



Universidad Autónoma de Querétaro
Facultad de informática
Maestría en Ingeniería de Software Distribuido

Modelación y simulación de la planeación de trayectorias para un manipulador flexible
con actuador neumático de dos grados de libertad

Tesis

Que como parte de los requisitos para obtener el grado de
Maestro en Ingeniería de Software Distribuido

Presenta

María Laura Gutiérrez Martínez

Dirigido por:

Dr. Juan Manuel Ramos Arreguín

Santiago de Querétaro, Qro; febrero de 2012



Universidad Autónoma de Querétaro
 Facultad de informática
 Especialidad Maestría en ingeniería de software

MODELACIÓN Y SIMULACIÓN DE LA PLANEACIÓN DE TRAYECTORIAS PARA UN
 MANIPULADOR FLEXIBLE CON ACTUADOR NEUMÁTICO DE DOS GRADOS DE LIBERTAD

TESIS

Que como parte de los requisitos para obtener el grado de
 Maestro en ingeniería de software Distribuido

Presenta:

María Laura Gutiérrez Martínez

Dirigido por:

Dr. Juan Manuel Ramos Arreguín

SINODALES

Dr. Juan Manuel Ramos Arreguín
 Presidente

Dr. José Emilio Vargas Soto
 Secretario

Dr. Jesús Carlos Pedraza Ortega
 Vocal

Dra. Sandra Luz Canchola Magdaleno
 Suplente

Dr. Efrén Gorrostieta Hurtado
 Suplente

M.C. Ruth Angélica Rico Hernández
 Nombre y Firma
 Director de la Facultad

Firma

Firma

Firma

Firma

Firma

Dr. Irineo Torres Pacheco
 Nombre y Firma
 Director de Investigación y Posgrado

Centro Universitario
 Querétaro, Qro.
 Febrero del 2012
 México

RESUMEN

En este trabajo se presenta la modelación y simulación gráfica de la planeación de trayectorias utilizando la metodología y la interfaz multilenguaje propuesta por el Dr. Juan Manuel Ramos Arreguín para poder obtener la generación de la planeación de trayectorias para un manipulador flexible con actuador neumático de dos grados de libertad.

Este proyecto es la continuación de un trabajo y una aportación importante es la planeación de trayectoria del robot, dado que lo desarrollado hasta el momento, solamente contempla la respuesta al primer escalón.

La generación de trayectoria es uno de los principales problemas en el área de la robótica. Partiendo de un análisis cinemático del manipulador y de la planeación de trayectorias, conseguir el desarrollo de una tarea comprende la definición de los puntos que habrán de conformar dicha trayectoria como generar la línea, círculo; utilizando las herramientas de software con C++, MatLab y OpenGL para lograr la simulación en 2-D que son aproximados a los que el robot realice en el mundo real. El modelado del robot requiere desarrollar la cinemática del brazo con dos grados de libertad. Pero existen dos problemas fundamentales: la cinemática directa y la cinemática inversa para poder llegar a realizar la trayectoria que se desea.

Se usó la computadora para demostrar la simulación del desarrollo de la planeación de trayectoria, cuyo objetivo principal es que el manipulador se mueva de un punto inicial a un punto final de una manera suave y controlada. Para ello sus articulaciones deben moverse coordinadamente, siendo de suma importancia calcular las ecuaciones. Cabe mencionar que se ignoran las propiedades dinámicas del robot.

(Palabras clave: cinemática directa, cinemática inversa, dinámica, modelación, simulación, robot, manipulador, actuador neumático).

SUMMARY

This work presents modeling and graphical simulation of the planning of trajectories using the methodology and the Multilingual interface proposed by Dr. Juan Manuel Ramos Arreguín to qualify for the generation of the planning of trajectories for a flexible manipulator with two degrees of freedom pneumatic actuator.

This project is the sequence of a work and an important contribution to the planning of the robot trajectory, as developed so far, only provides the answer to the first step.

The generation of trajectory is one of the main problems in the area of robotics. On the basis of an analysis of the manipulator and the planning of cinematic trajectories, ensure the development of a task includes the definition of the points that will shape the path how to generate the line, circle; using C++, MatLab, and OpenGL software tools for simulation in 2-D are approximate to the robot perform and in the real world. The modeling of the robot requires developing the kinematics of the arm with two degrees of freedom. But there are two fundamental problems: the direct kinematics and the inverse kinematics to make the path that you want to.

The computer was used to demonstrate the simulation of the development of the planning of trajectory, whose main objective is that the handler moves from a starting point to an end point in a smooth and controlled manner. So their joints must move in coordination, being of the utmost importance to calculate equations. It is worth mentioning that the dynamic properties of the robot are ignored.

(Key words: direct cinematics, inverse cinematics, dynamic, modelation, simulation, robot, manipulator, pneumatic actuator).

DEDICATORIA

A Dios por darme la oportunidad de concluir otra meta importante en mi vida.

A mis padres y hermanos por brindarme su colaboración y apoyo a lo largo de mis estudios y levantarme el ánimo en los momentos en que creía desfallecer.

A mi madre por su apoyo, dedicación y ánimo en los momentos difíciles de mi vida aunque ya no está conmigo siempre la voy a tener presente y dedicarle esta tesis que ya no lo vio culminado.

A mi esposo y a mis hijos por comprenderme y darme tiempo a seguir estudiando ya que ellos son la razón de mi vida, gracias a los cuales, he podido superar los momentos difíciles encontrados a lo largo de estos años de trabajo.

A mi amigo y compañero de estudio Guadalupe por su amistad y apoyo incondicional.

A mi director de tesis por compartir sus conocimientos.

A los maestros que me brindaron su enseñanza y guía a lo largo de mi carrera. Al Maestro David Cortés Arellano por brindarme su apoyo en la revisión de la traducción en inglés.

A todas las demás personas que de una u otra forma colaboraron desinteresadamente en la consecución de este sueño.

A TODOS ELLOS....GRACIAS

AGRADECIMIENTOS

Deseo expresar mi gratitud a todos aquellos que me apoyaron para realizar esta investigación, especialmente al Doctor Juan Manuel Ramos Arreguín que siempre me ha brindado su confianza para participar en el proyecto, dentro del cual se realizó este trabajo; además de que es mi director de tesis a quien debo mi interés por este proyecto y me permitió lograr los objetivos trazados, al Doctor Marco Antonio Aceves Fernández por su ayuda en orientarme y contactarme con el Doctor de tesis en este proyecto y darme su apoyo incondicional desde la División de Investigación y Posgrado de la Facultad de Informática; al Doctor Jesús Carlos Pedraza Ortega por motivarme y orientarme en OpenGL ya que desconocía esta herramienta y el me ayudo a motivarme para aprenderlo así como al Doctor Efrén Gorrostieta Hurtado por el apoyo y darme ánimos de continuar y terminarlo. Al Doctor José Emilio Vargas Soto y a la Doctora Sandra Luz Canchola Magdaleno por su paciencia para leer los borradores y sus acertadas sugerencias y a Guadalupe compañero de estudios y andanzas.

Mi agradecimiento al Doctor del proyecto, de quienes recibí amistad, así como múltiples y novedosos conocimientos e ideas.

Mi agradecimiento y amistad a la MC. Ruth Angélica Rico Hernández, Director de la facultad de Informática quien con su luz como mi maestra de carrera me enseñó que para lograr las metas y logros se tiene uno que esmerar hasta alcanzar lo que uno desea. A las personas responsables de la biblioteca y al Maestro David Cortés Arellano, mi gratitud por su amistad y apoyo.

A toda mi familia, especialmente a mi esposo y a mis hijos por su apoyo, y sobre todo por permitirme compartir con ellos su mundo y sus juegos.

Finalmente una disculpa a todos aquellos que de alguna manera colaboraron en este trabajo y que inevitablemente he omitido, apelo a su comprensión e indulgencia.

INDICE

	Página
Resumen	i
Summary	ii
Dedicatorias	iii
Agradecimientos	iv
Índice	v
Índice de figuras	vii
Índice de tablas	viii
I.INTRODUCCIÓN	1
1.1 Objetivo general	2
1.2 Objetivos específicos	2
1.3 Historia de la robótica	3
1.3.1 Definición de Robótica	6
1.3.2 Robot industrial	8
1.3.3 Clasificación de los robots	9
1.3.3.1 Robots manipuladores	10
1.4 Elementos de un robot Industrial	14
14.1 Tipo de actuadores	14
1.5 Manipuladores flexibles	18
1.5.1 Arquitectura del manipulador	19
1.5.2 Tipos de la junta	21
1.5.3 Características de los robots	22
1.6 Organización de la tesis	23
II. MODELADO DEL ROBOT NEUMATICO Y	24
GENERACION DE TRAYECTORIA	
2.1 Modelado del robot	24
2.1.1 . Cinemática directa	26
2.1.1.1 Primer mecanismo de cuatro barras	27
2.1.1.2 Segundo mecanismo de cuatro barras	32
2.1.2 . Cinemática inversa	38
2.1.2.1 Primer mecanismo de cuatro barras	42
2.1.2.2 Segundo mecanismo de cuatro barras	47
2.1.3 . Planeación de trayectoria	52

III. DESARROLLO DEL SOFTWARE	57
3.1 Metodología	57
3.2 Interfaz multilenguaje	60
3.2.1 Interfaz entre MatLab y C++	61
3.2.2 MatLab	62
3.2.3 C++	64
3.2.4 OpenGL	67
3.3 Estructura del robot	71
3.4 Programa y Librerías del software	73
IV. PRUEBAS Y RESULTADO	76
V. CONCLUSIONES	79
5.1 Trabajo futuro	80
BIBLIOGRAFIA	81
GLOSARIO	87
ANEXO A CODIGO DEL PROGRAMA	89

INDICE DE FIGURAS

1 Fotografía de Isaac Asimoc	4
2 Robot Industrial	9
3 Herramientas matemáticas	13
4 Robot en la industria su actividad es soldar	17
5 Los avances de los robot (IEEE,2009)	18
6 Escenario del futuro	18
7. Articulaciones y eslabones	19
8. Elementos de un brazo manipulador a) b)	20
9. Diagrama de bloques que indica las relaciones entre estos dos problemas	25
10. Dimensiones de los sistemas mecánicos.	25
11. Primer mecanismo corredera-manivela de cuatro barras, con dos cinemáticas cerradas	27
12. Cadenas cinemáticas cerradas del primer mecanismo	28
13. Segundo mecanismo corredera-manivela de cuatro barras, con dos cinemáticas cerradas	32
14. Cadenas cinemáticas cerradas del segundo mecanismo	33
15. Dimensiones de los elementos que se utilizaran para el cálculo inicial.	39
16. Triángulos rectángulos útiles para el cálculo inicial donde se obtiene θ_4 y θ_{14}	40
17. Primer mecanismo corredera- manivela de cuatro barras, con dos cadenas cinemáticas cerradas	42
18. Cadenas cinemáticas cerradas del primer mecanismo	43
19. Segundo mecanismo corredera-manivela de cuatro barras, con dos cadenas cinemáticas cerradas	47
20. Cadenas cinemáticas cerradas del segundo mecanismo	48
21. Tipos de trayectoria	54
22. Trayectoria de una línea recta	55
23. La metodología para la simulación grafica	58
24. El desarrollo del software	59
25. Proceso para obtener la animación grafica de la trayectoria	60
26. El desarrollo de interfaz multilenguaje	61
27. Interfaz entre MatLab y C++	62
28 Análisis para poder obtener los θ 's	63
29. El círculo	63
30. La línea	64
31. Muestra (a) la traslación, (b) rotación y (c) la vista del objeto	69
32. Diagrama de flujo	70
33. Manipulador flexible a) variables del sistema b) vista general	71
34. Objetos formados por primitivas OpenGL para el manipulador flexible	72
35. Robot manipulador flexible de dos grados de libertad	72
36. Posición inicial del manipulador	76
37. Trayectoria de un círculo	77
38. Trayectoria de una línea	77

INDICE DE TABLAS

1.1 Leyes de la robótica	5
1.2 Clasificación del robot industrial	6
1.3 Manipuladores	7
1.4 (a) Clasificación de los robots manipuladores	10
1.4 (b) Clasificación de los robots manipuladores	11
1.5 Clasificación por estructura mecánica de los robots	12
1.6 Fuentes de suministro de energía para el manipulador	15
1.7 Ventajas y desventajas actuadores eléctricos	16
1.8 Ventajas y desventajas de los neumáticos e hidráulicos	16
1.9 Ventajas y desventajas de la robótica industrial	17
2.10 Dimensiones del mecanismo corredera-manivela-4 barras	26
3.11 Variables del manipulador flexible	71
3.12 Opciones del menú	75

I. INTRODUCCIÓN

A lo largo de la historia, el hombre se ha sentido fascinado por máquinas y dispositivos capaces de imitar las funciones y los movimientos de los seres vivos [12]. Por esta razón, la inteligencia del hombre se ha destacado por crear incontables herramientas capaces de mejorar la forma de vida de las personas, ejemplo de ello son los llamados robots.

El campo de la robótica es interdisciplinario, ya que va desde el diseño de componentes mecánicos y electrónicos hasta tecnología de sensores, sistemas computacionales e inteligencia artificial [13].

La robótica ya no es parte de nuestro futuro sino de nuestro presente tangible [14].

La generación de trayectoria es importante porque nos permite establecer cuales son la trayectoria que debe seguir cada articulación del robot la posición y orientación del extremo del robot, para poder aprovechar mejor las ventajas que nos da un sistema neumático.

Para poder realizar la generación de trayectorias es necesario conocer las plataformas que se necesitan para poder realizar los cálculos matemáticos, realizar las gráficas y la animación. La interfaz multilenguaje nos permite aprovechar cada uno de los lenguajes que se van a utilizar como son MatLab, C++, OpenGL [44].

1.1 OBJETIVOS GENERALES

Desarrollar un software para la modelación y simulación de la planeación de trayectorias para un manipulador flexible con actuador neumático utilizando herramientas matemáticas y de modelación y simulación, para finalmente mostrar los resultados planteados y analizar la factibilidad de su uso.

1.2 OBJETIVOS ESPECIFICOS

- Analizar la planeación de trayectoria del manipulador, utilizando el modelo matemático.
- Obtener el modelo cinemático para el robot de 2 grados de libertad con actuador neumático.
- Implementar la simulación de la planeación de la trayectoria para un manipulador flexible del neumático de un robot brazo flexible con MatLab, C++ y OpenGL.
- Desarrollar diferentes herramientas tanto matemáticas como de software, para simular el comportamiento del robot flexible de 2 grados de libertad con accionamiento neumático.

1.3 HISTORIA DE LA ROBÓTICA

El término robótica procede de la palabra robot. “La robótica es, por lo tanto, la ciencia o rama de la ciencia que se ocupa del estudio, desarrollo y aplicaciones de los robots. Los robots son dispositivos compuestos de sensores que reciben datos de entrada y que pueden estar conectados a la computadora. Esta, al recibir la información de entrada, ordena al robot que efectúe una determinada acción. Puede ser que los propios robots dispongan de microprocesadores que reciben la señal de los sensores y que estos microprocesadores ordenen al robot la ejecución de las acciones para las cuales está concebido. En este último caso, el propio robot es a su vez una computadora” [15].

Karel Capek invento la palabra robot en los años 20’s. Capek era checoslovaco y escribió una obra de teatro llamada R.U.R. (Robots Universales de Rusia) [16][17].

Robot, esta palabra tiene su origen en el idioma checo con dos palabras: Robota y Robotnik, que significan “trabajo forzado” y “esclavo” respectivamente [18]. La palabra “Robot” aparece por primera vez en una obra satírica denominada: “The Rossum Universal Robots”, escrita por el dramaturgo checo Karel Capel, en 1921. En dicha obra los robots son unos personajes con apariencia humanoide que son fieles obedientes a las órdenes de su maestro.

El término robótica tiene su origen en la palabra robot. De esta forma, lo que hoy conocemos como Robótica se refiere al área del conocimiento asociado al estudio, desarrollo y aplicación de los robots.

Los egipcios hicieron figuras humanas en sus relojes hace aproximadamente 3400 años, las cuales eran impulsadas por agua para hacer sonar las campanas para dar la hora.

Más de mil años después El griego arquitas manufacturo una paloma de madera que movía sus alas.

Mil años después, el emperador Constantino VII, tenía un árbol con pájaros mecánicos, los cuales movían sus alas y cantaban.

En el siglo XIII, Alberto Magno invento un robot, algunas historias dicen que hablaba, otros que caminaba. El científico Ingles Roger Bacón, construyó un robot para que informara de los secretos del universo; producía humo, sus ojos brillaban e incluso movía la quijada.

En el siglo XVIII, James Watt inventó la máquina de vapor y pronto las fábricas estaban usando más sistemas mecánicos. Thomas Alva Edison descubrió varias formas de utilizar la electricidad.

En la década de los 30, las personas ya estaban listas para tomar en serio los robots, pues tenían muchas clases de máquinas [19]. En 1932, en Londres se mostró un robot llamado Alfa, podía leer, inclinarse, decir la hora, cantar y fumar puros.

En 1939 Nueva York en la feria Elektro, fue presentado otro que podía caminar, contar con los dedos y dar órdenes a su perro Sparky, que movía la cola y ladrar.

En 1948 William Gray Walter construyó dos tortugas robots, Elmer y Elesie se movían sin chocar con las cosas [20].

En 1950, Isaac Asimov presenta su famosa novela “Yo, Robot”, en donde define las tres leyes de la robótica [12] [21] [22] como se muestra en la siguiente tabla 1.1. En la figura 1 se muestra una foto suya.



Figura 1. Fotografía de Isaac Asimov.

Tabla 1.1 Leyes de la robótica.

Ley	Descripción
Primera	Un robot no puede lastimar a un ser humano o por su inacción dejar que un ser humano sea lastimado.
Segunda	Un robot debe obedecer las órdenes dadas por un ser humano excepto cuando estas órdenes estén en oposición con la primera ley.
Tercera	Un robot debe proteger su propia existencia siempre y cuando ésta protección no difiera con las dos primeras leyes.

Sin embargo ambos escritores estaban hablando de algo que no existía en aquel momento. Los primeros robots fueron construidos en los años 50 por Joseph F. Engelberger y George C. Devol, que se inspiraron en las obras de Asimov, consiguiendo solo un brazo robótico, conocido como manipulador, que ha sido una de las herramientas de la industria moderna más útiles, a tal grado que al referirse a un robot, en el siglo XX se hablaba de un manipulador.

En 1952 Claude Shannon construyó un robot ratón, el cual podía encontrar su camino en un laberinto [20].

Según la Robot Institute of América: Un robot es un manipulador reprogramable multifuncional, diseñado para mover materiales, piezas, o dispositivos especializados por medio de movimientos programados variables para la realización de una variedad de tareas [24].

En 1954 George Devol registró una patente llamada programmed article transfer (transferencia automática de archivos), que condujo al primer robot industrial, fabricado en 1962 por la Unimation, Inc. Fundada por Joseph Engleberger en 1958, el robot industrial ha generado una multitud de definiciones; de ellas la adoptada por el Robot Institut of América (instituto Norteamericano de Robots): “un robot industrial es un manipulador reprogramable con funciones múltiples diseñado para mover materiales, partes, herramientas o dispositivos especializados a través de movimientos programados para el desempeño de una gran diversidad de tareas, es la más común.

Palabras clave que distinguen a los robots de otras máquinas Manipulador y Reprogramable” [25].

1.3.1 DEFINICIÓN DE ROBÓTICA

Esta definición, ligeramente modificada, ha sido adoptada por la Organización Internacional de Estándares (ISO) que define al robot industrial como: Manipulador multifuncional reprogramable con varios grados de libertad, capaz de manipular materias, piezas, herramientas o dispositivos especiales según trayectorias variables programadas para realizar tareas diversas (Asociación de Industrias Robóticas: RIA).

Tabla 1.2 Clasificación del robot industrial

Nombre	Ejemplo
1ª Generación	Repite la tarea programada secuencialmente. No toma en cuenta las posibles alteraciones de su entorno.
2ª Generación	Adquiere información limitada de su entorno y actúa en consecuencia. Puede localizar, clasificar (visión) y detectar esfuerzos y adaptar sus movimientos en consecuencia.
3ª Generación	Su programación se realiza mediante el empleo de un lenguaje natural. Posee capacidad para la planificación automática de tareas.

Se incluye en esta definición la necesidad de que el robot tenga varios grados de libertad. Otra, más completa es la establecida por la Asociación Francesa de Normalización (AFNOR), que define primero el manipulador y, basándose en dicha definición, el robot:

Manipulador: mecanismo formado generalmente por elementos en serie, articulados entre sí, destinado al agarre y desplazamiento de objetos. Es

multifuncional y puede ser gobernado directamente por un operador humano o mediante dispositivo lógico.

En el caso de los robots manipuladores, podemos encontrar diversas aplicaciones tales como procesos industriales, traslado de objetos, procesos de automatización, procesos de medicina, etc. Es por ello que este tipo de robots son estudiados en diversas maneras [4]. En la tabla 1.3 se muestra la clasificación de los manipuladores que pueden ser tres clasificaciones.

Tabla 1.3 Manipuladores

Nombre	Ejemplo
De secuencia fija	Aquellos que efectúan cada etapa de una operación dada según un esquema de movimientos predeterminados, que no pueden cambiarse sin una modificación física.
Reprogramable	Aquellos en los que los movimientos programados o las funciones auxiliares pueden cambiarse sin modificación física.
Multi-aplicación	Aquellos que puede adaptarse a una aplicación diferente con modificación física.

1.3.2 ROBOT INDUSTRIAL

Manipulador *multiaplicación reprogramable* con varios grados de libertad, capaz de manipular materias, piezas, herramientas o dispositivos especiales según trayectorias variables programadas para realizar tareas diversas.

Robot: manipulador automático servo-controlado, reprogramable, polivalente, capaz de posicionar y orientar piezas, útiles o dispositivos especiales, siguiendo trayectoria variables reprogramables, para la ejecución de tareas variadas. Normalmente tiene la forma de uno o varios brazos terminados en una muñeca. Su unidad de control incluye un dispositivo de memoria y ocasionalmente de percepción del entorno. Normalmente su uso es el de realizar una tarea de manera cíclica, pudiéndose adaptar a otra sin cambios permanentes en su material.

Robot Institute of América: Un robot industrial es: “un manipulador multifuncional y reprogramable, diseñado para mover materiales piezas, herramientas o dispositivos especiales, mediante movimientos programables y variables que permitan llevar a cabo diversas tareas” [27].

Asociación japonesa de robótica Industrial (JIRA), los robots son dispositivos capaces de moverse de modo flexible, análogo al que poseen los organismos vivos, con o sin funciones intelectuales, lo que permite la realización de operaciones en respuesta a órdenes recibidas por humanos [14].

El robot industrial. “Nace de la unión de una estructura mecánica articulada y de un sistema electrónico de control en el que se integra una computadora. Que permite la programación y control de los movimientos del robot y la memorización de las diversas secuencias de trabajo” [26]. La figura 2 muestra un ejemplo de robot industrial.



Figura 2. Robot industrial.

La capacidad del robot industrial para reconfigurar su ciclo de trabajo, unida a la versatilidad y variedad de sus elementos terminales (pinzas, garras, herramientas, etc.), le permite adaptarse fácilmente a la evolución o cambio de los procesos de producción, facilitando su reconversión.

La definición más comúnmente aceptada posiblemente sea la de la Asociación de Industrias Robóticas (RIA), [16] [17] según la cual:

Un robot industrial es un manipulador multifuncional reprogramable, capaz de mover materias, piezas, herramientas, o dispositivos especiales, según trayectorias variables, programadas para realizar tareas diversas.

1.3.3 CLASIFICACION DE LOS ROBOTS

Es difícil establecer una clasificación de los robots. Ya que existen varias se revisan las diferentes clasificaciones de los robots basados en varios criterios como son: etapas de desarrollo, generaciones y propiedades. En su forma o aplicación en que pueden ser catalogados los manipuladores robóticos industriales se mencionan algunas de ellas:

1.3.3.1 ROBOTS MANIPULADORES

En la tabla 1.1 se describe la clasificación de los robots manipuladores, y se muestra un ejemplo de los mismos.

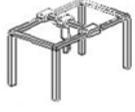
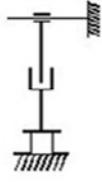
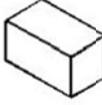
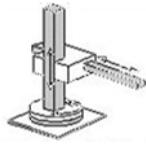
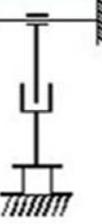
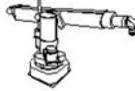
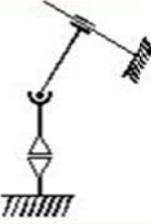
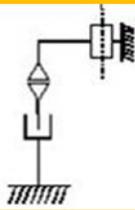
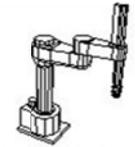
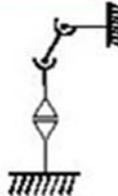
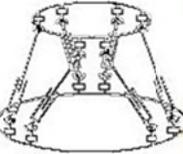
Tabla 1.4 (a) Clasificación de los robots manipuladores.

Nombre	Descripción	Ejemplo
Manipuladores	<p>Son sistemas mecánicos multifuncionales, con un sencillo sistema de control, que permite gobernar el movimiento de sus elementos, de los siguientes modos:</p> <p>Manual es cuando el operario controla directamente la tarea del manipulador.</p> <p>De secuencia fija: cuando se repite, de forma invariable, el proceso de trabajo preparado previamente.</p> <p>De secuencia variable: Se pueden alterar algunas características de los ciclos de trabajo.</p> <p>Existen muchas operaciones básicas que pueden ser realizadas óptimamente mediante manipuladores, por lo que se debe considerar seriamente el empleo de estos dispositivos, cuando las funciones de trabajo sean sencillas y repetitivas. En la ejemplo se muestra las partes de un manipulador.</p>	 <p style="text-align: center;">Brazo Manipulador</p>
Robots de repetición o aprendizaje	<p>Son manipuladores que se limitan a repetir una secuencia de movimientos, previamente ejecutada por un operador humano, haciendo uso de un controlador manual o un dispositivo auxiliar. En este tipo de robots, el operario en la fase de enseñanza, se vale de una pistola de programación con diversos pulsadores o teclas, o bien, de joysticks, o bien utiliza un maniquí, o a veces, desplaza directamente la mano del robot. Los robots de aprendizaje son los más conocidos, hoy día, en los ambientes industriales y el tipo de programación que incorporan, recibe el nombre de "gestual". Como se muestra en el ejemplo.</p>	 <p style="text-align: center;">Robot de repetición</p>

Tabla 1.4 (b) Clasificación de los robots manipuladores.

Nombre	Descripción	Ejemplo
Robots con control por computador	<p>Son manipuladores o sistemas mecánicos multifuncionales, controlados por un computador, que habitualmente suele ser un microordenador. En este tipo de robots, el programador no necesita mover realmente el elemento de la máquina, (Ver ejemplo) cuando la prepara para realizar un trabajo. El control por computador dispone de un lenguaje <i>específico</i>, compuesto por varias instrucciones adaptadas al robot, con las que se puede confeccionar un programa de aplicación utilizando solo el terminal del computador, no el brazo. A esta programación se le denomina textual y se crea sin la intervención del manipulador.</p> <p>Las grandes ventajas que ofrecen este tipo de robots, hacen que se vayan imponiendo en el mercado rápidamente, lo que exige la preparación urgente de personal cualificado, capaz de desarrollar programas similares a los de tipo informático.</p>	 <p>Robot controlado por la computadora</p>
Robots inteligentes	<p>Son similares a los del grupo anterior, pero, además, son capaces de relacionarse con el mundo que les rodea a través de sensores y tomar decisiones en tiempo real (auto programable).</p> <p>De momento, son muy poco conocidos en el mercado y se encuentran en fase experimental, en la que se esfuerzan los grupos investigadores por potenciarles y hacerles más efectivos, al mismo tiempo que más asequibles.</p> <p>La visión artificial, el sonido de máquina y la inteligencia artificial, son las ciencias que más están estudiando para su aplicación en los robots inteligentes. [27]</p>	 <p>Robot inteligente que está tocando una melodía.</p>
Micro-robots	<p>Con fines educacionales, de entretenimiento o investigación.</p>	 <p>Micro-robots con fines educacionales</p>

Tabla 1.5 Clasificación por estructura mecánica de los robots.

TIPO	DESCRIPCIÓN	PRINCIPIO	ESTRUCTURA CINEMÁTICA	ESPACIO DE TRABAJO	EJEMPLO
Robot cartesiano	Robot donde el brazo) está constituido por tres articulaciones prismáticas donde los ejes están dispuestos según un sistema de coordenadas cartesianas.				
Robot cilíndrico	Robot donde el brazo está constituido por al menos una articulación rotativa y de una articulación prismática donde los ejes están dispuestos según un sistema de coordenadas cilíndricas.				
Robot esférico (o polar)	Robot cuyos ejes forman un sistema de coordenadas polar.				
Robot SCARA(*)	Robot que tiene dos articulaciones rotativas con ejes paralelos, para proporcionar conformidad en un plano dado.				
Robot articulado	Robot cuyo brazo tiene al menos articulaciones rotativas				
Robot paralelo	Robot donde el brazo (ejes principales) están constituidos por tres articulaciones prismáticas concurrentes				

(*): SCARA viene de Selectively Compliant Arm for Robotic Assembly.

La manipulación es el acto de cambiar su posición y orientación en el espacio como se muestra en la siguiente figura 3 se necesitan herramientas matemáticas para poder lograr las trayectorias.

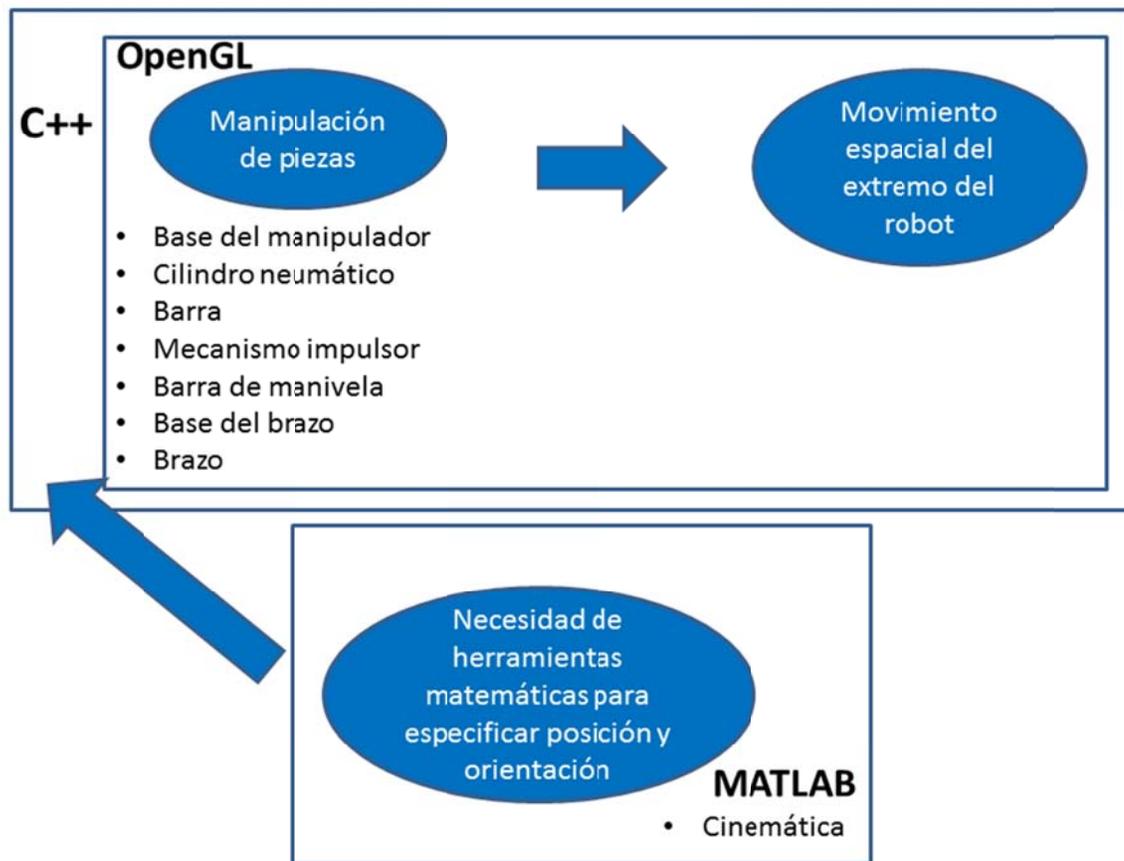


Figura 3. Herramientas Matemáticas

En el caso de los robots manipuladores, podemos encontrar diversas aplicaciones tales como procesos industriales, traslado de objetos, procesos de automatización, procesos de medicina, etc. Es por ello que este tipo de robots son estudiados en diversas maneras [1].

Un robot precisa de elementos motrices capaces de convertir señales de control en movimientos con una potencia de actuación acorde con las tareas que tenga encomendadas, los motores que crean movimientos en las articulaciones de los robots, se llaman actuadores.

1.4 ELEMENTOS DE UN ROBOT INDUSTRIAL

Los elementos que forman un robot son:

- Una computadora
- Dispositivos de medición
- Un generador de potencia eléctrica, hidráulica o neumática,
- Un manipulador,
- Posiblemente ruedas.

Los actuadores son las partes del robot que se encargan de transformar algún tipo inicial de energía en variaciones de energía cinética, es decir, en movimientos o paradas del robot [18].

1.4.1 TIPO DE ACTUADORES

Los elementos motrices que generan el movimiento de las articulaciones pueden ser, según la energía que consuman, de tipo hidráulico, neumático o eléctrico.

Los actuadores de tipo hidráulico se destinan a tareas que requieren una gran potencia y grandes capacidades de carga. Dado el tipo de energía que emplean, se construyen con mecánica de precisión y su coste es elevado. Los robots hidráulicos, se diseñan formando un conjunto compacto la central hidráulica, la cabina electrónica de control y el brazo del manipulador.

La energía neumática dota a sus actuadores de una gran velocidad de respuesta, junto a un bajo coste, pero su empleo está siendo sustituido por elementos eléctricos.

Los motores eléctricos, que cubren la gama de media y baja potencia, acaparan el campo de la Robótica, por su gran precisión en el control de su movimiento y las ventajas inherentes a la energía eléctrica que consumen.

Tabla 1.6 Fuentes de suministro de energía para el manipulador

Descripción	Ejemplo
<p>HIDRÁULICOS</p> <p>Son utilizados donde los sistemas robotizados requieren una alta capacidad de carga, como aplicaciones nucleares y bajo el agua. Trabajan con aceite especial y el sistema requiere de líneas de retorno del aceite a un recipiente especial donde una bomba utiliza el mismo aceite para enviar la presión al actuador. La compresibilidad de los líquidos usados es mínima, por lo que, desde el punto de vista de control, se convierte en un sistema lineal. Sin embargo, los mayores problemas en el uso de estos sistemas, es la contaminación del fluido dentro del sistema así como en el medio ambiente. [2]</p>	 <p>Robot hidráulico</p>
<p>ELÉCTRICOS</p> <p>Normalmente hablamos de motores, y la energía que requieren puede ser almacenada fácilmente, especialmente en el uso de baterías, sin requerir de bombas de presión, líneas de retorno, válvulas controladoras del sentido de movimiento. Los motores eléctricos se pueden adaptar más fácilmente al medio ambiente, y todo el sistema de control ocupa menos espacio que un sistema hidráulico ó neumático.</p>	 <p>Robot eléctrico</p>
<p>NEUMÁTICOS</p> <p>Utilizan gases compresibles como medio de transmisión de energía, y generalmente es el aire del medio ambiente. La presión en las cámaras es menor que en los sistemas hidráulicos, lo que resulta en una menor capacidad de fuerza. Los sistemas neumáticos son similares a los hidráulicos, pero cuando se usa neumática, no es necesario líneas de retorno de aire. Los sistemas neumáticos han sido usados en robots cuando se requiere que el robot sea ligero, pequeño, y con una relación de potencia-peso relativamente alta. La neumática es limpia y no es inflamable, el mantenimiento es sencillo y a menor costo. Debido a la compresibilidad del fluido, el sistema de control es diferente de un actuador hidráulico, derivando en un control no lineal.</p>	 <p>Robot neumático</p>

Eléctricos son la mayoría de los actuadores utilizados para el movimiento de los robots.

Tabla 1.7 Ventajas y Desventajas actuadores eléctricos

Ventajas	Desventajas
<ul style="list-style-type: none"> ❑ Por la facilidad de requerir solamente energía eléctrica. ❑ Se adaptan fácilmente al medio ambiente. ❑ Ocupan menos espacio. ❑ Son fáciles de instalar y de limpiar. <p>No generan tanto ruido</p>	<ul style="list-style-type: none"> ❑ Producir un torque demasiado pequeño, comparado con su tamaño, peso y consumo eléctrico.

Neumáticos e hidráulicos requieren un compresor de aire y una bomba hidráulica

Tabla 1.8 Ventajas y desventajas de los neumáticos e hidráulicos

Ventajas	Desventajas
<ul style="list-style-type: none"> ❑ Utiliza gases comprensibles como el medio de transmisión de energía. Y generalmente es el aire del medio ambiente. ❑ Cuando el robot es ligero, pequeño y con una relación potencia-peso. ❑ Es limpia ❑ No es inflamable ❑ El mantenimiento es sencillo ❑ Menor costo 	<ul style="list-style-type: none"> ❑ La presión en las cámaras es menor a los sistemas hidráulicos. ❑ La uniformidad del movimiento, perdiendo exactitud en el posicionamiento.

Beneficios y desventajas de la robótica en la industria

Tabla 1.9 Ventajas y Desventajas de la robótica industrial

Ventajas	Desventajas
<ul style="list-style-type: none">❑ La productividad aumenta.❑ Y la eficiencia.❑ Más barato al realizar trabajos peligrosos.❑ Reducir el desperdicio de materiales.❑ Reduce el costo de inversiones en equipo.❑ Disminuir la rotación de personal.❑ Reducción de costos de mano de obra.❑ Mejorar la calidad del producto	<ul style="list-style-type: none">❑ No actúan a situaciones imprevistas.❑ Realiza tareas repetitivas.❑ Requieren de la capacidad motora y perceptiva humana (planeación, decisiones y evaluación).

Debido a la gran importancia hoy en día la robótica va en auge y es un campo de gran interés en el proceso de modernización e innovación de la industria. El hombre posee una inteligencia y creatividad que gracias a ello transforma la naturaleza y su ingenio es tan asombroso que crea instrumentos ejemplo de ello son los robots que le pueden ayudar a realizar ciertas actividades que puede ser vital o destructora. Como se puede mostrar en la siguiente figura 4.



Figura 4. Robot en la industria su actividad es soldar.

En la figura 5 se muestra una gráfica que detalla el incremento de los robots en el mundo, y en la figura 6 se muestra un escenario del futuro de los robots [6].

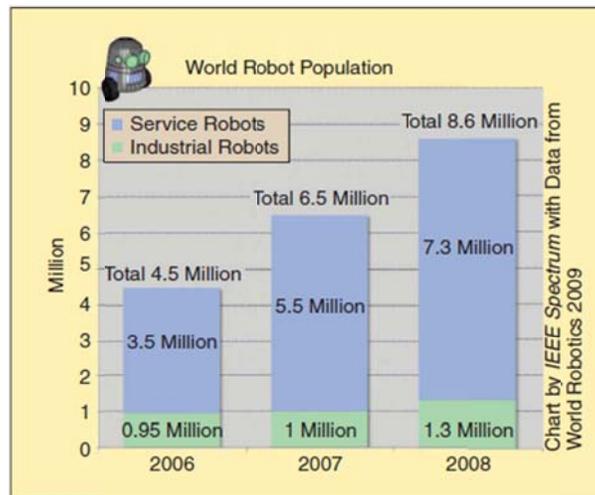


Figura 5. Avances de los robot (IEEE, 2009).



Figura 6. Escenario del futuro.

1.5 MANIPULADORES FLEXIBLES

La modelación de los manipuladores flexibles. Es muy complicada, algunos investigadores consideran como una primera etapa, un solo eslabón como flexible [5]. Sin embargo, el uso de computadoras potentes ha permitido considerar sistemas de varios eslabones flexibles [7], [8], [9], [10], [11]. La ventaja de los actuadores neumáticos es su bajo costo, limpieza de operación y una relación

potencia y peso por lo que permite construir manipuladores compactos, ligeros, sencillos y de bajo costo.

En la modelación y simulación se aplicaron las metodologías propuestas de los grandes investigadores y se basara en una interfaz multilenguaje donde está la más actual realizada por grandes investigadores [1], [2], [3], [4].

1.5.1 ARQUITECTURA DEL MANIPULADOR

En esta sección se trata de las características del brazo manipulador, el cual está compuesto por dos eslabones, que para fines prácticos se consideran rígidos, como se muestra en la figura 7. Un grado de libertad es un tipo de movimiento que puede hacer un robot, en términos generales, cada articulación aporta un grado de libertad. En algunos casos, se agrupan varios grados de libertad (GDL) en un mismo punto, lo cual se puede comprender como eslabones sucesivos de longitud cero.

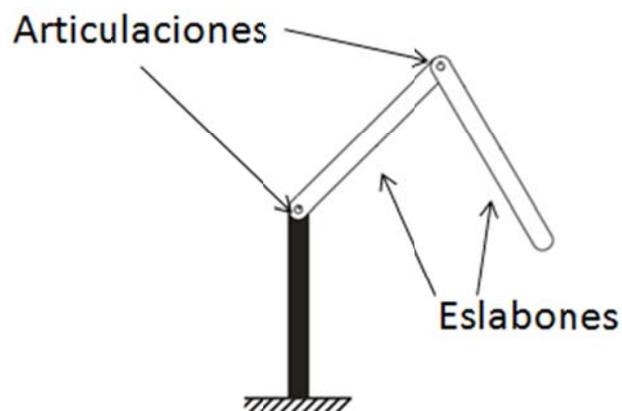


Figura 7. Articulaciones y eslabones

Los robots industriales pueden animarse con potencia neumática, eléctrica hidráulica. La potencia neumática está limitada en términos generales a robots que seleccionan y colocan artículos, en los que los actuadores se pueden mover

rápidamente hasta que son detenidos por fines de carrera mecánicos. Los sistemas de robot más complejos generalmente usan actuadores hidráulicos, estos son compactos y capaces de producir fuerzas o pares de gran magnitud. Aunque los actuadores eléctricos son aún más comunes están limitados al 30 % de la población existente en robots. Los robots animados por energía eléctrica son más precisos que los hidráulicos, solo que el sistema eléctrico requiere reductores en sus motores y esto hace más costosos el sistema. La figura 8 muestra los elementos principales de un robot manipulador industrial a), b).

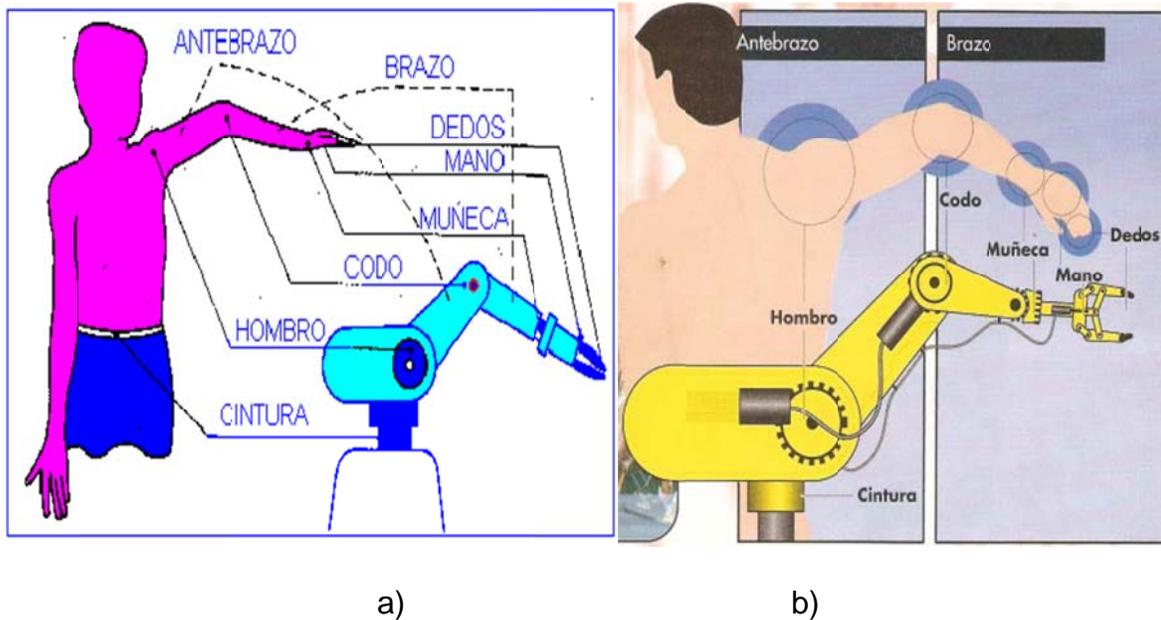
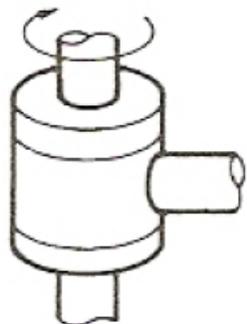


Figura 8. Elementos de un brazo manipulador a), b)

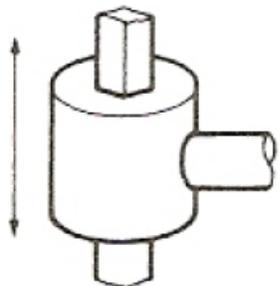
Los manipuladores son, esencialmente, brazos articulados de forma más precisa, un manipulador industrial convencional es una cadena cinemática abierta formada por un conjunto de eslabones o elementos de la cadena interrelacionados mediante articulaciones o pares cinemáticos.

1.5.2 TIPOS DE LA JUNTURA

- Rotación. Junturas rotatorias a menudo manejadas por los motores eléctricos y la cadena, el cinturón o las transmisiones del motor, o por los cilindros hidráulicos y palancas.

ESQUEMA	ARTICULACIÓN	GRADOS LIBERTAD
	Rotación	1

- Prismático: juntas del deslizador en que el eslabón se apoya en un deslizador llevar lineal, y linealmente actúa por los tornillos de la pelota y motores o cilindros.

ESQUEMA	ARTICULACION	GRADOS LIBERTAD
	Prismática	1

1.5.3 CARACTERÍSTICAS DE LOS ROBOTS DETERMINAN EL MODELO Y LAS APLICACIONES.

- Grado de libertad. Son los parámetros que se precisan para determinar la posición y la orientación del elemento terminal del manipulador. También se pueden definir los grados de libertad, como los posibles movimientos básicos (giratorios y de desplazamiento) independientes.
- Zonas de trabajo y dimensiones del manipulador. Las dimensiones de los elementos del manipulador, junto a los grados de libertad, definen la zona de trabajo del robot, característica fundamental en las fases de selección e implantación del modelo adecuado.
- Capacidad de carga. El peso en kilogramos que puede transportar la garra del manipulador recibe el nombre de capacidad de carga.
- Exactitud. La exactitud de punto es más importante al realizar fuera de la línea programando, porque se usan las coordenadas absolutas.
- Repetibilidad. La repetibilidad de punto es a menudo más pequeña que la exactitud.
- Resolución. Está basada en un número limitado de puntos que el robot puede alcanzar para estos se muestran aquí como los puntos negros. Estos puntos están típicamente separados por un milímetro o menor, dependiendo del tipo de robot.
- Coordenadas de los movimientos. La estructura del manipulador y la relación entre sus elementos proporcionan una configuración mecánica, que da origen al establecimiento de los parámetros que hay que conocer para definir la posición y orientación del elemento terminal.

Los actuadores son las partes del robot que se encargan de transformar algún tipo inicial de energía en variaciones de energía cinética, es decir, en movimientos o paradas del robot.

1.6 ORGANIZACIÓN DE LA TESIS

CAPÍTULO 1.- En este capítulo se muestran los antecedentes se dan el objetivo general y se listan los objetivos específicos que se pretenden lograr para el proyecto de la modelación y simulación de la planeación de trayectorias para un manipulador flexible con actuador neumático de dos grados de libertad, así como, la historia de los robots y el estudio de viabilidad, los alcances y limitaciones y la organización de la tesis.

CAPÍTULO 2.- En este capítulo se muestran el modelado de robot y el desarrollo de la planeación de trayectorias para un manipulador flexible con actuador neumático de dos grados de libertad se debe analizar la cinemática cuyo objetivo son las trayectorias que deben de seguir las articulaciones y para ello se menciona la cinemática inversa y directa. Se aplica el Método Newton-Raphson.

CAPÍTULO 3.- Este capítulo se muestra el desarrollo de software de la modelación y simulación de la planeación de trayectorias para un manipulador flexible con actuador neumático de dos grados de libertad donde se mencionan las herramientas para la realización del proyecto.

CAPÍTULO 4.- Pruebas y resultados.

CAPÍTULO 5.-Conclusiones

Bibliografía.

Glosario

Anexo A Código de la animación

2. MODELADO DEL ROBOT NEUMÁTICO Y GENERACION DE TRAYECTORIA

El modelado del manipulador, solo contempla la modelación cinemática, se analiza la cinemática directa e inversa del manipulador para lograr obtener las trayectorias deseadas que debe seguir cada articulación del robot.

2.1 MODELADO DEL ROBOT

La cinemática de un manipulador 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 coordenadas articulares [8].

Las ecuaciones cinemáticas son no lineales, su solución no siempre es fácil o posible incluso en una forma cerrada; las cuales se resolvieron utilizando el método de Newton-Raphson.

La cinemática se interesa por la descripción analítica del desplazamiento espacial del robot como una función del tiempo, en particular de las relaciones entre la posición de las variables de articulación y la posición y orientación del efector final del brazo del robot. [12]

Existen dos problemas fundamentales en la cinemática del robot que son: cinemática directa e indirecta se muestra en la figura 9 la relación.

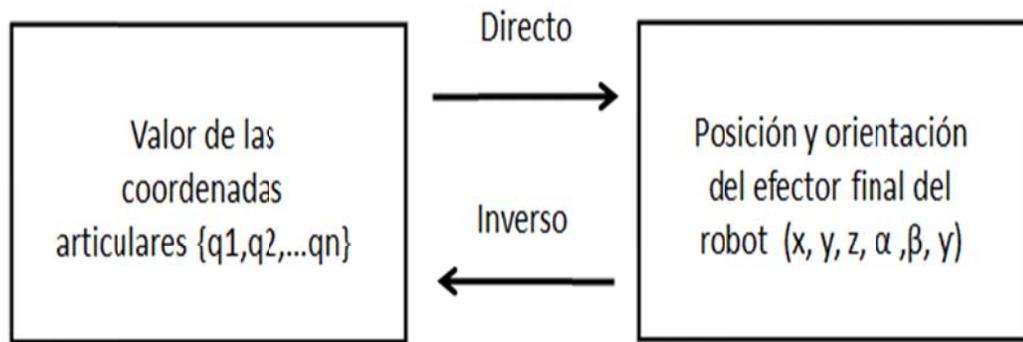


Figura 9. Diagrama de bloques que indica las relaciones entre estos dos problemas.

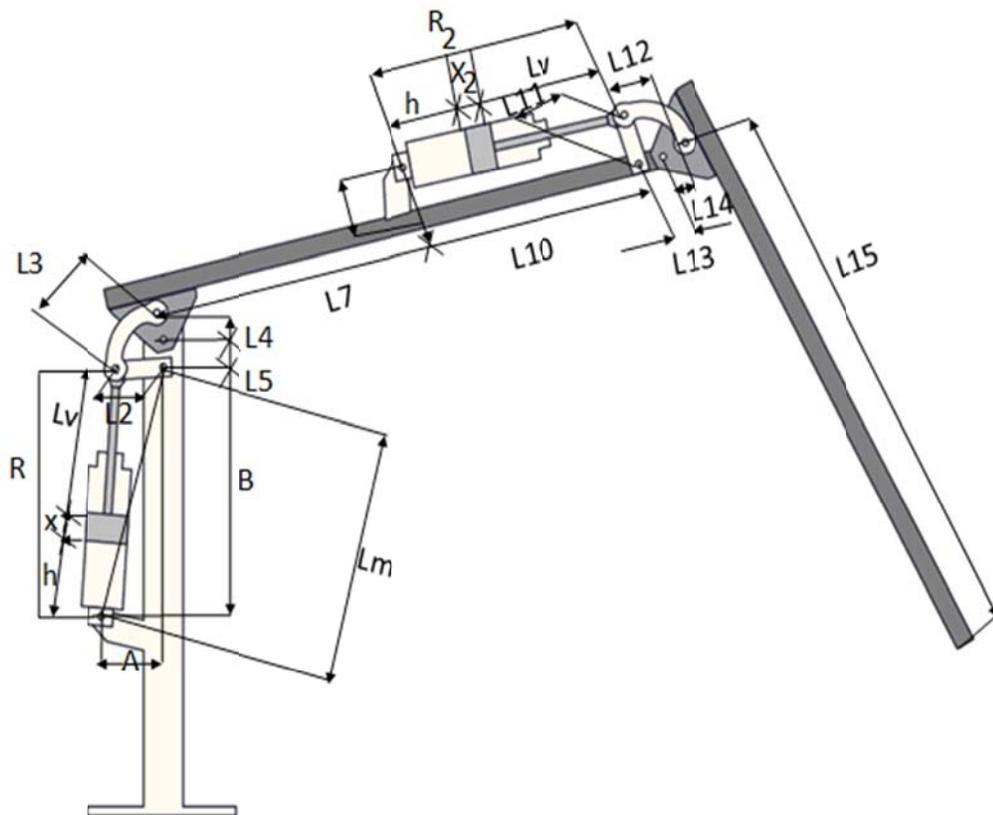


Figura 10. Dimensiones de los sistemas mecánicos; longitud de los eslabones. Las dimensiones de los dos mecanismos son iguales.

Tabla 2.10: Dimensiones del mecanismo corredera – manivela – 4 barras.

Desnivel entre el cilindro y la manivela	A	0.108 m
Distancia entre pivotes cilindro - manivela	B, L10	0.450 m
Distancia entre pivotes manivela - seguidora	L5, L13	0.041 m
Distancia del pivote al fondo del cilindro	H	0.042 m
Longitud ensamble pistón - vástago	Pv	0.357 m
Desplazamiento del vástago del pistón del primer mecanismo	x	
Desplazamiento del vástago del pistón del segundo mecanismo	x_2	
Longitud de la manivela	L2,L11	0.074 m
Longitud de la barra acopladora	L3, L12	0.099 m
Longitud de la barra seguidora	L4, L14	0.043 m
Desnivel entre el cilindro y la manivela	L8	0.078 m
Longitud de la barra que une a los dos mecanismos	L7	0.966 m
Longitud de la barra final	L15	1.500 m
Distancia entre pivotes del cilindro y la manivela	Lm	0.463

2.1.1 CINEMÁTICA DIRECTA

Determinar la posición y orientación del efector final del robot, con respecto a un sistema de coordenadas de referencia, conocidos los ángulos de las articulaciones y los parámetros geométricos de los elementos del robot.

El problema cinemático directo se resolverá el primer mecanismo de cuatro barras y posteriormente el segundo.

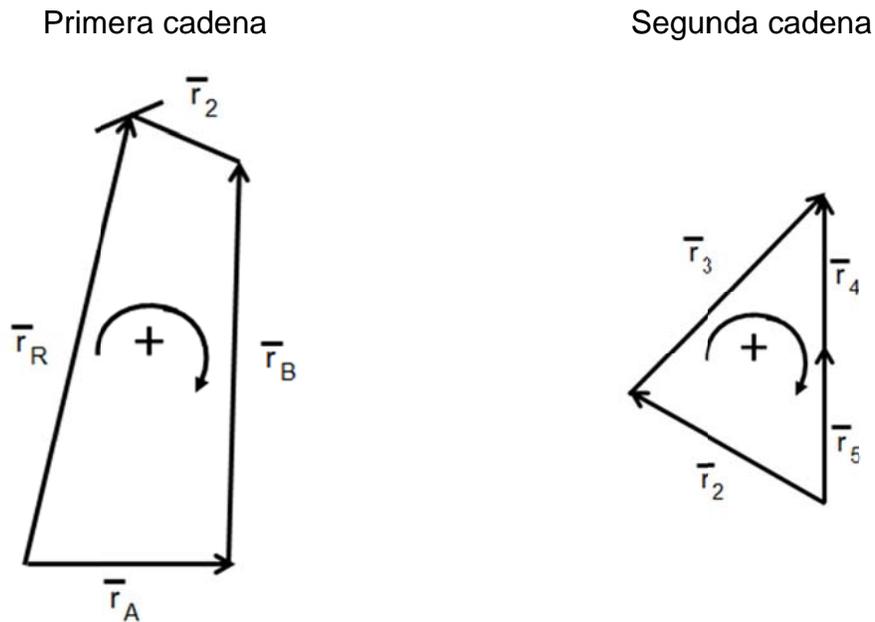


Figura 12. Cadenas cinemáticas cerradas del primer mecanismo.

De la primera cadena cinemática se obtenemos la ecuación vectorial (ver figura 12):

$$\vec{r}_R + \vec{r}_2 - \vec{r}_B - \vec{r}_A = 0 \quad (2.30)$$

Se sustituyen las componentes en i y j de cada vector en la ecuación (2.30), como resultado 2 ecuaciones:

$$R \cos \theta_1 - L_2 \cos \theta_2 - A = 0 \quad (2.31)$$

$$R \sin \theta_1 - L_2 \sin \theta_2 - B = 0 \quad (2.32)$$

De las ecuaciones (2.31) y (2.32), se despeja, se despeja el término que tenga θ_1 , se eleva al cuadrado ambos lados y sumamos las dos ecuaciones. Como resultado dará la siguiente ecuación:

$$R^2 = L_2^2 + 2L_2A \cos \theta_2 + 2L_2B \sin \theta_2 + A^2 + B^2 \quad (2.33)$$

Aplicando el método de Newton – Raphson, podemos conocer con (2.33) mediante iteraciones a θ_2 .

El método de Newton – Raphson es el siguiente:

$$X_{i+1} = X_i - \frac{f(X)}{f'(X)} \quad (2.34)$$

donde $f(X)=0$ resolviendo para (2.33):

$$f(\theta_2) = L_2^2 + 2L_2A \cos \theta_2 + 2L_2B \text{sen } \theta_2 + A^2 + B^2 - R^2 \quad (2.35)$$

$$f'(\theta_2) = -2L_2A \text{sen } \theta_2 + 2L_2B \cos \theta_2 \quad (2.36)$$

Sustituyendo (2.35) y (2.36) en (2.34):

$$\theta_{2i+1} = \theta_{2i} - \frac{L_2^2 + 2L_2A \cos \theta_{2i} + 2L_2B \text{sen } \theta_{2i} + A^2 + B^2 - R^2}{-2L_2A \text{sen } \theta_{2i} + 2L_2B \cos \theta_{2i}} \quad (2.37)$$

Calculando θ_1 :

De las ecuaciones (2.31) y (2.32), se despeja el término que tenga θ_2 , se eleva al cuadrado ambos lados y sumamos las dos ecuaciones. Nos dará como resultado la siguiente ecuación:

$$L_2^2 = R^2 - 2RA \cos \theta_1 - 2RB \text{sen } \theta_1 + A^2 + B^2 \quad (2.38)$$

Aplicando el método de Newton – Raphson, podemos conocer con (2.38) mediante iteraciones a θ_1 .

Resolviendo para (2.38):

$$f(\theta_1) = R^2 - 2RA \cos \theta_1 - 2RB \text{sen } \theta_1 + A^2 + B^2 - L_2^2 \quad (2.39)$$

$$f'(\theta_1) = 2RA \text{sen } \theta_1 - 2LRB \cos \theta_1 \quad (2.40)$$

Sustituyendo (2.39) y (2.40) en (2.34):

$$\theta_{i+1} = \theta_{i} - \frac{R^2 - 2RA \cos \theta_{i} - 2RB \sin \theta_{i} + A^2 + B^2 - L_2^2}{2RA \sin \theta_{i} - 2LRB \cos \theta_{i}} \quad (2.41)$$

De la segunda cadena cinemática obtenemos la siguiente ecuación vectorial (ver figura 12):

$$\bar{r}_2 + \bar{r}_3 - \bar{r}_4 - \bar{r}_5 = 0 \quad (2.42)$$

Si sustituimos las componentes en i y j de cada vector en la ecuación (2.42), nos dará como resultado 2 ecuaciones:

$$L_2 \cos \theta_2 + L_3 \cos \theta_3 - L_4 \cos \theta_4 = 0 \quad (2.43)$$

$$L_2 \sin \theta_2 + L_3 \sin \theta_3 - L_4 \sin \theta_4 - L_5 = 0 \quad (2.44)$$

De las ecuaciones (2.43) y (2.44), se despeja el término que tenga θ_3 , se eleva al cuadrado ambos lados y sumamos las dos ecuaciones. Nos dará como resultado la siguiente ecuación:

$$L_3^2 = L_5^2 + L_4^2 + L_2^2 - 2L_4L_2 \cos \theta_4 \cos \theta_2 - 2L_4L_2 \sin \theta_4 \sin \theta_2 + 2L_5L_4 \sin \theta_4 - 2L_5L_2 \sin \theta_2 \quad (2.45)$$

Aplicando el método de Newton – Raphson, podemos conocer con (2.45) mediante iteraciones a θ_4 .

Resolviendo para (2.45):

$$f(\theta_4) = L_5^2 + L_4^2 + L_2^2 - 2L_4L_2 \cos \theta_4 \cos \theta_2 - 2L_4L_2 \sin \theta_4 \sin \theta_2 + 2L_5L_4 \sin \theta_4 - 2L_5L_2 \sin \theta_2 - L_3^2 \quad (2.46)$$

$$f'(\theta_4) = 2L_4L_2 \sin \theta_4 \cos \theta_2 - 2L_4L_2 \cos \theta_4 \sin \theta_2 + 2L_5L_4 \cos \theta_4 \quad (2.47)$$

Sustituyendo (2.46) y (2.47) en (2.34):

$$\theta_{4i+1} = \theta_{4i} - \frac{L_5^2 + L_4^2 + L_2^2 - 2L_4L_2 \cos \theta_4 \cos \theta_2 - 2L_4L_2 \text{sen} \theta_4 \text{sen} \theta_2}{2L_4L_2 \text{sen} \theta_4 \cos \theta_2 - 2L_4L_2 \cos \theta_4 \text{sen} \theta_2} \quad (2.48)$$

$$\frac{+ 2L_5L_4 \text{sen} \theta_4 - 2L_5L_2 \text{sen} \theta_2 - L_3^2}{+ 2L_5L_4 \cos}$$

Calculando θ_3 :

De las ecuaciones (2.43) y (2.44), se despeja el término que tenga θ_4 , se eleva al cuadrado ambos lados y sumamos las dos ecuaciones, nos dará como resultado la siguiente ecuación:

$$L_4^2 = \left[\begin{array}{l} L_5^2 + L_3^2 + L_2^2 + 2L_3L_2 \cos \theta_3 \cos \theta_2 + 2L_3L_2 \text{sen} \theta_3 \text{sen} \theta_2 \\ - 2L_5L_3 \text{sen} \theta_3 - 2L_5L_2 \text{sen} \theta_2 \end{array} \right] \quad (2.49)$$

Aplicando el método de Newton – Raphson, podemos conocer con (2.49) mediante iteraciones a θ_3 .

Resolviendo para (2.49):

$$f(\theta_3) = \left[\begin{array}{l} L_5^2 + L_3^2 + L_2^2 + 2L_3L_2 \cos \theta_3 \cos \theta_2 + 2L_3L_2 \text{sen} \theta_3 \text{sen} \theta_2 \\ - 2L_5L_3 \text{sen} \theta_3 - 2L_5L_2 \text{sen} \theta_2 - L_4^2 \end{array} \right] \quad (2.50)$$

$$f'(\theta_3) = -2L_3L_2 \text{sen} \theta_3 \cos \theta_2 + 2L_3L_2 \cos \theta_3 \text{sen} \theta_2 - 2L_5L_3 \cos \theta_3 \quad (2.51)$$

Sustituyendo (2.50) y (2.51) en (2.34):

$$\theta_{3i+1} = \theta_{3i} - \frac{f(\theta_3)}{f'(\theta_3)} \quad (2.52)$$

2.1.1.2 SEGUNDO MECANISMO DE CUATRO BARRAS.

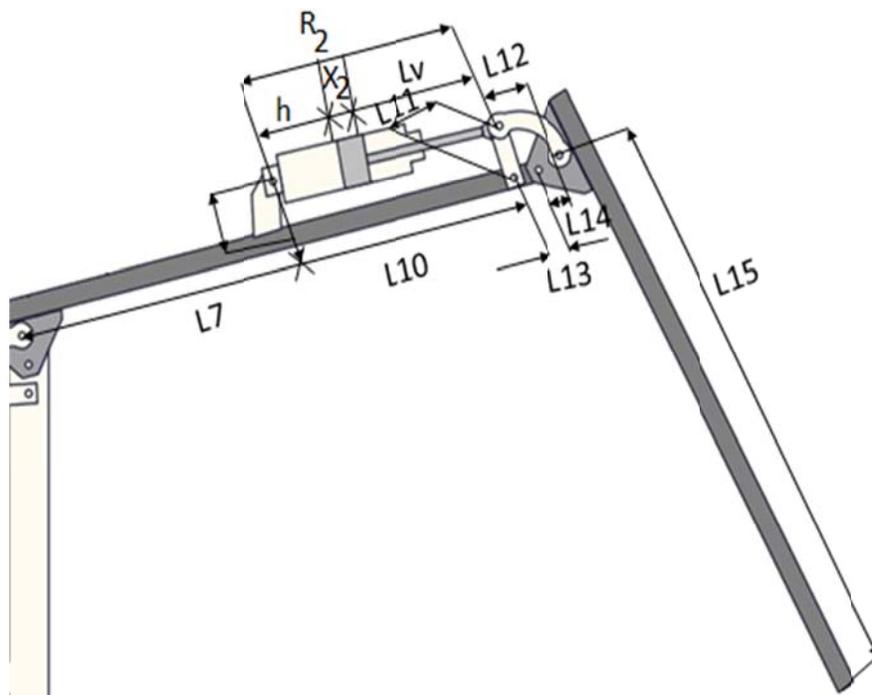


Figura 13. Segundo mecanismo corredera – manivela de cuatro barras, con dos cadenas cinemáticas cerradas.

Del segundo mecanismo se ven dos cadenas cinemáticas cerradas. La primera cadena la forman L8, L10, L11 y R2; la segunda cadena está formada por L13, L14, L12 y L11 (ver figura 13)

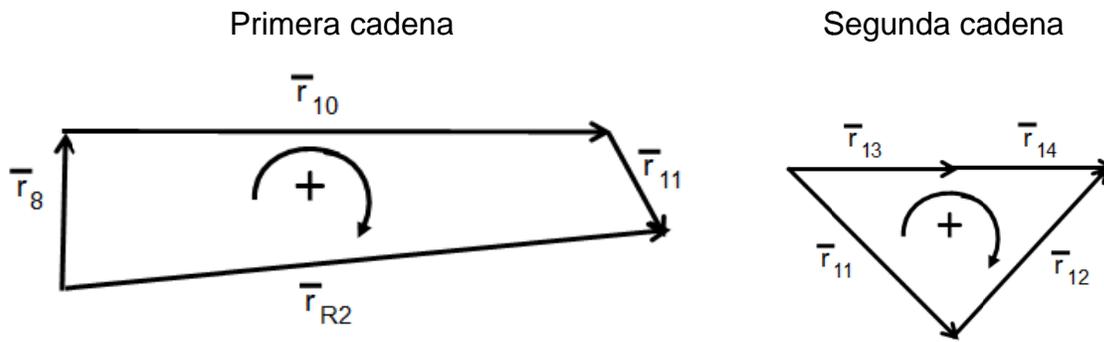


Figura 14. Cadenas cinemáticas cerradas del segundo mecanismo.

De la primera cadena cinemática obtenemos la siguiente ecuación vectorial (ver figura 14):

$$\vec{r}_8 + \vec{r}_{10} + \vec{r}_{11} - \vec{r}_{R2} = 0 \quad (2.53)$$

Si sustituimos las componentes en i y j de cada vector en la ecuación (2.53) nos dará como resultado 2 ecuaciones:

$$L_8 \cos \theta_8 + L_{10} \cos \theta_{10} + L_{11} \cos \theta_{11} - R_2 \cos \theta_9 = 0 \quad (2.54)$$

$$L_8 \text{sen} \theta_8 + L_{10} \text{sen} \theta_{10} + L_{11} \text{sen} \theta_{11} - R_2 \text{sen} \theta_9 = 0 \quad (2.55)$$

De las ecuaciones (2.54) y (2.55), se despeja el término que tenga θ_9 , se eleva al cuadrado ambos lados y sumamos las dos ecuaciones, tomando en cuenta que $\theta_8 = \theta_4$ y $\theta_{10} = (\theta_4 + 270)$. Nos dará como resultado la siguiente ecuación:

$$R_2^2 = \left[\begin{aligned} &L_8^2 + L_{10}^2 + L_{11}^2 + 2L_8L_{10} \cos \theta_4 \cos (\theta_4 + 270) \\ &+ 2L_8L_{10} \text{sen} \theta_4 \text{sen} (\theta_4 + 270) + 2L_8L_{11} \cos \theta_4 \cos \theta_{11} \\ &+ 2L_8L_{11} \text{sen} \theta_4 \text{sen} \theta_{11} + 2L_{11}L_{10} \cos \theta_{11} \cos (\theta_4 + 270) \\ &+ 2L_{11}L_{10} \text{sen} \theta_{11} \text{sen} (\theta_4 + 270) \end{aligned} \right] \quad (2.56)$$

Aplicando el método de Newton – Raphson, podemos conocer con (2.56) mediante iteraciones a θ_{11} . El ángulo θ_4 ya se calculó, por lo tanto es conocido.

Resolviendo para (2.56):

$$\begin{aligned}
 f(\theta_{11}) = & L_8^2 + L_{10}^2 + L_{11}^2 + 2L_8L_{10} \cos \theta_4 \cos(\theta_4 + 270) \\
 & + 2L_8L_{10} \text{sen} \theta_4 \text{sen}(\theta_4 + 270) + 2L_8L_{11} \cos \theta_4 \cos \theta_{11} \\
 & + 2L_8L_{11} \text{sen} \theta_4 \text{sen} \theta_{11} + 2L_{11}L_{10} \cos \theta_{11} \cos(\theta_4 + 270) \\
 & + 2L_{11}L_{10} \text{sen} \theta_{11} \text{sen}(\theta_4 + 270) - R_2^2
 \end{aligned} \tag{2.57}$$

$$\begin{aligned}
 f'(\theta_{11}) = & -2L_8L_{11} \cos \theta_4 \text{sen} \theta_{11} + 2L_8L_{11} \text{sen} \theta_4 \cos \theta_{11} \\
 & - 2L_{11}L_{10} \text{sen} \theta_{11} \cos(\theta_4 + 270) + 2L_{11}L_{10} \cos \theta_{11} \text{sen}(\theta_4 + 270)
 \end{aligned} \tag{2.58}$$

Sustituyendo (2.57) y (2.58) en (2.34):

$$\theta_{11i+1} = \theta_{11i} - \frac{f(\theta_{11i})}{f'(\theta_{11i})} \tag{2.59}$$

Calculando θ_9 :

De las ecuaciones (2.54) y (2.55), se despeja el término que tenga θ_{11} , se eleva al cuadrado ambos lados y sumamos las dos ecuaciones, tomando en cuenta que $\theta_8 = \theta_4$ y $\theta_{10} = (\theta_4 + 270)$. Nos dará como resultado la siguiente ecuación:

$$\begin{aligned}
 L_{11}^2 = & L_8^2 + L_{10}^2 + R_2^2 + 2L_8L_{10} \cos \theta_4 \cos(\theta_4 + 270) \\
 & + 2L_8L_{10} \text{sen} \theta_4 \text{sen}(\theta_4 + 270) - 2L_8R_2 \cos \theta_4 \cos \theta_9 \\
 & - 2L_8R_2 \text{sen} \theta_4 \text{sen} \theta_9 - 2R_2L_{10} \cos \theta_9 \cos(\theta_4 + 270) \\
 & - 2R_2L_{10} \text{sen} \theta_9 \text{sen}(\theta_4 + 270)
 \end{aligned} \tag{2.60}$$

Aplicando el método de Newton – Raphson, podemos conocer con (2.60) mediante iteraciones a θ_9 . El ángulo θ_4 ya se calculó, por lo tanto es conocido.

Resolviendo para (2.60):

$$\begin{aligned}
 f(\theta_9) &= L_8^2 + L_{10}^2 + R_2^2 + 2L_8L_{10} \cos \theta_4 \cos (\theta_4 + 270) \\
 &+ 2L_8L_{10} \operatorname{sen} \theta_4 \operatorname{sen} (\theta_4 + 270) - 2L_8R_2 \cos \theta_4 \cos \theta_9 \\
 &- 2L_8R_2 \operatorname{sen} \theta_4 \operatorname{sen} \theta_9 - 2R_2L_{10} \cos \theta_9 \cos (\theta_4 + 270) \\
 &- 2R_2L_{10} \operatorname{sen} \theta_9 \operatorname{sen} (\theta_4 + 270) - L_{11}^2
 \end{aligned} \tag{2.61}$$

$$\begin{aligned}
 f'(\theta_9) &= 2L_8R_2 \cos \theta_4 \cos \theta_9 - 2L_8R_2 \operatorname{sen} \theta_4 \cos \theta_9 \\
 &+ 2R_2L_{10} \operatorname{sen} \theta_9 \cos (\theta_4 + 270) - 2R_2L_{10} \cos \theta_9 \operatorname{sen} (\theta_4 + 270)
 \end{aligned} \tag{2.62}$$

Sustituyendo (2.61) y (2.62) en (2.34):

$$\theta_{9i+1} = \theta_{9i} - \frac{f(\theta_9)}{f'(\theta_9)} \tag{2.63}$$

De la segunda cadena cinemática obtenemos la siguiente ecuación vectorial (ver figura 14):

$$\bar{r}_{13} + \bar{r}_{14} - \bar{r}_{12} - \bar{r}_{11} = 0 \tag{2.64}$$

Si sustituimos las componentes en i y j de cada vector en la ecuación (2.64) nos dará como resultado 2 ecuaciones:

$$L_{13} \cos \theta_{13} + L_{14} \cos \theta_{14} - L_{12} \cos \theta_{12} - L_{11} \cos \theta_{11} = 0 \tag{2.65}$$

$$L_{13} \operatorname{sen} \theta_{13} + L_{14} \operatorname{sen} \theta_{14} - L_{12} \operatorname{sen} \theta_{12} - L_{11} \operatorname{sen} \theta_{11} = 0 \tag{2.66}$$

De las ecuaciones (2.65) y (2.66), se despeja el término que tenga θ_{12} , se eleva al cuadrado ambos lados y sumamos las dos ecuaciones, tomando en cuenta que $\theta_{13} = (\theta_4 + 270)$. Nos dará como resultado la siguiente ecuación (2.67).

$$\begin{aligned}
L_{12}^2 &= L_{13}^2 + L_{14}^2 + L_{11}^2 + 2L_{13}L_{14} \cos \theta_{14} \cos(\theta_4 + 270) \\
&+ 2L_{13}L_{14} \operatorname{sen} \theta_{14} \operatorname{sen}(\theta_4 + 270) - 2L_{13}L_{11} \cos(\theta_4 + 270) \cos \theta_{11} \\
&- 2L_{13}L_{11} \operatorname{sen}(\theta_4 + 270) \operatorname{sen} \theta_{11} - 2L_{14}L_{11} \cos \theta_{14} \cos \theta_{11} \\
&- 2L_{14}L_{11} \operatorname{sen} \theta_{14} \operatorname{sen} \theta_{11}
\end{aligned} \tag{2.67}$$

Aplicando el método de Newton – Raphson, podemos conocer con (2.67) mediante iteraciones a θ_{14} . El ángulo θ_4 y θ_{11} ya se calcularon, por lo tanto son conocidos.

Resolviendo para (2.67):

$$\begin{aligned}
f(\theta_{14}) &= L_{13}^2 + L_{14}^2 + L_{11}^2 + 2L_{13}L_{14} \cos \theta_{14} \cos(\theta_4 + 270) \\
&+ 2L_{13}L_{14} \operatorname{sen} \theta_{14} \operatorname{sen}(\theta_4 + 270) - 2L_{13}L_{11} \cos(\theta_4 + 270) \cos \theta_{11} \\
&- 2L_{13}L_{11} \operatorname{sen}(\theta_4 + 270) \operatorname{sen} \theta_{11} - 2L_{14}L_{11} \cos \theta_{14} \cos \theta_{11} \\
&- 2L_{14}L_{11} \operatorname{sen} \theta_{14} \operatorname{sen} \theta_{11} - L_{12}^2
\end{aligned} \tag{2.68}$$

$$\begin{aligned}
f'(\theta_{14}) &= -2L_{13}L_{14} \operatorname{sen} \theta_{14} \cos(\theta_4 + 270) + 2L_{13}L_{14} \cos \theta_{14} \operatorname{sen}(\theta_4 + 270) \\
&+ 2L_{14}L_{11} \operatorname{sen} \theta_{14} \cos \theta_{11} - 2L_{14}L_{11} \cos \theta_{14} \operatorname{sen} \theta_{11}
\end{aligned} \tag{2.69}$$

Sustituyendo (2.68) y (2.69) en (2.34):

$$\theta_{14i+1} = \theta_{14i} - \frac{f(\theta_{14i})}{f'(\theta_{14i})} \tag{2.70}$$

Calculando θ_{12} :

De las ecuaciones (2.65) y (2.66), se despeja el término que tenga θ_{14} , se eleva al cuadrado ambos lados y sumamos las dos ecuaciones, tomando en cuenta que $\theta_{13} = (\theta_4 + 270)$. Nos dará como resultado la siguiente ecuación (2.71).

$$\begin{aligned}
L_{14}^2 &= L_{13}^2 + L_{12}^2 + L_{11}^2 - 2L_{13}L_{12} \cos \theta_{12} \cos (\theta_4 + 270) \\
&- 2L_{13}L_{12} \text{sen } \theta_{12} \text{sen } (\theta_4 + 270) - 2L_{13}L_{11} \cos (\theta_4 + 270) \cos \theta_{11} \\
&- 2L_{13}L_{11} \text{sen } (\theta_4 + 270) \text{sen } \theta_{11} + 2L_{12}L_{11} \cos \theta_{12} \cos \theta_{11} \\
&+ 2L_{12}L_{11} \text{sen } \theta_{12} \text{sen } \theta_{11}
\end{aligned} \tag{2.71}$$

Aplicando el método de Newton – Raphson, podemos conocer con (2.71) mediante iteraciones a θ_{12} . El ángulo θ_4 y θ_{11} ya se calcularon, por lo tanto son conocidos.

Resolviendo para (2.71):

$$\begin{aligned}
f(\theta_{12}) &= L_{13}^2 + L_{12}^2 + L_{11}^2 - 2L_{13}L_{12} \cos \theta_{12} \cos (\theta_4 + 270) \\
&- 2L_{13}L_{12} \text{sen } \theta_{12} \text{sen } (\theta_4 + 270) - 2L_{13}L_{11} \cos (\theta_4 + 270) \cos \theta_{11} \\
&- 2L_{13}L_{11} \text{sen } (\theta_4 + 270) \text{sen } \theta_{11} + 2L_{12}L_{11} \cos \theta_{12} \cos \theta_{11} \\
&+ 2L_{12}L_{11} \text{sen } \theta_{12} \text{sen } \theta_{11} - L_{14}^2
\end{aligned} \tag{2.72}$$

$$\begin{aligned}
f'(\theta_{12}) &= 2L_{13}L_{12} \text{sen } \theta_{12} \cos (\theta_4 + 270) \\
&- 2L_{13}L_{12} \cos \theta_{12} \text{sen } (\theta_4 + 270) - 2L_{12}L_{11} \text{sen } \theta_{12} \cos \theta_{11} \\
&+ 2L_{12}L_{11} \cos \theta_{12} \text{sen } \theta_{11}
\end{aligned} \tag{2.73}$$

Sustituyendo (2.72) y (2.73) en (2.34):

$$\theta_{12i+1} = \theta_{12i} - \frac{f(\theta_{12})}{f'(\theta_{12})} \tag{2.74}$$

Para calcular la posición del efector final (x, y) nos apoyaremos en la figura 16. Los ángulos α_3 y β_3 son constantes, esto porque los eslabones que forman los triángulos están unidos uno de otro perpendicularmente. Los ángulos α y β son variables.

$$\alpha = \theta_4 - \alpha_3 \tag{2.75}$$

$$\beta = \theta_{14} + \beta_3 \tag{2.76}$$

Por lo tanto la posición es:

$$x = a \cos \alpha + b \cos \beta \quad (2.77)$$

$$y = a \sin \alpha + b \sin \beta \quad (2.78)$$

Las ecuaciones (2.37), (2.41), (2.48), (2.52), (2.59), (2.63), (2.70), (2.74), (2.75), (2.76), (2.77) y (2.78) nos servirán para hacer el algoritmo de programación del manipulador, esto se verá en la parte de simulación.

2.1.2 CINEMÁTICA INVERSA

Determinar la configuración que debe adoptar el robot para una posición y orientación del extremo conocidas.

El método utilizado es la aplicación de las ecuaciones resultantes del cálculo de la cinemática directa del robot manipulador.

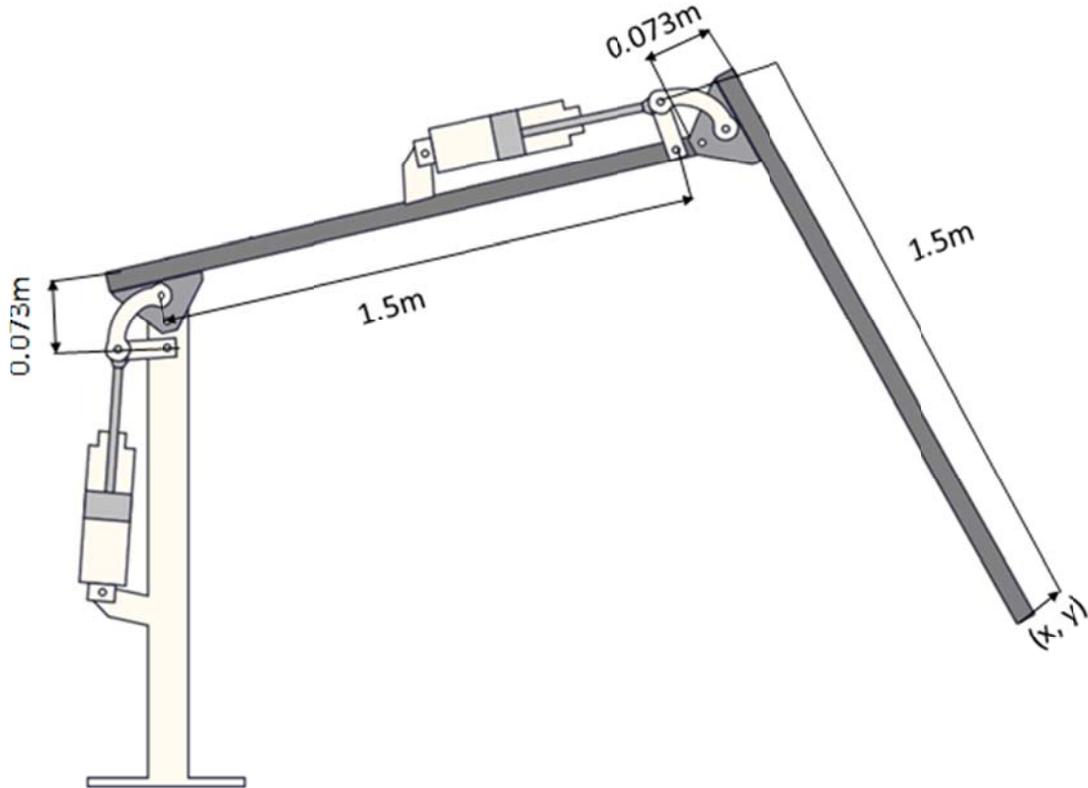


Figura 15. Dimensiones de los elementos que se utilizaran para el cálculo inicial.

De la figura 15, observamos que están dimensionados solo unos elementos, esto es por qué solo estos serán utilizados para nuestro análisis inicial. En la figura 16 se observa con mayor claridad los dos triángulos rectángulos que nos servirán para calcular θ_4 , θ_{14} , θ_5 y θ_{15} , y posteriormente mediante el método de Newton – Raphson el resto de los ángulos.

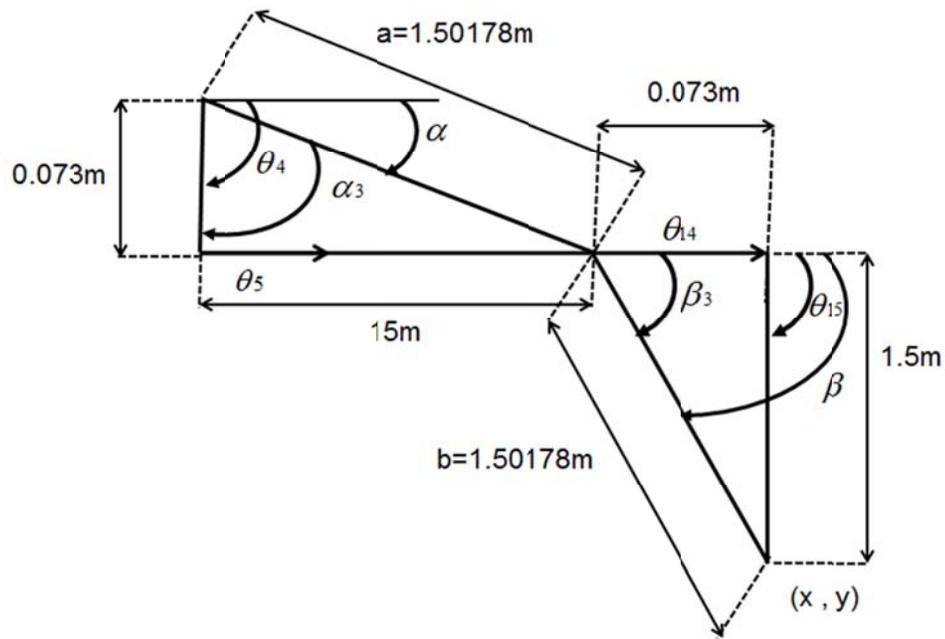


Figura 16. Triángulos rectángulos útiles para el cálculo inicial donde se obtienen θ_4 y θ_{14} ,

Los ángulos α_3 y β_3 son constantes, esto porque los eslabones que forman los triángulos están unidos uno de otro perpendicularmente. Los ángulos α y β son variables y dependen de la posición del efector final.

Partiendo de la posición de efector final, tenemos (ver figura 16):

$$x = a \cos \alpha + b \cos \beta \quad (2.79)$$

$$y = a \sin \alpha + b \sin \beta \quad (2.80)$$

De las ecuaciones (2.79) y (2.80), se despeja el término que tenga β , se eleva al cuadrado ambos lados y sumamos las dos ecuaciones, x y y son conocidos. Nos dará como resultado la siguiente ecuación:

$$b^2 = a^2 + x^2 + y^2 - 2ax \cos \alpha - 2ay \sin \alpha \quad (2.81)$$

Aplicando el método de Newton – Raphson, podemos conocer con (2.81) mediante iteraciones a α .

Resolviendo para (2.81):

$$f(\alpha) = a^2 + x^2 + y^2 - 2ax \cos \alpha - 2aysen \alpha - b^2 \quad (2.82)$$

$$f'(\alpha) = 2axsen \alpha - 2ay \cos \alpha \quad (2.83)$$

Sustituyendo (2.82) y (2.83) en (2.34):

$$\alpha_{i+1} = \alpha_i - \frac{a^2 + x^2 + y^2 - 2ax \cos \alpha - 2aysen \alpha - b^2}{2axsen \alpha - 2ay \cos \alpha} \quad (2.84)$$

Calculando β :

De las ecuaciones (2.79) y (2.80), se despeja el término que tenga α , se eleva al cuadrado ambos lados y sumamos las dos ecuaciones, x y y son conocidos. Nos dará como resultado la siguiente ecuación:

$$a^2 = b^2 + x^2 + y^2 - 2bx \cos \alpha - 2bysen \alpha \quad (2.85)$$

Aplicando el método de Newton – Raphson, podemos conocer con (2.85) mediante iteraciones a β .

Resolviendo para (2.85):

$$f(\beta) = b^2 + x^2 + y^2 - 2bx \cos \alpha - 2bysen \alpha - a^2 \quad (2.86)$$

$$f'(\beta) = 2bxsen \alpha - 2by \cos \alpha \quad (2.87)$$

Sustituyendo (2.86) y (2.87) en (2.34):

$$\beta_{i+1} = \beta_i - \frac{b^2 + x^2 + y^2 - 2bx \cos \beta - 2bysen \beta - a^2}{2bxsen \beta - 2by \cos \beta} \quad (2.88)$$

De la figura 16 observamos:

$$\theta_4 = \alpha + \alpha_3 \quad (2.89)$$

$$\theta_5 = \theta_4 + 270 \quad (2.90)$$

$$\theta_{14} = \beta - \beta_3 \quad (2.91)$$

$$\theta_{15} = \theta_{14} + 90 \quad (2.92)$$

Para conocer el resto de los ángulos se realizara el análisis del manipulador en dos etapas; primero se analizara el primer mecanismo y posteriormente el segundo.

2.1.2.1 PRIMER MECANISMO DE CUATRO BARRAS.

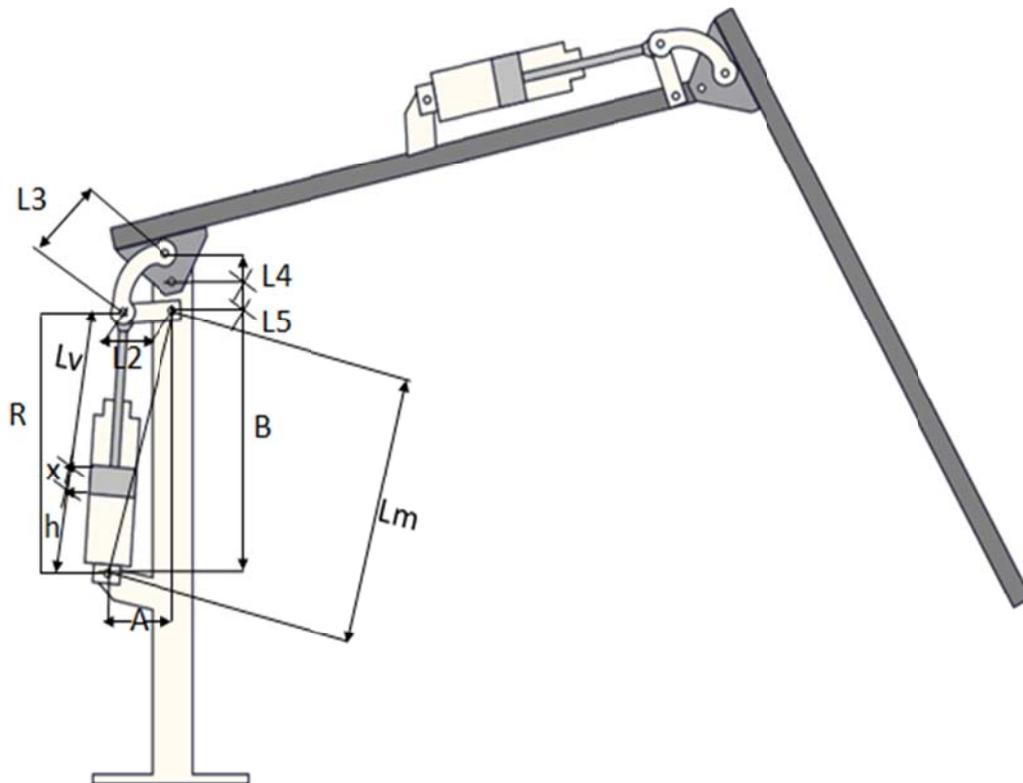


Figura 17. Primer mecanismo corredera – manivela de cuatro barras, con dos cadenas cinemáticas cerradas

Del primer mecanismo se observan dos cadenas cinemáticas cerradas, las cuales utilizaremos para obtener los ángulos θ_1 , θ_2 y θ_3 . La primera cadena la forman L2, L3, L4 y L5; la segunda cadena está formada por R, L2 y Lm, (ver figura 17)

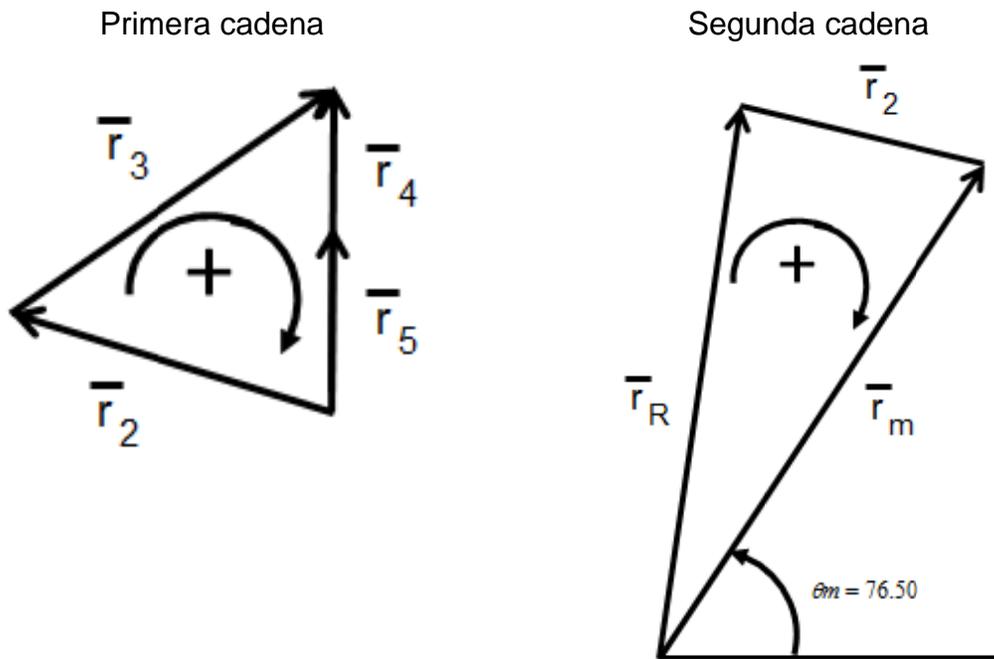


Figura 18. Cadenas cinemáticas cerradas del primer mecanismo.

De la primera cadena cinemática obtenemos la siguiente ecuación vectorial (ver figura 18):

$$\bar{r}_2 + \bar{r}_3 - \bar{r}_4 - \bar{r}_5 = 0 \quad (2.93)$$

Si sustituimos las componentes en i y j de cada vector en la ecuación (2.93) nos dará como resultado 2 ecuaciones:

$$L_2 \cos \theta_2 + L_3 \cos \theta_3 - L_4 \cos \theta_4 = 0 \quad (2.94)$$

$$L_2 \text{sen} \theta_2 + L_3 \text{sen} \theta_3 - L_4 \text{sen} \theta_4 - L_5 = 0 \quad (2.95)$$

De las ecuaciones (2.94) y (2.95), se despeja el término que tenga θ_2 , se eleva al cuadrado ambos lados y sumamos las dos ecuaciones, tomando en cuenta que θ_2 es conocido, ya fue calculado anteriormente. Nos dará como resultado la siguiente ecuación:

$$L_2^2 = L_5^2 + L_4^2 + L_3^2 - 2L_4L_3 \cos \theta_4 \cos \theta_3 - 2L_4L_3 \text{sen } \theta_4 \text{sen } \theta_3 + 2L_5L_4 \text{sen } \theta_4 - 2L_5L_3 \text{sen } \theta_3 \quad (2.96)$$

Aplicando el método de Newton – Raphson, podemos conocer con (2.96) mediante iteraciones a θ_3 .

Resolviendo para (2.96):

$$f(\theta_3) = L_5^2 + L_4^2 + L_3^2 - 2L_4L_3 \cos \theta_4 \cos \theta_3 - 2L_4L_3 \text{sen } \theta_4 \text{sen } \theta_3 + 2L_5L_4 \text{sen } \theta_4 - 2L_5L_3 \text{sen } \theta_3 - L_2^2 \quad (2.97)$$

$$f'(\theta_3) = 2L_4L_3 \cos \theta_4 \text{sen } \theta_3 - 2L_4L_3 \text{sen } \theta_4 \cos \theta_3 - 2L_5L_3 \cos \theta_3 \quad (2.98)$$

Sustituyendo (2.97) y (2.98) en (2.34):

$$\theta_{3i+1} = \theta_{3i} - \frac{f(\theta_3)}{f'(\theta_3)} \quad (2.99)$$

Calculando θ_2 :

De las ecuaciones (2.94) y (2.95), se despeja el término que tenga θ_3 , se eleva al cuadrado ambos lados y sumamos las dos ecuaciones, tomando en cuenta que θ_2 es conocido, ya fue calculado anteriormente. Nos dará como resultado la siguiente ecuación:

$$L_3^2 = L_5^2 + L_4^2 + L_2^2 - 2L_4L_2 \cos \theta_4 \cos \theta_2 - 2L_4L_2 \text{sen } \theta_4 \text{sen } \theta_2 + 2L_5L_4 \text{sen } \theta_4 - 2L_5L_2 \text{sen } \theta_2 \quad (2.100)$$

Aplicando el método de Newton – Raphson, podemos conocer con (2.100) mediante iteraciones a θ_2 .

Resolviendo para (2.100):

$$f(\theta_2) = L_5^2 + L_4^2 + L_2^2 - 2L_4L_2 \cos \theta_4 \cos \theta_2 - 2L_4L_2 \text{sen} \theta_4 \text{sen} \theta_2 + 2L_5L_4 \text{sen} \theta_4 - 2L_5L_2 \text{sen} \theta_2 - L_3^2 \quad (2.101)$$

$$f'(\theta_2) = 2L_4L_2 \cos \theta_4 \text{sen} \theta_2 - 2L_4L_2 \text{sen} \theta_4 \cos \theta_2 - 2L_5L_2 \cos \theta_2 \quad (2.102)$$

Sustituyendo (2.101) y (2.102) en (2.34):

$$\theta_{2i+1} = \theta_{2i} - \frac{f(\theta_2)}{f'(\theta_2)} \quad (2.103)$$

De la segunda cadena cinemática obtenemos la siguiente ecuación vectorial (ver figura 18):

$$\vec{r}_R - \vec{r}_2 - \vec{r}_m = 0 \quad (2.104)$$

Si sustituimos las componentes en i y j de cada vector en la ecuación (2.104) nos dará como resultado 2 ecuaciones:

$$R \cos \theta_1 - L_2 \cos \theta_2 - L_m \cos \theta_m = 0 \quad (2.105)$$

$$R \text{sen} \theta_1 - L_2 \text{sen} \theta_2 - L_m \text{sen} \theta_m = 0 \quad (2.106)$$

De las ecuaciones (2.105) y (2.106), se despeja el término que tenga θ_1 , se eleva al cuadrado ambos lados y sumamos las dos ecuaciones. Nos dará como resultado la siguiente ecuación:

$$R^2 = L_2^2 + L_m^2 + 2L_2L_m \cos \theta_2 \cos \theta_m + 2L_2L_m \text{sen} \theta_2 \text{sen} \theta_m \quad (2.107)$$

Con la ecuación anterior podemos conocer R y con ésta, el desplazamiento (x) del vástago del pistón del primer mecanismo. Los ángulos θ_2 y θ_m son conocidos; θ_m es constante.

$$R = \left(L_2^2 + L_m^2 + 2L_2L_m \cos\theta_2 \cos\theta_m + 2L_2L_m \text{sen}\theta_2 \text{sen}\theta_m \right)^{\frac{1}{2}} \quad (2.108)$$

Por lo tanto el desplazamiento del vástago (x) es (ver figura 17):

$$x = R - Lv - h \quad (2.109)$$

Calculando θ_1 :

De las ecuaciones (2.105) y (2.106), se despeja el término que tenga θ_m , se eleva al cuadrado ambos lados y sumamos las dos ecuaciones. Nos dará como resultado la siguiente ecuación:

$$L_m^2 = L_2^2 + R^2 - 2L_2R \cos\theta_2 \cos\theta_1 - 2L_2R \text{sen}\theta_2 \text{sen}\theta_1 \quad (2.110)$$

Aplicando el método de Newton – Raphson, podemos conocer con (2.110) mediante iteraciones a θ_1 , considerando que θ_2 es conocido.

Resolviendo para (2.110):

$$f(\theta_1) = L_2^2 + R^2 - 2L_2R \cos\theta_2 \cos\theta_1 - 2L_2R \text{sen}\theta_2 \text{sen}\theta_1 - L_m^2 \quad (2.111)$$

$$f'(\theta_1) = 2L_2R \cos\theta_2 \text{sen}\theta_1 - 2L_2R \text{sen}\theta_2 \cos\theta_1 \quad (2.112)$$

Sustituyendo (2.111) y (2.112) en (2.34):

$$\theta_{li+1} = \theta_{li} - \frac{L_2^2 + R^2 - 2L_2R \cos\theta_2 \cos\theta_1 - 2L_2R \text{sen}\theta_2 \text{sen}\theta_1 - L_m^2}{2L_2R \cos\theta_2 \text{sen}\theta_1 - 2L_2R \text{sen}\theta_2 \cos\theta_1} \quad (2.113)$$

2.1.2.2 SEGUNDO MECANISMO DE CUATRO BARRAS.

Para conocer el resto de los ángulos se realizara el análisis del manipulador donde ahora se analizara el segundo manipulador como se muestra en la figura 19.

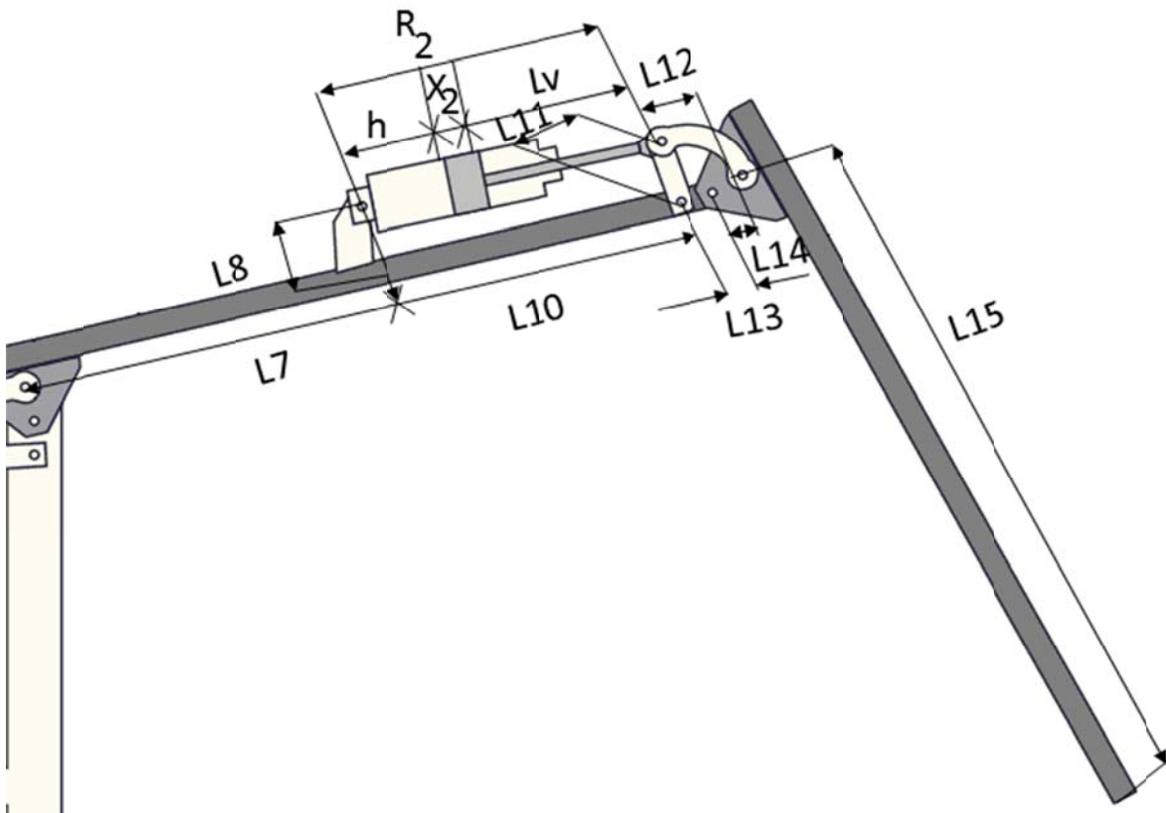


Figura 19. Segundo mecanismo corredera – manivela de cuatro barras, con dos cadenas cinemáticas cerradas.

Del segundo mecanismo se ven dos cadenas cinemáticas cerradas. La primera cadena la forman L13, L14, L12 y L11; la segunda cadena está formada por L8, L10, L11 y R_2 (ver figura 19)

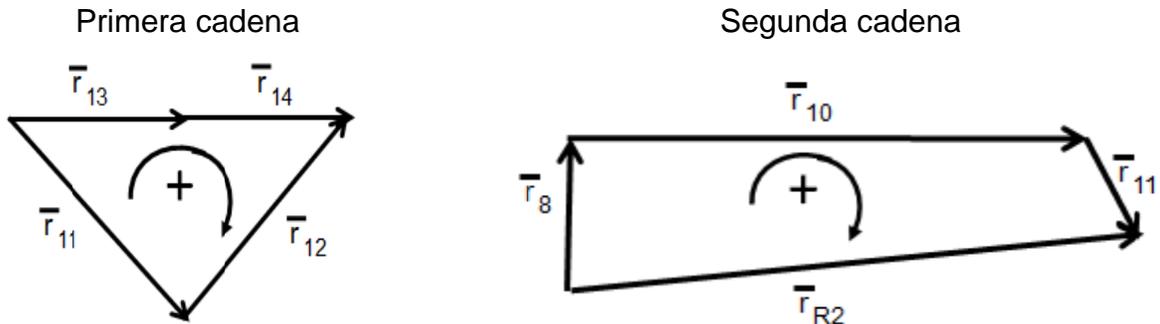


Figura 20. Cadenas cinemáticas cerradas del segundo mecanismo.

De la primera cadena cinemática obtenemos la siguiente ecuación vectorial (ver figura 20):

$$\vec{r}_{13} + \vec{r}_{14} - \vec{r}_{12} - \vec{r}_{11} = 0 \quad (2.114)$$

Si sustituimos las componentes en i y j de cada vector en la ecuación (2.114) nos dará como resultado 2 ecuaciones:

$$L_{13} \cos \theta_{13} + L_{14} \cos \theta_{14} - L_{12} \cos \theta_{12} - L_{11} \cos \theta_{11} = 0 \quad (2.115)$$

$$L_{13} \text{sen} \theta_{13} + L_{14} \text{sen} \theta_{14} - L_{12} \text{sen} \theta_{12} - L_{11} \text{sen} \theta_{11} = 0 \quad (2.116)$$

De las ecuaciones (2.115) y (2.116), se despeja el término que tenga θ_{12} , se eleva al cuadrado ambos lados y sumamos las dos ecuaciones, tomando en cuenta que $\theta_{13} = (\theta_4 + 270)$. Nos dará como resultado la siguiente ecuación:

$$\begin{aligned} L_{12}^2 &= L_{13}^2 + L_{14}^2 + L_{11}^2 + 2L_{13}L_{14} \cos \theta_{14} \cos (\theta_4 + 270) \\ &+ 2L_{13}L_{14} \text{sen} \theta_{14} \text{sen} (\theta_4 + 270) - 2L_{13}L_{11} \cos (\theta_4 + 270) \cos \theta_{11} \\ &- 2L_{13}L_{11} \text{sen} (\theta_4 + 270) \text{sen} \theta_{11} - 2L_{14}L_{11} \cos \theta_{14} \cos \theta_{11} \\ &- 2L_{14}L_{11} \text{sen} \theta_{14} \text{sen} \theta_{11} \end{aligned} \quad (2.117)$$

Aplicando el método de Newton – Raphson, podemos conocer con (2.117) mediante iteraciones a θ_{11} , el ángulo θ_4 y θ_{14} ya se calcularon, por lo tanto son conocidos.

Resolviendo para (2.117):

$$\begin{aligned}
 f(\theta_{11}) &= L_{13}^2 + L_{14}^2 + L_{11}^2 + 2L_{13}L_{14} \cos \theta_{14} \cos(\theta_4 + 270) \\
 &+ 2L_{13}L_{14} \text{sen } \theta_{14} \text{sen}(\theta_4 + 270) - 2L_{13}L_{11} \cos(\theta_4 + 270) \cos \theta_{11} \\
 &- 2L_{13}L_{11} \text{sen}(\theta_4 + 270) \text{sen} \theta_{11} - 2L_{14}L_{11} \cos \theta_{14} \cos \theta_{11} \\
 &- 2L_{14}L_{11} \text{sen} \theta_{14} \text{sen} \theta_{11} - L_{12}^2
 \end{aligned} \tag{2.118}$$

$$\begin{aligned}
 f'(\theta_{11}) &= 2L_{13}L_{11} \cos(\theta_4 + 270) \text{sen} \theta_{11} - 2L_{13}L_{11} \text{sen}(\theta_4 + 270) \cos \theta_{11} \\
 &+ 2L_{14}L_{11} \cos \theta_{14} \text{sen} \theta_{11} - 2L_{14}L_{11} \text{sen} \theta_{14} \cos \theta_{11}
 \end{aligned} \tag{2.119}$$

Sustituyendo (2.118) y (2.119) en (2.34):

$$\theta_{11i+1} = \theta_{11i} - \frac{f(\theta_{11i})}{f'(\theta_{11i})} \tag{2.120}$$

Calculando θ_{12} :

De las ecuaciones (2.115) y (2.116), se despeja el término que tenga θ_{11} , se eleva al cuadrado ambos lados y sumamos las dos ecuaciones, tomando en cuenta que $\theta_{13} = (\theta_4 + 270)$. Nos dará como resultado la siguiente ecuación:

$$\begin{aligned}
 L_{11}^2 &= L_{13}^2 + L_{12}^2 + L_{14}^2 + 2L_{13}L_{14} \cos(\theta_4 + 270) \cos \theta_{14} \\
 &+ 2L_{13}L_{14} \text{sen}(\theta_4 + 270) \text{sen} \theta_{14} - 2L_{13}L_{12} \cos \theta_{12} \cos(\theta_4 + 270) \\
 &- 2L_{13}L_{12} \text{sen} \theta_{12} \text{sen}(\theta_4 + 270) - 2L_{12}L_{14} \cos \theta_{12} \cos \theta_{14} \\
 &- 2L_{12}L_{14} \text{sen} \theta_{12} \text{sen} \theta_{14}
 \end{aligned} \tag{2.121}$$

Aplicando el método de Newton – Raphson, podemos conocer con (2.121) mediante iteraciones a θ_{12} . El ángulo θ_4 y θ_{11} ya se calcularon, por lo tanto son conocidos.

Resolviendo para (2.121):

$$\begin{aligned}
 f(\theta_{12}) = & L_{13}^2 + L_{12}^2 + L_{14}^2 + 2L_{13}L_{14} \cos(\theta_4 + 270) \cos \theta_{14} \\
 & + 2L_{13}L_{14} \text{sen}(\theta_4 + 270) \text{sen} \theta_{14} - 2L_{13}L_{12} \cos \theta_{12} \cos(\theta_4 + 270) \\
 & - 2L_{13}L_{12} \text{sen} \theta_{12} \text{sen}(\theta_4 + 270) - 2L_{12}L_{14} \cos \theta_{12} \cos \theta_{14} \\
 & - 2L_{12}L_{14} \text{sen} \theta_{12} \text{sen} \theta_{14} - L_{11}^2
 \end{aligned} \tag{2.122}$$

$$\begin{aligned}
 f'(\theta_{12}) = & 2L_{13}L_{12} \text{sen} \theta_{12} \cos(\theta_4 + 270) - 2L_{13}L_{12} \cos \theta_{12} \text{sen}(\theta_4 + 270) \\
 & + 2L_{12}L_{14} \text{sen} \theta_{12} \cos \theta_{14} - 2L_{12}L_{14} \cos \theta_{12} \text{sen} \theta_{14} - L_{11}^2
 \end{aligned} \tag{2.123}$$

Sustituyendo (2.122) y (2.123) en (2.34):

$$\theta_{12i+1} = \theta_{12i} - \frac{f(\theta_{12})}{f'(\theta_{12})} \tag{2.124}$$

De la segunda cadena cinemática obtenemos la siguiente ecuación vectorial (ver figura 20):

$$\bar{r}_8 + \bar{r}_{10} + \bar{r}_{11} - \bar{r}_{R2} = 0 \tag{2.125}$$

Si sustituimos las componentes en i y j de cada vector en la ecuación (2.96) nos dará como resultado 2 ecuaciones:

$$L_8 \cos \theta_8 + L_{10} \cos \theta_{10} + L_{11} \cos \theta_{11} - R_2 \cos \theta_9 = 0 \tag{2.126}$$

$$L_8 \text{sen} \theta_8 + L_{10} \text{sen} \theta_{10} + L_{11} \text{sen} \theta_{11} - R_2 \text{sen} \theta_9 = 0 \tag{2.127}$$

De las ecuaciones (2.126) y (2.127), se despeja el término que tenga θ_9 , se eleva al cuadrado ambos lados y sumamos las dos ecuaciones, tomando en cuenta que $\theta_8=\theta_4$, $\theta_{10} = (\theta_4+270)$. Nos dará como resultado la siguiente ecuación:

$$\begin{aligned}
 R_2^2 = & L_8^2 + L_{10}^2 + L_{11}^2 + 2L_8L_{10} \cos \theta_4 \cos (\theta_4 + 270) \\
 & + 2L_8L_{10} \text{sen } \theta_4 \text{sen } (\theta_4 + 270) + 2L_8L_{11} \cos \theta_4 \cos \theta_{11} \\
 & + 2L_8L_{11} \text{sen } \theta_4 \text{sen } \theta_{11} + 2L_{11}L_{10} \cos \theta_{11} \cos (\theta_4 + 270) \\
 & + 2L_{11}L_{10} \text{sen } \theta_{11} \text{sen } (\theta_4 + 270)
 \end{aligned} \tag{2.128}$$

Con la ecuación anterior podemos conocer R_2 y con ésta, el desplazamiento (x_2) del vástago del pistón del segundo mecanismo. Los ángulos θ_4 y θ_{11} son conocidos.

$$R_2 = \left[\begin{aligned}
 & L_8^2 + L_{10}^2 + L_{11}^2 + 2L_8L_{10} \cos \theta_4 \cos (\theta_4 + 270) \\
 & + 2L_8L_{10} \text{sen } \theta_4 \text{sen } (\theta_4 + 270) + 2L_8L_{11} \cos \theta_4 \cos \theta_{11} \\
 & + 2L_8L_{11} \text{sen } \theta_4 \text{sen } \theta_{11} + 2L_{11}L_{10} \cos \theta_{11} \cos (\theta_4 + 270) \\
 & + 2L_{11}L_{10} \text{sen } \theta_{11} \text{sen } (\theta_4 + 270)
 \end{aligned} \right]^{\frac{1}{2}} \tag{2.129}$$

Por lo tanto el desplazamiento del vástago (x_2)

$$x_2 = R_2 - Lv - h \tag{2.130}$$

Las ecuaciones (2.85), (2.88), (2.89), (2.90), (2.91), (2.92), (2.99), (2.103), (2.109), (2.113), (2.120), (2.124), (2.129) y (2.130) nos servirán para hacer el algoritmo de programación del manipulador, esto se verá en la parte de simulación. [13]

2.1.3 PLANEACIÓN DE TRAYECTORIA

El objetivo que el manipulador se mueva de un punto a otro de una manera suave y controlada para realizar una línea o un círculo [12]; el robot debe mover las articulaciones de un punto inicial a un punto final coordinadamente [8].

La generación de trayectorias tiene la función de unir una sucesión de puntos en el espacio articular por los que se quiere que pasen las articulaciones del robot en un instante determinado.

Cada trayectoria consta de uno o más segmentos delimitados por puntos conocidos denominados nodos, entre los cuales se calcula un número variable de puntos intermedios, de acuerdo con la selección efectuada por quien define la trayectoria [32].

Para determinar las trayectorias existen tres posibilidades:

1. Trayectoria punto a punto

Cada articulación evoluciona desde su posición inicial hasta la final, sin realizar consideración alguna sobre el estado o evolución de las demás articulaciones. El movimiento puede ser:

- eje por eje Solo se mueve un eje cada vez. Comenzará a moverse la primera articulación, una vez que haya alcanzado su punto final lo hará la segunda, y así sucesivamente. Ventaja es un menor consumo de potencia instantánea por parte de los actuadores.
- simultáneo de ejes Actuadores se mueven simultáneamente las articulaciones del robot a una velocidad específica para cada una de ellas. El movimiento del robot no acabara hasta que se alcance el punto final [31].

Solo tiene interés práctico cuando los puntos están suficientemente separados ya que, en caso contrario, la especificación sería muy laboriosa. Por otra parte, los puntos tampoco pueden estar muy separados ya que entonces el riesgo de que se generen movimientos imprevisibles o no controlados es grande. En este control punto a punto el sistema de control automático del robot debe realizar la interpolación entre los puntos especificados, de forma tal que, posteriormente, sea posible realizar el control de movimientos para que el robot pase por dichos puntos.

2. Trayectorias Coordinadas o isócronas.

Todas las articulaciones se coordinan comenzando y acabando su movimiento a la vez, adaptándose todas a la más lenta. Son aquellas donde se toma como base al eje más lento, averiguando cuál es esta articulación y cuánto tiempo invertirá, para que el resto de los ejes realicen su movimiento en el mismo tiempo. Se tiene así que todas las articulaciones se coordinan comenzando y terminado su movimiento a la vez, adaptándose todas a la más lenta.

3. Trayectoria continuas.

Son aquellas en las que el usuario determina la trayectoria que desea que el robot describa, determinada por líneas o curvas.

El resultado será que cada articulación sigue un movimiento aparentemente complicado con posibles cambios de dirección y sin aparente coordinación con el resto de las articulaciones, teniendo como resultado que el robot describirá la trayectoria deseada. El sistema de control debe hacer que el robot reproduzca lo más fielmente posible la trayectoria especificada. La figura 21 representa las trayectorias articulares $q_1(t)$ y $q_2(t)$ y el resultado final en el espacio del área (x, y) para un robot tipo Scara correspondientes a la figura 21 que muestra a los cuatro tipos de trayectorias indicadas.

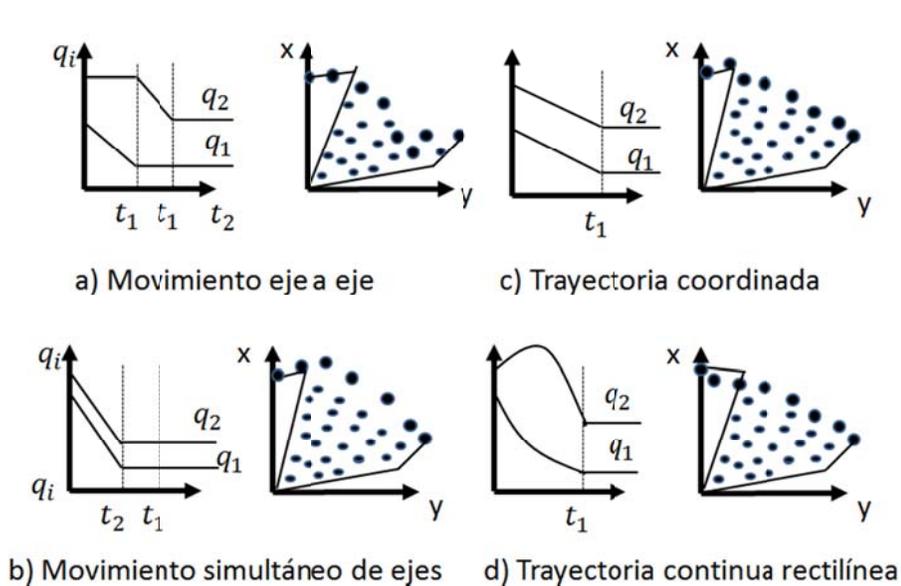


Figura 21. Tipos de trayectorias

En este caso para la realización del proyecto se toma muy en cuenta la segunda opción se denomina control de trayectoria continua. En este caso, el sistema de control debe hacer que el robot reproduzca lo más fielmente posible la trayectoria especificada.

En robots manipuladores la generación de trayectorias articulares suele realizarse en tiempo del orden de los milisegundos o decenas de milisegundos. [33]

Modo XY el brazo realiza los movimientos en los planos x, y, tomando como base el sistema cartesiano del robot.

A continuación se plantea y se obtienen las ecuaciones generales necesarias para desplazamientos lineales.

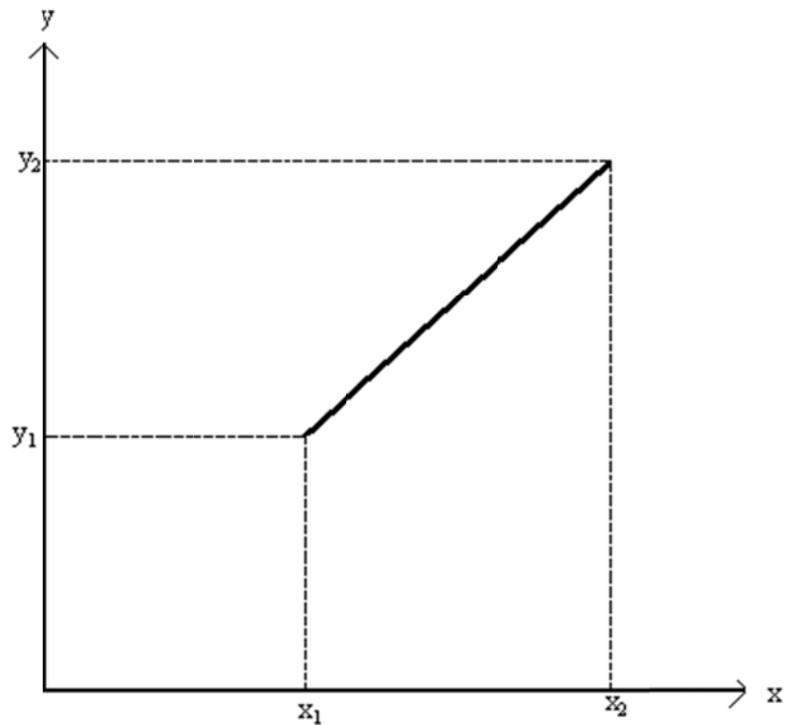


Figura 22. Trayectoria de una línea recta.

Partiendo de la ecuación general para una línea recta tenemos (ver figura 22) :

$$y = mx + b \quad (2.3.1)$$

$$m = \frac{y_1 - y_2}{x_1 - x_2} \quad (2.3.2)$$

$$b = y_1 - \left(\frac{y_1 - y_2}{x_1 - x_2} \right) x_1 \quad (2.3.3)$$

Sustituyendo (2.3.2) y (2.3.3) en (2.3.1), obtenemos una ecuación general para y que depende de x .

$$y(x) = \left(\frac{y_1 - y_2}{x_1 - x_2} \right) x + \left[y_1 - \left(\frac{y_1 - y_2}{x_1 - x_2} \right) x_1 \right] \quad (2.3.4)$$

DESARROLLO DEL SOFTWARE

En este capítulo se detalla la aplicación de la metodología para integrar las aplicaciones matemáticas complejas, transfiriendo la información a la aplicación en C++, utilizando la facilidad de las herramientas que otorga las librerías de OpenGL para mostrar la simulación en 2 dimensión.

Con la simulación se pretende diseñar y desarrollar un modelo computarizado de la cinemática con el propósito de entender el comportamiento de la cinemática de manera ideal.

3.1 METODOLOGÍA

La metodología para la simulación gráfica se muestra en la figura 23. El origen de cada objeto es establecido por el usuario de acuerdo a la trayectoria que se requiere para realizar una traslación de coordenadas para mover el origen de las piezas en forma coordinada, a su punto de rotación y que se pueda realizar la simulación gráfica. Primero se requiere haber obtenido previamente los parámetros del modelo de la cinemática una vez que se cuente con los valores se realizan los cálculos en Matlab donde se evalúa la ley de control para la trayectoria y la salida es usada como entrada al proceso de simulación grafica a través de la interfaz en lenguaje C++, donde OpenGL toma dichas valores para generar la animación grafica de la trayectoria. Los resultados son analizados por el usuario y en caso de que no corresponda el comportamiento deseado de la trayectoria, se realizan ajustes a los parámetros de control y se analiza la ley de control de la trayectoria de esta manera los resultados deben de corresponder a la trayectoria deseada.

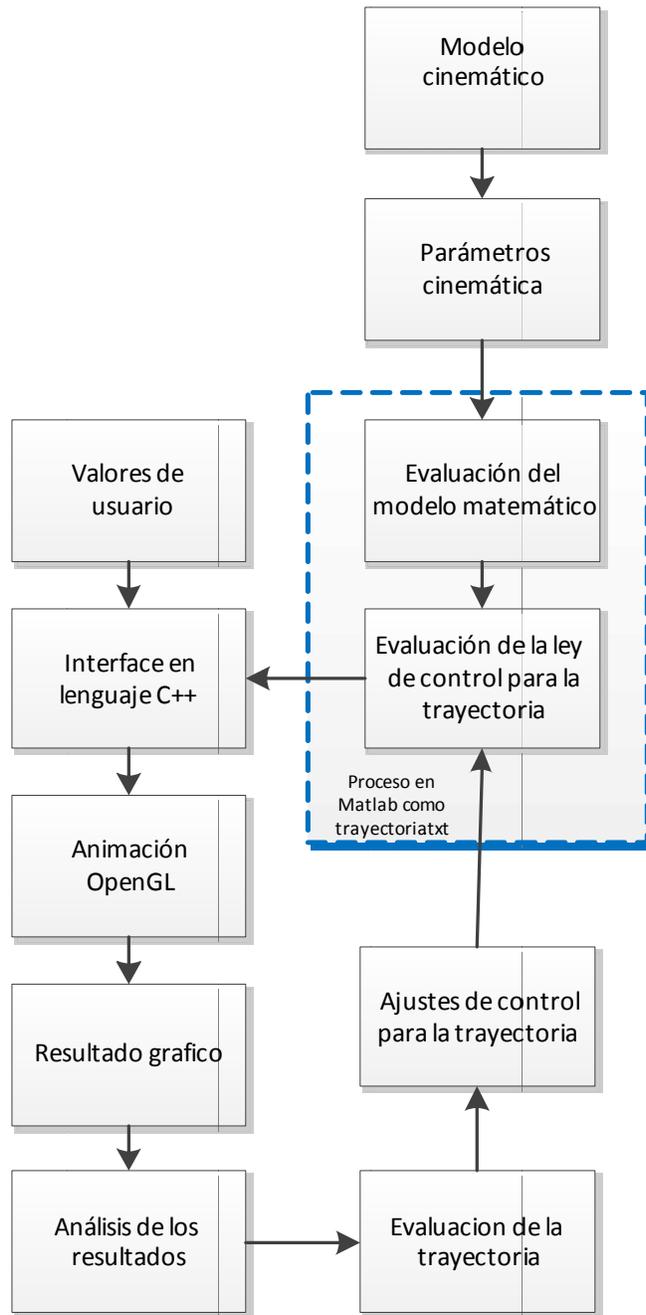


Figura 23. Metodología para la simulación gráfica.

El desarrollo del software se muestra en la figura 24.

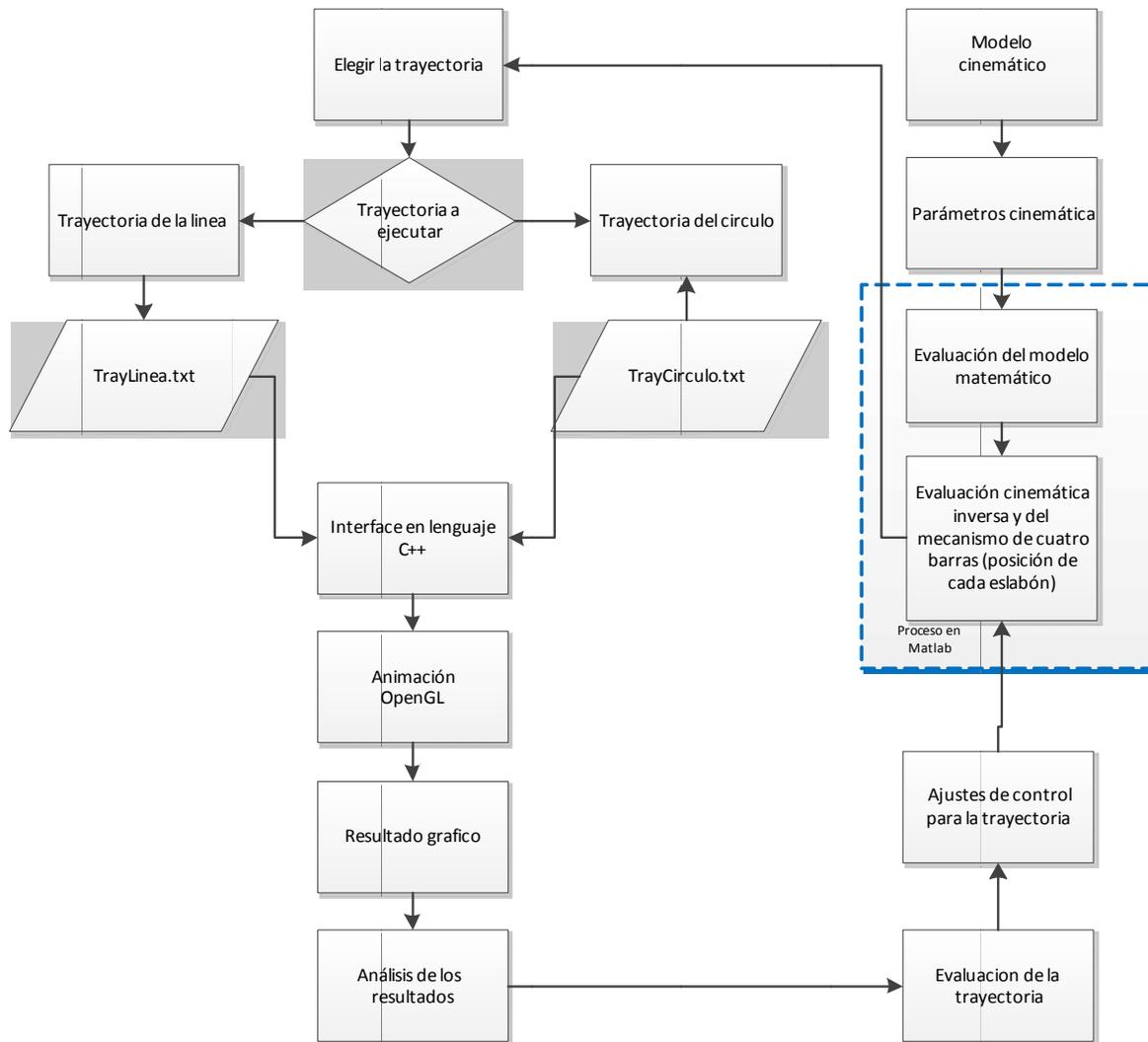


Figura 24. El desarrollo del software.

El proceso para obtener la descripción gráfica de la trayectoria de para un manipulador flexible con actuador neumático de dos grados de libertad como se muestra en la figura 25.

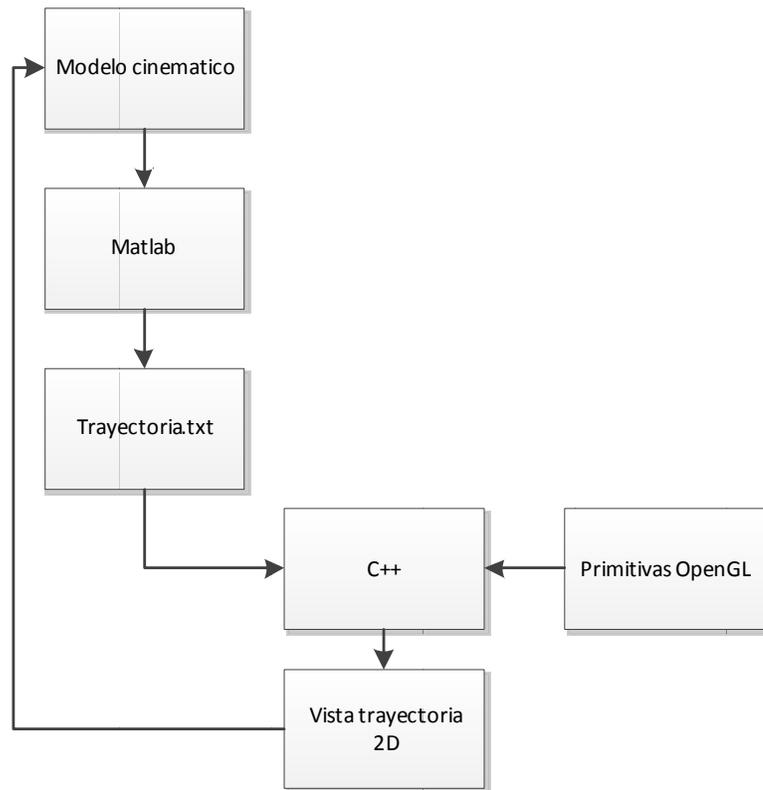


Figura 25. Proceso para obtener la animación gráfica de la trayectoria

3.2 INTERFAZ MULTILENGUAJE

Para la simulación de este proyecto denominado Modelación y simulación de la planeación de trayectorias para un manipulador flexible con actuador neumático de dos grados de libertad. Se basa en una interfaz multilenguaje como muestra Dr. Ramos [1] [2] [3] [4] Como se muestra en la siguiente figura 26.

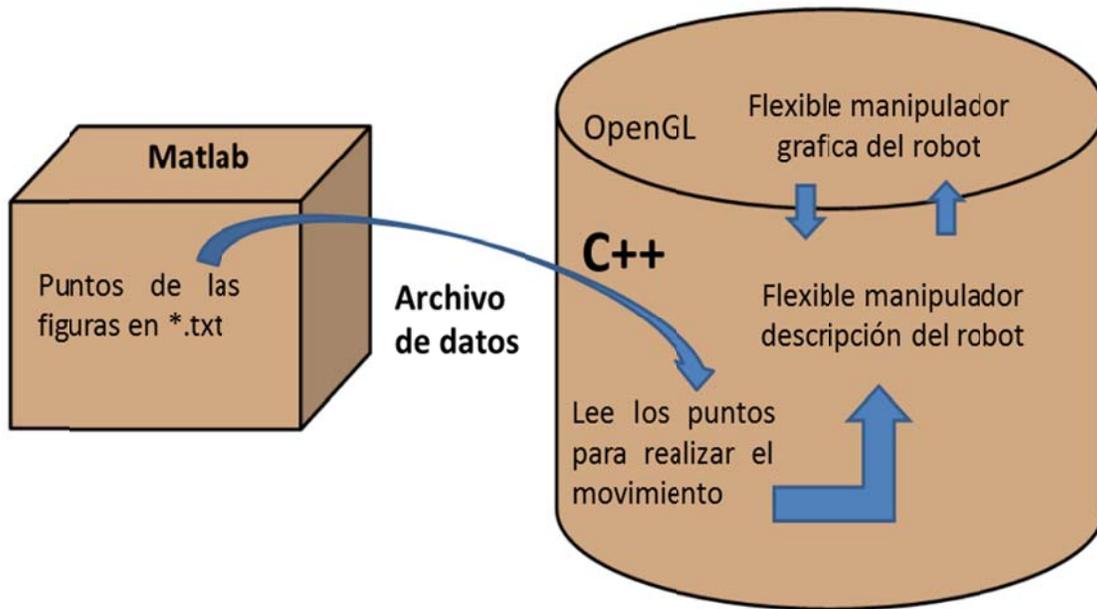


Figura 26. El desarrollo de interfaz multilenguaje

3.2.1 INTERFAZ ENTRE MATLAB Y C++

Cabe mencionar que no se pudo realizar directamente la interfaz ya que aún no existen las herramientas para la Versión del Visual Studio 2008. Por lo que se realizaron los cálculos en MatLab 2010 y se pasaron a un archivo *.txt para que en C se puedan leer y con ayuda del OpenGL se pueda graficar y realizar los movimientos con la ayuda de los datos de MatLab.

Por otro lado la interfaz entre Matlab y C++ que se utilizó no fue más que un archivo de texto *.txt, ya que no era necesario pasar la información en tiempo real entre ambos programas y el archivo de texto es la forma para pasar la información.

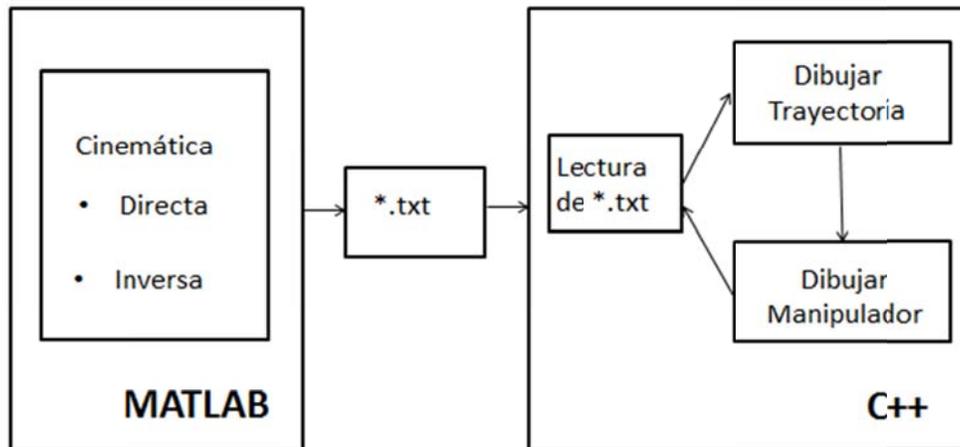


Figura 27. Interfaz entre MatLab y C++.

Utilizando estas herramientas se desarrolló la simulación en C++. Primero se realizan todos los cálculos necesarios en MatLab, hacemos la interface con C++, para finalmente simular con las librerías de OpenGL [40].

3.2.2 MATLAB

MatLab es un lenguaje de programación matemático, utilizado para implementar operaciones matemáticas computacionalmente complejas, de una más rápida que lenguajes tradicionales como C++.

Los modelos cinemáticos son implementados en MatLab, debido a que los resultados se obtienen en un menor tiempo a que el lenguaje C++.

En MatLab se obtienen la posición de los vástagos, los ángulos y posteriormente se calcula la posición del efector final de los dos actuadores.

Se analizó en el plano x, y el valor de $\theta_1=0$ obteniendo los valores de θ_2, θ_3 como se muestra en el código en base a la siguiente figura 28 desarrollándose las ecuaciones características del movimiento cinemático.

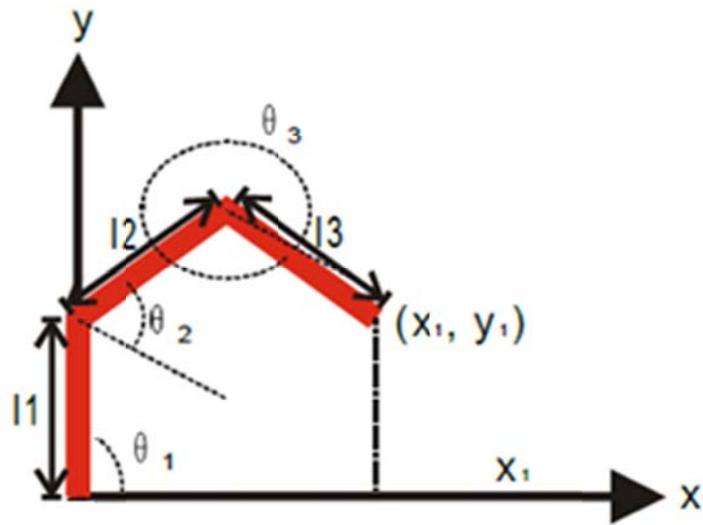


Figura 28. Análisis para poder obtener los θ_2 y θ_3 .

Como se muestra en las siguientes figuras 29 el círculo y en la figura 30 la línea para verificar que los datos sean correctos así obteniendo los puntos para guardarlos en *.txt

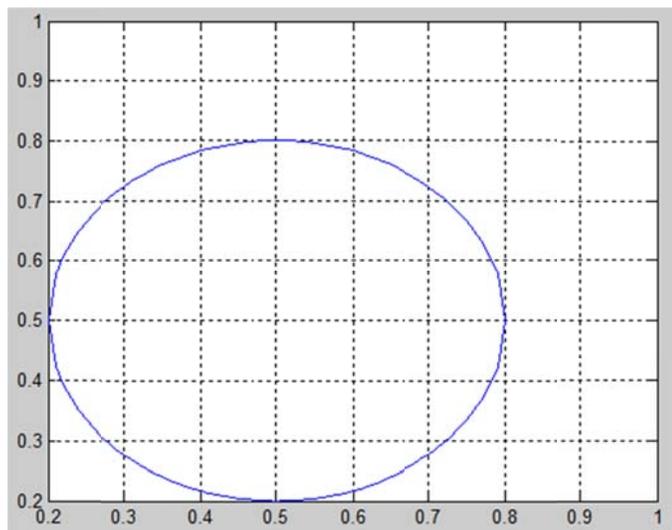


Figura 29. El círculo

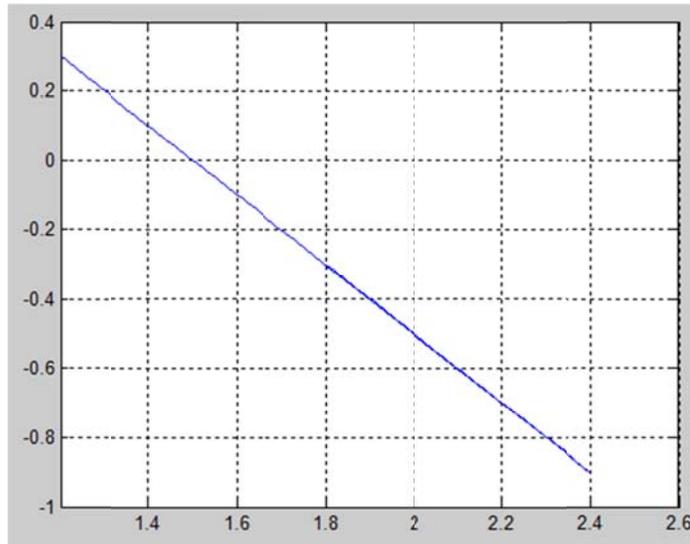


Figura 30. La línea

3.2.3 C++

El lenguaje con el que se va a programar es con C++ es un software orientado a objetos además de que tiene la facilidad de utilizar las interfaces de aplicación grafica (API), a través de las librerías graficas abiertas (OpenGL) [26]. Decisión en C++ para conseguir un ejecutable ligero, eficiente, que llama directamente los API's nativos de Windows. Como entorno de desarrollo y compilador se escogió Visual Studio 2008 ya que es sencillo, facilidad en uso de ciclos, subrutinas y los más importantes la posibilidad de poder usar OpenGL para el manejo de gráficos y visualizaciones.

El lenguaje C++ es usado para llamar a las rutinas de OpenGL. Y el archivo *.txt que es generado por MatLab., de esta manera se logra la simulación grafica con animación visual para mostrar las trayectorias del círculo, línea y el cuadrado.

En el programa principal se ponen las librerías y o include al iniciar el programa

```
//Encabezados necesarios para iniciar un programa en OpenGL
#include <windows.h>
#include <GL/glu.h>
#include <GL/glut.h>
#include <GL/glaux.h>
```

La función main, manda llamar a iniciar, manda desplegar el menú y por ultimo manda al Loop.

```
//Programa principal, y su lógica
int main(int argc, char **argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize(600,600);
    glutInitWindowPosition(100,100);
    glutCreateWindow("Titulo");
    init();
    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutMainLoop();
    return 0;
}
```

Partes del código.

En la parte de encabezados, se debe incluir los siguientes: *windows.h*, *gl.h*, *glu.h* y *glut.h*.

A continuación, se describe el uso de cada sección dentro de la rutina *main()*.

- *glutInit(int *argc, char **argv)*. Inicializa el proceso de GLUT para la interpretación de líneas de comandos. Esta función debe ser llamada antes de cualquier otra rutina GLUT.
- *glutInitDisplayMode(unsigned int mode)*. Especifica el modelo de color a utilizar, RGBA ó índice de color. Se puede especificar el tipo de buffer a utilizar en la ventana (doble ó sencillo); se puede también especificar si se desea que la ventana tenga asociado un grosor (depth), estarcido (stencil), muestreo múltiple, y/o búfer de acumulación. Los valores posibles de *mode* está definido por la combinación de los valores siguientes:

Colores:	GLUT_RGB ó GLUT_INDEX
Búfer:	GLUT_DOUBLE ó GLUT_SINGLE
Banderas de habilitación del búfer:	GLUT_DEPTH, GLUT_STENCIL ó GLUT_ACCUM

- *glutInitWindowSize(int width, int size)*. Especifica en pixeles, el tamaño de la ventana.

-
- *glutInitWindowPosition(int x, int y)*. Especifica la posición en la pantalla, para la esquina superior izquierda de la ventana.
 - *glutCreateWindow(char *string)*. Crea una ventana con un contexto OpenGL, utilizando el contenido de la cadena *string* como el título de la misma.
 - *init()*. Esta función es utilizada para la inicialización de parámetros del ambiente OpenGL.
 - *glutDisplayFunc(void (*func)(void))*. Especifica la función que será llamada cuando el contenido de la ventana necesite ser redibujado, como cuando es abierta inicialmente, al modificar el tamaño de la ventana, y cuando se llama a la función *glutPostRedisplay*.
 - *glutReshapeFunc(void (*func)(int width, int height))*. Especifica la función a ser llamada cuando la ventana modifique su tamaño. Los argumentos que espera la función son ancho y alto de la ventana.
 - *glutMainLoop()*. Esta la función más importante, pues la ventana creada no será mostrada hasta que se ejecute esta función, que crea un lazo, y que únicamente se regresa de la misma, cuando se sale del programa.

Archivos Include.

Los archivos cabecera que deben ser incluidos para el uso de las rutinas de OpenGL son: *gl.h*, *glu.h* y *glut.h*. El archivo *windows.h*, debe ser usado antes de las 3 anteriores en el caso de trabajar en sistema operativo Microsoft Windows, pues estos archivos hacen referencias a elementos que están declarados en el encabezado *windows.h*. Esto se puede obtener de la página <http://www.opengl.org>.

Se manejan otros para realizar el dibujo del brazo neumático los cuales son:

```
#include "variables.h"  
#include "luz.h"  
#include "base.h"  
#include "cilindro.h"  
#include "vastago.h"  
#include "impulso1.h"  
#include "impulso2.h"  
#include "impulso3.h"  
#include "brazo.h"  
#include "puntos.h"
```

Animación.

Para generar el efecto de animación, OpenGL maneja dos tipos de buffer: uno mantiene la imagen actual a mostrar en la pantalla, y el otro contiene la próxima imagen a mostrar. El buffer que contiene la próxima imagen, cuando termina de generar la imagen a mostrar, se lleva a cabo el intercambio de buffer, pasando la próxima imagen a ser la actual, y buffer de próxima imagen comienza a trabajar en la obtención de la siguiente imagen.

La función main que se llama automáticamente al iniciar la ejecución del programa.

```
void main(int argc, char **argv)
{
    pedir();
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA);
    glutInitWindowPosition(0,0);
    glutInitWindowSize(1275,735);
    glutCreateWindow("Flexible Manipulator Robot");
    HDC = wglGetCurrentDC();
    BuildFont();
    init();
    glutDisplayFunc(renderScene);
    glutKeyboardFunc(keyboard);
    glutSpecialFunc(keyboard2);
    glutReshapeFunc(changeSize);
    glutMainLoop();
}
```

3.2.4 OPENGL

OpenGL es una Librería grafica escrita originalmente en C que permite la manipulación de gráficos 2D.

OpenGL “Desde el punto de vista del programador OpenGL es una API para interactuar con dispositivos gráficos y aceleradores 2D” [12] [38].

OpenGL es un estándar sobre gráficos por computadora. Hoy en día es uno de los estándares gráficos más conocidos del mundo. OpenGL está disponible en www.opengl.org.

La representación de la imagen del robot, además del modelo matemático es necesario la manipulación de gráficos mucho más rápidos y eficientes se utilizó OpenGL que es una API para interactuar con dispositivos gráficos y aceleradoras 2D. Contiene varios comandos que nos ayudan a definir objetos. OpenGL es una librería gráfica. La desventaja que se puede observar es que no tiene funciones para el control de audio, red o control de entrada.

Gracias a la portabilidad de OpenGL nuestras aplicaciones podrán ejecutarse en una amplia variedad de arquitecturas y de soportes gráficos, sin que el resultado se vuelva inconsistente.

Debido a los cambios en el hardware gráfico, y a que OpenGL es básicamente un interface de abstracción del hardware, parece muy probable que se quede anticuado pronto, es decir, podríamos suponer que en cuanto aparezca una nueva prestación (por ejemplo en las aceleradores) OpenGL no sería capaz de proporcionar unas funciones al programador para que este pudiese hacer uso de estas nuevas características en sus programas. Pues bien, OpenGL se diseñó desde el principio para ser capaz de hacer frente a este problema, y gracias a sus mecanismos de extensión, las nuevas funcionalidades se pueden ir introduciendo sin problemas mientras que se respeta la compatibilidad con las versiones anteriores. [41]

En OpenGL se realizó varias funciones para poder ver más a detalle el movimiento del robot y verla en sus posiciones (x,y). Como se muestra en la figura 29 en este proyecto solamente se va a manejar en 2D donde muestra mover la cámara usando los comandos de transformación de vista de cualquier de las siguientes maneras *glTranslatef()* y *glRotatef()* y usando la rutina *gluLookAt()* de la librería de utilidades para definir una línea de visión. Esta rutina encapsula una serie de comandos de translación y rotación [42].

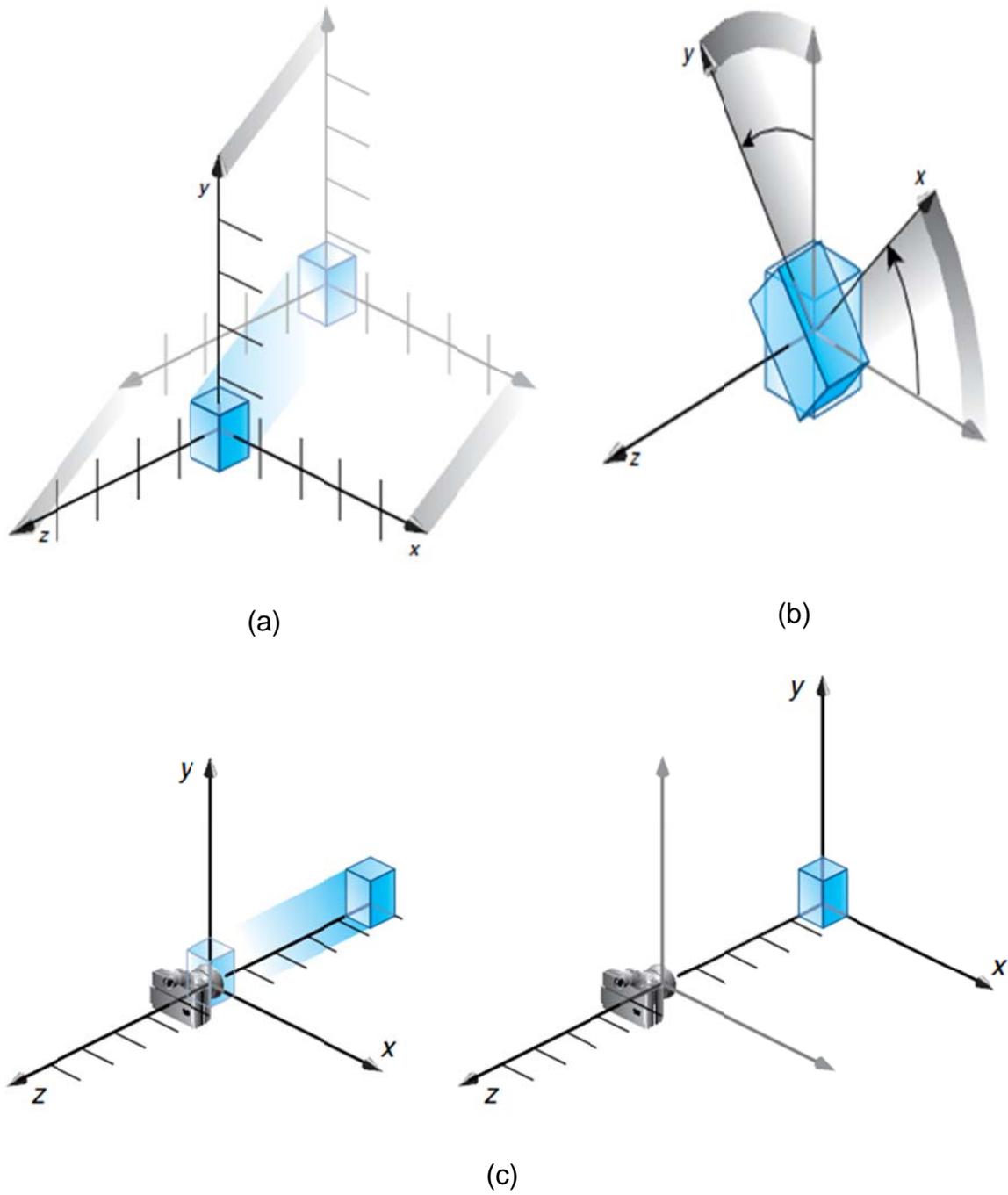


Figura 31. Muestra (a) la traslación,(b) rotación y (c) la vista del objeto

El diagrama de flujo nos muestra las operaciones ejecutadas dentro del sistema y la secuencia en que se ejecutan. Como se muestra en la figura 31.

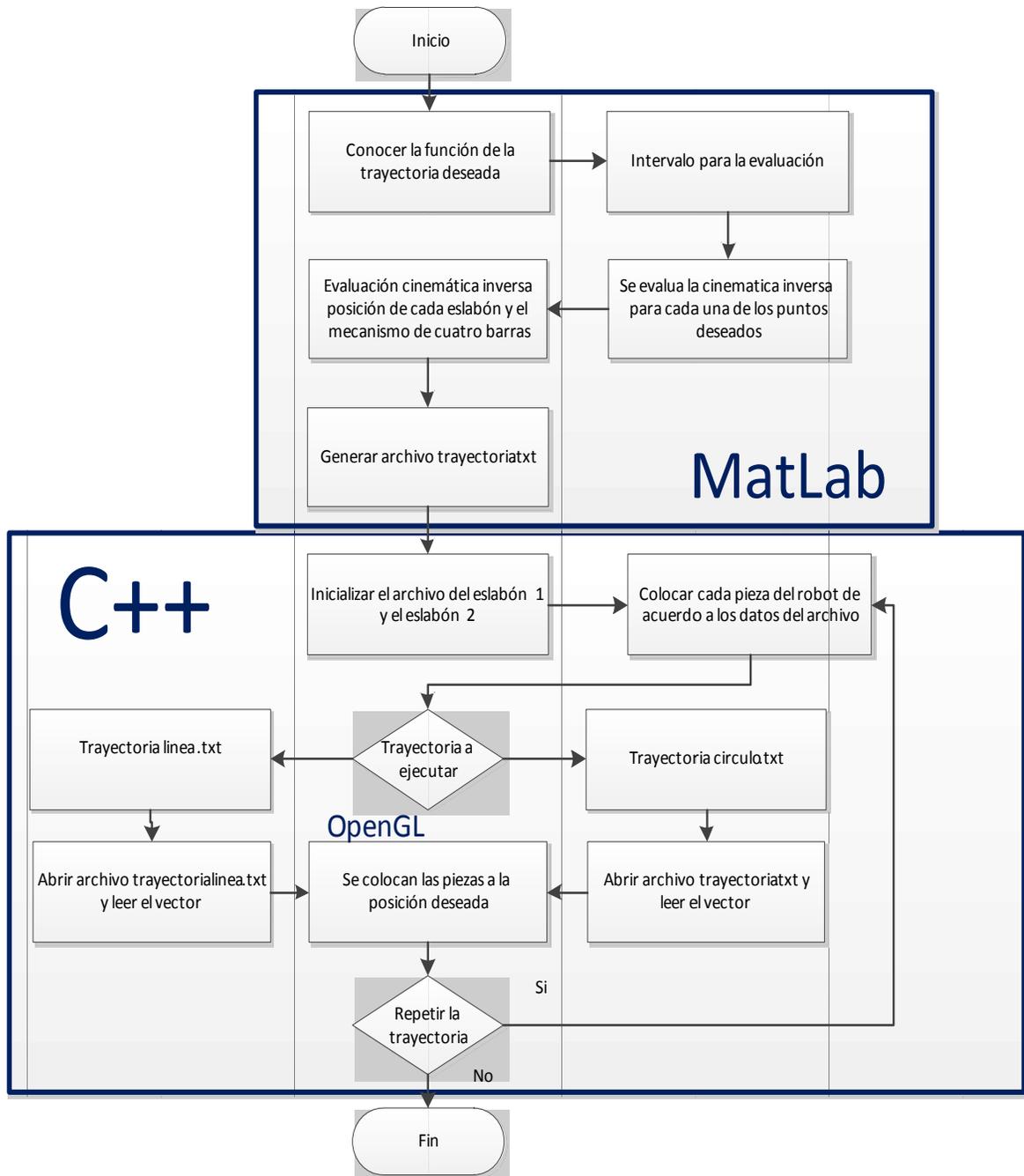


Figura 32. Diagrama de flujo

3.3 ESTRUCTURA DEL ROBOT

El un manipulador flexible con actuador neumático de dos grados de libertad se muestra en la siguiente figura 33.

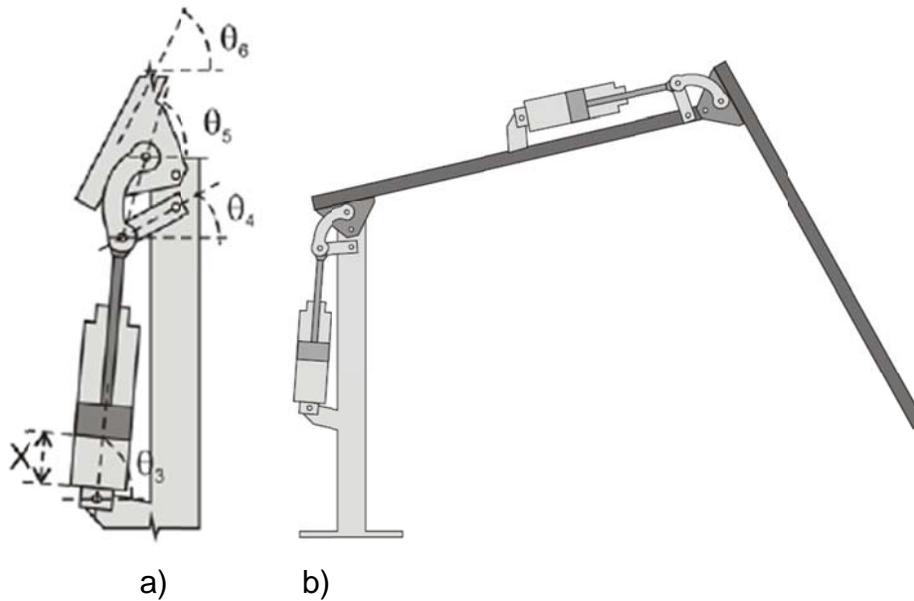


Figura 33. Manipulador flexible a) Variables del sistema b) vista general

Definición de las variables usadas en el sistema del manipulador flexible. [36]

Tabla 3.11 Variables del manipulador flexible.

Descripción	Unidades	in
Desplazamiento del vástago	Metros	X
Angulo del cilindro	Grados	θ_3
Angulo de la palanca impulsora	Grados	θ_4
Angulo de la barra de acoplamiento	Grados	θ_5
Angulo del brazo	Grados	θ_6

Las partes del manipulador se muestran en la siguiente figura 34 que es la estructura y el diseño:

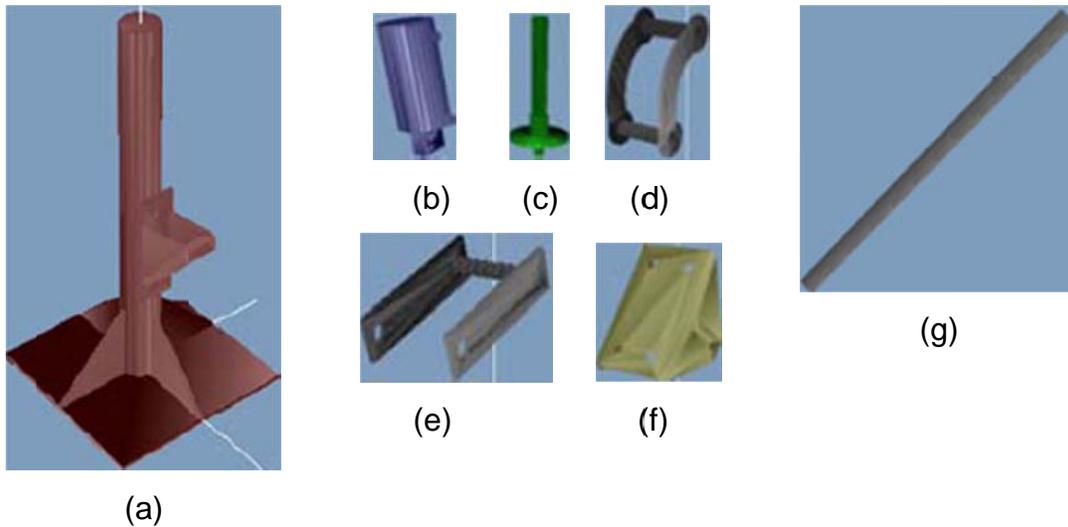


Figura 34. Objetos formados por primitivas OpenGL para el manipulador flexible. (a) Base del manipulador. (b) cilindro neumático. (c) barra (d) mecanismo impulsor. (e) Barra de manivela. (f) base del brazo. (g) Brazo.

Se muestra en la siguiente figura 35 ya el brazo ensamblado completo del robot manipulador flexible de dos grados de libertad, obtenido en OpenGL.

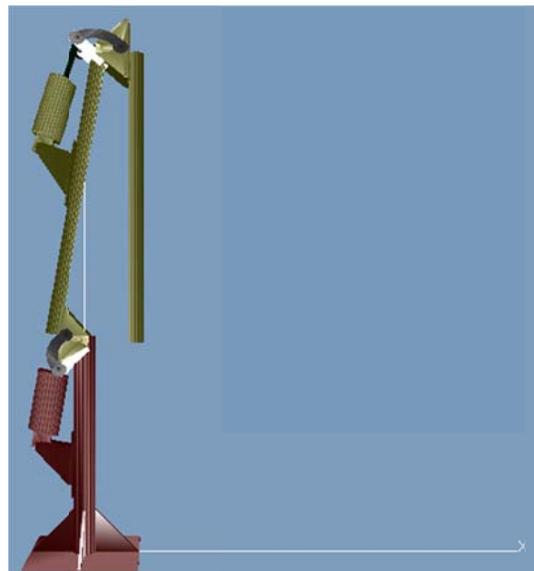


Figura 35. Robot manipulador flexible de dos grados de libertad.

La simulación nos ayuda a descubrir el comportamiento de la cinemática del manipulador, a postular teorías o hipótesis que expliquen el comportamiento y predecir el comportamiento futuro del manipulador.

Como resultado se obtendrá el software de la simulación en la que se mostrara el brazo del robot y su movimiento.

3.4 PROGRAMAS Y LIBRERÍAS DEL SOFTWARE

Para poder ejecutar el programa es necesario instalar en la maquina las Librerías gráficas, OpenGL.

Al instalar el software de Microsoft Visual Studio 2008 se deben de poner las librerías de OpenGL.

En C: \Archivos de programa\Microsoft Visual Studio 9.0\VC\lib



..\lib\opengl32.lib

..\lib\glu32.lib

..\lib\glaux.lib

En C: \Archivos de programa\Microsoft Visual Studio 9.0\VC\include



O

En C: \Archivos de programa\Microsoft Visual Studio 9.0\VC\include\gl

Se crea la carpeta gl y se pone en los programas es decir gl/gl.h

```
..\include\GL\gL.h
..\include\GL\glaux.h
..\include\GL\glu.h
```

La primera opción es mejor.

En C: \Windows\System32



En caso de que si existiera alguno ya en la ubicación debemos de conservar ambos archivos.

```
..\windows\system\glu32.dll
..\windows\system\opengl32.dll
```

La primera opción es mejor.

Tabla 3.12 Opciones del menú

Opción	Símbolo	Observaciones
Zoom	+	Hace más grande el brazo
Zoom	-	Hace más pequeño el brazo
Mover eje X	+	Mueve el brazo adelante respecto al eje x
Mover eje X	-	Mueve el brazo atrás respecto al eje x
Mover eje Y	+	Mueve el brazo arriba respecto al eje Y
Mover eje Y	-	Mueve el brazo abajo respecto al eje Y
Fig. solido	+	El brazo lo dibuja solido
Fig. líneas	-	El brazo lo dibuja en líneas
Trayectoria del circulo	+	Realiza la trayectoria del circulo
Trayectoria de la línea	-	Realiza la trayectoria de la línea
Girar Eje Y	+	Gira el brazo a la derecha del eje y
Girar Eje Y	-	Gira el brazo a la izquierda del eje y
Girar Eje X	+	Gira el brazo a delante del eje X
Girar Eje X	-	Gira el brazo atrás del eje X
Resetear	+	Deja el brazo en la última posición
Salir	-	Salir

4 PRUEBAS Y RESULTADOS

Los resultados del modelo de un sistema neumático compuesto por dos eslabones realizando trayectorias de un círculo y una línea. Que se llegaron a cabo en este simulador y el movimiento que siguió el robot de acuerdo al programa realizado.

En la siguiente figura 36 se observa la posición del brazo así como los ángulos y la posición del efector final (x, y) de cada uno de los eslabones y el valor de las thetas así como la rotación.



Figura 36. Posición inicial del manipulador

En las siguientes figuras se muestra la trayectoria a seguir del brazo del robot en el caso de la figura 37 muestra la trayectoria del círculo y en la figura 38 muestra la trayectoria de una línea.

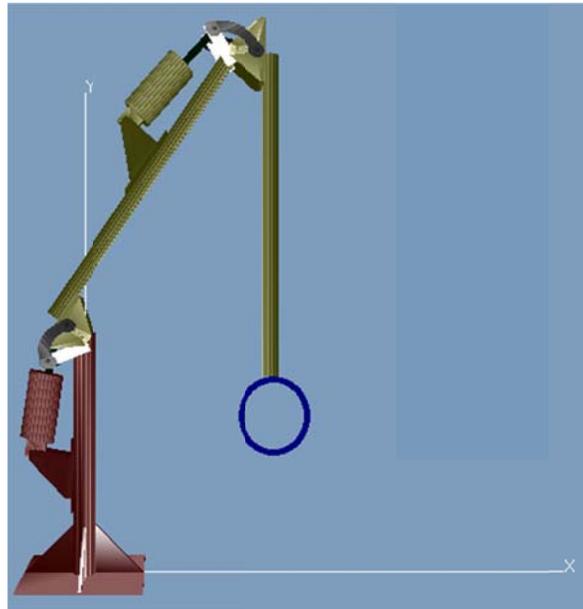


Figura 37. Trayectoria de un círculo

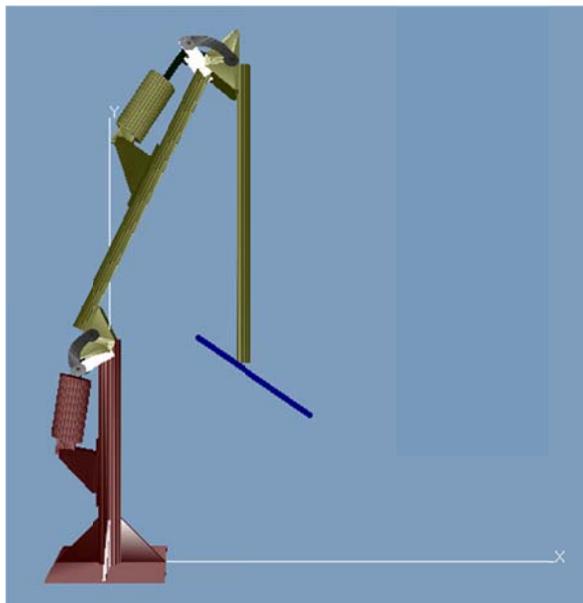


Figura 38. Trayectoria de una línea

Con la simulación se pretende diseñar y desarrollar un modelo computarizado de la cinemática con el propósito de entender el comportamiento de la cinemática de manera ideal. Donde podemos ver la posición de los dos actuadores, el ángulo de cada uno de los eslabones y la posición del efector final al trazar la trayectoria deseada.

El desarrollo de este software es muy importante ya que por medio de la simulación se pueden ver posibles fallas que podrían ocasionarse en el futuro, así como la realización de las trayectorias a seguir, además de que si se realiza físicamente puede realizar trabajos que no arriesguen la vida de los seres humanos además de que son ligeros, pequeños y con una relación potencia-peso, lo interesante es que es de menor costo y el mantenimiento es sencillo.

5 CONCLUSIONES

Se utilizaron como base algunas investigaciones importantes ya que solamente se contaba con un solo eslabón y en este trabajo se contemplan 2 grado el segundo eslabón así como las trayectorias.

Se ajustó la modelación de la cinemática directa e inversa de la arquitectura del robot, basados en otras investigaciones.

Se ha estudiado, comprendido y adaptado para dos grados de libertad el comportamiento de los elementos neumáticos del manipulador.

Se desarrolló el programa para la modelación y simulación gráfica donde se contempla la cinemática directa e inversa y la planeación de trayectorias utilizando C++, OpenGL y MatLab para un manipulador de dos grados.

En este trabajo se presenta el desarrollo de un simulador para planeación de trayectorias.

Las pruebas y resultados de las trayectorias se analizan mediante la simulación del manipulador.

La animación grafica nos ayuda a conocer el comportamiento que tendrá en forma física.

Aprendí el manejo de multilenguaje para simular el comportamiento de un robot manipulador de flexible con actuador neumático de dos grados de libertad

Darme la oportunidad de contribuir en esta propuesta de trabajo ya que fue una propuesta de una metodología para darle continuación al segundo eslabón.

Simular gráficamente un proceso permite implementar proyectos en menos tiempo y calidad y sobre todo identificar los problemas que puedan surgir.

5.1 TRABAJO FUTURO

Escribir documentos asociados a esta investigación para ser propuestos para su publicación en revistas.

Presentar la investigación en congresos.

Este trabajo puede servir de base para el desarrollo para tres grados de libertad.

Interesar a los programadores a desarrollar en esta metodología propuesta por los investigadores.

Aprovechar las diversas plataformas y librerías de programación.

En esta aproximación de la simulación del brazo neumático con dos grados de libertad, se trabajó con la cinemática no incluyendo la dinámica, sin embargo sería conveniente que se continúe con ello.

BIBLIOGRAFIA

- [1] Ramos, J.M.; Gorrostieta, E.; Vargas, E.; Pedraza, J.C.; Romero, R.J.; Aceves, M.A.; Canchola, S. Dynamic Behavior of a Pneumatic Manipulator with Two Degrees of Freedom. Autorizado y pendiente de imprimir. (PENDIENTE DE DEFINIR).
- [2] González, C.A.; Zúñiga, A.; Pedraza, C.; Gorrostieta, E.; Ramos, J.M.; Aceves, M.; Olmos C.A.; Santoyo, A.; Valerio S.; Rico, R.; Modeling and Simulation of the Dynamic Behavior applied to a Manipulator Robot. International Journal of Factory Automation. Robotica and Soft Computing, Issue 3, July 2009; ISSN 1828-6984.
- [3] Ramos, J.M.; Pedraza, J.C.; García, M.C.; Gorrostieta, E.; Ramos, C.A.; Delgado, M. Metodología para la simulación gráfica de aplicaciones mecatrónicas usando técnicas multilenguaje. VII Congreso Internacional en Innovación y Desarrollo Tecnológico, 2009. Autorizado y pendiente de imprimir. (PENDIENTE DE DEFINIR).
- [4] Ramos-Arreguin, J.M.; Pedraza-Ortega, C.; Gorrostieta, E.; Ramos, J.M.; Aceves, M.; Olmos C.A.; Santoyo, A.; Valerio S.; Rico, R.; Simulation Methodology for Mechatronic Applications using Multilanguage Techniques. 14th International Conference on Methods and Models in Automation and Robotics, 19 - 21 August 2009, Miedzyzdroje, Poland.
- [5] Jorge J. Feliu, Vicente Feliu, Member, IEEE, and Carlos Cerrada. Load Adaptive Control of Single-Link Flexible Arms Based on a New Modeling Technique. *IEEE Transactions on Robotics and Automation*, Vol. 15, NO.5 OCTOBER 1999

-
- [6] Publishers Preview IEEE Robotics and Automation Magazine, Edition 16244, Latest#:1, Vo. 17. No.4, ISSM 1070-9932 http://qtools.qmags.com/EditionPreview/p196/i5507/e16244/rev_1/16244.pdf
December 2010. P. 27
- [7] Wayne. J. Book. Recursive Lagrangian Dynamics of Flexible Manipulator Arms. Via Transformation Matrices. *International Journal of Robotics Research* Vol. 3, NO 3,1984.
- [8] Barrientos A. y Balaguer C., "Fundamentos de Robótica", Mc Graw Hill, Primera Edición, España, 1997.
- [9] W. J. Book and K. Oberfell. Practical models for practical flexible arms. Proceeding of the 2000 IEEE International Conference on Robotics & Automation, 2000.
- [10] J. Wang, D.J. Wang, P.R. Moore, and J. Pu. Modeling study, analysis and robust servocontrol of pneumatic cylinder actuator systems. *IEEE Proc. Control Theory Appl.*, 148(1):35–42, January 2001.
- [11] Rogerio Rodrigues dos Santos, Valder Steffen Jr. Optimal Path Planning and Task Adjustment For Cooperative Flexible Manipulator, ABCM Symposium Series in Mechatronics- Vol.3- pp.236-245, 2008
- [12] Antonio Barrientos, Luis Felipe Peñin, Carlos Balaguer, Rafael Aracil Santoja Fundamentos de Robótica por Barrientos. McGraw-Hill, 2ª edición, 2007

-
- [13] Ing. Juan Manuel Hernández Duran. Tesis Modelación de un robot neumático de dos grados de libertad., ITM, Maestría en Ingeniería mecánica 2006
- [14] Gonzalo Zabala. Robótica. 1ª. Ed. Banfield-Lomas de Zamora: Gradi, 2007 (Robotica_Guia_teoría_y_practica.pdf)
- [15] Historia del arte de la robótica: tipos de robotsroboticspot.com, <http://robotik-jilq.blogspot.com/2009/03/tipos-de-robots-2.html>. 2009 Consultada el 13 de diciembre de 2009
- [16] Nehmzow, U. Mobile Robotics: a practical introduction, Spring, Inglaterra,2000
- [17] Mark W. Spong, M. Vidyasgar, Robot Dynamics and Control, Wiley, 1989
- [18] <http://robotica2.galeon.com/> Consultada el 13 de diciembre de 2009.
- [19] Greene C. Los Robots, CONACYT, México. 1987
- [20] Caballero H, Diseño de agentes inteligentes, UTM, México,1999
- [21] <http://www.yorobot.com/> /, Ediciones Nowtilus, Febrero del 2006.
- [22] Roger Clarke, “The 1940 Laws of Robotics” <http://www.rogerclarke.com/SOS/Asimov.html> Consultado 6 de enero del 2011

-
- [23] Groover Mikell P, Weiss Mitchel, Ángel Roger N., Odrey Nicholas G. *Robótica Industrial: Tecnología, programación y aplicaciones*. Ed. Mx Graw Hill, México, 1995
- [24] K. S. Fu, R. C. González, C. S. G. Lee, *ROBÓTICA, Control, detección, visión e inteligencia*, México: Mc Graw Hill, 1990.
<http://es.scribd.com/doc/11502967/Robotica-Control-Deteccion-Vision-e-Inteligencia-Fu-Gonzalez-Lee> Consultada 18 febrero 2011
- [25] A. J. Koivo *Fundamentals for control of robotic manipulators*, Ed. Wiley. 1989. E.U. p2-11
- [26] http://www.geocities.com/loboduke2001/Tipos_robots.html7 Consultada el 13 de diciembre de 2009.
- [27] J. Ma. Angulo Rafael Avilés. *Cursos de Robótica*. Ed. Paraninfo.2005
- [28] *Libro blanco de la robótica: De la investigación al desarrollo tecnológico y aplicaciones futuras*. Editado por CEA-GTRob, 2007 (22681234-libro-robotica.pdf)
- [29] FEIRER, John L. *METALISTERIA ARTE Y CIENCIA DEL TRABAJO CON MERALES*, México. McGraw-Hill 1990.
- [30] REHG, James A. *INTRODUCTION TO ROBOTICS IN CIM SYSTEMS*. New Jersey. Prentice-Hall 2000
- [31] John J. Craig, *Robótica*, Pearson Prentice Hall, Tercera edición, 2006

-
- [32] <http://graphics.cs.ucdavis.edu/education/CAGDNotes/Reparameterizing-Bezier-Curves.pdf>]
- [33] Aníbal Ollero Baturone, Robótica Manipuladores y robots móviles, Marcombo S. A, Boixareu editores., 2001
- [34] Tesis GENERACIÓN AUTOMÁTICA DE TRAYECTORIAS PARA UN ROBOT MANIPULADOR UTILIZANDO PROCESAMIENTO DE IMÁGENES Ing. Jorge Alberto Soto Cajiga CIDESI, 2006
- [35] McKerrow,P.J. "introduction to robotics". Addison Wesley,1991.
- [36] CIINDET 2009 VII Congreso Internacional en Innovación y desarrollo tecnológico, 7 al 9 octubre de 2009, Cuernavaca , Morelos. Metodología para la simulación grafica de aplicaciones mecatronicas usando técnicas multilinguaje J.M. Ramos-Arreguin , J.C. Pedraza Ortega, M.C. García López, C.A. Ramos Arreguin, E. Gorrostieta Hurtado, señor member, IEEE, M. Delgado Rosas., 2009],
- [37] http://usuarios.lycos.es/tutoriales/andromeda/Tutorial_0_introducción_General.htm , 2009
- [38] Munshi, Aftab, et. al. OpenGL Programming Guide. Addison Wesley, 2009. USA.
- [39] Lever, Nick. Real-time 3D Character Animation with Visual C++. Wiley Interscience, 2005. USA.

-
- [40] <http://www.mecatronica.net/emilio/papers/4cnm-32.pdf> Consultada el 13 de diciembre de 2009.
- [41] Interactive Computer Graphics. Angel, Edward. Addison Wesley 4° (2005) ed. De Shreiner, Woo, Neider y Davis.
- [42] OpenGL Programming Guide Seventh Edition. The official Guide to Learning OpenGL, version 3.0 and 3.1. Dave Shreiner, The khronos OpenGL ARB Workin Group, Copyright 2010 Education, Inc.]
- [43] Holzbecher, Ekkehard. Environmental Modeling with MATLAB. Springer, 2007. USA.
- [44] Gamiño M. Pedraza JC, Gorrostieta E. MatLab-C++ Interface for a Flexible Arm Manipulator Simulation using Multilenguaje Techniques, Fifth Mexican International Conference on Artificial Intelligence (MICAI), 2006.

GLOSARIO

Cinemática. Rama de la mecánica que estudia los movimientos de los cuerpos, con independencia de las fuerzas que los producen.

Cadena cinemática. Sucesión de eslabones relacionados entre sí para transmitir movimiento.

Coordenada. Cada una de las líneas que sirven para determinar la posición de un punto, y de los ejes y planos a que se refieren dichas líneas.

Coordenadas generalizadas. Son cada uno de los parámetros, independientes entre sí, que determinan la configuración del sistema respecto a un eslabón fijo.

Dinámica. Rama de la mecánica que estudia las relaciones de las fuerzas y los movimientos.

Efecto final. Último eslabón de un manipulador robótico, diseñado para realizar una tarea específica.

Eslabón. Todos y cada uno de los elementos que forman la cadena cinemática de un manipulador robótico.

Fuerza. Acción o influencia capaz de modificar el estado de movimiento de un cuerpo.

Geometría. Disciplina matemática que tiene por objeto es estudio riguroso del espacio y de las formas.

Grados de libertad. Cantidad de coordenadas generalizadas que determinan completamente la configuración del mecanismo.

Jacobiano. Representación de la geometría de los elementos de un mecanismo en el tiempo.

Manipulador robótico. Mecanismo reprogramable.

Marcos de referencia. Ejes de coordenadas con respecto a los cuales se describen los eslabones de un manipulador.

Matriz de transformación. Descripción de un par cinemática de acuerdo a la metodología de Denavit-Hartenberg.

Robot. Conjunto de eslabones seriados automáticos y programables, capaces de realizar una tarea.

Robótica. Conjunto de técnicas utilizadas para el diseño y construcción de robots industriales y la puesta en práctica de sus aplicaciones.

Trayectoria. Lina descrita por un punto material en movimiento.

Vector de posición. Vector que ubica la posición del centro de masa de un eslabón.

ANEXO A CODIGO DEL PROGRAMA

```
////////////////////////////////////
// Universidad Autónoma de Querétaro
// Facultad de Informática
// Maestría en ingeniería de software distribuido
// Proyecto de tesis: Modelación y simulación de la planeación de
// trayectorias para un manipulador flexible con actuador neumático de
// dos grados de libertad.
// Nombre: Robot2.cpp
// Director de Tesis: Dr. Juan Manuel Ramos Arreguín
// Alumna: María Laura Gutierrez Martinez
// Modificación del 20 de Junio
////////////////////////////////////

#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <math.h>
#include "glut.h"
#include <windows.h>
#include "variables.h"
#include "luz.h"
#include "base.h"
#include "cilindro.h"
#include "vastago.h"
#include "impulso1.h"
#include "impulso2.h"
#include "impulso3.h"
#include "brazo.h"
#include "puntos.h"

FILE *animation;

////menu
HDC          hDC=NULL;           // Private GDI Device Context
HGLRC        hRC=NULL;           // Permanent Rendering Context
HWND         hWnd=NULL;         // Holds Our Window Handle
HINSTANCE     hInstance;         // Holds The Instance Of The Application

GLuint  base;                    // Base Display List For The Font Set
GLfloat rot;                     // Used To Rotate The Text

bool    keys[256];               // Array Used For The Keyboard Routine
bool    active=TRUE;             // Window Active Flag Set To TRUE By Default
bool    fullscreen=TRUE;        // Fullscreen Flag Set To Fullscreen Mode By Default

GLYPHMETRICSFLOAT gmf[256];     // Storage For Information About Our Outline Font Characters

GLvoid BuildFont(GLvoid)        // Build Our Bitmap Font
{
    HFONT font;                  // Windows Font ID
    base = glGenLists(256);      // Storage For 256 Characters
    font = CreateFont( -12,      // Height Of Font
        0,                      // Width Of Font
        0,                      // Angle Of Escapement
        0,                      // Orientation Angle
        FW_BOLD,                // Font Weight
        FALSE,                  // Italic
        FALSE,                  // Underline
        FALSE,                  // Strikeout
        ANSI_CHARSET,           // Character Set Identifier
        OUT_TT_PRECIS,          // Output Precision
        CLIP_DEFAULT_PRECIS,    // Clipping Precision
```

```

ANTIALIASED_QUALITY, // Output Quality
FF_DONTCARE|DEFAULT_PITCH, // Family And Pitch
"Verdana"); // Font Name

SelectObject(hDC, font); // Selects The Font We Created
wglUseFontOutlines(hDC, // Select The Current DC
0, // Starting Character
255, // Number Of Display Lists To Build
base, // Starting Display Lists
0.0f, // Deviation From The True Outlines
0.2f, // Font Thickness In The Z Direction
WGL_FONT_POLYGONS, // Use Polygons, Not Lines
gmf); // Address Of Buffer To Recieve Data
}

GLvoid KillFont(GLvoid) // Delete The Font
{
    glDeleteLists(base, 256); // Delete All 256 Characters
}
GLvoid glPrint(const char *fmt, ...) // Custom GL "Print" Routine
{
    float length=0; // Used To Find The Length Of The Text
    char text[256]; // Holds Our String
    va_list ap; // Pointer To List Of Arguments
    if (fmt == NULL) // If There's No Text
        return; // Do Nothing
    va_start(ap, fmt); // Parses The String For Variables
    vsprintf(text, fmt, ap); // And Converts Symbols To Actual Numbers
    va_end(ap); // Results Are Stored In Text
    for (unsigned int loop=0; loop<(strlen(text)); loop++) // Loop To Find Text Length
    {
        length+=gmf[text[loop]].gmfCellIncX; // Increase Length By Each Characters Width
    }
    glTranslatef(-length/2, 0.0f, 0.0f); // Center Our Text On The Screen

    glPushAttrib(GL_LIST_BIT); // Pushes The Display List Bits
    glListBase(base); // Sets The Base Character to 0
    glCallLists(strlen(text), GL_UNSIGNED_BYTE, text); // Draws The Display List Text
    glPopAttrib(); // Pops The Display List Bits
}

void pedir()
{
    theta[4]=theta[0]=0;
    theta[5]=theta[1]=0;
    theta[6]=theta[2]=0;
    theta[7]=theta[3]=0;
}

void spinDisplay()
{
    char *p,c,s[12];
    int j;
    float dato[11];

    c=fgetc(animation);
    p=s;
    for(j=0;j<11;j++)
    {
        s[0]=0;
        while(!strchr("0123456789+-.",c) && j!=11 && !feof(animation))
            c=fgetc(animation);
        while(strchr("0123456789+-.",c) && !feof(animation))
        {
            *p++=c;
            c=fgetc(animation);
        }
        *p=0;
        p=s;
        dato[j]=atof(s);
    }
}

```

```

    }
    if(feof(animation))
    {
        fclose(animation);
        glutIdleFunc(NULL);
    }
    else
    {
        My1=dato[1];
        theta[0]=dato[2];
        theta[1]=dato[3];
        theta[2]=dato[4];
        theta[3]=dato[5];
        My2=dato[6];
        theta[4]=dato[7];
        theta[5]=dato[8];
        theta[6]=dato[9];
        theta[7]=dato[10];
        glutPostRedisplay();
    }
}

void puntos_tray()
{
    char *p,c,s[12];
    int j;
    float dato[11];
    FILE *fid;

    glBegin(GL_POINTS);
    fid=fopen("trayectoria.txt","r");
    c=fgetc(fid);
    p=s;

    while(!feof(fid))
    {
        for(j=0;j<3;j++)
        {
            s[0]=0;
            while(!strchr("0123456789+-.",c) && j!=11 && !feof(fid))
                c=fgetc(fid);
            while(strchr("0123456789+-.",c) && !feof(fid))
            {
                *p++=c;
                c=fgetc(fid);
            }
            *p=0;
            p=s;
            dato[j]=atof(s);
        }
        tx=dato[0];
        tz=dato[1];
        ty=dato[2];
        glVertex3f(-tx, ty, tz);
    }
    fclose(fid);
    glEnd();
}

static void init ()
{
    init_light();
    glEnable(GL_DEPTH_TEST);
    glShadeModel(GL_SMOOTH);

    startList=glGenLists(9);

    //////////// 1. SOPORTE DEL SISTEMA
    glNewList(startList, GL_COMPILE);
    base7();
}

```

```

glEndList();

////////// 2. CILINDRO DEL ACTUADOR NEUMATICO
glNewList(startList+1, GL_COMPILE);
    cilindro0();
    cilindro1();
    cilindro2();
    cilindro3();
    cilindro4();
    cilindro5();
    cilindro6();
glEndList();

////////// 3. Vastago
glNewList(startList+2, GL_COMPILE);
    vastago0();
glEndList();

////////// 4. IMPULSO 1
glNewList(startList+3, GL_COMPILE);
    impulso10();
    impulso11();
    impulso12();
glEndList();

////////// 5. IMPULSO 2
glNewList(startList+4, GL_COMPILE);
    impulso20();
    impulso21();
    impulso22();
    impulso23();
glEndList();

////////// 6. IMPULSO 3
glNewList(startList+5, GL_COMPILE);
    impulso30();
    impulso31();
    impulso32();
    impulso33();
    impulso34();
glEndList();

////////// 7. BRAZO FLEXIBLE
glNewList(startList+6, GL_COMPILE);
    brazo0();
glEndList();

////////// 8. SOPORTE DEL CILINDRO
glNewList(startList+7, GL_COMPILE);
    base0();
    base5();
    base6();
    base8();
glEndList();

////////// 9. SOPORTE DE LA BASE DEL ROBOT
glNewList(startList+8, GL_COMPILE);
    base1();
    base2();
    base3();
    base4();
    base9();
glEndList();

////////// 10. SE DIBUJA LA TRAYECTORIA DESEADA
glNewList(startList+9, GL_COMPILE);
    puntos_tray();
glEndList();

```

////////// 10. MENU

```
glNewList(startList+10, GL_COMPILE);
glBegin(GL_QUADS);
  SetMaterial(mat_specularBLUE, mat_ambientBLUE, mat_diffuseBLUE, mat_shininessBLUE);

  glColor3f(1.0, 0.0, 0.0);
  glVertex3f(0.0f, 0.0f, 0.0f);
  glVertex3f(250.0f, 0.0f, 0.0f);
  glVertex3f(250.0f, 30.0f, 0.0f);
  glVertex3f(0.0f, 30.0f, 0.0f);

  glVertex3f(0.0f, 0.0f, -105.0f);
  glVertex3f(250.0f, 0.0f, -105.0f);
  glVertex3f(250.0f, 30.0f, -105.0f);
  glVertex3f(0.0f, 30.0f, -105.0f);
  SetMaterial(mat_specularWHITE, mat_ambientWHITE, mat_diffuseWHITE, mat_shininessWHITE);
  //tapas
  glColor3f(0.0, 1.0, 0.0);
  glVertex3f(0.0f, 0.0f, 0.0f);
  glVertex3f(250.0f, 0.0f, 0.0f);
  glVertex3f(250.0f, 0.0f, -105.0f);
  glVertex3f(0.0f, 0.0f, -105.0f);

  glVertex3f(250.0f, 30.0f, 0.0f);
  glVertex3f(0.0f, 30.0f, 0.0f);
  glVertex3f(0.0f, 30.0f, -105.0f);
  glVertex3f(250.0f, 30.0f, -105.0f);

  SetMaterial(mat_specularRED, mat_ambientRED, mat_diffuseRED, mat_shininessRED);
  glVertex3f(0.0f, 0.0f, 0.0f);
  glVertex3f(0.0f, 30.0f, 0.0f);
  glVertex3f(0.0f, 30.0f, -105.0f);
  glVertex3f(0.0f, 0.0f, -105.0f);

  glVertex3f(250.0f, 0.0f, 0.0f);
  glVertex3f(250.0f, 30.0f, 0.0f);
  glVertex3f(250.0f, 30.0f, -105.0f);
  glVertex3f(250.0f, 0.0f, -105.0f);

  glEnd();
glEndList();
```

////////// 10.

```
glNewList(startList+11, GL_COMPILE);
glBegin(GL_QUADS);
  SetMaterial(mat_specularRED, mat_ambientRED, mat_diffuseRED, mat_shininessRED);

  glColor3f(1.0, 0.0, 0.0);
  glVertex3f(0.0f, 0.0f, 0.0f);
  glVertex3f(250.0f, 0.0f, 0.0f);
  glVertex3f(250.0f, 30.0f, 0.0f);
  glVertex3f(0.0f, 30.0f, 0.0f);
  glVertex3f(0.0f, 0.0f, -105.0f);
  glVertex3f(250.0f, 0.0f, -105.0f);
  glVertex3f(250.0f, 30.0f, -105.0f);
  glVertex3f(0.0f, 30.0f, -105.0f);
  SetMaterial(mat_specularWHITE, mat_ambientWHITE, mat_diffuseWHITE, mat_shininessWHITE);
  //tapas
  glColor3f(0.0, 1.0, 0.0);
  glVertex3f(0.0f, 0.0f, 0.0f);
  glVertex3f(250.0f, 0.0f, 0.0f);
  glVertex3f(250.0f, 0.0f, -105.0f);
  glVertex3f(0.0f, 0.0f, -105.0f);

  glVertex3f(250.0f, 30.0f, 0.0f);
  glVertex3f(0.0f, 30.0f, 0.0f);
  glVertex3f(0.0f, 30.0f, -105.0f);
  glVertex3f(250.0f, 30.0f, -105.0f);
```

```

        glVertex3f(0.0f,0.0f ,0.0f);
        glVertex3f(0.0f,30.0f ,0.0f);
        glVertex3f(0.0f,30.0f ,-105.0f);
        glVertex3f(0.0f,0.0f ,-105.0f);

        glVertex3f(250.0f,0.0f ,0.0f);
        glVertex3f(250.0f,30.0f ,0.0f);
        glVertex3f(250.0f,30.0f ,-105.0f);
        glVertex3f(250.0f,0.0f ,-105.0f);

        glEnd();
    glEndList();

    glNewList(startList+12, GL_COMPILE);
    SetMaterial(mat_specularGREEN, mat_ambientGREEN, mat_diffuseGREEN,
    mat_shininessGREEN);

    glColor3f(1.0,1.0,0.0);
    glScalef(20,20,20);

    glTranslatef(15,2.5,5);
    glRotatef(-15,0.0,1.0,0.0);
    glRotatef(0,1.0,0.0,0.0);
    glPrint("Mover Eje X (+/-)",10);
    glTranslatef(-3.4,1.7,0);
    glPrint("Zoom (+/-)",10);
    glTranslatef(-3.3,-3.4,0);
    glPrint("Mover Eje Y (+/-)",10);
    glTranslatef(-4.2,-1.7,0);
    glPrint("Fig solido/lineas(+/-)",10);
    glTranslatef(-4.7,-1.7,0);
    glPrint("Tray circulo/linea (+/-)",10);
    glTranslatef(-5,-1.0,1.8);
    glPrint("Girar Eje Y (+/-)",10);
    glTranslatef(-4.0,-1.7,0);
    glPrint("Girar Eje X(+/-)",10);
    glTranslatef(-4,-1.7,0);
    glPrint("Resetear/Salir(+/-)",10);

    glEndList();

    glNewList(startList+13, GL_COMPILE);
    SetMaterial(mat_specularGREEN, mat_ambientGREEN, mat_diffuseGREEN,
    mat_shininessGREEN);

    glColor3f(1.0,1.0,0.0);
    glScalef(20,20,20);
    glTranslatef(-13,17,6);
    glPrint("Manipulador flexible",1);

    glEndList();
}
void changeSize(int w, int h)
{
    // Prevent a divide by zero, when window is too short
    // (you cant make a window of zero width).
    if(h == 0)
        h = 1;

    // Reset the coordinate system before modifying
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();

    // Set the viewport to be the entire window
    glViewport(0, 0, w, h);

    // Set the correct perspective.
    gluPerspective(45,1.0* w / h,1,2500);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    gluLookAt(300.0,300.0,900.0,
        0.0,0.0,0.0,
        0.0f,1.0f,0.0f);
}

```

```

}

void renderBitmapCharacher(float x, float y, float z, void *font,char *string)
{
    char *c;
    glRasterPos3f(x, y,z);
    for (c=string; *c != '\0'; c++)
    {
        glutBitmapCharacter(font, *c);
    }
}

void renderScene(void)
{
    float z,y;
    float z1,y1,x1;
    float z2,y2,x2;
    float z3,y3,x3;
    char s[100];
    float pi=3.14159;
    float L1=0.5;
    float L2=0.65;
    float L3=0.65;
    float theta1,theta2,theta3;

    theta1=0;
    //theta1=Rotacion*pi/180;
    theta2=theta[3]*pi/180;
    theta3=theta[7]*pi/180;

    x1=0;
    y1=L1;
    z1=0;

    x2=L2*cos(theta2);
    y2=L1+L2*sin(theta2);
    z2=x2/tan(theta1);

    x3=(L2*cos(theta2)+L3*cos(theta3));
    y3=L1+L2*sin(theta2)+L3*sin(theta3);
    z3=x3/tan(theta1);

    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    SetMaterial(mat_specularWHITE, mat_ambientWHITE, mat_diffuseWHITE, mat_shininessWHITE);

    sprintf(s,"x2=%.4f",x2);renderBitmapCharacher(169,280,0,(void *)font[2],s);
    sprintf(s,"y2=%.4f",y2);renderBitmapCharacher(168,290,0,(void *)font[2],s);
    sprintf(s,"theta1=%.1f",0);renderBitmapCharacher(165,300,0,(void *)font[2],s);
    sprintf(s,"theta2=%.1f",theta[3]);renderBitmapCharacher(164,310,0,(void *)font[2],s);
    sprintf(s,"theta3=%.1f",theta[7]);renderBitmapCharacher(163,320,0,(void *)font[2],s);
    sprintf(s,"x3=%.4f",x3);renderBitmapCharacher(162,330,0,(void *)font[2],s);
    sprintf(s,"y3=%.4f",y3);renderBitmapCharacher(161,340,0,(void *)font[2],s);

    glPushMatrix();
    glCallList(startList+12);
    glPopMatrix();

    glPushMatrix();
    glCallList(startList+13);
    glPopMatrix();

    glPushMatrix();
    glRotatef(-15,0,0,1,0,0,0);
    glRotatef(-12,1,0,0,0,0,0);
    glTranslatef(200,33,0);
    glCallList(startList+10);
    glTranslatef(0,33,0);
    glCallList(startList+10);
    glTranslatef(0,-66,0);

```

```

        glVertex3f(0,0,0);
        glVertex3f(1,0,0);
        glVertex3f(0,0,0);
        glVertex3f(0,1,0);
        glEnd();
        sprintf(s,"X");renderBitmapCharacher(1,0,0,(void *)font[2],s);
        sprintf(s,"Y");renderBitmapCharacher(0,1,0,(void *)font[2],s);

        // Se dibujan los ejes principales
        glBegin(GL_LINES);
            glVertex3f(0,0,0);
            glVertex3f(1,0,0);
            glVertex3f(0,0,0);
            glVertex3f(0,1,0);
        glEnd();

        // Se dibuja la base del robot
        glPushMatrix();
            SetMaterial(mat_specularRED, mat_ambientRED, mat_diffuseRED, mat_shininessRED);
            glCallList(startList+8);
            glCallList(startList+9);
        glPopMatrix();

        // Se rota todo el sistema
        glPushMatrix();
            glRotatef(90,0,1,0);
            glCallList(startList+9);

        // Se dibujan los ejes principales
        glBegin(GL_LINES);
            glVertex3f(0,0,0);
            glVertex3f(0,1,0);
        glEnd();

        // Se dibuja el primer eslabon
        glPushMatrix();
            SetMaterial(mat_specularRED, mat_ambientRED, mat_diffuseRED,
            mat_shininessRED);
            glCallList(startList);
            glCallList(startList+7);
            glPushMatrix();// Se dibuja el cilindro
                glTranslatef(0,L1/2,-0.1025);
                glScalef(1,1.3,1);
                glRotatef(90,0,1,0);
                glRotatef(4-theta[0],0,0,1);
                glCallList(startList+1);
            glPopMatrix();
            glPushMatrix(); // Movimiento del vastago
                SetMaterial(mat_specularGREEN, mat_ambientGREEN,
                mat_diffuseGREEN, mat_shininessGREEN);
                glTranslatef(0,L1/2,-0.1025);
                glRotatef(4-theta[0],1,0,0);
                glTranslatef(0,0.023276+My1,0);
                glCallList(startList+2);

```

```

        glPopMatrix();
glPopMatrix();
glPushMatrix();    // Movimiento del impulsor 1
    SetMaterial(mat_specularWHITE, mat_ambientWHITE, mat_diffuseWHITE,
mat_shininessWHITE);
    glTranslatef(0,L1-0.03,0.0);
    glPushMatrix();
        glRotatef(90,0,0,1);
        glRotatef(90,0,1,0);
        glRotatef(theta[1],0,1,0);
        glCallList(startList+3);

glPopMatrix();
SetMaterial(mat_specularGRAY, mat_ambientGRAY, mat_diffuseGRAY,
mat_shininessGRAY);
glPushMatrix();
    y=0.075*cos(theta[1]*3.1415/180);
    z=0.075*sin(theta[1]*3.1415/180);
    glScalef(1.5,0.8,1);
    glTranslatef(0.0,y-0.01,-z);
    glRotatef(180,0,1,0);
    glRotatef(-45+theta[2],1,0,0);
    glRotatef(180,0,0,1);
    glRotatef(90,1,0,0);
    glCallList(startList+4);

glPopMatrix();
glPopMatrix();
glPushMatrix();    // Movimiento del impulsor 2
    SetMaterial(mat_specularYELLOW, mat_ambientYELLOW, mat_diffuseYELLOW,
mat_shininessYELLOW);
    glTranslatef(0.0,L1-0.01,0.0);
    glRotatef(-theta[3],1,0,0);
    glCallList(startList+5);
    glTranslatef(0,0.06,0);
    glCallList(startList+6);
    //////////////// SEGUNDO ESLABON
    //////////////// Se dibuja la base del segundo eslabón
    glRotatef(90,1,0,0);
    glTranslatef(0,0.17,-0.01);
    glCallList(startList+7);
    glPushMatrix();// Se dibuja el cilindro
        glTranslatef(0,L1/2,-0.1025);
        glScalef(1,1.3,1);
        glRotatef(90,0,1,0);
        glRotatef(4-theta[4],0,0,1);
        glCallList(startList+1);

glPopMatrix();
glPushMatrix();    // Movimiento del vastago
    SetMaterial(mat_specularGREEN, mat_ambientGREEN,
mat_diffuseGREEN, mat_shininessGREEN);
    glTranslatef(0,L1/2,-0.1025);
    glRotatef(3-theta[4],1,0,0);
    glTranslatef(0,0.01+My2,0);
    glCallList(startList+2);

glPopMatrix();
glPushMatrix();    // Movimiento del impulsor 1
    SetMaterial(mat_specularWHITE, mat_ambientWHITE,
mat_diffuseWHITE, mat_shininessWHITE);
    glTranslatef(0,L1-0.03,0.0);
    glPushMatrix();
        glRotatef(90,0,0,1);
        glRotatef(90,0,1,0);
        glRotatef(theta[5]+10,0,1,0);
        glCallList(startList+3);

glPopMatrix();
SetMaterial(mat_specularGRAY, mat_ambientGRAY, mat_diffuseGRAY,
mat_shininessGRAY);
glPushMatrix();
    y=0.075*cos(theta[5]*3.1415/180);
    z=0.075*sin(theta[5]*3.1415/180);
    glScalef(1.5,0.8,1);

```

```

        glTranslatef(0.0,y-0.01,-z);
        glRotatef(180,0,1,0);
        glRotatef(-45+theta[6],1,0,0);
        glRotatef(180,0,0,1);
        glRotatef(90,1,0,0);
        glCallList(startList+4);
    glPopMatrix();
glPopMatrix();
glPushMatrix(); // Movimiento del impulsor 2
    SetMaterial(mat_specularYELLOW, mat_ambientYELLOW,
    mat_diffuseYELLOW, mat_shininessYELLOW);
    glTranslatef(0.0,L1-0.01,0.0);
    glRotatef(-theta[7],1,0,0);
    glCallList(startList+5);
    glTranslatef(0,0.06,0);
    glCallList(startList+6);
glPopMatrix();
glPopMatrix();
glPopMatrix();
glutSwapBuffers();
}

void keyboard2(int key, int x, int y){
    switch(key)
    {
    case GLUT_KEY_UP:
        subir+=33;
        if(subir>33*7) subir-=33;
        glutPostRedisplay();
        break;
    case GLUT_KEY_DOWN:
        subir-=33;
        if(subir<0) subir=0;
        glutPostRedisplay();
        break;
    }
}

void keyboard(unsigned char key, int x, int y)
{
    switch(key)
    {
        case '+':
            switch((int)subir/33){
                case 7: Escala+=20; glutPostRedisplay(); break;
                case 6: POSX+=10; glutPostRedisplay(); break;
                case 5: POSY+=10; glutPostRedisplay(); break;
                case 4: Tipo=GL_POLYGON;init(); glutPostRedisplay(); break;
                case 3: animation=fopen("trayectoriaCIRCULO2d.txt","r");
                glutIdleFunc(spinDisplay);break;
                case 2:angley++;glutPostRedisplay(); break;
                case 1: anglex++; glutPostRedisplay();break;
                case 0: Escala=500;anglex=0;angley=0;anglez=0;POSX=-250.0;
                POSY=-250.0;POSZ=0.0;glutPostRedisplay(); break;
                break;
            default:
                break;
            }
        case '-':
            switch((int)subir/33){
                case 7: Escala-=20; glutPostRedisplay(); break;
                case 6: POSX-=10; glutPostRedisplay(); break;
                case 5: POSY-=10; glutPostRedisplay(); break;
                case 4: Tipo=GL_LINE_LOOP;init();glutPostRedisplay(); break;
                case 3: animation=fopen("trayectorialinea2D.txt","r");
                glutIdleFunc(spinDisplay);
                case 2: angley-=2;glutPostRedisplay(); break;
                break;
            }
    }
}

```

```

        case 1:  anglex-=2; glutPostRedisplay();      break;
        case 0: glutIdleFunc(spinDisplay); exit(0);break;
                break;
        default:
                break;
    }
    break;
}
}

void main(int argc, char **argv)
{
    pedir();
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA);
    glutInitWindowPosition(0,0);
    glutInitWindowSize(1275,735);
    glutCreateWindow("Modelacion y simulacion de la planeacion de trayectorias para un manipulador flexible con
actuador neumatico de dos grados de libertad");
    hDC = wglGetCurrentDC();
    BuildFont();
    init();
    glutDisplayFunc(renderScene);
    glutKeyboardFunc(keyboard);
    glutSpecialFunc(keyboard2);
    glutReshapeFunc(changeSize);
    glutMainLoop();
}

```