacio.

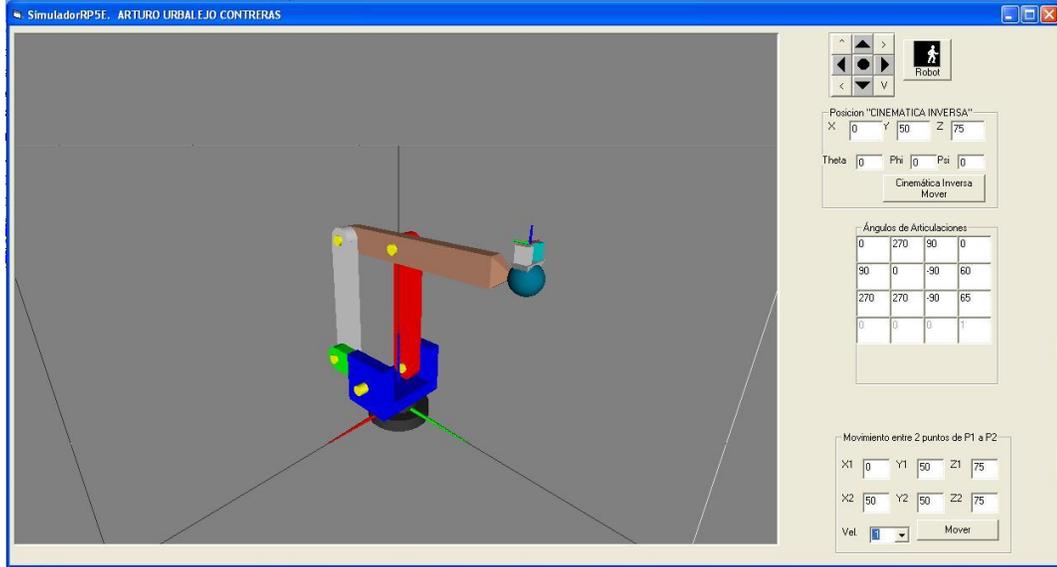


Figura 4.4. Configuración Inicial del robot

La configuración mostrada en la figura anterior es llamada “home” o configuración de referencia la cual queda caracterizada por: las coordenadas $(X_p, Y_p, Z_p) = (0, 50, 75)$ con orientación $(\psi, \theta, \phi) = (0, 0, 0)$.

La figura 4.5 muestra una nueva configuración del robot la cual se obtiene de variar el valor de la X a 20 unidades, la base θ_1 gira el ángulo correspondiente manteniendo los demás actuadores en la misma posición.

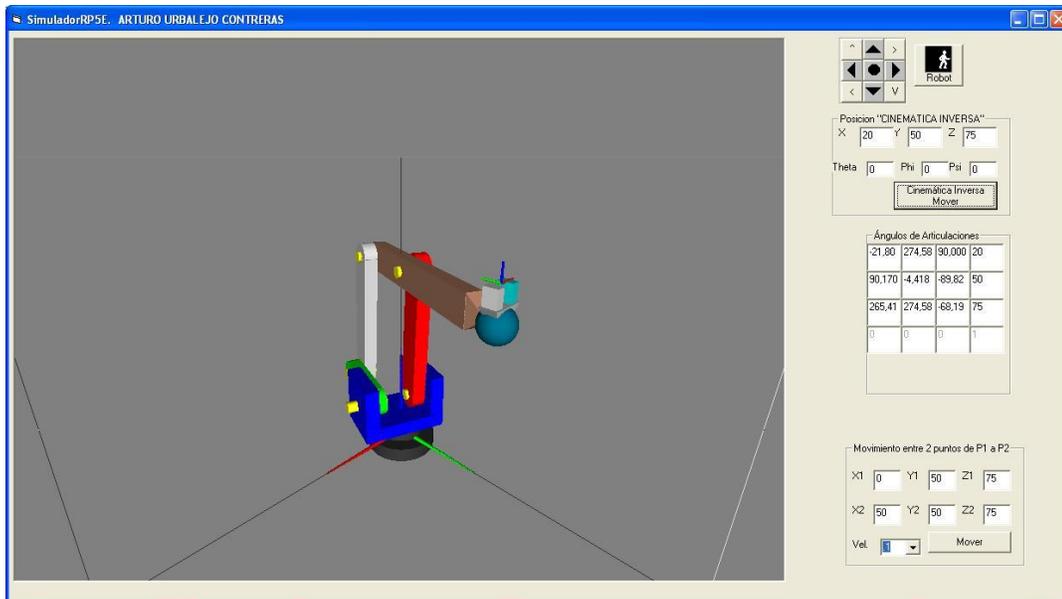


Figura 4.5 Configuración de posición (20,50,75) orientación(0,0,0)

Tomando en cuenta los valores anteriores se disminuye el valor de Z en 25 unidades (de 75 a 50) y se puede observar como el órgano efector final del manipulador solamente disminuyo la altura. Ver fig. 4.6

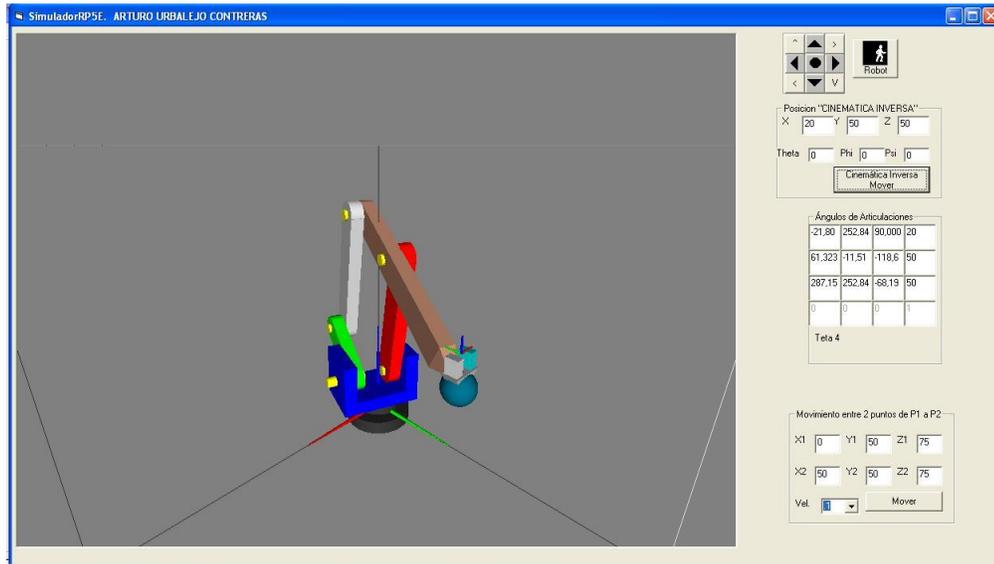


Figura 4.6 Configuración de posición (20,50,50) orientación(0,0,0)

Ahora si se incrementa en 20 unidades el valor de Y (de 50 a 70) y se puede observar como el órgano efector final del manipulador solamente se desplaza en el eje Y. Ver figura. 4.7

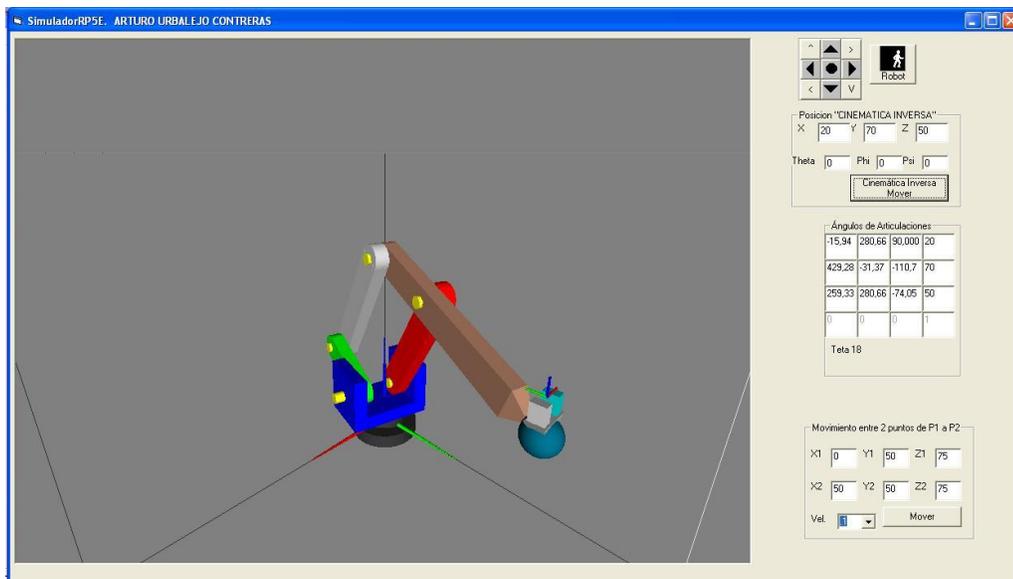


Figura 4.7 Configuración de posición (20,70,50) orientación(0,0,0)

En la figura 4.8 se muestra la posición de inicio del robot (0,50,75) con una orientación de (30,0,0) y con el punto de vista rotado de tal manera que se aprecie bien la rotación

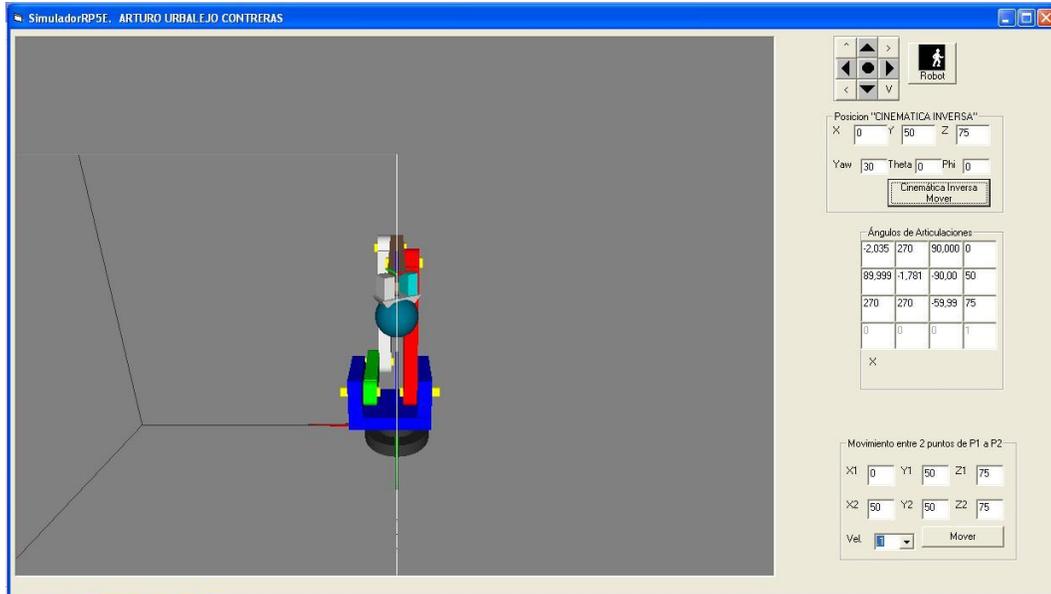


Figura 4.8 Configuración de posición (0,50,75) orientación(30,0,0)

Tomando en cuenta la posición y orientación anterior, se modifica el segundo ángulo de orientación de cero a treinta grados (30,0,0) a (30,30,0) ver figura 4.9

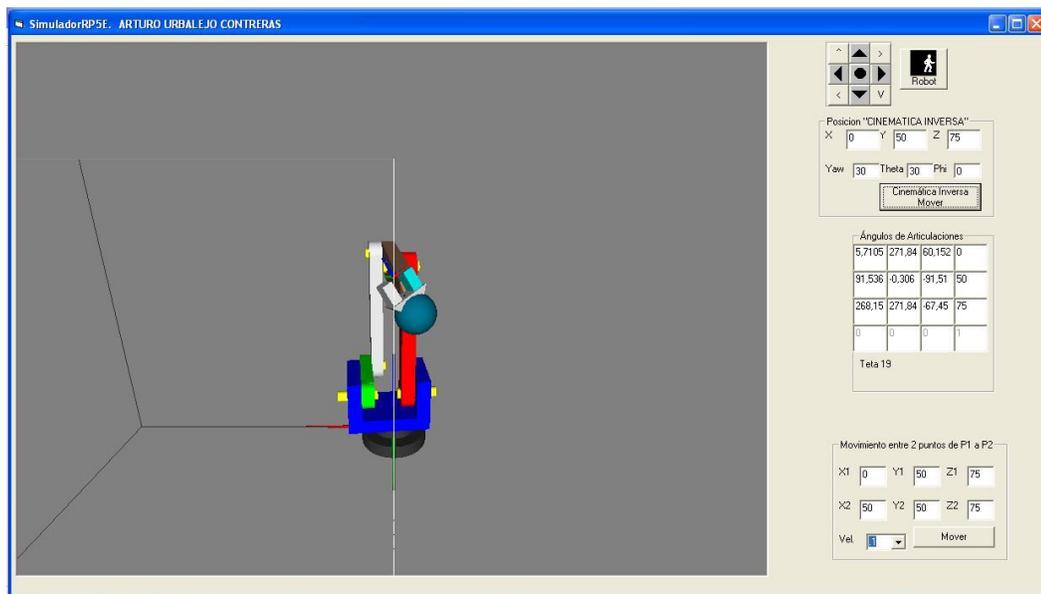


Figura 4.9 Configuración de posición (0,50,75) orientación(30,30,0)

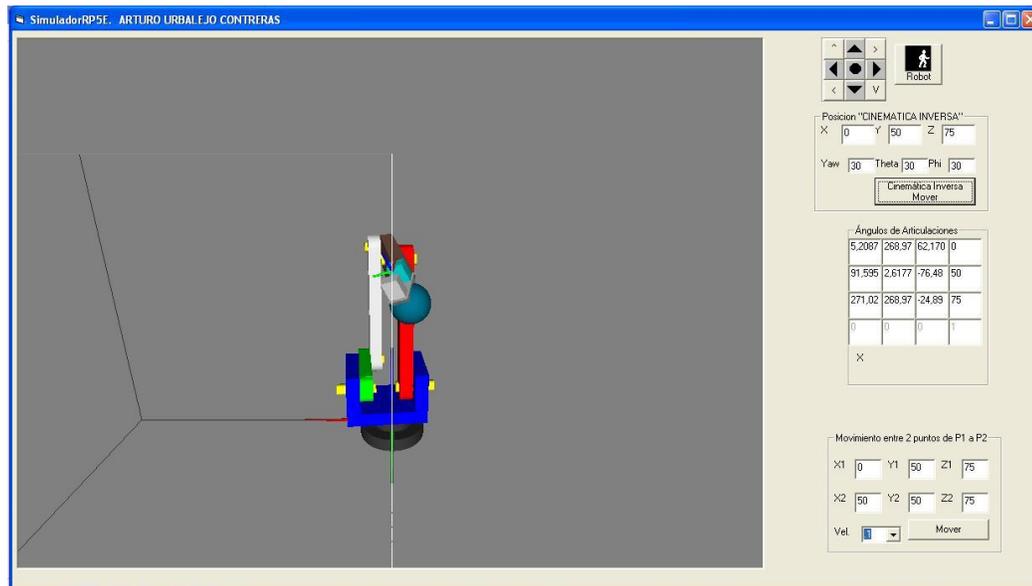


Figura 4.10 Configuración de posición (0,50,75) orientación(30,30,30)

De acuerdo con las figuras anteriores se puede ver que el simulador se posiciona correctamente en el espacio en base a las coordenadas del punto y la orientación especificados, pero como no se restringieron los movimientos de cada articulación en el simulador también es posible que cuando no se conoce la orientación que se esté especificando el órgano efector final se posiciona, encima del manipulador como se muestra en la figura 4.11

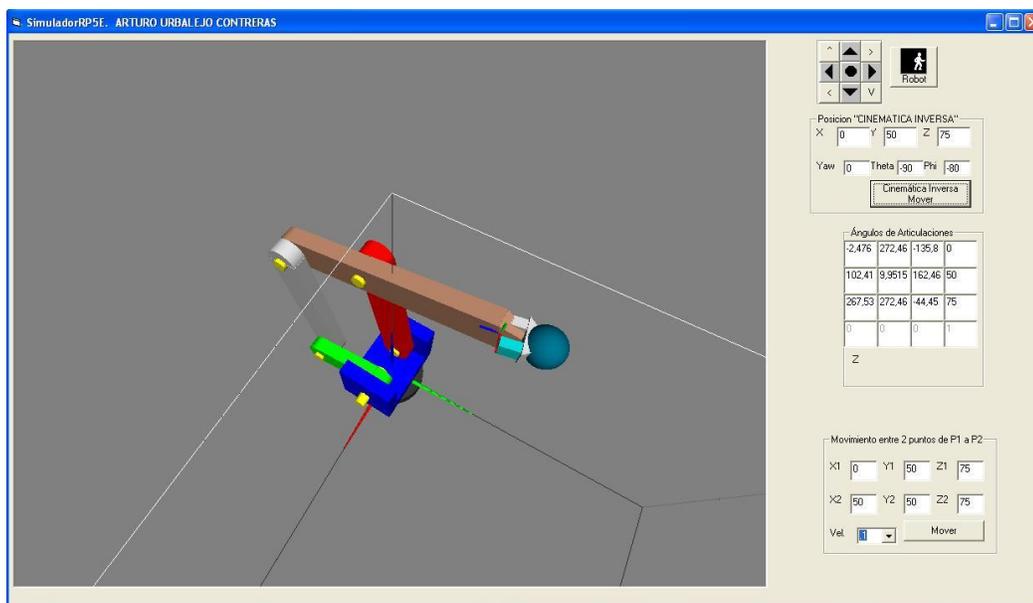


Figura 4.11 Configuración de posición (0,50,75) orientación(0,-90,-80)

También hay puntos en el espacio que el manipulador no puede alcanzar debido a que al resolver la cinemática inversa uno o varios de sus ángulos no tienen solución a este problema se le denomina singularidad en robótica.

Matriz de Articulaciones

En esta sección se pueden ver los ángulos de todas las articulaciones una vez posicionado el robot en el punto en el espacio, ver figura 4.12. Se posiciona el ratón sobre el valor del ángulo en la matriz y se despliega el nombre del ángulo como se muestra en la imagen el ángulo teta 20.

| | | | |
|-----|-----|-----|----|
| 0 | 270 | 90 | 0 |
| 90 | 0 | -90 | 50 |
| 270 | 270 | -90 | 75 |
| 0 | 0 | 0 | 1 |

Teta 20

Figura 4.12 Matriz de articulaciones

Trayectoria Lineal

En esta sección se selecciona el punto de inicio ($X1, Y1, Z1$) y el punto final ($X2, Y2, Z2$) y se da click al botón Mover (ver figura 4.13) y el simulador mueve el robot en una trayectoria lineal entre los dos puntos, en la opción Vel se puede cambiar la velocidad de simulación.

Movimiento entre 2 puntos de P1 a P2

X1 Y1 Z1

X2 Y2 Z2

Vel.

Figura 4.13 Trayectoria Lineal

4.3 Programación de las ecuaciones en Visual Basic 6.0

El simulador RP5E se desarrolló en la plataforma Visual Basic 6.0 y en esta sección se muestran las ecuaciones que gobiernan el modelo de posición del robot paralelo. A continuación se presenta el código de programación del modelo cinemático inverso que se obtuvo en el capítulo 3:

```
Fi = Val(txtFi.Text) * (Pi / 180)
Psi = Val(txtPsi.Text) * (Pi / 180)
teta = Val(txtTeta.Text) * (Pi / 180)

a11 = Cos(teta) * Cos(Fi) * Cos(Psi) - Sin(Fi) * Sin(Psi)
a12 = -Cos(teta) * Cos(Psi) * Sin(Fi) - Cos(Fi) * Sin(Psi)
a13 = Cos(Psi) * Sin(teta)
a14 = -a13 * d21 + Val(txtX.Text)

a21 = Cos(Psi) * Sin(Fi) + Cos(teta) * Cos(Fi) * Sin(Psi)
a22 = Cos(Fi) * Cos(Psi) - Cos(teta) * Sin(Fi) * Sin(Psi)
a23 = Sin(teta) * Sin(Psi)
a24 = -a23 * d21 + Val(txtY.Text)

a31 = -Cos(Fi) * Sin(teta)
a32 = Sin(teta) * Sin(Fi)
a33 = Cos(teta)
a34 = -a33 * d21 + Val(txtZ.Text)

*****TETA1*****
A1 = a13 * d21 - Val(txtX.Text)
B1 = a23 * d21 - Val(txtY.Text)
C1 = d15 - d12

Teta1 = Atan2(B1, A1) + ACos(-C1 / Sqr(A1 ^ 2 + B1 ^ 2))

*****TETA 13*****
A2 = 2 * (d14 / d17 ^ 2) * (D2 + a33 * d21 - Val(txtZ.Text))
B2 = 2 * (d14 / d17 ^ 2) * ((Val(txtY.Text) - a23 * d21) * Cos(Teta1) + (a13 * d21 - Val(txtX.Text)) * Sin(Teta1))
C2 = (d14 ^ 2 / d17 ^ 2) - 1
E2 = C2 + AD2

Teta13 = Atan2(B2, A2) - ACos(-E2 / Sqr(A2 ^ 2 + B2 ^ 2))

*****TETA 16*****
A3 = 1 * (d14 + (D2 + a33 * d21 - Val(txtZ.Text)) * Cos(Teta13) + ((Val(txtY.Text) - a23 * d21) * Cos(Teta1) + (a13 * d21 - Val(txtX.Text)) * Sin(Teta1)) * Sin(Teta13))
B3 = (a23 * d21 - Val(txtY.Text)) * Cos(Teta1) * Cos(Teta13) + (Val(txtX.Text) - a13 * d21) * Sin(Teta1) * Cos(Teta13) + (D2 + a33 * d21 - Val(txtZ.Text)) * Sin(Teta13)

Teta16 = Atan2(B3, -A3) + 360 * (Pi / 180)
```

***** TETA 4*****

$$A7 = (2 * d5 / d8 ^ 2) * (d11 * \text{Cos}(\text{Teta13} + \text{Teta16}) - d14 * \text{Cos}(\text{Teta13}))$$

$$B7 = (2 * d5 / d8 ^ 2) * (d11 * \text{Sin}(\text{Teta13} + \text{Teta16}) - d14 * \text{Sin}(\text{Teta13}))$$

$$C7 = (1 / d8 ^ 2) * (d5 ^ 2 - d8 ^ 2 + d11 ^ 2 + d14 ^ 2 - 2 * d11 * d14 * \text{Cos}(\text{Teta16}))$$

$$\text{Teta4} = \text{Atan2}(B7, A7) - \text{ACos}(-C7 / \text{Sqr}(A7 ^ 2 + B7 ^ 2)) + 2 * \text{Pi}$$

***** TETA 7*****

$$A8 = d14 * \text{Cos}(\text{Teta13} - \text{Teta4}) - d11 * \text{Cos}(\text{Teta13} + \text{Teta16} - \text{Teta4}) - d5$$

$$B8 = d14 * \text{Sin}(\text{Teta13} - \text{Teta4}) - d11 * \text{Sin}(\text{Teta13} + \text{Teta16} - \text{Teta4})$$

$$\text{Teta7} = \text{Atan2}(B8, A8) + 2 * \text{Pi}$$

***** TETA 10*****

$$A9 = d14 * \text{Cos}(\text{Teta13} - \text{Teta4} - \text{Teta7}) - d5 * \text{Cos}(\text{Teta7}) - d8$$

$$B9 = d5 * \text{Sin}(\text{Teta7}) + d14 * \text{Sin}(\text{Teta13} - \text{Teta4} - \text{Teta7})$$

$$\text{Teta10} = \text{Atan2}(B9, A9) + 2 * \text{Pi}$$

***** TETA 18*****

$$A4 = a13 * \text{Cos}(\text{Teta1}) + a23 * \text{Sin}(\text{Teta1})$$

$$B4 = ((a13 * \text{Sin}(\text{Teta1}) - a23 * \text{Cos}(\text{Teta1})) * (\text{Sin}(\text{Teta13}) - \text{Cot}(\text{Teta16}) * \text{Cos}(\text{Teta13})) + a33 * (\text{Cos}(\text{Teta13}) + \text{Cot}(\text{Teta16}) * \text{Sin}(\text{Teta13}))) * \text{Sin}(\text{Teta16})$$

$$\text{Teta18} = -\text{Atan2}(B4, A4)$$

***** TETA 19*****

$$A5 = a33 * \text{Cos}(\text{Teta13} + \text{Teta16}) + (a13 * \text{Sin}(\text{Teta1}) - a23 * \text{Cos}(\text{Teta1})) * \text{Sin}(\text{Teta13} + \text{Teta16})$$

$$B5 = \text{Cos}(\text{Teta13} + \text{Teta16}) * (-a13 * \text{Sin}(\text{Teta1}) + a23 * \text{Cos}(\text{Teta1})) + a33 * \text{Sin}(\text{Teta13} + \text{Teta16})$$

$$E5 = \text{Sin}(\text{Teta18}) * A5$$

$$\text{Teta19} = \text{Atan2}(B5, E5)$$

***** TETA 20*****

$$A6 = \text{Cos}(\text{Teta18}) * \text{Cos}(\text{Teta19}) * (a11 * \text{Cos}(\text{Teta1}) + a21 * \text{Sin}(\text{Teta1})) - (a12 * \text{Cos}(\text{Teta1}) + a22 * \text{Sin}(\text{Teta1})) * \text{Sin}(\text{Teta18})$$

$$B6 = -\text{Cos}(\text{Teta18}) * \text{Cos}(\text{Teta19}) * (a12 * \text{Cos}(\text{Teta1}) + a22 * \text{Sin}(\text{Teta1})) - (a11 * \text{Cos}(\text{Teta1}) + a21 * \text{Sin}(\text{Teta1})) * \text{Sin}(\text{Teta18})$$

$$\text{Teta20} = \text{Atan2}(B6, A6)$$

Las siguientes conversiones son importantes:

$$\text{Teta1} = \text{Teta1} * (180 / \text{Pi})$$

$$\text{Teta13} = (180 / \text{Pi}) * \text{Teta13}$$

$$\text{Teta16} = (180 / \text{Pi}) * \text{Teta16}$$

$$\text{Teta18} = (180 / \text{Pi}) * \text{Teta18}$$

$$\text{Teta19} = (180 / \text{Pi}) * \text{Teta19}$$

$$\text{Teta20} = (180 / \text{Pi}) * \text{Teta20}$$

$$\text{Teta4} = (180 / \text{Pi}) * \text{Teta4}$$

```
Teta7 = (180 / Pi) * Teta7
Teta10 = (180 / Pi) * Teta10
```

Dichas conversiones están asociadas con las relaciones entre los ángulos, esto es, de radianes a grados.

4.4 Librería OpenGL

Para la visualización de los gráficos en 3D se utilizó la librería OpenGL la cual cuenta con funciones específicas para crear objetos en 3D y manipularlos en el espacio. A continuación se muestra el código para trabajar con la librería OpenGL:

```
Option Explicit
Dim hRC As Long

Sub EnableOpenGL(ghDC As Long)
Dim pfd As PIXELFORMATDESCRIPTOR, PixFormat As Long

ZeroMemory pfd, Len(pfd)
pfd.nSize = Len(pfd)
pfd.nVersion = 1
pfd.dwFlags = PFD_DRAW_TO_WINDOW Or PFD_SUPPORT_OPENGL Or PFD_DOUBLEBUFFER
Or PFD_GENERIC_FORMAT
pfd.iPixelFormat = PFD_TYPE_RGBA
pfd.cColorBits = 24
pfd.cDepthBits = 16
pfd.cStencilBits = 1
pfd.iLayerType = PFD_MAIN_PLANE

PixFormat = ChoosePixelFormat(ghDC, pfd)
If PixFormat = 0 Then GoTo ee
SetPixelFormat ghDC, PixFormat, pfd
hRC = wglCreateContext(ghDC)
wglMakeCurrent ghDC, hRC

Exit Sub
ee: MsgBox "Can't create OpenGL context!", vbCritical, "Error"
End Sub

Sub DisableOpenGL()
wglMakeCurrent 0, 0
wglDeleteContext hRC
End Sub

'se habilita la librería OpenGL en el área de gráficos
EnableOpenGL Picture1.hDC

glEnable GL_LIGHTING
```

'Establece un vector 3D para la posición de la luz y establece una luz direccional
'con "0" en el último parámetro

```
PosLuz(0) = 0  
PosLuz(1) = 0  
PosLuz(2) = 100  
PosLuz(3) = 0
```

'Usa una luz blanca "GL_LIGHT0" en la posición establecida
glLightfv GL_LIGHT0, GL_POSITION, PosLuz(0)

'Activa la luz a usar
glEnable GL_LIGHT0

'Permite que se iluminen ambas caras, frontal y posterior de los polígonos
glLightModeli GL_LIGHT_MODEL_TWO_SIDE, GL_TRUE

'Activa la utilización de la función "glColorMaterial"
glEnable GL_COLOR_MATERIAL

'Establece las caras frontal o posterior a iluminar
glColorMaterial GL_FRONT_AND_BACK, GL_AMBIENT_AND_DIFFUSE

'Establece la forma como los materiales reflejan la luz
glMaterialfv GL_FRONT_AND_BACK, GL_SHININESS, 20

'Permite que "glCullFace" pueda seleccionar la cara frontal o posterior para no dibujar
glEnable GL_CULL_FACE

'Establece el color con el que se limpia la pantalla con "glClear GL_COLOR_BUFFER_BIT"
glClearColor 0.5, 0.5, 0.5, 0

'Establece que se dibujen los píxeles más cercanos al observador cuando se active
'la prueba de profundidad con "glEnable GL_DEPTH_TEST"
glDepthFunc GL_LESS

'Activa la prueba de profundidad de acuerdo al parámetro establecido por "glDepthFunc"
glEnable GL_DEPTH_TEST

```
PVistaOriginal = AVect(100, 100, 100)  
PVistaActual = PVistaOriginal  
CVista = AVect(0, 0, 50)
```

```
Ax1 = 0  Ax2 = 0  Ax3 = 0  Ax4 = 0  Ax5 = 0  Ax6 = 0  
EjeX = AVect(1, 0, 0)  
EjeY = AVect(0, 1, 0)  
EjeZ = AVect(0, 0, 1)  
Eje1Actual = TimH(EjeX)  
AxisZOriginal = AVect(0, 0, 1)  
AxisXOriginal = VectNormal(CruzAB(AxisZOriginal, PVistaOriginal))  
AxisYOriginal = VectNormal(CruzAB(PVistaOriginal, AxisXOriginal))  
AxisX = AxisXOriginal  
AxisY = AxisYOriginal  
AxisZ = AxisZOriginal
```

```

glViewport 0, 0, Picture1.Width, Picture1.Height
glMatrixMode GL_PROJECTION
glLoadIdentity
gluPerspective 65!, Picture1.Width / Picture1.Height, 1!, 500! 'Calcula el aspecto de la ventana

xAngle = 0  yAngle = 0  zAngle = 0

Teta1 = 0  Teta4 = 90  Teta10 = 270  Teta7 = 270  Teta13 = 0  Teta16 = 270  Teta18 = 90
Teta19 = -90  Teta20 = -90
Me.Show

```

El siguiente código es un módulo donde se encuentran las subrutinas llamadas primitivas que son los objetos base para los gráficos en 3D:

```

Public Sub Esfera(Radio As GLdouble, Gajos As GLint, Rebanadas As GLint, X As Double, Y As Double, Z
As Double, R As Double, G As Double, B As Double)
  Dim quadObj As GLUquadric
  glColor3f R, G, B
  quadObj = gluNewQuadric()
  gluQuadricDrawStyle quadObj, GLU_FILL
  gluQuadricNormals quadObj, GLU_SMOOTH
  glTranslated X, Y, Z
  gluSphere quadObj, Radio, Gajos, Rebanadas
  glTranslated -X, -Y, -Z
End Sub

Public Sub Cilindro(BasRad As GLdouble, TopRad As GLdouble, Alto As GLdouble, Gajos As GLint,
Rebanadas As GLint, X As Double, Y As Double, Z As Double, R As Double, G As Double, B As Double)
  Dim quadObj As GLUquadric
  glColor3f R, G, B
  quadObj = gluNewQuadric()
  gluQuadricDrawStyle quadObj, GLU_FILL
  gluQuadricNormals quadObj, GLU_SMOOTH
  glTranslated X, Y, Z
  gluCylinder quadObj, BasRad, TopRad, Alto, Gajos, Rebanadas
  glRotated 180, 0, 1, 0
  gluDisk quadObj, 0, BasRad, Gajos, 1
  glRotated 180, 0, 1, 0
  glTranslated 0, 0, Alto
  gluDisk quadObj, 0, TopRad, Gajos, 1
  glTranslated 0, 0, -Alto
  glTranslated -X, -Y, -Z
End Sub

Public Sub Cono(Base As GLdouble, Alto As GLdouble, Gajos As GLint, X As Double, Y As Double, Z As
Double, R As Double, G As Double, B As Double)
  glColor3f R, G, B
  glTranslated X, Y, Z
  glutSolidCone Base, Alto, Gajos, 1
  glTranslated -X, -Y, -Z
End Sub

```

```

Public Sub Cubo(Lado As GLdouble, X As Double, Y As Double, Z As Double, R As Double, G As Double,
B As Double)
    glColor3f R, G, B
    glTranslated X, Y, Z
    glutSolidCube Lado
    glTranslated -X, -Y, -Z
End Sub

```

```

Public Sub PrismaCuad(Ancho As Double, Alto As Double, Largo As Double, X As Double, Y As Double, Z
As Double, R As Double, G As Double, B As Double)
    glPushMatrix
    glEnable GL_NORMALIZE
    glColor3f R, G, B
    glTranslated X, Y, Z + Largo / 2
    glScaled Ancho, Alto, Largo
    glutSolidCube 1
    glDisable GL_NORMALIZE
    glPopMatrix
End Sub

```

Y por último el siguiente código llamado RENDER es donde se forma el robot y se posiciona según los ángulos obtenidos en el modelo de cinemática inversa:

```

Private Sub Render()
    glClear GL_COLOR_BUFFER_BIT Or GL_DEPTH_BUFFER_BIT Or GL_STENCIL_BUFFER_BIT
    glPushMatrix
    glCallList Cubo2
    glCallList EjesBase
    glRotated Teta1, 0, 0, 1 'Aquí se da el ángulo de rotación teta1
    glCallList Base
    glCallList Base2
    glTranslated 12.5, 0, 15
    glRotated Teta4, 1, 0, 0 'Aquí se da el ángulo de rotación teta4
    glCallList Brazo1 'BRAZO DE COLOR VERDE
    glRotated -Teta4, 1, 0, 0
    glTranslated -15, 0, 0
    glRotated Teta13, 1, 0, 0 'Aquí se da el ángulo de rotación teta13
    glCallList Brazo4 'BRAZO DE COLOR ROJO
    glRotated -Teta13, 1, 0, 0
    glTranslated 15, 0, 0
    glRotated Teta4, 1, 0, 0
    glTranslated -5, 0, 25
    glRotated Teta7, 1, 0, 0 'Aquí se da el ángulo de rotación teta7
    glCallList Brazo2 'BRAZO DE COLOR GRIS
    glTranslated -5, 0, 50
    glRotated Teta10, 1, 0, 0 'Aquí se da el ángulo de rotación teta10
    glCallList Brazo3 'ESTE ES EL BRAZO CAFE
    glTranslated -2.5, 0, 75
    glRotated -Teta18, 0, 0, 1 'Aquí se da el ángulo de rotación teta18
    glRotated -Teta19, 0, 1, 0 '0, 1, 0 'Aquí se da el ángulo de rotación teta19
    glRotated Teta20, 0, 0, 1 '0, 0, 1 'Aquí se da el ángulo de rotación teta20
    glCallList Gripper

```

```
glTranslated 0, 0, 10  
glCallList Ejes  
glPopMatrix  
SwapBuffers Picture1.hDC  
End Sub
```

4.5 Requerimientos de Hardware y Software

Los requerimientos de hardware y software para el funcionamiento del simulador son los siguientes:

- Sistema Operativo Windows XP de 32 bits
- Procesador Pentium IV o superiores
- Memoria Ram de 512 MB como mínimo
- Disco Duro al menos 50MB
- Referencia a VB OpenGL API 1.2
- Librerías glu32.dll y glut32.dll

CAPÍTULO 5 DESARROLLO DE UN PROTOTIPO

En este capítulo se presenta el desarrollo de un prototipo del robot motivo de estudio en este trabajo de tesis. A continuación se presenta toda la información necesaria para el desarrollo del prototipo [9].

5.1 Descripción del prototipo

Para poder probar la representatividad del modelo matemático y del simulador desarrollado en el capítulo anterior, fue necesario construir un prototipo del robot en estudio. Dicho prototipo se muestra en la figura 5.1

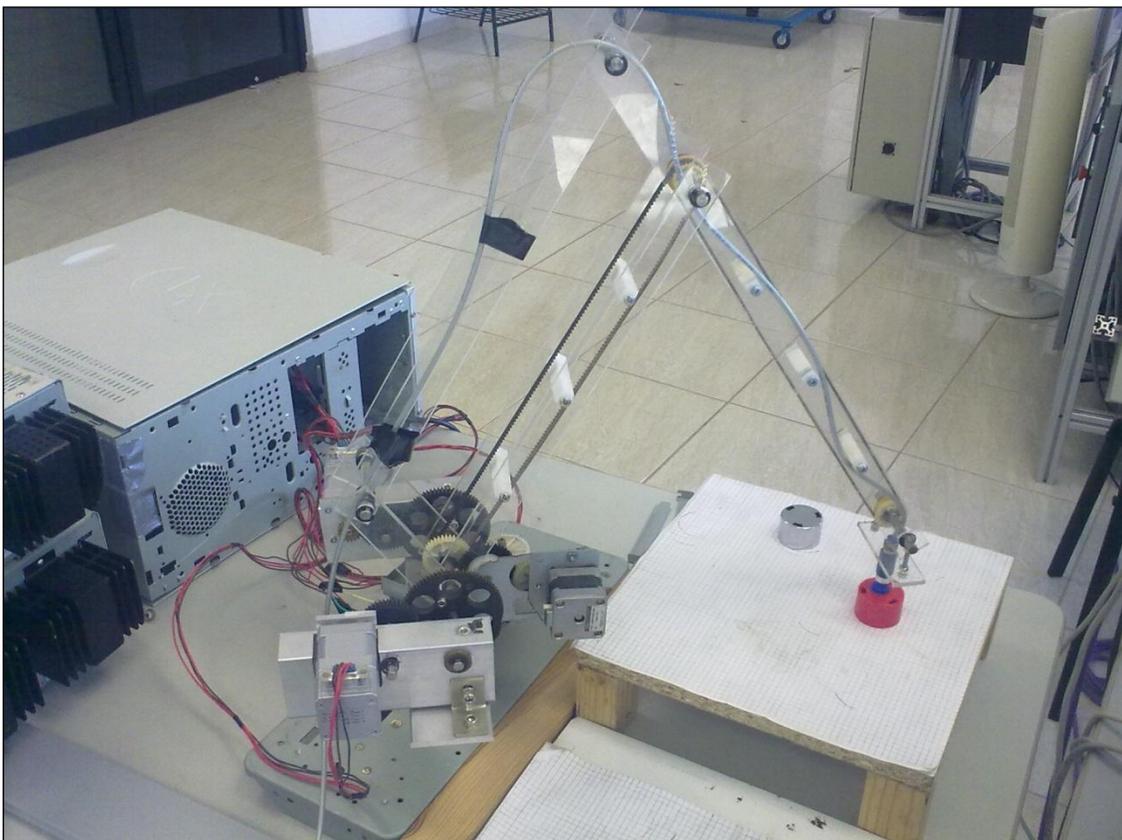


Figura 5.1. Prototipo desarrollado

El robot mostrado en la figura anterior, consta de 5 eslabones y su configuración es tipo paralela. Consta de:

- 1) Eslabones contruidos de material acrílico.
- 2) Fueron usados cuatro motores paso a paso, tanto unipolar como bipolar.
- 3) Se usaron trenes de engranes rectos para las transmisiones y bandas dentadas.
- 4) Sobre el extremo del eslabón final fue colocada un dispositivo de succión llamado ventosa, el cual es usado para mover objetos de una posición a otra.

Algunas consideraciones sobre las dimensiones, los actuadores y las trasmisiones del robot se resumen a continuación:

Para la construcción física del mecanismo en estudio se utilizaron materiales reciclados de copadoras y alimentadores de hojas para copadoras, las dimensiones originales del manipulador se tuvieron que escalar de 1 a 0.714 debido al sistema de transmisión por bandas para el grado de libertad de la muñeca quedando con las medidas mostradas en la figura 5.2.

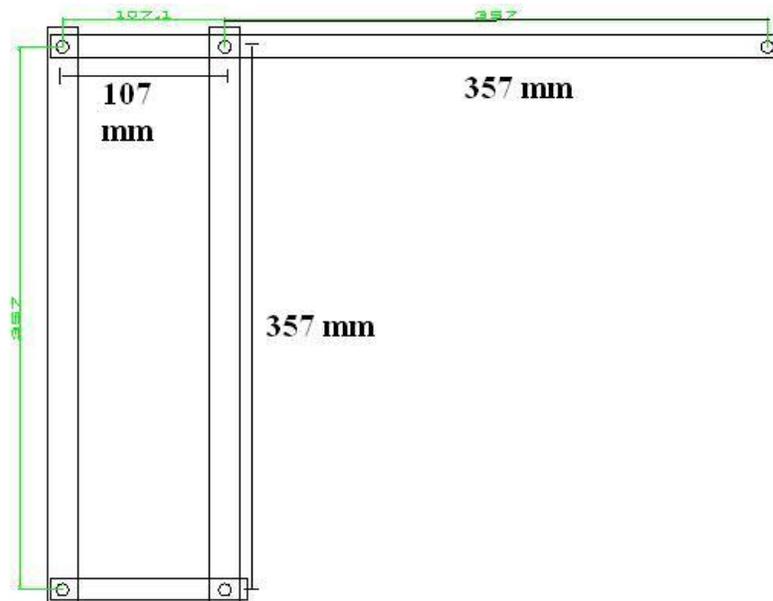


Figura 5.2. Dimensiones principales del prototipo

Por otro lado, para obtener los movimientos del hombro y el codo se usaron engranes y reductores extraídos de las copadoras. Una transmisión reductora en desplazamiento incrementa el torque y disminuye la velocidad y el desplazamiento con las siguientes características: engrane A de 19 dientes, engrane B de 42 dientes, engrane C de 21 dientes y

engrane D de 75 dientes dando como relación de transmisión 1 : 7.89, esto es (ver figura 5.3):

| RELACION PARA TORQUE | RELACION PARA DESPLAZAMIENTO Y VELOCIDAD |
|--------------------------------------|--|
| $RT = (B/A) \times (D/C)$ | $RT = (A/B) \times (C/D)$ |
| $RT = (42/19) \times (75/21) = 7.89$ | $RT = (19/42) \times (21/75) = 0.1266$ |

Esto significa que si el motor tiene un desplazamiento angular de 1.8 grados, a la salida de la transmisión se tendrá un desplazamiento de 0.228 grados por que $1.8 \times 0.1266 = 0.228$

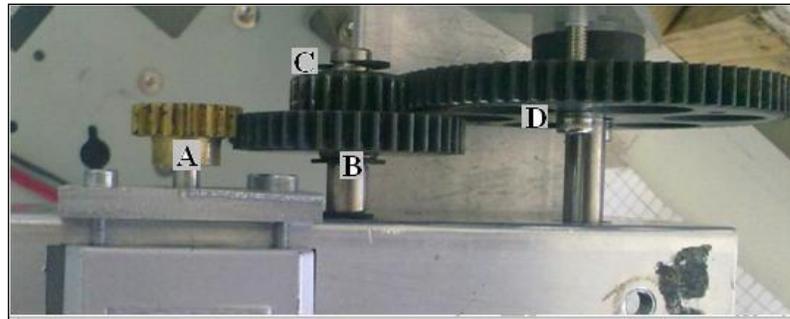


Figura 5.3. Tren de engranes

Para el movimiento de la base se utilizó una transmisión con relación de torque 1 : 9.09 y de desplazamiento de 0.11 en la cual por cada paso que da el motor de 1.8 grados a la salida se tiene un desplazamiento de 0.198 grados. Estos datos se ingresaron en la programación del simulador para controlar el desplazamiento de cada grado de libertad. Por otro lado, para fabricar los eslabones y las bases de las transmisiones se utilizó un centro de maquinado robodrill fanuc 3i, con el cual se le dio una precisión de fabricación de +/- 0.1 mm.

Cabe mencionar que con la construcción del prototipo, se pueden visualizar y comprender los siguientes aspectos:

- 1) Se pueden visualizar las interrelaciones entre la mecánica, la electrónica y la computación.
- 2) Se puede evaluar la funcionalidad y eficiencia del simulador.
- 3) Se pueden dar datos reales y con ello probar el simulador.
- 4) Se puede ver la complejidad de la parte mecánica.
- 5) Se diseña el sistema de control para los actuadores.

Para el caso de los movimientos del manipulador se utilizaron motores paso a paso unipolar y bipolar, este tipo de motores, dependiendo de sus características físicas son usados para movimientos de precisión donde se puede controlar el sentido de giro, el desplazamiento

angular, velocidad y el “torque”. Los motores utilizados en este robot tienen un paso de 1.8 grados, esto significa que para que el eje del motor de un giro de 360 grados el motor tiene que moverse 200 pasos pues:

$$\frac{360 \text{ grados}}{1.8 \text{ grados/paso}} = 200 \text{ pasos}$$

La figura 5.4 muestra uno de los motores a pasos utilizado para mover el robot. Se usaron transmisiones de engranajes rectos con la finalidad de dar mayor potencia al mecanismo y sobre todo para hacer más ilustrativo el uso de sistemas mecánicos, como los engranes, en el desarrollo de prototipos de robot.

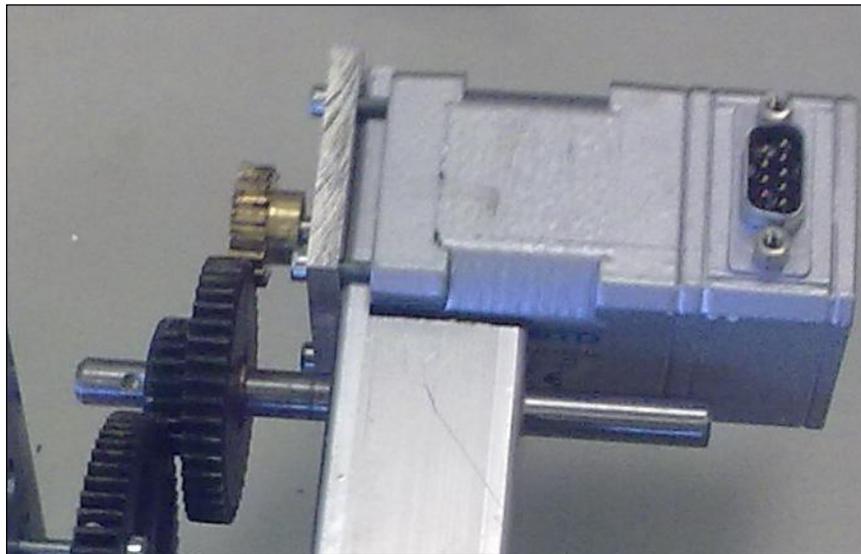


Figura 5.4. Motor paso a paso y sistema de engranaje

5.2 Aspectos importantes sobre el control del prototipo

Para el control de los motores paso a paso se elaboraron drivers en base a las especificaciones de las hojas de datos de los integrados L297 y L298N mostrados en la figura 5.5. Con el integrado L297 se puede controlar la corriente con la que trabajará el motor (pin 15), habilitar o deshabilitar el motor (pin 10), hacer que gire un paso o medios pasos (pin 19), hacer que gire en sentido horario o antihorario (pin 17). Una vez que se seleccionó el modo de operación se le envía un tren de pulsos en el pin 18 y por cada flanco de subida el motor dará un paso o medio paso según se haya configurado, la velocidad de giro está directamente relacionada con la frecuencia de los pulsos. El integrado L298N es el encargado de energizar los devanados del motor en base a la secuencia que este enviando el circuito de control L297. También se encarga de la potencia del motor ya que puede manejar hasta 2 amp. de corriente. La figura 5.6 muestra el circuito físico.

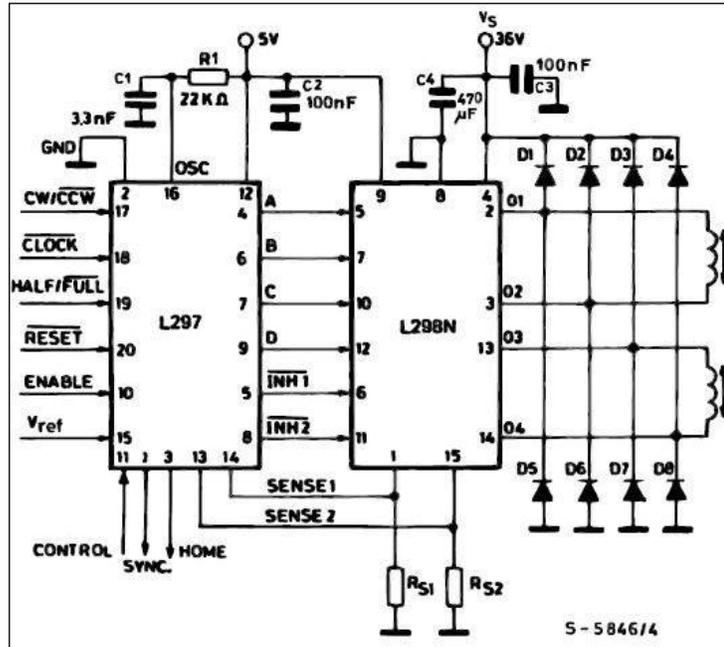


Figura 5.5. Circuitos integrados

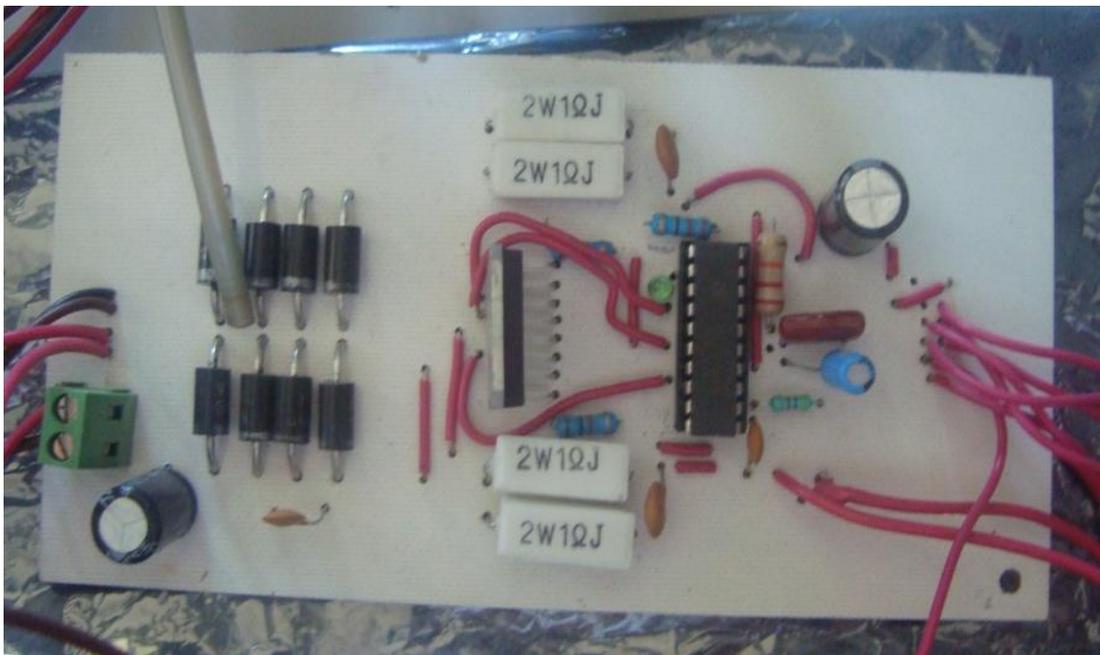


Figura 5.6. Circuito impreso.

Finalmente, la figura 5.7 muestra el algoritmo de control del prototipo.

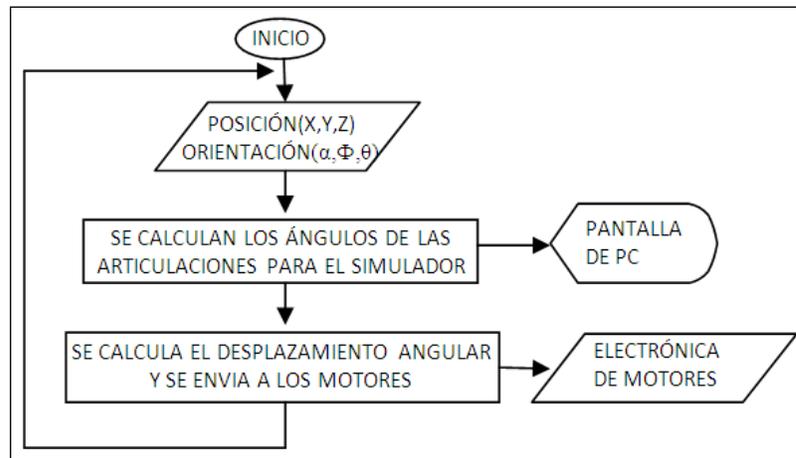


Figura 5.7. Algoritmo de control

5.3 Aspectos importantes sobre la funcionalidad del robot

En esta sección se presentan algunos aspectos importantes relacionados con el funcionamiento del robot.

Este prototipo fue desarrollado para probar el modelo cinemático de posición (implementando solo 4 GDL, tres de posición y uno de orientación) en el cual a través de un punto en el espacio y una orientación del órgano efector final (gripper) se obtiene la cinemática inversa del manipulador. El proceso es el siguiente: en el simulador en visual basic ingresamos la posición y orientación espacial, este simulador contiene implementadas las ecuaciones de cinemática inversa que se obtuvieron en el capítulo 3.2 arrojándonos como resultado los ángulos de posición teta 1, teta 13 y teta 16 y de orientación para este prototipo el ángulo teta 19, aquí mismo en base a los datos de transmisión se calcula el desplazamiento angular de cada GDL y la cantidad de pasos que hay que enviarle a cada motor, una vez obtenidos estos cálculos se envía el resultado al simulador y también se envía un tren de pulsos por el puerto paralelo para los controladores del robot y este se posiciona en el punto seleccionado. En la figura 5.8 el simulador está en la posición (0,50,55) y el prototipo en la figura 5.9 se encuentra en la misma posición.

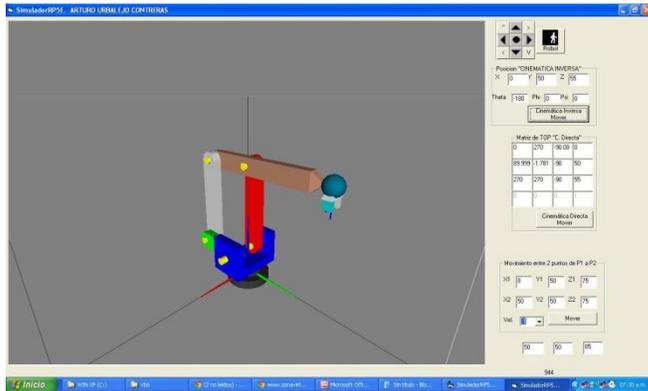


Figura 5.8 Simulador en (0,50,55)



Figura 5.9 Prototipo en (0,50,55)

En la figura 5.10 el simulador se encuentra en la posición (-20,60,25) y el prototipo en la figura 5.11 se encuentra en la misma posición.

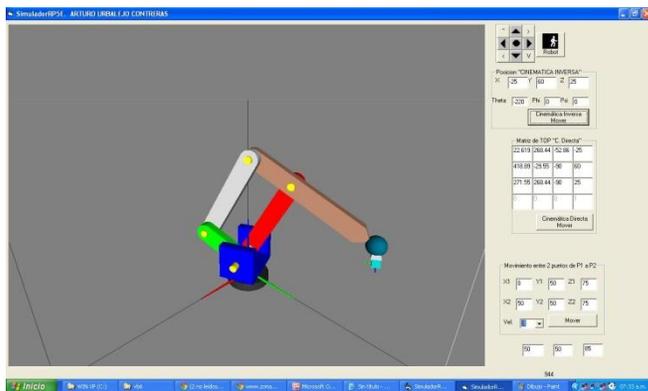


Figura 5.10 Simulador en (-20,60,25)

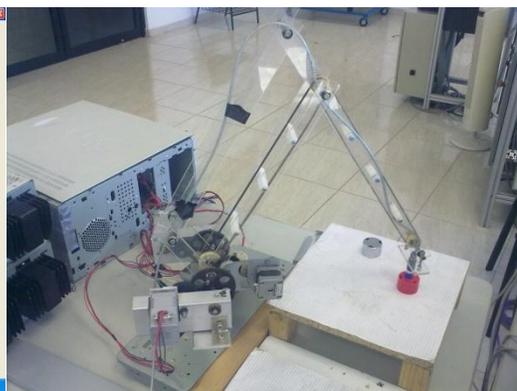


Figura 5.11 Prototipo en (-20,60,25)

Al observar las imágenes anteriores se puede inferir que el prototipo si es representativo del modelo matemático de posición obtenido en esta investigación.

Finalmente, al no utilizar un sensor de desplazamiento angular en cada grado de libertad el sistema utilizado en este proyecto, entonces el control es de lazo abierto, si por alguna razón el mecanismo de desplazamiento se traba, el sistema de control no tiene manera de verificar que el prototipo llegue a la posición solicitada.

CONCLUSIONES

A través de este trabajo de tesis se cumplieron los objetivos siguientes:

- 1) Se modeló la cinemática de posición de un robot paralelo espacial.
- 2) El modelo generado fue programado en el paquete Visual Basic V6.
- 3) Fueron simulados los movimientos del robot.
- 4) Se validó el modelo en un prototipo.

Los resultados y conclusiones se resumen en los puntos siguientes:

- En el desarrollo de este proyecto se utilizaron ecuaciones de lazo matricial para relacionar los ángulos de cada una de las articulaciones del manipulador, construyendo el modelo que representó la cinemática del robot.
- Utilizando el método de las matrices homogéneas se determinaron las ecuaciones de cada uno de estos ángulos tomando en cuenta las dos cadenas cinemáticas, esto fue de gran experiencia para entender como es que los robots industriales pueden posicionarse en el espacio y también para entender errores como el de singularidad.
- En la programación del modelo virtual se utilizó el software visual Basic y se usó una librería para manejar objetos en tercera dimensión llamada OpenGL. Esta plataforma de programación resultó ser muy útil para la visualización del robot y la simulación.
- El simulador desarrollado puede ser utilizado en clases de robótica para comprobar modelos matemáticos sin importar el método que se utilice para el cálculo de los ángulos, solo hay que hacer algunos ajustes en base a la configuración del robot.
- Debido a que no tiene restricciones de movimiento, ninguno de los ángulos del simulador puede tomar posiciones en las cuales se puede apreciar que el manipulador se ataca a sí mismo.
- Es necesario desarrollar prototipos en donde los alumnos puedan poner en práctica el conocimiento teórico de la mecatrónica. El desarrollo del robot paralelo permitió conocer las interrelaciones entre la mecánica, la electrónica y la computación, así como el uso de las matemáticas.
- Gracias a este proyecto, se desarrollaron 3 artículos en congresos nacionales [7] [8] [9], donde se habla del modelo matemático, el simulador y el prototipo elaborado. Además, se elaboró un artículo internacional [10] de servomotores para mejoras a este mismo proyecto.

RECOMENDACIONES

- Algunas de las mejoras que se propone para trabajos futuros a este simulador son que se delimite el área de trabajo del simulador restringiendo cada una de las articulaciones en base a lo que físicamente el manipulador puede alcanzar.
- Otra mejora es colocar un sistema de ejes coordenados en el punto central de la herramienta para que se ubique claramente la orientación ya que en esta versión es complicado visualizarlo, también se propone que el simulador pueda representar movimientos entre dos puntos en el espacio y para esto requeriría hacerse un análisis dinámico del manipulador.
- Elaborar un segundo prototipo utilizando servomotores para tener un sistema de lazo cerrado y asegurar los desplazamientos.
- Por último cabe agregar que en la universidad en la que actualmente trabajo existe la posibilidad de darle seguimiento a este proyecto y construir físicamente el manipulador.

BIBLIOGRAFÍA

LIBROS

1. SPONG Mark W. “Robot Dynamics and Control”, 1989
2. CRAIG John j. “Introduction to Robotics Mechanics & Control, Addison-wesley, 1986.
3. Merlet Jean-Pierre, “Les Robots Paralleles”. Hermes Paris, 1990
4. BARRIENTOS Antonio, “Fundamentos de Robótica”, Ed. McGraw Hill, 1997
5. Jacobo Peña Javier. “Desarrollo de un programa de simulación de un modelo cinemático de un robot articulado de 6 grados de libertad CRS A465 del ITESCA”. Tesis de Maestría en Ingeniería Mecánica. Universidad Nacional Autónoma de México. México D.F. 2006.

ARTÍCULOS EN INTERNET

6. Ilian Bonev , “The True Origins of Parallel Robots”. <http://www.ParallelMic.org>

ARTÍCULOS EN CONGRESOS

7. Urbalejo, A, Jiménez, F, Jiménez, E, Jacobo, J, Castro, J, *Modelación y simulación de un robot paralelo de 5 barras y 6 GLD usando matrices homogéneas*. 9º Congreso Nacional de Mecatrónica, Octubre 13 al 15, 2010. Puebla, Puebla.
8. Urbalejo A., Jacobo J., Jiménez E., Quintero A., López R., Castro J. *Simulación, control y construcción de robot paralelo de 5 barras y 6 gdl para propósitos didácticos*. 10º Congreso Nacional de Mecatrónica. Noviembre 3 y 4, 2011. Puerto Vallarta, Jalisco.
9. Urbalejo, A, Jiménez, E, Quintero A., *Desarrollo de un prototipo de robot paralelo didáctico: modelación, simulación, construcción y control*. Congreso Nacional de Tecnología Industrial 2012. 26 y 27 de abril Mérida Yucatán.
10. Jacobo J., Urbalejo A., Jiménez E., Guerrero J., López J., Luna A. *Desarrollo de una interface hombre-máquina con LABVIEW para controlar un servomotor industrial*. XII *International Conference on Engineering and Technology Education*. March 11 -14. Dili, East Timor. (2012).