



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

**Modelación y simulación de un robot neumático de  
tres grados de libertad**

**T e s i s**

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

**Presenta**

**José Guadalupe Gutiérrez Morales**

**Dirigido por:**

**Dr. Juan Manuel Ramos Arreguín**

**Santiago de Querétaro, Qro; febrero de 2012**



Universidad Autónoma de Querétaro  
Facultad de Informática  
Maestría

Modelación y simulación de un robot neumático de tres grados de libertad

**TESIS**

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

**Presenta:**

José Guadalupe Gutiérrez Morales

**Dirigido por:**

Dr. Juan Manuel Ramos Arreguín

**SINODALES**

Dr. Juan Manuel Ramos Arreguín

Presidente

Firma

Dr. Saúl Tovar Arriaga

Secretario

Firma

Dr. Jesús Carlos Pedraza Ortega

Vocal

Firma

Dra. Sandra Luz Canchola Magdaleno

Suplente

Firma

Dr. Efrén Gorrostieta Hurtado

Suplente

Firma

MC. Rait Angélica Rico Hernández

Director de la Facultad

Dr. Irineo Torres Pacheco

Director de Investigación y  
Posgrado

Centro Universitario  
Querétaro, Qro.  
febrero de 2012  
México



## Resumen.

En este trabajo de tesis, se propone la modelación y simulación gráfica del comportamiento de un robot manipulador flexible con actuador neumático, con tres grados de libertad. Cabe mencionar, que el presente trabajo está basado en trabajos de investigación realizados previamente, en los cuales se hacen aportaciones relevantes para uno y dos grados de libertad.

La modelación y simulación de un robot manipulador es una actividad primordial y necesaria para encontrar y definir sus movimientos y estrategias de control. Previo a la implementación física del robot, se debe contar con un sistema de software confiable y eficiente que permita analizar el comportamiento del robot, así como realizar las correcciones necesarias, con lo cual se reducen los tiempos y costos de dicha implementación.

En este trabajo, se presenta la modelación matemática de la cinemática directa y la cinemática inversa del robot, así como la modelación matemática de la dinámica para este tipo de robot industrial. Se realiza el análisis desde el punto de vista mecánico de un robot atendiendo exclusivamente a sus movimientos (estudio cinemático) y atendiendo además a las fuerzas y momentos que actúan sobre sus partes (estudio dinámico) debidas a los elementos actuadores y a la carga transportada por el elemento terminal.

Como producto final de esta tesis, se elaboró un sistema de software basado en la programación multilenguaje, utilizando Matlab, OpenGL y lenguaje C++. Dicho sistema resulta ser eficiente y flexible para la modelación y simulación del robot, aparte de ser una alternativa de bajo costo comparado con sistemas de software de este tipo.

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



## **Summary.**

In this thesis, both the modelation and graphic simulation of the behavior of a manipulating flexible robot with pneumatic actuator, with three degrees of freedom are proposed. It is important to mention, that the present work is based in works of investigation accomplished previously, in the ones that they are done in relevant contributions for one and two degrees of freedom.

The modelation and simulation of a manipulating robot it is a primary and necessary activity to find and to define their motions and strategies of control. Previous to the physical implementation of the robot, we should have a system of reliable and efficient software that it allows analyzing the behavior of the robot, as well as accomplishing the necessary corrections, with which times and costs of that implementation decrease.

In this work, the mathematical modelation of the direct cinematics and the inverse cinematics of the robot are presented, as well as the mathematical modelation of the dynamics for this type of industrial robot. The mechanical analysis of the robot attending exclusively its motions (cinematics study) and attending besides to forces and moments that actuators perform on on their parts (study dynamic ) due to the actuators and to the load transported by the terminal element.

As a final product of this thesis, a system of software was developed, based on multi-language programming, using Matlab, OpenGL and C++ language. The mentioned system turns out to be efficient and flexible for the modelation and simulation of the robot, aside from being a low-cost alternative in comparison to systems of suchlike software.

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



## DEDICATORIA.

A Dios por todas sus bendiciones y su infinita bondad y amor, por permitirme culminar este importante proyecto.

A mis padres, por brindarme la maravillosa oportunidad de cursar estudios superiores y por su gran ejemplo de superación y amor que siempre me dieron.

A mis hermanos, familiares y amigos, que tuvieron palabras de apoyo y aliento para mí durante mis estudios.

A mi director de tesis, por su apoyo incondicional en la elaboración de esta tesis.

A mi pequeña hija, Mayra Guadalupe, quien es el motor de mi vida y mi ángel enviado por Dios, que me da fuerzas cada día para luchar y continuar contra las adversidades que se presentan en la vida.



## **AGRADECIMIENTOS.**

Agradezco sinceramente a todas aquellas personas que me brindaron sus conocimientos y apoyo para hacer posible la culminación de este proyecto de investigación, especialmente al Dr. Juan Manuel Ramos Arreguín, mi director de tesis, por su apoyo decidido, amistad y paciencia durante la elaboración de este proyecto.

Mi agradecimiento también a los doctores Marco Antonio Aceves Fernández, Jesús Carlos Pedraza Ortega y Efrén Gorrostieta Hurtado por su apoyo y orientación para retomar mi proyecto de titulación así como darme ánimos de continuar y terminarlo. A los doctores Sandra Luz Canchola Magdaleno y Saúl Tovar Arriaga por su paciencia para leer los borradores por participar como mis sinodales.

Y finalmente, mi agradecimiento a las autoridades universitarias, y el personal de la Facultad de Informática, que tuvieron un gesto de amistad y apoyo durante mis estudios.

Gracias a todos, Dios los bendiga.



## ÍNDICE

	<b>Página</b>
Resumen	i
Summary	ii
Dedicatoria	iii
Agradecimientos	iv
Índice	v
Índice de figuras	vii
Índice de tablas	ix
1. INTRODUCCIÓN.	1
1.1 Antecedentes y justificación de la investigación	2
1.2 Historia de la robótica	4
1.3 Definición del robot industrial	8
1.4 Clasificación del robot industrial	19
1.5 Aplicaciones	21
1.6 Organización de la tesis	21
2. MODELACIÓN.	23
2.1 Modelo cinemático del robot	24
2.1.1 Modelo cinemático directo	25
2.1.2 Modelo cinemático inverso	37
2.2 Modelo dinámico del manipulador	49
3. SIMULACIÓN	56
3.1 Programación Interfaz multilenguaje.	57
3.2 Lenguaje C++	59
3.3 Matlab	61
3.4 OpenGL	62
3.5 Metodología propuesta	63



4. RESULTADOS, CONCLUSIONES Y TRABAJOS A FUTURO	66
4.1 Resultados	67
4.2 Programas y librerías del software	74
4.3 Conclusiones	76
4.4 Trabajos a futuro	76
BIBLIOGRAFÍA	78
GLOSARIO	82
ANEXOS	84





## ÍNDICE DE FIGURAS

	Figura	Página
Figura 1.	Elementos estructurales de un robot industrial.	11
Figura 2.	Distintos tipos de articulaciones de un robot: a) lineal, b) rotacionales.	11
Figura 3.	Punto terminal de un manipulador.	12
Figura 4.	Semejanza de un brazo manipulador con la anatomía humana.	13
Figura 5.	Distintos grados de libertad de un brazo de robot.	14
Figura 6.	Dimensiones de los sistemas mecánicos; longitud de los eslabones. Las dimensiones de los dos mecanismos son iguales.	25
Figura 7.	Primer mecanismo corredera – manivela de cuatro barras, con dos cadenas cinemáticas cerradas.	27
Figura 8.	Cadenas cinemáticas cerradas del primer mecanismo.	28
Figura 9.	Segundo mecanismo corredera – manivela de cuatro barras, con dos cadenas cinemáticas cerradas.	32
Figura 10.	Cadenas cinemáticas cerradas del segundo mecanismo.	33
Figura 11.	Dimensiones de los elementos que se utilizaran para el cálculo inicial.	38
Figura 12.	Triángulos rectángulos útiles para el cálculo inicial donde se obtienen $\theta_4$ y $\theta_{14}$ .	39
Figura 13.	Primer mecanismo corredera – manivela de cuatro barras, con dos cadenas cinemáticas cerradas.	41
Figura 14.	Cadenas cinemáticas cerradas del primer mecanismo.	42
Figura 15.	Segundo mecanismo corredera – manivela de cuatro barras, con dos cadenas cinemáticas cerradas.	46



Figura 16.	Cadenas cinemáticas cerradas del segundo mecanismo.	46
Figura 17.	Estructura propuesta de un robot manipulador con tres grados de libertad.	53
Figura 18.	Diagrama de bloques para el desarrollo de una interfaz multilenguaje.	57
Figura 19.	Metodología propuesta para la simulación gráfica de un proceso mecatrónico.	64
Figura 20.	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.	68
Figura 21.	Imagen del sistema de simulación del robot manipulador de tres grados de libertad.	69
Figura 22.	Imagen inicial del sistema de simulación del robot manipulador de tres grados de libertad.	70
Figura 23.	Gráficas de la dinámica de los actuadores del robot neumático.	71
Figura 24.	Diagrama de flujo del programa principal del robot neumático.	72
Figura 25.	Diagrama de flujo de la rutina para dibujar el robot manipulador neumático.	73
Figura 26.	Diagrama de flujo de la rutina para dibujar el movimiento del robot manipulador neumático.	74



## ÍNDICE DE TABLAS

	Tabla	Página
Tabla 1.	Resumen de procesos y sus características.	8
Tabla 2.	Tipos de sensores utilizados en robótica.	16
Tabla 3.	Diferentes tipos de actuadores.	18
Tabla 4.	Dimensiones del mecanismo corredera – manivela 4 barras.	26

# **CAPÍTULO 1**

## **INTRODUCCIÓN**



## 1. INTRODUCCIÓN.

### 1.1 Antecedentes y justificación de la investigación.

La robótica es un campo relativamente joven de tecnología moderna, que cruza los límites de la ingeniería tradicional.

Entender la complejidad de los robots y sus aplicaciones requiere conocimientos de ingeniería eléctrica, ingeniería mecánica, ingeniería industrial y de sistemas, ciencias computacionales, economía y matemáticas [1]. Nuevas disciplinas de ingeniería tales como ingeniería de manufactura, ingeniería de aplicaciones y la ingeniería del conocimiento han emergido para abordar la complejidad de la robótica y la automatización industrial.

El auge actual de la robótica tiene varias causas, entre las que podemos destacar: la alta velocidad de procesamiento de los microprocesadores actuales y la baja de costos de los componentes de computación. A esto debemos agregar todo el conocimiento acumulado por más de 50 años de investigación en Inteligencia Artificial (IA) y también el uso de Internet que permite compartir experiencias entre investigadores de distintas partes del mundo. [2]

Un robot manipulador industrial, está definido por el estándar ISO 8373 como un sistema multipropósito, reprogramable y de retroalimentación controlada. [3] Son programables en tres o más grados de libertad y son usados en procesos de automatización industrial.

En el caso particular de los robots manipuladores, podemos encontrar diversas aplicaciones tales como procesos industriales, traslado de objetos, procesos de automatización, procesos de medicina, etc. [4]. Actualmente, podemos encontrar el número más grande de robots manipuladores en la industria automotriz en donde la proporción de trabajadores humanos y robots es de 6:1, [5] especialmente útiles en ambientes peligrosos o agresivos. Es por ello que este tipo de robots son estudiados en diversas maneras.



No obstante, la mayoría de los trabajos reportados emplean actuadores eléctricos o hidráulicos, los cuales tienen un comportamiento lineal y el control es mucho más fácil que los actuadores neumáticos. La principal desventaja de los actuadores eléctricos es su baja relación fuerza-peso, las grandes cantidades de corriente que necesita y además son muy pesados. En el caso de los actuadores hidráulicos, su desventaja es que no son ecológicos, necesitan aceite y líneas de retroalimentación de bombeo, lo cual eleva los costos y el peso físico de los sistemas.

Por otro lado, y en forma contraria a los actuadores eléctricos e hidráulicos, los actuadores neumáticos son limpios, económicos, ligeros, rápidos, tienen una gran relación fuerza-peso y no necesita líneas de retroalimentación. Sin embargo, los actuadores neumáticos no son utilizados en el desarrollo de manipuladores flexibles debido a su alto comportamiento no lineal.

En el presente trabajo, basado en investigaciones realizadas previamente, como en [4], [6], [7] y [8], en los cuales se hacen aportaciones muy importantes con uno y dos grados de libertad, se propone la modelación y simulación gráfica del comportamiento de un robot manipulador flexible con actuador neumático, con tres grados de libertad, utilizando técnicas de programación multilenguaje. Para lograr este objetivo, primero se realiza el análisis del sistema de un robot neumático de tres grados de libertad, para la obtención de su modelo matemático, posteriormente se obtiene el modelo cinemático y dinámico del robot neumático de tres grados de libertad para enseguida implementar la simulación del posicionamiento de un pistón neumático de un robot brazo flexible con C++, OpenGL y Matlab.

La modelación y simulación de un robot manipulador es una premisa necesaria para encontrar y definir sus movimientos y estrategias de control. [9] Los robots se componen de diferentes dispositivos de alta precisión que se dañan fácilmente si no se manejan adecuadamente, y por esta razón es conveniente contar con un sistema confiable que permita familiarizarse con el comportamiento del robot, así



como corregir posibles fallas, antes del proceso de implementación física del robot.

## 1.2 Historia de la robótica.

El ciudadano industrializado que vive los inicios del siglo XXI se ha visto en la necesidad de asumir, en escasos 30 años, el significado de un buen número de nuevos términos marcados por su alto contenido tecnológico. De ellos sin duda el más relevante ha sido la computadora. Ésta está introducida hoy en día en su versión personal en multitud de hogares, y el ciudadano medio va conociendo en creciente proporción, además de su existencia, su modo de uso y buena parte de sus posibilidades.

Pero dejando de lado esta verdadera revolución social, existen otros conceptos procedentes del desarrollo tecnológico que han superado las barreras impuestas por las industrias y centros de investigación, incorporándose en cierta medida al lenguaje coloquial. Es llamativo como entre éstas destaca el concepto de **robot**. Pero el robot industrial, que se conoce y emplea en nuestros días, no surge como consecuencia de la tendencia o afición de reproducir a imagen y semejanza a los seres vivientes, sino de la necesidad. Fue la necesidad la que dio origen a la agricultura, el pastoreo, la caza, la pesca, etc. Más adelante, la necesidad provoca la primera revolución industrial con el descubrimiento de la máquina de vapor de Watt y, actualmente, la necesidad ha cubierto de ordenadores la faz de la tierra. Inmersos en la era de la informatización, la imperiosa necesidad de aumentar la productividad y mejorar la calidad de los productos, ha hecho insuficiente la automatización industrial rígida, dominante en las primeras décadas del siglo XX.

Se llama mecanización a la incorporación de máquinas en la realización de determinadas tareas. Así, se habla de la mecanización del campo, cuando a las tareas agrícolas se han incorporado máquinas por todos conocidas como el tractor, el arado o la cosechadora. También se habla del mecanizado de piezas en un taller, cuando para su fabricación se utilizan máquinas como tornos o fresadoras, es decir, cuando se abandona la fabricación manual y se sustituye por



procesos mecanizados que permiten mejores acabados y mayor rapidez en la confección de elementos. Ya en la antigua Grecia se encontraban algunos sistemas mecánicos de control, como los relojes de agua de Ktesibios y de Platón.

Un paso más adelante se encuentra la automatización, o control automático, proceso cuyo alcance se puede comprender a partir de la definición que el Diccionario de la Lengua Española de la Real Academia realiza de Automática:

**“Disciplina que trata de los métodos y procedimientos cuya finalidad es la sustitución del operador humano por un operador artificial en la ejecución de una tarea física o mental previamente programada”.**

En casi todas las actividades humanas actuales existen dispositivos que ayudan al desempeño automático de tareas que antes realizaban las personas. En la primera parte del siglo XX la automatización tuvo su origen en el campo de la ingeniería de control y de los servosistemas (capaces de conocer su propio estado y de introducir las modificaciones precisas para llegar al estado requerido). Pero su ámbito se ha ido ampliando progresivamente con el desarrollo de otras disciplinas, como las ciencias de la computación, la teoría de la información, el procesamiento de señales, la inteligencia artificial, etc. Antes de 1960, los automatismos se realizaban con elementos electromecánicos o neumáticos. Se trataba de sistemas no programados, denominados automatismos de primera generación. Se conocen autómatas que fueron ya construidos por los griegos en el templo de Dédalo. Sin embargo, uno de los casos más difundidos es el del Pato de Jacques de Vaucanson (Grenoble 1709-París 1782) construido en 1738, que era un pato artificial capaz de batir las alas, zambullirse, nadar, tragar grano e incluso expeler una sustancia parecida al excremento. Otro caso de autómatas célebres aunque más próximo a nuestro tiempo es el del jugador de ajedrez de Torres Quevedo, construido en 1912, capaz de jugar finales de partida (rey contra rey y torre). Pero debe ponerse de manifiesto que los autómatas siempre repiten el mismo modelo de actuación y no son capaces de variar sus acciones en función del entorno o la situación.





El control numérico es la supervisión y regulación de determinadas tareas mecánicas de precisión, realizadas por una máquina herramienta. El control de estas tareas se realiza de forma automática para evitar, de este modo, que el control se lleve a cabo por un operario que, en ocasiones, podría verse sometido a ciertos riesgos en un proceso donde es imposible erradicar los errores humanos. De esta forma, se ajusta al máximo la precisión en la confección de piezas estandarizadas y se libera al operario de su control, mejorando la calidad y la cantidad del trabajo realizado. Un ejemplo de control automático es el control de la velocidad de giro de un taladro o la velocidad y control de avance de un torno o fresadora. Otro sistema de control automático es el servocontrol. Consiste en controlar de forma automática las acciones de una máquina en función de unos parámetros definidos y sus variaciones. Por ejemplo, el servocontrol se puede utilizar para controlar la velocidad de giro de un torno, de forma que se mantenga fija dentro de unos límites. Otro ejemplo de servocontrol podría ser el del freno de algunos vehículos en los que la fuerza transmitida a las ruedas es proporcional a la fuerza aplicada por el conductor sobre el pedal del freno; de esta función se encarga un mecanismo servocontrolado que se llama servofreno.

El gran avance de la electrónica en las décadas de los 60 y 70 preparó el terreno para la aparición de los automatismos de segunda generación: sistemas programados mediante lógica cableada destinados para aplicaciones específicas. Se trataba de sistemas de programación rígida puesto que pequeños cambios en la programación original requerían la desconexión y nueva conexión de numerosos elementos eléctricos y electrónicos (por ejemplo, los cuadros de mando de los antiguos ascensores repletos de relés). A partir de mediados de la década de los años 70, la comercialización de los microprocesadores dio paso a los autómatas programables y a las computadoras digitales, que usan lógica programada y posibilitan el diálogo persona-máquina. Estos sistemas, que constituyen **automatismos de tercera generación**, permiten una **programación flexible** de sus funciones.



La **robotización** es también una automatización de procesos sin la intervención humana, pero en este caso se da un paso más; hay desplazamiento de cargas, manipulación de objetos y un fuerte componente de realimentación. Es decir, este tipo de automatización permite la manipulación automática y programable de acciones y objetos. La **realimentación** es un proceso imprescindible en la robotización, ya que dota a un proceso de capacidad para captar información que, una vez procesada por la máquina, permite modificar su comportamiento (sus acciones). Una máquina que posea la capacidad de realimentación es capaz de modificar sus respuestas en función de las variaciones de su entorno.

Centrando el análisis en las diferencias que existen entre automatización y robotización, puede decirse que una máquina automatizada (autómata) responde siempre de igual manera ante sucesos de idéntica naturaleza. Mientras que por el contrario un robot, es decir, una máquina robotizada, se caracteriza porque puede manejar objetos y, lo más interesante, es un dispositivo multifuncional y reprogramable. Una máquina robotizada es capaz de hacer trabajos totalmente diferentes y adaptarse al medio, ya que puede tomar decisiones en función de las condiciones exteriores.

La totalidad de los procesos de mejora y control de la producción pueden sintetizarse en tres fundamentales:

**procesos de mecanización,**  
**procesos de automatización, y**  
**procesos de robotización.**

Las principales características de cada uno de estos tres grupos se exponen de forma resumida en la tabla 1.

La Robótica es una nueva tecnología, que surgió como tal, hacia 1960 [23]. Han transcurrido pocos años y el interés que ha despertado, desborda cualquier previsión. Quizás, al nacer la Robótica en la era de la información, una propaganda desmedida ha propiciado una imagen irreal a nivel popular y, al igual



que sucede con el microprocesador, la mitificación de esta nueva máquina, que de todas formas, nunca dejara de ser eso, una máquina.

PROCESO	CARACTERÍSTICAS
<b>MECANIZACIÓN</b>	<ul style="list-style-type: none"> <li>• Incorporación de máquinas:</li> <li>▪ Realizan procesos repetitivos</li> <li>▪ Sustituyen el esfuerzo humano</li> <li>▪ Manejadas por operador humano</li> </ul>
<b>AUTOMATIZACIÓN</b>	<ul style="list-style-type: none"> <li>• Sustitución parcial o total de intervención humana</li> <li>• Sistemas capaces de autorregulación</li> </ul>
<b>ROBOTIZACIÓN</b>	<ul style="list-style-type: none"> <li>• Reprogramación informática</li> <li>• Coordinación de automatismos</li> <li>• Adaptabilidad a diversas tareas (polivalencia)</li> </ul>

Tabla 1. Resumen de procesos y sus características.

El auge de la Robótica y la imperiosa necesidad de su implantación en numerosas instalaciones industriales, requiere el concurso de un buen número de especialistas en la materia. La Robótica es una tecnología multidisciplinar. Hace uso de todos los recursos de vanguardia de otras ciencias afines entre las que se pueden mencionar las siguientes: Mecánica, Cinemática, Dinámica, Matemáticas, Automática, Electrónica, Informática, Energía y actuadores eléctricos, neumáticos e hidráulicos, Visión Artificial, Sonido de máquinas, Inteligencia Artificial.

Realmente la Robótica es una combinación de todas las disciplinas expuestas, más el conocimiento de la aplicación a la que se enfoca, por lo que su estudio se hace especialmente indicado en las carreras de Ingeniería Superior y Técnica y en los centros de formación profesional, como asignatura práctica.

### 1.3 Definición del robot industrial.

Entre los robots considerados de más utilidad en la actualidad se encuentran los **robots industriales o manipuladores**. Existen ciertas dificultades a la hora de establecer una definición formal de lo que es un robot industrial. La primera de



ellas surge de la diferencia conceptual entre el mercado japonés y el euro-americano de lo que es un robot y lo que es un manipulador. Así, mientras que para los japoneses un robot industrial es cualquier dispositivo mecánico dotado de articulaciones móviles destinado a la manipulación, el mercado occidental es más restrictivo, exigiendo una mayor complejidad, sobre todo en lo relativo al control. En segundo lugar, y centrándose ya en el concepto occidental, aunque existe una idea común acerca de lo que es un robot industrial, no es fácil ponerse de acuerdo a la hora de determinar una definición formal. Además, la evolución de la robótica ha ido obligando a diferentes actualizaciones de su definición.

La definición más comúnmente aceptada posiblemente sea la de la Asociación de Industrias de Robótica (RIA, *Robotic Industry Association*), 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"

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"

Se incluye en esta definición la necesidad de que el robot tenga varios grados de libertad. Una definición 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.



**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.

Por último, la Federación Internacional de Robótica (IFR, *International Federation of Robotics*) distingue entre robot industrial de manipulación y otros robots:

"Por robot industrial de manipulación se entiende una máquina de manipulación automática, reprogramable y multifuncional con tres o más ejes que pueden posicionar y orientar materias, piezas, herramientas o dispositivos especiales para la ejecución de trabajos diversos en las diferentes etapas de la producción industrial, ya sea en una posición fija o en movimiento"

En esta definición se debe entender que la reprogramabilidad y la multifunción se consiguen sin modificaciones físicas del robot.

Común en todas las definiciones anteriores es la aceptación del robot industrial como un brazo mecánico con capacidad de manipulación y que incorpora un control más o menos complejo. Un sistema robotizado, en cambio, es un concepto más amplio. Engloba todos aquellos dispositivos que realizan tareas de forma automática en sustitución de un ser humano y que pueden incorporar o no a uno o varios robots, siendo esto último lo más frecuente.

### 1.3.1 Estructura de los robots industriales.

Un manipulador robótico consta de una secuencia de elementos estructurales rígidos, como se muestra en la figura 1, denominados enlaces o **eslabones**, conectados entre sí mediante juntas o **articulaciones**, que permiten el movimiento relativo de cada dos eslabones consecutivos.



Figura 1. Elementos estructurales de un robot industrial.

Una articulación puede ser, como se muestra en la figura 2:

- **Lineal** (deslizante, traslacional o prismática), si un eslabón desliza sobre un eje solidario al eslabón anterior.
- **Rotacional**, en caso de que un eslabón gire en torno a un eje solidario al eslabón anterior.

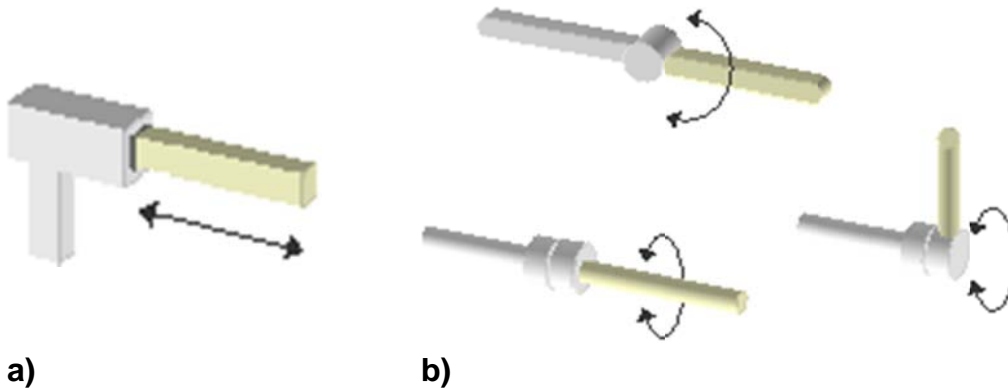


Figura 2. Distintos tipos de articulaciones de un robot: a) lineal, b) rotacionales.

El conjunto de eslabones y articulaciones se denomina **cadena cinemática**. Se dice que una cadena cinemática es abierta si cada eslabón se conecta mediante articulaciones exclusivamente al anterior y al siguiente, exceptuando el primero, que se suele fijar a un soporte, y el último, cuyo extremo final queda libre. A éste se puede conectar un **elemento terminal** o **actuador final**: una herramienta

especial que permite al robot de uso general realizar una aplicación particular, que debe diseñarse específicamente para dicha aplicación: una herramienta de sujeción, de soldadura, de pintura, etc. El punto más significativo del elemento terminal se denomina **punto terminal (PT)**, como se muestra en la figura 3. En el caso de una pinza, el punto terminal vendría a ser el centro de sujeción de la misma.



Figura 3. Punto terminal de un manipulador.

Los elementos terminales pueden dividirse en dos categorías:

- **pinzas** (*gripper*)
- **herramientas**

Las **pinzas** se utilizan para tomar un objeto, normalmente la pieza de trabajo, y sujetarlo durante el ciclo de trabajo del robot. Hay una diversidad de métodos de sujeción que pueden utilizarse, además de los métodos mecánicos obvios de agarre de la pieza entre dos o más dedos. Estos métodos suplementarios incluyen el empleo de casquillos de sujeción, imanes, ganchos, y cucharas.

Una **herramienta** se utiliza como actuador final en aplicaciones en donde se exija al robot realizar alguna operación sobre la pieza de trabajo. Estas aplicaciones incluyen la soldadura por puntos, la soldadura por arco, la pintura por pulverización y las operaciones de taladro. En cada caso, la herramienta particular está unida a la muñeca del robot para realizar la operación.

A los manipuladores robóticos se les suele denominar también **brazos de robot** por la analogía que se puede establecer, en muchos casos, con las extremidades superiores del cuerpo humano, figura 4.

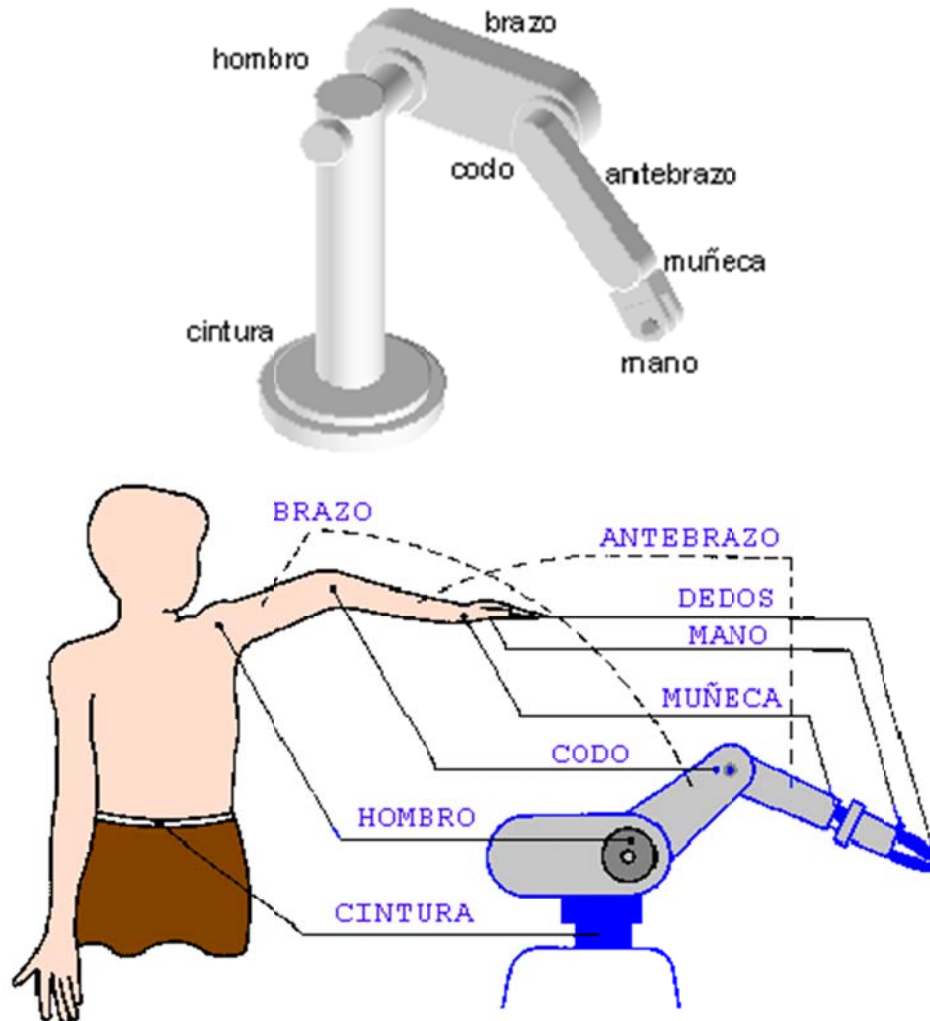


Figura 4. Semejanza de un brazo manipulador con la anatomía humana.

Se denomina **grado de libertad (g.d.l.)** figura 5, a cada una de las coordenadas independientes que son necesarias para describir el estado del sistema mecánico del robot (posición y orientación en el espacio de sus elementos). Normalmente, en cadenas cinemáticas abiertas, cada par eslabón-articulación tiene un solo grado de libertad, ya sea de rotación o de traslación. Pero una articulación podría tener dos o más g.d.l. que operan sobre ejes que se cortan entre sí.





Figura 5. Distintos grados de libertad de un brazo de robot.

Para describir y controlar el estado de un brazo de robot es preciso determinar:

- La posición del punto terminal (o de cualquier otro punto) respecto de un sistema de coordenadas externo y fijo, denominado el sistema mundo.
- El movimiento del brazo cuando los elementos actuadores aplican sus fuerzas y momentos.

El análisis desde el punto de vista mecánico de un robot se puede efectuar atendiendo exclusivamente a sus movimientos (estudio cinemático) o atendiendo además a las fuerzas y momentos que actúan sobre sus partes (estudio dinámico) debidas a los elementos actuadores y a la carga transportada por el elemento terminal.

### 1.3.2 Configuraciones morfológicas y parámetros característicos de los robots industriales.

Un robot es esencialmente un sistema organizado de forma tal que responde con una acción inteligente a los estímulos que es capaz de percibir. Se puede considerar como la síntesis de varios subsistemas, entre los que destacan:

- el **sistema sensorial**, compuesto por los **sensores** que recogen información acerca del estado del propio robot y de su entorno;



- el **sistema de accionamiento**, compuesto por los elementos **actuadores** (motores, por ejemplo) que permiten llevar a cabo las acciones programadas;
- la **unidad de control**, formada por los **elementos computacionales y el software** que regulan el comportamiento global del robot.

Además, a los anteriores hay que añadir:

- los **elementos estructurales**, que confieren rigidez y estabilidad al robot.
- los **elementos de transmisión y conversión del movimiento**. Cambios en el eje o la dirección de rotación, ampliación/reducción de velocidad, etc.
- los **elementos terminales**, según la aplicación a la que se destine al robot (pinzas, electrodos de soldadura, inyectores de pintura, etc.).
- los **dispositivos auxiliares de entrada/salida**, que permiten la comunicación del usuario con la unidad de control.
- los **sistemas de alimentación**, de tipo eléctrico, neumático o hidráulico.

## 1. Sensores

Para asegurar que el robot sigue una determinada trayectoria y alcanza la posición final deseada en el instante preciso, deben conocerse la posición, la velocidad y la aceleración de los elementos que lo constituyen. Los sensores que proporcionan esta información y, en general, todos aquéllos que producen información sobre el estado interno del robot, se denominan **sensores internos**.

Por otra parte, en la mayoría de las tareas es necesario conocer datos del mundo que rodea al robot: distancias a objetos (o contacto con ellos), fuerza ejercida en las operaciones de prensión o ejercida por objetos externos (peso), etc. Esta información puede obtenerse con dispositivos muy variados, desde los más simples (microinterruptores) a los más complejos (cámaras de TV). Éstos se denominan **sensores externos**. Sin ellos, cualquier suceso inesperado podría bloquear o dañar al robot. Asimismo, la imprecisión que afecta a las magnitudes



que intervienen en cualquier tarea (por ejemplo, el posicionado de piezas) malograría todo intento de ejecución fiable.

En la siguiente tabla 2 se muestran algunos de los principales sensores utilizados en robótica:

<b>Sensores internos</b>	<b>De posición</b>	Eléctricos: potenciómetros, sincros y <i>resolvers</i> Ópticos: optointerruptores, codificadores absolutos e incrementales ( <i>encoders</i> )
	<b>De velocidad</b>	Eléctricos: dinamos tacométricas Ópticos: con <i>encoder</i>
	<b>Acelerómetros</b>	
<b>Sensores externos</b>	<b>De proximidad</b>	De contacto: microinterruptores Sin contacto: resistivos, de efecto Hall, de fibra óptica, de ultrasonidos ...
	<b>De tacto</b>	De fotodetectores, de presión neumática, de polímeros (piel artificial) ...
	<b>De fuerza</b>	Por corriente en el motor, por deflexión de los dedos
	<b>De visión</b>	Cámaras de tubo, cámaras CCD

Tabla 2. Tipos de sensores utilizados en robótica.

## 2. Actuadores.

Los elementos actuadores son los dispositivos que ejercen fuerzas y momentos sobre las partes de un robot haciendo que éstas se muevan. Transforman en energía mecánica algún otro tipo de energía y, para que sean útiles en Robótica, deben poder ser controlados con rapidez y precisión. Los actuadores que se utilizan actualmente son de tres tipos:

- **Hidráulicos**, que aprovechan la circulación de fluidos, normalmente aceite especial. Son controlados mediante servoválvulas que regulan el flujo de fluido, el cual provoca un desplazamiento lineal de un cilindro o pistón. Los actuadores hidráulicos son recomendables en los manipuladores que tienen una gran capacidad de carga, y requieren una precisa regulación de velocidad.



- **Neumáticos.** Su principio de funcionamiento es similar al de los hidráulicos, pero emplean aire, altamente compresible, a diferencia de los aceites especiales. Los actuadores neumáticos resultan muy indicados en el control de movimientos rápidos, pero de precisión limitada.
- **Eléctricos.** Son los más utilizados actualmente en robots comerciales y experimentales. Se trata, principalmente, de motores de corriente continua (c.c.) y de motores paso a paso. Ambos convierten energía eléctrica en movimiento rotacional. Los motores c.c. controlados por armadura se comportan en sí mismos como un sistema realimentado, lo que los hace especialmente útiles. Los motores paso a paso permiten realizar giros de paso definido, con precisiones de aproximadamente  $\pm 1.8^\circ$ . Tienen un elevado momento a bajas velocidades y no necesitan codificadores de posición (*encoders*), es decir, dispositivos que informen continuamente de la posición instantánea.

### **Servomotores.**

Cada uno de estos sistemas presenta características diferentes, siendo preciso evaluarlas a la hora de seleccionar el tipo de actuador más conveniente. Las características a considerar son, entre otras:

- Potencia
- Controlabilidad
- Peso y volumen
- Precisión
- Velocidad
- Mantenimiento
- Coste

En la siguiente tabla 3, se proporciona un resumen comparativo de los actuadores utilizados en robótica:



	Neumáticos	Hidráulicos	Eléctricos
<b>Energía</b>	Aire a presión (5-10 bar)	Aceite mineral (50-100 bar)	Corriente eléctrica
<b>Opciones</b>	Cilindros Motor de paletas Motor de pistón	Cilindros Motor de paletas Motor de pistones axiales	Corriente continua Corriente alterna Motor paso a paso Servomotor
<b>Ventajas</b>	Baratos Rápidos Sencillos Robustos	Rápidos Alta relación potencia-peso Autolubricantes Alta capacidad de carga Estabilidad frente a cargas estáticas	Precisos Fiables Fácil control Sencilla instalación Silenciosos
<b>Desventajas</b>	Dificultad de control continuo Instalación especial (compresor, filtros) Ruidoso	Difícil mantenimiento Instalación especial (filtros, eliminación aire) Frecuentes fugas Caros	Potencia limitada

Tabla 3. Diferentes tipos de actuadores.

### 3. Unidad de control

Se trata del "cerebro" del robot, el órgano de tratamiento de la información. Es el responsable de determinar los movimientos precisos de cada parte del mecanismo para que el elemento terminal pueda ser movido a la posición y orientación requeridas en el espacio. Puede tratarse de un PLC (*Programmable Logic Controller*) en los modelos menos avanzados o de un sistema basado en microprocesadores en los más avanzados. En su memoria contiene un modelo físico del propio robot, un modelo de su entorno y los programas necesarios para desarrollar los algoritmos de control.



## 1.4 Clasificación del Robot Industrial.

La maquinaria para la automatización rígida dio paso al robot con el desarrollo de controladores rápidos, basados en el microprocesador, así como un empleo de servos en bucle cerrado, que permiten establecer con exactitud la posición real de los elementos del robot y establecer el error con la posición deseada. Esta evolución ha dado origen a una serie de tipos de robots, que se citan a continuación:

### 1. Manipuladores:

Son sistemas mecánicos multifuncionales, con un sencillo sistema de control, que permite gobernar el movimiento de sus elementos, de los siguientes modos:

- a. *Manual*: Cuando el operario controla directamente la tarea del manipulador.
- b. *De secuencia fija*: cuando se repite, de forma invariable, el proceso de trabajo preparado previamente.
- c. *De secuencia variable*: Se pueden alterar algunas características de los ciclos de trabajo.

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.

### 2. Robots de repetición o aprendizaje:

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".

### **3. Robots con control por computador:**

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, cuando la prepara para realizar un trabajo. El control por computador dispone de un lenguaje específico, 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. 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.

### **4. Robots inteligentes:**

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).

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. 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.

### **5. Micro-robots:**

Con fines educacionales, de entretenimiento o investigación, existen numerosos robots de formación o micro-robots a un precio muy asequible y, cuya estructura y funcionamiento son similares a los de aplicación industrial.



## **1.5. Aplicaciones del robot industrial.**

En la actualidad los robots se usan de manera extensa en la industria, siendo un elemento indispensable en una gran parte de los procesos de manufactura. Impulsados principalmente por el sector del automóvil, los robots han dejado de ser máquinas misteriosas propias de la ciencia-ficción para ser un elemento más de muchos de los talleres y líneas de producción.

Por su propia definición el robot industrial es multifuncional, esto es, puede ser aplicado a un número, en principio ilimitado, de funciones. No obstante, la práctica ha demostrado que su adaptación es óptima en determinados procesos (soldadura, palatización, etc.) en los que hoy día el robot es sin duda alguna, la solución más rentable.

Junto con estas aplicaciones ya arraigadas hay otras novedosas que si bien la utilización del robot no se realiza a gran escala, si se justifica su aplicación por las condiciones intrínsecas del medio de trabajo (ambientes contaminados, salas asépticas, construcción, etc.) o la elevada exigencia en cuanto a calidad de los resultados (medicina, etc.).

Estos robots se han venido llamando robots de servicio. La Federación Internacional de la Robótica (IFR) estableció una clasificación de las aplicaciones de la Robótica en el sector manufacturero:

1. Manipulación en fundición, 2. Manipulación en moldeo de plásticos, 3. Manipulación en tratamientos térmicos, 4. Manipulación en la forja y estampación, 5. Soldadura, 6. Aplicación de materiales, 7. Mecanización, 8. Otros procesos 9. Montaje.

## **1.6 Organización de la tesis.**

El presente documento de tesis está organizado en cuatro capítulos de la siguiente manera:





**Capítulo 1.** En este capítulo se muestran los antecedentes y justificación de este trabajo de tesis y los objetivos a lograr para el proyecto de la modelación y simulación cinemática y dinámica, así como la historia de los robots, sus elementos principales, aplicaciones y la organización de la tesis.

**Capítulo 2.** Se describe el modelo matemático del robot manipulador, considerando específicamente tres aspectos: la modelación cinemática directa, la modelación cinemática inversa y la modelación dinámica.

**Capítulo 3.** Se describe la metodología empleada para el desarrollo del software de modelación y simulación para un manipulador flexible con actuador neumático de tres grados de libertad y se describen las herramientas para la realización del proyecto.

**Capítulo 4.** Se describen las pruebas y resultados obtenidos, mostrándose el detalle del software realizado, así como las conclusiones y trabajos a futuro relacionados con esta tesis.

También se incluye un glosario de términos importantes en el proyecto, así como dos anexos que incluyen el diagrama de bloques del sistema de simulación y el código fuente del programa principal de la animación del robot.

# **CAPÍTULO 2**

## **MODELACIÓN**



## 2. Modelación.

Para describir y controlar el estado de un brazo de robot es preciso determinar:

- La posición del punto terminal (o de cualquier otro punto) respecto de un sistema de coordenadas externo y fijo, denominado el sistema mundo.
- El movimiento del brazo cuando los elementos actuadores aplican sus fuerzas y momentos.

El análisis desde el punto de vista mecánico de un robot se puede efectuar atendiendo exclusivamente a sus movimientos (estudio cinemático) o atendiendo además a las fuerzas y momentos que actúan sobre sus partes (estudio dinámico) debidas a los elementos actuadores y a la carga transportada por el elemento terminal.

### 2.1 Modelo cinemático del robot.

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.

En la figura 6, se muestra el dibujo del prototipo del robot manipulador de tres grados de libertad, así como también se muestran sus dimensiones en la tabla 4.

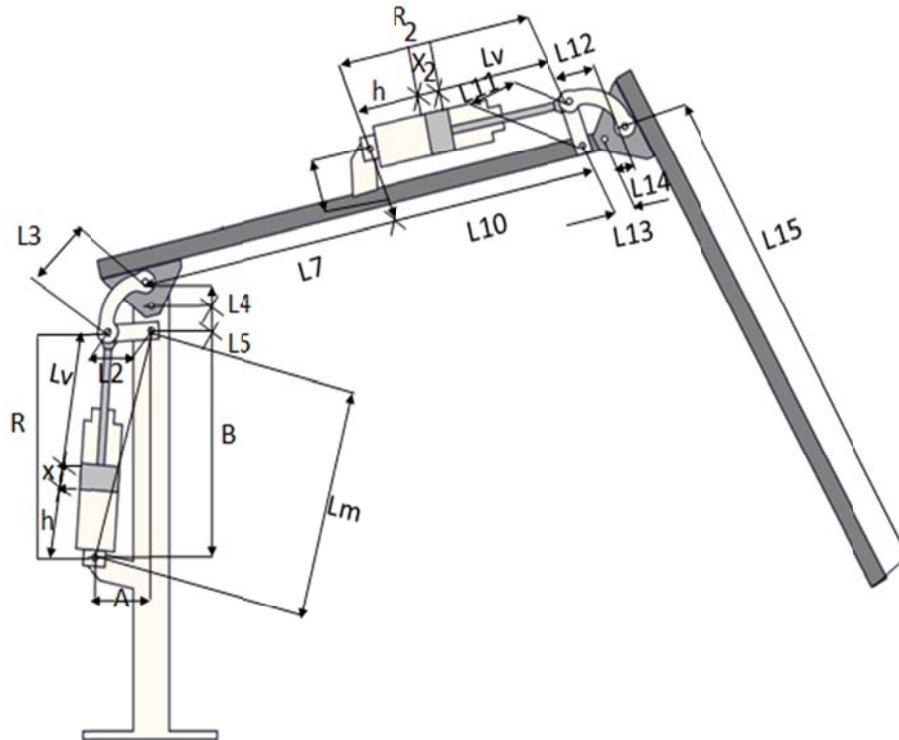


Figura 6. Dimensiones de los sistemas mecánicos; longitud de los eslabones. Las dimensiones de los dos mecanismos son iguales.

### 2.1.1 Modelo cinemático directo.

El problema cinemático directo, consiste en determinar cuál es la posición y orientación del extremo final del robot, con respecto a un sistema de coordenadas que se toma como referencia, conocidos los valores de las articulaciones y los parámetros geométricos de los elementos del robot.

Determina la posición y orientación del extremo 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 resolvió en dos etapas; primero se analiza el primer mecanismo de cuatro barras y posteriormente el segundo.

En la tabla 4 se muestran las variables y dimensiones del mecanismo corredera – manivela utilizados en el diseño del manipulador.



Descripción	Variable	Medida
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 <sub>2</sub>	
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
	Lm	0.463 m

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

Estos cálculos están basados en [25], y se adaptaron para este tipo de robot:

### Cinemática del primer mecanismo.

Del primer mecanismo se observan dos cadenas cinemáticas cerradas; (figura 7). La primera cadena la forman R, L2, A y B, la segunda cadena está formada por L2, L3, L4 y L5, (ver figura 8).

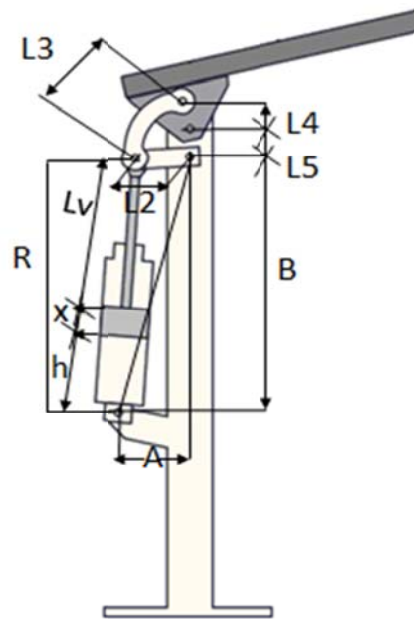


Figura 7. Primer mecanismo corredera – manivela de cuatro barras, con dos cadenas cinemáticas cerradas.

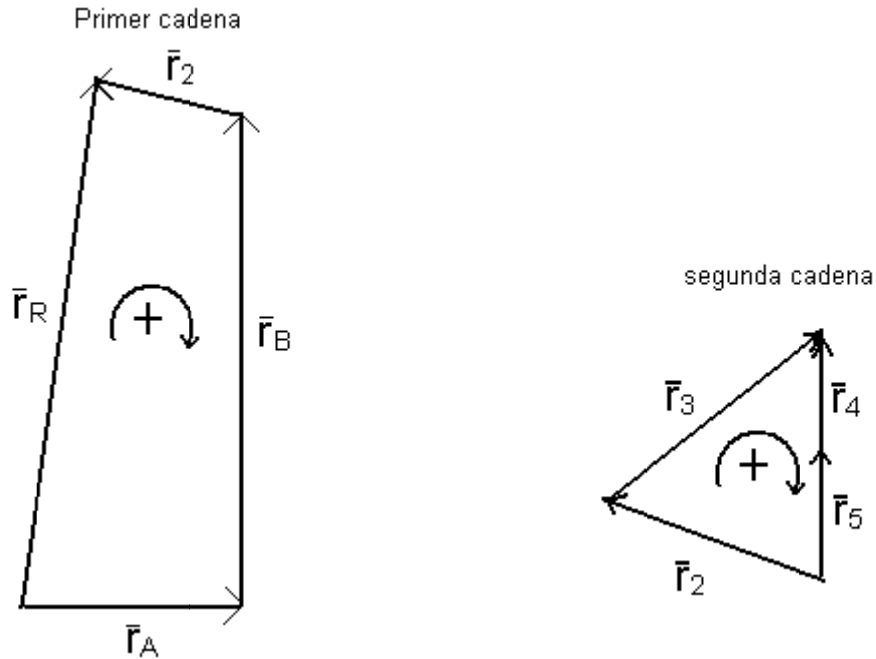


Figura 8. Cadenas cinemáticas cerradas del primer mecanismo.

De la primera cadena cinemática obtenemos la siguiente ecuación vectorial (ver figura 8):

$$\vec{r}_R - \vec{r}_2 - \vec{r}_B - \vec{r}_A = 0 \quad (2.1)$$

Si sustituimos las componentes en  $i$  y  $j$  de cada vector en la ecuación (2.1), nos dará como resultado 2 ecuaciones:

$$R \cos \theta_1 - L_2 \cos \theta_2 - A = 0 \quad (2.2)$$

$$R \sin \theta_1 - L_2 \sin \theta_2 - B = 0 \quad (2.3)$$

De las ecuaciones (2.2) y (2.3), se despeja el término que tenga  $\theta_1$ , se eleva al cuadrado ambos lados y sumamos las dos ecuaciones. Nos dará como resultado la siguiente ecuación:

$$R^2 = L_2^2 + 2L_2 A \cos \theta_2 + 2L_2 B \sin \theta_2 + A^2 + B^2 \quad (2.4)$$



Aplicando el método de Newton – Raphson, podemos conocer con (2.4) mediante iteraciones a  $\theta_2$ .

El método de Newton – Raphson es el siguiente:

$$X_{i+1} = X_i - \frac{f(X)}{f'(X)} \quad (2.5)$$

donde  $f(X)=0$

Resolviendo para (2.4):

$$f(\theta_2) = L_2^2 + 2L_2A \cos \theta_2 + 2L_2B \sin \theta_2 + A^2 + B^2 - R^2 \quad (2.6)$$

$$f'(\theta_2) = -2L_2A \sin \theta_2 + 2L_2B \cos \theta_2 \quad (2.7)$$

Sustituyendo (2.6) y (2.7) en (2.5):

$$\theta_{2i+1} = \theta_{2i} - \frac{L_2^2 + 2L_2A \cos \theta_{2i} + 2L_2B \sin \theta_{2i} + A^2 + B^2 - R^2}{-2L_{2i}A \sin \theta_{2i} + 2L_{2i}B \cos \theta_{2i}} \quad (2.8)$$

**Calculando  $\theta_1$ :**

De las ecuaciones (2.2) y (2.3), se despeja el término que tenga  $\theta_2$ , se eleva al cuadrado ambos lados y sumamos las dos ecuaciones. Nos dará como resultado la siguiente ecuación:

$$L_2^2 = R^2 - 2RA \cos \theta_1 - 2RB \sin \theta_1 + A^2 + B^2 \quad (2.9)$$

Aplicando el método de Newton – Raphson, podemos conocer con (2.9) mediante iteraciones a  $\theta_1$ .

Resolviendo para (2.9):

$$f(\theta_1) = R^2 - 2RA \cos \theta_1 - 2RB \sin \theta_1 + A^2 + B^2 - L_2^2 \quad (2.10)$$

$$f'(\theta_1) = 2RA \sin \theta_1 - 2LRB \cos \theta_1 \quad (2.11)$$





Sustituyendo (2.10) y (2.11) en (2.5):

$$\theta_{i+1} = \theta_i - \frac{R^2 - 2RA\cos\theta_i - 2RB\sin\theta_i + A^2 + B^2 - L_2^2}{2RA\sin\theta_i - 2LRB\cos\theta_i} \quad (2.12)$$

De la segunda cadena cinemática obtenemos la siguiente ecuación vectorial (ver figura 10):

$$\vec{r}_2 + \vec{r}_3 - \vec{r}_4 - \vec{r}_5 = 0 \quad (2.13)$$

Si sustituimos las componentes en i y j de cada vector en la ecuación (2.13), nos dará como resultado 2 ecuaciones:

$$L_2 \cos\theta_2 + L_3 \cos\theta_3 - L_4 \cos\theta_4 = 0 \quad (2.14)$$

$$L_2 \sin\theta_2 + L_3 \sin\theta_3 - L_4 \sin\theta_4 - L_5 = 0 \quad (2.15)$$

De las ecuaciones (2.14) y (2.15), se despeja el término que tenga  $\theta_3$ , se eleva al cuadrado ambos lados y sumamos las dos ecuaciones. Nos dará como resultado la siguiente ecuación:

$$L_3^2 = L_5^2 + L_4^2 + L_2^2 - 2L_4L_2 \cos\theta_4 \cos\theta_2 - 2L_4L_2 \sin\theta_4 \sin\theta_2 + 2L_5L_4 \sin\theta_4 - 2L_5L_2 \sin\theta_2 \quad (2.16)$$

Aplicando el método de Newton – Raphson, podemos conocer con (2.16) mediante iteraciones a  $\theta_4$ .

Resolviendo para (2.16):

$$f(\theta_4) = L_5^2 + L_4^2 + L_2^2 - 2L_4L_2 \cos\theta_4 \cos\theta_2 - 2L_4L_2 \sin\theta_4 \sin\theta_2 + 2L_5L_4 \sin\theta_4 - 2L_5L_2 \sin\theta_2 - L_3^2 \quad (2.17)$$

$$f'(\theta_4) = 2L_4L_2 \sin\theta_4 \cos\theta_2 - 2L_4L_2 \cos\theta_4 \sin\theta_2 + 2L_5L_4 \cos\theta_4 \quad (2.18)$$



Sustituyendo (2.17) y (2.18) en (2.5):

$$\theta_{4i+1} = \theta_{4i} - \frac{L_5^2 + L_4^2 + L_2^2 - 2L_4L_2 \cos \theta_4 \cos \theta_2 - 2L_4L_2 \text{sen} \theta_4 \text{sen} \theta_2}{2L_4L_2 \text{sen} \theta_4 \cos \theta_2 - 2L_4L_2 \cos \theta_4 \text{sen} \theta_2 + 2L_5L_4 \text{sen} \theta_4 - 2L_5L_2 \text{sen} \theta_2 - L_3^2 + 2L_5L_4 \cos \theta_4} \quad (2.19)$$

**Calculando  $\theta_3$ :**

De las ecuaciones (2.14) y (2.15), se despeja el término que tenga  $\theta_4$ , se eleva al cuadrado ambos lados y sumamos las dos ecuaciones, nos dará como resultado la siguiente ecuación:

$$L_4^2 = \left[ \begin{array}{l} L_5^2 + L_3^2 + L_2^2 + 2L_3L_2 \cos \theta_3 \cos \theta_2 + 2L_3L_2 \text{sen} \theta_3 \text{sen} \theta_2 \\ - 2L_5L_3 \text{sen} \theta_3 - 2L_5L_2 \text{sen} \theta_2 \end{array} \right] \quad (2.20)$$

Aplicando el método de Newton – Raphson, podemos conocer con (2.20) mediante iteraciones a  $\theta_3$ .

Resolviendo para (2.20):

$$f(\theta_3) = \left[ \begin{array}{l} L_5^2 + L_3^2 + L_2^2 + 2L_3L_2 \cos \theta_3 \cos \theta_2 + 2L_3L_2 \text{sen} \theta_3 \text{sen} \theta_2 \\ - 2L_5L_3 \text{sen} \theta_3 - 2L_5L_2 \text{sen} \theta_2 - L_4^2 \end{array} \right] \quad (2.21)$$

$$f'(\theta_3) = -2L_3L_2 \text{sen} \theta_3 \cos \theta_2 + 2L_3L_2 \cos \theta_3 \text{sen} \theta_2 - 2L_5L_3 \cos \theta_3 \quad (2.22)$$

Sustituyendo (2.21) y (2.22) en (2.5):

$$\theta_{3i+1} = \theta_{3i} - \frac{f(\theta_3)}{f'(\theta_3)} \quad (2.23)$$

### Cinemática del Segundo mecanismo.

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 9).

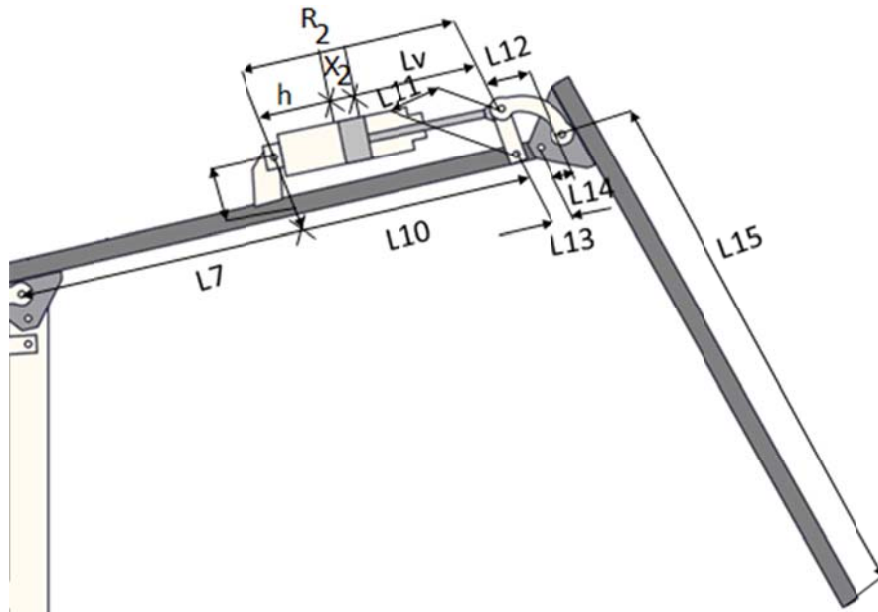


Figura 9. Segundo mecanismo corredera – manivela de cuatro barras, con dos cadenas cinemáticas cerradas.

De la primera cadena cinemática obtenemos la siguiente ecuación vectorial (ver figura 10):

$$\vec{r}_8 + \vec{r}_{10} + \vec{r}_{11} - \vec{r}_{R2} = 0 \quad (2.24)$$

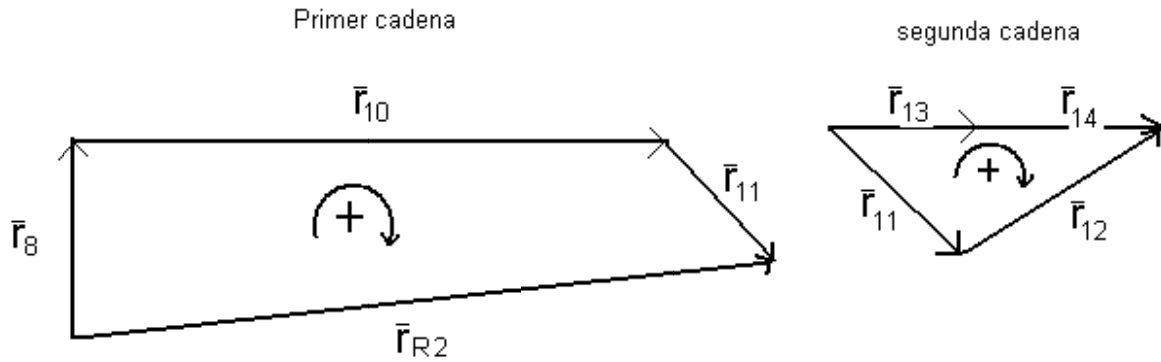


Figura 10. Cadenas cinemáticas cerradas del segundo mecanismo.

Si sustituimos las componentes en  $i$  y  $j$  de cada vector en la ecuación (2.24) nos dará como resultado 2 ecuaciones:

$$L_8 \cos \theta_8 + L_{10} \cos \theta_{10} + L_{11} \cos \theta_{11} - R_2 \cos \theta_9 = 0 \quad (2.25)$$

$$L_8 \text{sen} \theta_8 + L_{10} \text{sen} \theta_{10} + L_{11} \text{sen} \theta_{11} - R_2 \text{sen} \theta_9 = 0 \quad (2.26)$$

De las ecuaciones (2.25) y (2.26), se despeja el término que tenga  $\theta_9$ , 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:

$$R_2^2 = \left[ \begin{array}{l} L_8^2 + L_{10}^2 + L_{11}^2 + 2L_8L_{10} \cos \theta_4 \cos (\theta_4 + 270) \\ + 2L_8L_{10} \text{sen} \theta_4 \text{sen} (\theta_4 + 270) + 2L_8L_{11} \cos \theta_4 \cos \theta_{11} \\ + 2L_8L_{11} \text{sen} \theta_4 \text{sen} \theta_{11} + 2L_{11}L_{10} \cos \theta_{11} \cos (\theta_4 + 270) \\ + 2L_{11}L_{10} \text{sen} \theta_{11} \text{sen} (\theta_4 + 270) \end{array} \right] \quad (2.27)$$

Aplicando el método de Newton – Raphson, podemos conocer con (2.27) mediante iteraciones a  $\theta_{11}$ . El ángulo  $\theta_4$  ya se calculó, por lo tanto es conocido.

Resolviendo para (2.27):



$$\begin{aligned}
 f(\theta_{11}) = & L_8^2 + L_{10}^2 + L_{11}^2 + 2L_8L_{10} \cos \theta_4 \cos(\theta_4 + 270) \\
 & + 2L_8L_{10} \text{sen} \theta_4 \text{sen}(\theta_4 + 270) + 2L_8L_{11} \cos \theta_4 \cos \theta_{11} \\
 & + 2L_8L_{11} \text{sen} \theta_4 \text{sen} \theta_{11} + 2L_{11}L_{10} \cos \theta_{11} \cos(\theta_4 + 270) \\
 & + 2L_{11}L_{10} \text{sen} \theta_{11} \text{sen}(\theta_4 + 270) - R_2^2
 \end{aligned} \tag{2.28}$$

$$\begin{aligned}
 f'(\theta_{11}) = & -2L_8L_{11} \cos \theta_4 \text{sen} \theta_{11} + 2L_8L_{11} \text{sen} \theta_4 \cos \theta_{11} \\
 & - 2L_{11}L_{10} \text{sen} \theta_{11} \cos(\theta_4 + 270) + 2L_{11}L_{10} \cos \theta_{11} \text{sen}(\theta_4 + 270)
 \end{aligned} \tag{2.29}$$

Sustituyendo (2.28) y (2.29) en (2.5):

$$\theta_{11i+1} = \theta_{11i} - \frac{f(\theta_{11i})}{f'(\theta_{11i})} \tag{2.30}$$

### Calculando $\theta_9$ :

De las ecuaciones (2.25) y (2.26), se despeja el término que tenga  $\theta_{11}$ , 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:

$$\begin{aligned}
 L_{11}^2 = & L_8^2 + L_{10}^2 + R_2^2 + 2L_8L_{10} \cos \theta_4 \cos(\theta_4 + 270) \\
 & + 2L_8L_{10} \text{sen} \theta_4 \text{sen}(\theta_4 + 270) - 2L_8R_2 \cos \theta_4 \cos \theta_9 \\
 & - 2L_8R_2 \text{sen} \theta_4 \text{sen} \theta_9 - 2R_2L_{10} \cos \theta_9 \cos(\theta_4 + 270) \\
 & - 2R_2L_{10} \text{sen} \theta_9 \text{sen}(\theta_4 + 270)
 \end{aligned} \tag{2.31}$$

Aplicando el método de Newton – Raphson, podemos conocer con (2.31) mediante iteraciones a  $\theta_9$ . El ángulo  $\theta_4$  ya se calculó, por lo tanto es conocido.

Resolviendo para (2.31):

$$\begin{aligned}
 f(\theta_9) = & L_8^2 + L_{10}^2 + R_2^2 + 2L_8L_{10} \cos \theta_4 \cos(\theta_4 + 270) \\
 & + 2L_8L_{10} \text{sen} \theta_4 \text{sen}(\theta_4 + 270) - 2L_8R_2 \cos \theta_4 \cos \theta_9 \\
 & - 2L_8R_2 \text{sen} \theta_4 \text{sen} \theta_9 - 2R_2L_{10} \cos \theta_9 \cos(\theta_4 + 270) \\
 & - 2R_2L_{10} \text{sen} \theta_9 \text{sen}(\theta_4 + 270) - L_{11}^2
 \end{aligned} \tag{2.32}$$

$$\begin{aligned}
 f'(\theta_9) = & 2L_8R_2 \cos \theta_4 \cos \theta_9 - 2L_8R_2 \text{sen} \theta_4 \cos \theta_9 \\
 & + 2R_2L_{10} \text{sen} \theta_9 \cos(\theta_4 + 270) - 2R_2L_{10} \cos \theta_9 \text{sen}(\theta_4 + 270)
 \end{aligned} \tag{2.33}$$



Sustituyendo (2.32) y (2.33) en (2.5):

$$\theta_{9i+1} = \theta_{9i} - \frac{f(\theta_9)}{f'(\theta_9)} \quad (2.34)$$

De la segunda cadena cinemática obtenemos la siguiente ecuación vectorial (ver figura 10):

$$\bar{r}_{13} + \bar{r}_{14} - \bar{r}_{12} - \bar{r}_{11} = 0 \quad (2.35)$$

Si sustituimos las componentes en i y j de cada vector en la ecuación (2.35) nos dará como resultado 2 ecuaciones:

$$L_{13} \cos\theta_{13} + L_{14} \cos\theta_{14} - L_{12} \cos\theta_{12} - L_{11} \cos\theta_{11} = 0 \quad (2.36)$$

$$L_{13} \text{sen}\theta_{13} + L_{14} \text{sen}\theta_{14} - L_{12} \text{sen}\theta_{12} - L_{11} \text{sen}\theta_{11} = 0 \quad (2.37)$$

De las ecuaciones (2.36) y (2.37), se despeja el término que tenga  $\theta_{12}$ , 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

$$\begin{aligned} L_{12}^2 &= L_{13}^2 + L_{14}^2 + L_{11}^2 + 2L_{13}L_{14} \cos \theta_{14} \cos (\theta_4 + 270) \\ &+ 2L_{13}L_{14} \text{sen} \theta_{14} \text{sen} (\theta_4 + 270) - 2L_{13}L_{11} \cos (\theta_4 + 270) \cos \theta_{11} \\ &- 2L_{13}L_{11} \text{sen} (\theta_4 + 270) \text{sen} \theta_{11} - 2L_{14}L_{11} \cos \theta_{14} \cos \theta_{11} \\ &- 2L_{14}L_{11} \text{sen} \theta_{14} \text{sen} \theta_{11} \end{aligned} \quad (2.38)$$

Aplicando el método de Newton – Raphson, podemos conocer con (2.38) mediante iteraciones a  $\theta_{14}$ . El ángulo  $\theta_4$  y  $\theta_{11}$  ya se calcularon, por lo tanto son conocidos.

Resolviendo para (2.38):

$$\begin{aligned} f(\theta_{14}) &= L_{13}^2 + L_{14}^2 + L_{11}^2 + 2L_{13}L_{14} \cos \theta_{14} \cos (\theta_4 + 270) \\ &+ 2L_{13}L_{14} \text{sen} \theta_{14} \text{sen} (\theta_4 + 270) - 2L_{13}L_{11} \cos (\theta_4 + 270) \cos \theta_{11} \\ &- 2L_{13}L_{11} \text{sen} (\theta_4 + 270) \text{sen} \theta_{11} - 2L_{14}L_{11} \cos \theta_{14} \cos \theta_{11} \\ &- 2L_{14}L_{11} \text{sen} \theta_{14} \text{sen} \theta_{11} - L_{12}^2 \end{aligned} \quad (2.39)$$



$$f'(\theta_{14}) = -2L_{13}L_{14}\text{sen } \theta_{14} \cos(\theta_4 + 270) + 2L_{13}L_{14} \cos \theta_{14}\text{sen}(\theta_4 + 270) + 2L_{14}L_{11}\text{sen } \theta_{14} \cos \theta_{11} - 2L_{14}L_{11} \cos \theta_{14}\text{sen } \theta_{11} \quad (2.40)$$

Sustituyendo (2.39) y (2.40) en (2.5):

$$\theta_{14i+1} = \theta_{14i} - \frac{f(\theta_{14})}{f'(\theta_{14})} \quad (2.41)$$

**Calculando  $\theta_{12}$ :**

De las ecuaciones (2.36) y (2.37), se despeja el término que tenga  $\theta_{14}$ , 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:

$$L_{14}^2 = L_{13}^2 + L_{12}^2 + L_{11}^2 - 2L_{13}L_{12} \cos \theta_{12} \cos(\theta_4 + 270) - 2L_{13}L_{12}\text{sen } \theta_{12}\text{sen}(\theta_4 + 270) - 2L_{13}L_{11} \cos(\theta_4 + 270) \cos \theta_{11} - 2L_{13}L_{11}\text{sen}(\theta_4 + 270)\text{sen } \theta_{11} + 2L_{12}L_{11} \cos \theta_{12} \cos \theta_{11} + 2L_{12}L_{11}\text{sen } \theta_{12}\text{sen } \theta_{11} \quad (2.42)$$

Aplicando el método de Newton – Raphson, podemos conocer con (2.42) mediante iteraciones a  $\theta_{12}$ . El ángulo  $\theta_4$  y  $\theta_{11}$  ya se calcularon, por lo tanto son conocidos.

Resolviendo para (2.42):

$$f(\theta_{12}) = L_{13}^2 + L_{12}^2 + L_{11}^2 - 2L_{13}L_{12} \cos \theta_{12} \cos(\theta_4 + 270) - 2L_{13}L_{12}\text{sen } \theta_{12}\text{sen}(\theta_4 + 270) - 2L_{13}L_{11} \cos(\theta_4 + 270) \cos \theta_{11} - 2L_{13}L_{11}\text{sen}(\theta_4 + 270)\text{sen } \theta_{11} + 2L_{12}L_{11} \cos \theta_{12} \cos \theta_{11} + 2L_{12}L_{11}\text{sen } \theta_{12}\text{sen } \theta_{11} - L_{14}^2 \quad (2.43)$$

$$f'(\theta_{12}) = 2L_{13}L_{12}\text{sen } \theta_{12} \cos(\theta_4 + 270) - 2L_{13}L_{12} \cos \theta_{12}\text{sen}(\theta_4 + 270) - 2L_{12}L_{11}\text{sen } \theta_{12} \cos \theta_{11} + 2L_{12}L_{11} \cos \theta_{12}\text{sen } \theta_{11} \quad (2.44)$$



Sustituyendo (2.439) y (2.44) en (2.5):

$$\theta_{12i+1} = \theta_{12i} - \frac{f(\theta_{12i})}{f'(\theta_{12i})} \quad (2.45)$$

Para calcular la posición del efector final (x,y) nos apoyaremos en la figura 10. Los ángulos  $\alpha_3$  y  $\beta_3$  son constantes, esto porque los eslabones que forman los triángulos están unidos uno de otro perpendicularmente. Los ángulos  $\alpha$  y  $\beta$  son variables.

$$\alpha = \theta_4 - \alpha_3 \quad (2.46)$$

$$\beta = \theta_{14} + \beta_3 \quad (2.47)$$

Por lo tanto la posición es:

$$x = a \cos \alpha + b \cos \beta \quad (2.48)$$

$$y = a \sin \alpha + b \sin \beta \quad (2.49)$$

Las ecuaciones (2.8), (2.12), (2.19), (2.23), (2.30), (2.34), (2.41), (2.45), (2.46), (2.47), (2.48) y (2.49) nos servirán para hacer el algoritmo de programación del manipulador, esto se verá en la parte de simulación.

### 2.1.2. Modelo cinemático inverso del robot manipulador.

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

De la figura 11, observamos que están dimensionados solo unos elementos, esto es porque solo estos serán utilizados para nuestro análisis inicial. En la figura 12 se observa con mayor claridad los dos triángulos rectángulos que nos servirán para calcular  $\theta_4$ ,  $\theta_{14}$ ,  $\theta_5$  y  $\theta_{15}$ , y posteriormente mediante el método de Newton – Raphson el resto de los ángulos.



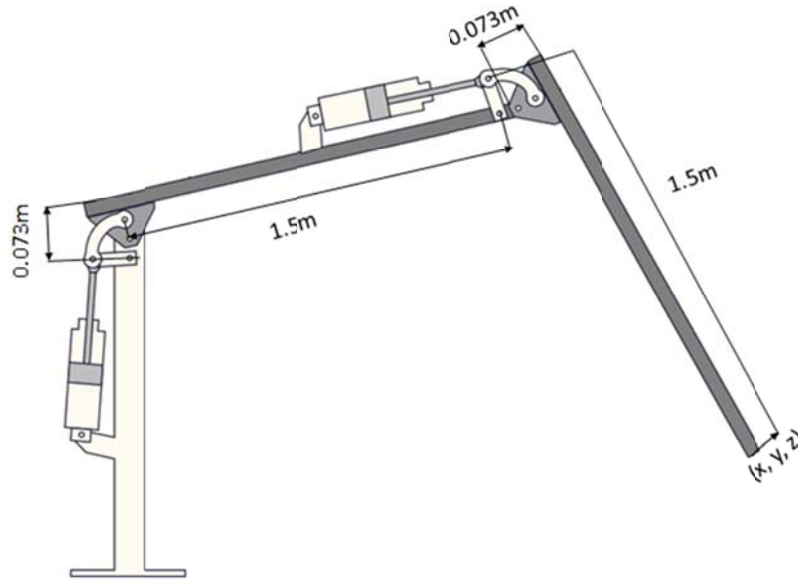


Figura 11. Dimensiones de los elementos que se utilizaran para el cálculo inicial.

Los ángulos  $\alpha_3$  y  $\beta_3$  son constantes, esto porque los eslabones que forman los triángulos están unidos uno de otro perpendicularmente. Los ángulos  $\alpha$  y  $\beta$  son variables y dependen de la posición del efector final.

Partiendo de la posición de efector final, tenemos (12):

$$x = a \cos \alpha + b \cos \beta \quad (2.50)$$

$$y = a \sin \alpha + b \sin \beta \quad (2.51)$$

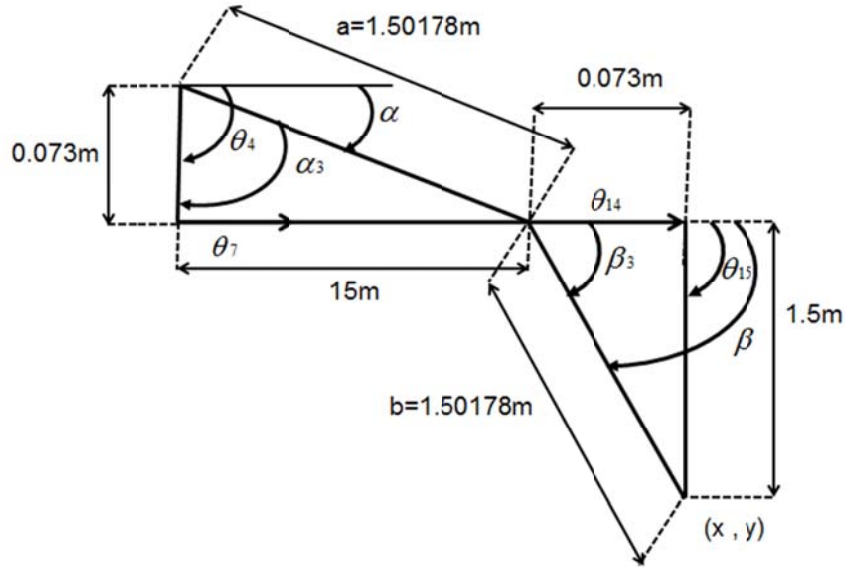


Figura 12. Triángulos rectángulos útiles para el cálculo inicial donde se obtienen  $\theta_4$  y  $\theta_{14}$ ,

De las ecuaciones (2.50) y (2.51), se despeja el término que tenga  $\beta$ , se eleva al cuadrado ambos lados y sumamos las dos ecuaciones,  $x$  y  $y$  son conocidos. Nos dará como resultado la siguiente ecuación:

$$b^2 = a^2 + x^2 + y^2 - 2ax \cos \alpha - 2aysen \alpha \quad (2.52)$$

Aplicando el método de Newton–Raphson, podemos conocer con (2.52) mediante iteraciones a  $\alpha$ .

Resolviendo para (2.52):

$$f(\alpha) = a^2 + x^2 + y^2 - 2ax \cos \alpha - 2aysen \alpha - b^2 \quad (2.53)$$

$$f'(\alpha) = 2axsen \alpha - 2ay \cos \alpha \quad (2.54)$$

Sustituyendo (2.53) y (2.54) en (2.5):

$$\alpha_{i+1} = \alpha_i - \frac{a^2 + x^2 + y^2 - 2ax \cos \alpha - 2aysen \alpha - b^2}{2axsen \alpha - 2ay \cos \alpha} \quad (2.55)$$



**Calculando  $\beta$ :**

De las ecuaciones (2.50) y (2.51), se despeja el término que tenga  $\alpha$ , se eleva al cuadrado ambos lados y sumamos las dos ecuaciones,  $x$  y  $y$  son conocidos. Nos dará como resultado la siguiente ecuación:

$$a^2 = b^2 + x^2 + y^2 - 2bx \cos \alpha - 2by \sin \alpha \quad (2.56)$$

Aplicando el método de Newton–Raphson, se puede conocer con (2.56) mediante iteraciones a  $\beta$ .

Resolviendo para (2.56):

$$f(\beta) = b^2 + x^2 + y^2 - 2bx \cos \alpha - 2by \sin \alpha - a^2 \quad (2.57)$$

$$f'(\beta) = 2bx \sin \alpha - 2by \cos \alpha \quad (2.58)$$

Sustituyendo (2.57) y (2.58) en (2.5):

$$\beta_{i+1} = \beta_i - \frac{b^2 + x^2 + y^2 - 2bx \cos \beta - 2by \sin \beta - a^2}{2bx \sin \beta - 2by \cos \beta} \quad (2.59)$$

De la figura 12 observamos:

$$\theta_4 = \alpha + \alpha_3 \quad (2.60)$$

$$\theta_5 = \theta_4 + 270 \quad (2.61)$$

$$\theta_{14} = \beta - \beta_3 \quad (2.62)$$

$$\theta_{15} = \theta_{14} + 90 \quad (2.63)$$

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.

### Primer mecanismo.

Del primer mecanismo se observan dos cadenas cinemáticas cerradas, las cuales utilizaremos para obtener los ángulos  $\theta_1$ ,  $\theta_2$  y  $\theta_3$ . La primera cadena la forman L2, L3, L4 y L5; la segunda cadena está formada por R, L2 y Lm, (ver figura 13)

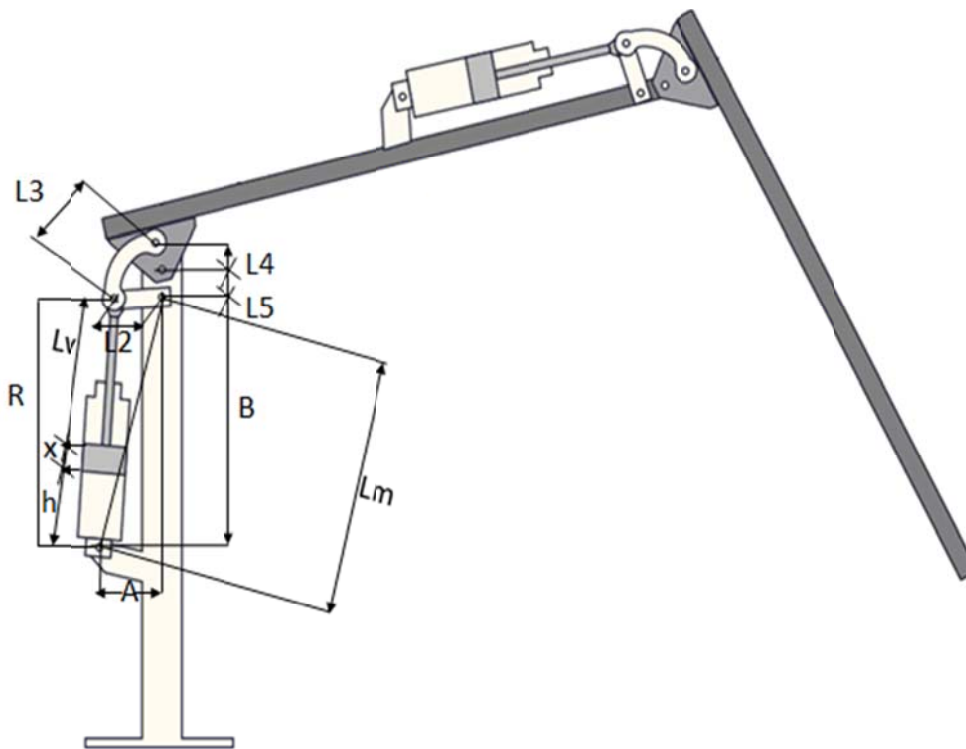


Figura 13. Primer mecanismo corredera – manivela de cuatro barras, con dos cadenas cinemáticas cerradas.

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

$$\vec{r}_2 + \vec{r}_3 - \vec{r}_4 - \vec{r}_5 = 0 \quad (2.64)$$

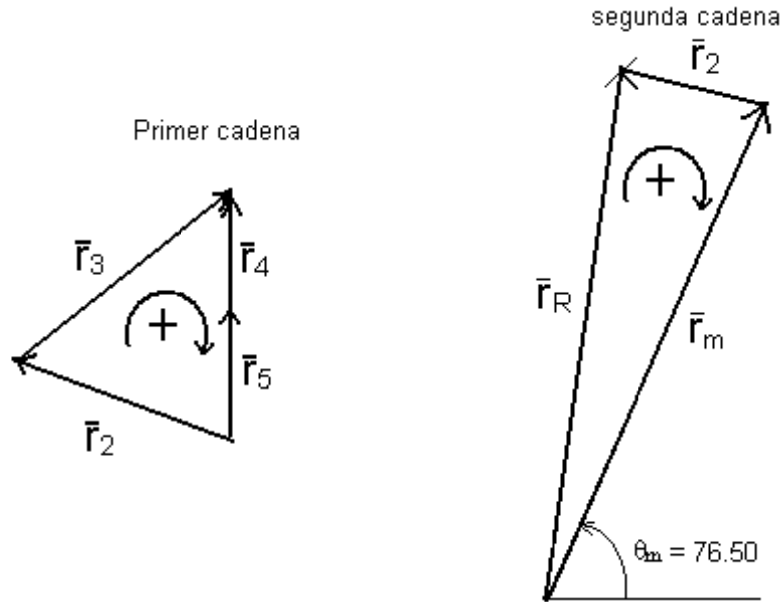


Figura 14. Cadenas cinemáticas cerradas del primer mecanismo.

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

$$L_2 \cos \theta_2 + L_3 \cos \theta_3 - L_4 \cos \theta_4 = 0 \quad (2.65)$$

$$L_2 \text{sen} \theta_2 + L_3 \text{sen} \theta_3 - L_4 \text{sen} \theta_4 - L_5 = 0 \quad (2.66)$$

De las ecuaciones (4.94) y (4.95), se despeja el término que tenga  $\theta_2$ , se eleva al cuadrado ambos lados y sumamos las dos ecuaciones, tomando en cuenta que  $\theta_2$  es conocido, ya fue calculado anteriormente. Nos dará como resultado la siguiente ecuación:

$$L_2^2 = L_5^2 + L_4^2 + L_3^2 - 2L_4L_3 \cos \theta_4 \cos \theta_3 - 2L_4L_3 \text{sen} \theta_4 \text{sen} \theta_3 + 2L_5L_4 \text{sen} \theta_4 - 2L_5L_3 \text{sen} \theta_3 \quad (2.67)$$

Aplicando el método de Newton–Raphson, podemos conocer con (2.67) mediante iteraciones a  $\theta_3$ .

Resolviendo para (2.67):



$$f(\theta_3) = L_5^2 + L_4^2 + L_3^2 - 2L_4L_3 \cos \theta_4 \cos \theta_3 - 2L_4L_3 \sin \theta_4 \sin \theta_3 + 2L_5L_4 \sin \theta_4 - 2L_5L_3 \sin \theta_3 - L_2^2 \quad (2.68)$$

$$f'(\theta_3) = 2L_4L_3 \cos \theta_4 \sin \theta_3 - 2L_4L_3 \sin \theta_4 \cos \theta_3 - 2L_5L_3 \cos \theta_3 \quad (2.69)$$

Sustituyendo (2.68) y (2.69) en (2.5):

$$\theta_{3i+1} = \theta_{3i} - \frac{f(\theta_3)}{f'(\theta_3)} \quad (2.70)$$

**Calculando  $\theta_2$ :**

De las ecuaciones (2.65) y (2.66), se despeja el término que tenga  $\theta_3$ , se eleva al cuadrado ambos lados y sumamos las dos ecuaciones, tomando en cuenta que  $\theta_2$  es conocido, ya fue calculado anteriormente. Nos dará como resultado la siguiente ecuación:

$$L_3^2 = L_5^2 + L_4^2 + L_2^2 - 2L_4L_2 \cos \theta_4 \cos \theta_2 - 2L_4L_2 \sin \theta_4 \sin \theta_2 + 2L_5L_4 \sin \theta_4 - 2L_5L_2 \sin \theta_2 \quad (2.71)$$

Aplicando el método de Newton – Raphson, podemos conocer con (2.71) mediante iteraciones a  $\theta_2$ .

Resolviendo para (2.71):

$$f(\theta_2) = L_5^2 + L_4^2 + L_2^2 - 2L_4L_2 \cos \theta_4 \cos \theta_2 - 2L_4L_2 \sin \theta_4 \sin \theta_2 + 2L_5L_4 \sin \theta_4 - 2L_5L_2 \sin \theta_2 - L_3^2 \quad (2.72)$$

$$f'(\theta_2) = 2L_4L_2 \cos \theta_4 \sin \theta_2 - 2L_4L_2 \sin \theta_4 \cos \theta_2 - 2L_5L_2 \cos \theta_2 \quad (2.73)$$

Sustituyendo (2.72) y (2.73) en (2.5):

$$\theta_{2i+1} = \theta_{2i} - \frac{f(\theta_2)}{f'(\theta_2)} \quad (2.74)$$



De la segunda cadena cinemática obtenemos la siguiente ecuación vectorial (ver figura 2.10):

$$\vec{r}_R - \vec{r}_2 - \vec{r}_m = 0 \quad (2.75)$$

Si sustituimos las componentes en i y j de cada vector en la ecuación (2.75) nos dará como resultado 2 ecuaciones:

$$R \cos \theta_1 - L_2 \cos \theta_2 - L_m \cos \theta_m = 0 \quad (2.76)$$

$$R \operatorname{sen} \theta_1 - L_2 \operatorname{sen} \theta_2 - L_m \operatorname{sen} \theta_m = 0 \quad (2.77)$$

De las ecuaciones (2.76) y (2.77), se despeja el término que tenga  $\theta_1$ , se eleva al cuadrado ambos lados y sumamos las dos ecuaciones. Nos dará como resultado la siguiente ecuación:

$$R^2 = L_2^2 + L_m^2 + 2L_2L_m \cos \theta_2 \cos \theta_m + 2L_2L_m \operatorname{sen} \theta_2 \operatorname{sen} \theta_m \quad (2.78)$$

Con la ecuación anterior podemos conocer R y con ésta, el desplazamiento (x) del vástago del pistón del primer mecanismo. Los ángulos  $\theta_2$  y  $\theta_m$  son conocidos;  $\theta_m$  es constante.

$$R = \left( L_2^2 + L_m^2 + 2L_2L_m \cos \theta_2 \cos \theta_m + 2L_2L_m \operatorname{sen} \theta_2 \operatorname{sen} \theta_m \right)^{\frac{1}{2}} \quad (2.79)$$

Por lo tanto el desplazamiento del vástago (x) es (ver figura 13):

$$x = R - Lv - h \quad (2.80)$$

### **Calculando $\theta_1$ :**

De las ecuaciones (2.76) y (2.77), se despeja el término que tenga  $\theta_m$ , se eleva al cuadrado ambos lados y sumamos las dos ecuaciones. Nos dará como resultado la siguiente ecuación:

$$L_m^2 = L_2^2 + R^2 - 2L_2R \cos \theta_2 \cos \theta_1 - 2L_2R \operatorname{sen} \theta_2 \operatorname{sen} \theta_1 \quad (2.81)$$

Aplicando el método de Newton – Raphson, podemos conocer con (2.81) mediante iteraciones a  $\theta_1$ , considerando que  $\theta_2$  es conocido.

Resolviendo para (2.81):

$$f(\theta_1) = L_2^2 + R^2 - 2L_2R \cos \theta_2 \cos \theta_1 - 2L_2R \sin \theta_2 \sin \theta_1 - L_m^2 \quad (2.82)$$

$$f'(\theta_1) = 2L_2R \cos \theta_2 \sin \theta_1 - 2L_2R \sin \theta_2 \cos \theta_1 \quad (2.83)$$

Sustituyendo (2.82) y (2.83) en (2.5):

$$\theta_{i+1} = \theta_i - \frac{L_2^2 + R^2 - 2L_2R \cos \theta_2 \cos \theta_i - 2L_2R \sin \theta_2 \sin \theta_i - L_m^2}{2L_2R \cos \theta_2 \sin \theta_i - 2L_2R \sin \theta_2 \cos \theta_i} \quad (2.84)$$

### Segundo mecanismo.

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 R2 (ver figura 15)

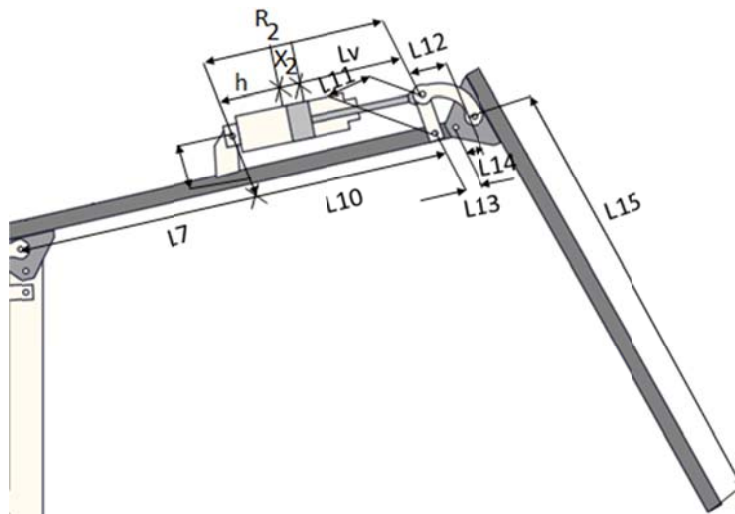


Figura 15 Segundo mecanismo corredera – manivela de cuatro barras, con dos cadenas cinemáticas cerradas.



De la primera cadena cinemática obtenemos la siguiente ecuación vectorial (ver figura 16):

$$\bar{r}_{13} + \bar{r}_{14} - \bar{r}_{12} - \bar{r}_{11} = 0 \quad (2.85)$$

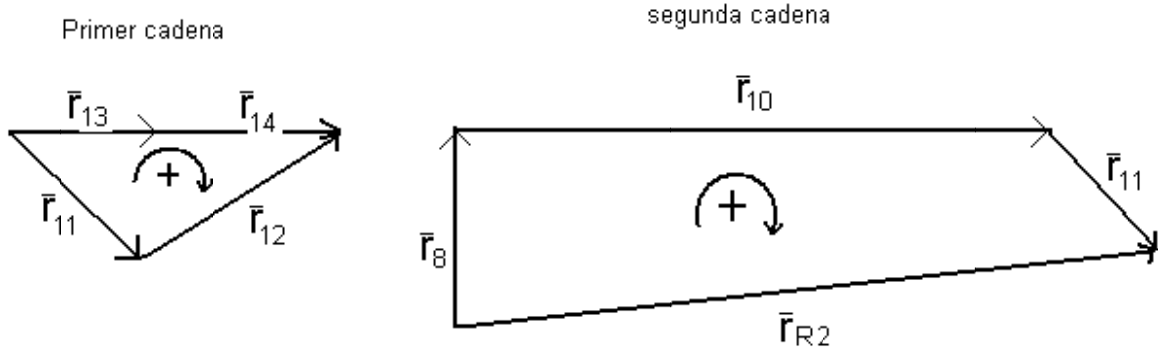


Figura 16. Cadenas cinemáticas cerradas del segundo mecanismo.

Si sustituimos las componentes en  $i$  y  $j$  de cada vector en la ecuación (2.85) nos dará como resultado 2 ecuaciones:

$$L_{13} \cos \theta_{13} + L_{14} \cos \theta_{14} - L_{12} \cos \theta_{12} - L_{11} \cos \theta_{11} = 0 \quad (2.86)$$

$$L_{13} \text{sen} \theta_{13} + L_{14} \text{sen} \theta_{14} - L_{12} \text{sen} \theta_{12} - L_{11} \text{sen} \theta_{11} = 0 \quad (2.87)$$

De las ecuaciones (2.86) y (2.87), se despeja el término que tenga  $\theta_{12}$ , 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:

$$\begin{aligned} L_{12}^2 &= L_{13}^2 + L_{14}^2 + L_{11}^2 + 2L_{13}L_{14} \cos \theta_{14} \cos (\theta_4 + 270) \\ &+ 2L_{13}L_{14} \text{sen} \theta_{14} \text{sen} (\theta_4 + 270) - 2L_{13}L_{11} \cos (\theta_4 + 270) \cos \theta_{11} \\ &- 2L_{13}L_{11} \text{sen} (\theta_4 + 270) \text{sen} \theta_{11} - 2L_{14}L_{11} \cos \theta_{14} \cos \theta_{11} \\ &- 2L_{14}L_{11} \text{sen} \theta_{14} \text{sen} \theta_{11} \end{aligned} \quad (2.88)$$



Aplicando el método de Newton - Raphson, podemos conocer con (2.88) mediante iteraciones a  $\theta_{11}$ , el ángulo  $\theta_4$  y  $\theta_{14}$  ya se calcularon, por lo tanto son conocidos.

Resolviendo para (2.88):

$$\begin{aligned}
 f(\theta_{11}) = & L_{13}^2 + L_{14}^2 + L_{11}^2 + 2L_{13}L_{14} \cos \theta_{14} \cos(\theta_4 + 270) \\
 & + 2L_{13}L_{14} \text{sen } \theta_{14} \text{sen}(\theta_4 + 270) - 2L_{13}L_{11} \cos(\theta_4 + 270) \cos \theta_{11} \\
 & - 2L_{13}L_{11} \text{sen}(\theta_4 + 270) \text{sen } \theta_{11} - 2L_{14}L_{11} \cos \theta_{14} \cos \theta_{11} \\
 & - 2L_{14}L_{11} \text{sen } \theta_{14} \text{sen } \theta_{11} - L_{12}^2
 \end{aligned} \tag{2.89}$$

$$\begin{aligned}
 f'(\theta_{11}) = & 2L_{13}L_{11} \cos(\theta_4 + 270) \text{sen } \theta_{11} - 2L_{13}L_{11} \text{sen}(\theta_4 + 270) \cos \theta_{11} \\
 & + 2L_{14}L_{11} \cos \theta_{14} \text{sen } \theta_{11} - 2L_{14}L_{11} \text{sen } \theta_{14} \cos \theta_{11}
 \end{aligned} \tag{2.90}$$

Sustituyendo (2.89) y (2.90) en (2.5):

$$\theta_{11i+1} = \theta_{11i} - \frac{f(\theta_{11i})}{f'(\theta_{11i})} \tag{2.91}$$

**Calculando  $\theta_{12}$ :**

De las ecuaciones (2.86) y (2.87), se despeja el término que tenga  $\theta_{11}$ , 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:

$$\begin{aligned}
 L_{11}^2 = & L_{13}^2 + L_{12}^2 + L_{14}^2 + 2L_{13}L_{14} \cos(\theta_4 + 270) \cos \theta_{14} \\
 & + 2L_{13}L_{14} \text{sen}(\theta_4 + 270) \text{sen } \theta_{14} - 2L_{13}L_{12} \cos \theta_{12} \cos(\theta_4 + 270) \\
 & - 2L_{13}L_{12} \text{sen } \theta_{12} \text{sen}(\theta_4 + 270) - 2L_{12}L_{14} \cos \theta_{12} \cos \theta_{14} \\
 & - 2L_{12}L_{14} \text{sen } \theta_{12} \text{sen } \theta_{14}
 \end{aligned} \tag{2.92}$$

Aplicando el método de Newton–Raphson, podemos conocer con (2.92) mediante iteraciones a  $\theta_{12}$ . El ángulo  $\theta_4$  y  $\theta_{11}$  ya se calcularon, por lo tanto son conocidos.



Resolviendo para (2.92):

$$\begin{aligned}
 f(\theta_{12}) &= L_{13}^2 + L_{12}^2 + L_{14}^2 + 2L_{13}L_{14} \cos(\theta_4 + 270) \cos \theta_{14} \\
 &+ 2L_{13}L_{14} \text{sen}(\theta_4 + 270) \text{sen} \theta_{14} - 2L_{13}L_{12} \cos \theta_{12} \cos(\theta_4 + 270) \\
 &- 2L_{13}L_{12} \text{sen} \theta_{12} \text{sen}(\theta_4 + 270) - 2L_{12}L_{14} \cos \theta_{12} \cos \theta_{14} \\
 &- 2L_{12}L_{14} \text{sen} \theta_{12} \text{sen} \theta_{14} - L_{11}^2
 \end{aligned} \tag{2.93}$$

$$\begin{aligned}
 f'(\theta_{12}) &= 2L_{13}L_{14} \text{sen} \theta_{12} \cos(\theta_4 + 270) - 2L_{13}L_{12} \cos \theta_{12} \text{sen}(\theta_4 + 270) \\
 &+ 2L_{12}L_{14} \text{sen} \theta_{12} \cos \theta_{14} - 2L_{12}L_{14} \cos \theta_{12} \text{sen} \theta_{14} - L_{11}^2
 \end{aligned} \tag{2.94}$$

Sustituyendo (2.93) y (2.94) en (2.5):

$$\theta_{12i+1} = \theta_{12i} - \frac{f(\theta_{12})}{f'(\theta_{12})} \tag{2.95}$$

De la segunda cadena cinemática obtenemos la siguiente ecuación vectorial (ver figura 16):

$$\bar{r}_8 + \bar{r}_{10} + \bar{r}_{11} - \bar{r}_{R2} = 0 \tag{2.96}$$

Si sustituimos las componentes en i y j de cada vector en la ecuación (2.4.96) nos dará como resultado 2 ecuaciones:

$$L_8 \cos \theta_8 + L_{10} \cos \theta_{10} + L_{11} \cos \theta_{11} - R_2 \cos \theta_9 = 0 \tag{2.97}$$

$$L_8 \text{sen} \theta_8 + L_{10} \text{sen} \theta_{10} + L_{11} \text{sen} \theta_{11} - R_2 \text{sen} \theta_9 = 0 \tag{2.98}$$

De las ecuaciones (2.97) y (2.98), se despeja el término que tenga  $\theta_9$ , 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:



$$\begin{aligned}
 R_2^2 = & L_8^2 + L_{10}^2 + L_{11}^2 + 2L_8L_{10} \cos \theta_4 \cos (\theta_4 + 270) \\
 & + 2L_8L_{10} \text{sen } \theta_4 \text{sen } (\theta_4 + 270) + 2L_8L_{11} \cos \theta_4 \cos \theta_{11} \\
 & + 2L_8L_{11} \text{sen } \theta_4 \text{sen } \theta_{11} + 2L_{11}L_{10} \cos \theta_{11} \cos (\theta_4 + 270) \\
 & + 2L_{11}L_{10} \text{sen } \theta_{11} \text{sen } (\theta_4 + 270)
 \end{aligned}
 \tag{2.99}$$

Con la ecuación anterior podemos conocer  $R_2$  y con ésta, el desplazamiento ( $x_2$ ) del vástago del pistón del segundo mecanismo. Los ángulos  $\theta_4$  y  $\theta_{11}$  son conocidos.

$$R_2 = \left[ \begin{aligned} & L_8^2 + L_{10}^2 + L_{11}^2 + 2L_8L_{10} \cos \theta_4 \cos (\theta_4 + 270) \\ & + 2L_8L_{10} \text{sen } \theta_4 \text{sen } (\theta_4 + 270) + 2L_8L_{11} \cos \theta_4 \cos \theta_{11} \\ & + 2L_8L_{11} \text{sen } \theta_4 \text{sen } \theta_{11} + 2L_{11}L_{10} \cos \theta_{11} \cos (\theta_4 + 270) \\ & + 2L_{11}L_{10} \text{sen } \theta_{11} \text{sen } (\theta_4 + 270) \end{aligned} \right]^{\frac{1}{2}}
 \tag{2.100}$$

Por lo tanto el desplazamiento del vástago ( $x_2$ ) es (ver figura 3.7):

$$x_2 = R_2 - Lv - h
 \tag{2.101}$$

Las ecuaciones (2.56), (2.59), (2.60), (2.61), (2.62), (2.63), (2.70), (2.74), (2.80), (2.84), (2.91), (2.95), (2.100) y (2.101) nos servirán para hacer el algoritmo de programación del manipulador, esto se verá en la parte de simulación.

## 2.2 Modelo dinámico del robot manipulador.

La dinámica se ocupa de la relación entre las fuerzas que actúan sobre un cuerpo y el movimiento que en él se origina. Por lo tanto, el modelo dinámico de un robot tiene por objeto conocer la relación entre el movimiento del robot y las fuerzas implicadas en el mismo.

Esta relación se obtiene mediante el denominado modelo dinámico, que relaciona matemáticamente:



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

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

El modelo dinámico debe ser resuelto entonces de manera iterativa mediante la utilización de un procedimiento numérico. El problema de la obtención del modelo dinámico de un robot es, por lo tanto, uno de los aspectos más complejos de la robótica, lo que ha llevado a ser obviado en numerosas ocasiones. Sin embargo, el modelo dinámico es imprescindible para conseguir los siguientes fines:

1. Simulación del movimiento del robot.
2. Diseño y evaluación de la estructura mecánica del robot.
3. Dimensionamiento de los actuadores.
4. Diseño y evaluación del control dinámico del robot.

Este último fin es evidentemente de gran importancia, pues de la calidad del control dinámico del robot depende la precisión y velocidad de sus movimientos. La gran complejidad ya comentada existente en la obtención del modelo dinámico



del robot, ha motivado que se realicen ciertas simplificaciones, de manera que así pueda ser utilizado en el diseño del controlador.

Es importante hacer notar que el modelo dinámico completo de un robot debe incluir no solo la dinámica de sus elementos (barras o eslabones) sino también la propia de sus sistemas de transmisión, de los actuadores y sus equipos electrónicos de mando. Estos elementos incorporan al modelo dinámico nuevas inercias, rozamientos, saturaciones de los circuitos electrónicos, etc. aumentando aún más su complejidad.

Por último, es preciso señalar que si bien en la mayor parte de las aplicaciones reales de robótica, las cargas e inercias manejadas no son suficientes como para originar deformaciones en los eslabones del robot, en determinadas ocasiones no ocurre así, siendo preciso considerar al robot como un conjunto de eslabones no rígidos. Aplicaciones de este tipo pueden encontrarse en la robótica espacial o en robots de grandes dimensiones.

En el desarrollo de este proyecto, es importante señalar que se tomará como base el modelo termo mecánico simplificado, para representar el comportamiento interno del actuador neumático, el cual se muestra en el conjunto de ecuaciones (2.102)[10]:

$$\dot{X} = \frac{d}{dt}X(1a)$$

$$D\dot{X} = \frac{d^2}{dt^2}X(1b)$$

Para  $0 \leq X \leq 0.01905$ :

$$\dot{P}_{a1} = g_{21}(X)(\dot{m}_{a1} - \dot{m}_{c1} - 9.176 \times 10^{-10}P_{a1}DX) \times 10^8(1c)$$

$$\dot{P}_{c1} = g_{31}(X)(\dot{m}_{c1} - 3.608 \times 10^8P_{c1}DX) \times 10^6(1d)$$

Para  $0.01905 \leq X \leq 0.1016$ :

$$\dot{P}_{a1} = g_{22}(X)(\dot{m}_{a1} - 3.7 \times 10^{-8}P_{a1}DX) \times 10^{11}(1e)$$

$$\dot{P}_{c1} = g_{32}(X)(\dot{m}_{c1} - 3.7 \times 10^{-8}P_{c1}DX) \times 10^{11}(1f)$$

Para  $0 \leq X \leq 0.07938$ :

$$\dot{P}_{c2} = g_{41}(X)(\dot{m}_{c2} + 3.469 \times 10^{-8}P_{c2}DX) \times 10^{11}(1g)$$

$$\dot{P}_{a2} = g_{51}(X)(\dot{m}_{a2} + 3.469 \times 10^{-8}P_{a2}DX) \times 10^{11}(1h)$$

Para  $0.07938 \leq X \leq 0.1016$ :

$$\dot{P}_{c2} = g_{42}(X)(\dot{m}_{c2} + 3.352 \times 10^{-8}P_{c2}DX) \times 10^{13}(1i)$$

$$\dot{P}_{a2} = g_{52}(X)[9.983 \times 10^3(\dot{m}_{a2} - \dot{m}_{c2}) + 1.168 \times 10^{-5}P_{a2}DX] \times 10^4(1j)$$

En la figura No. 17 se muestra la estructura simplificada del manipulador flexible, que se utilizó como base para el proyecto [4] y el cual se modificó para el objetivo buscado con tres grados de libertad:

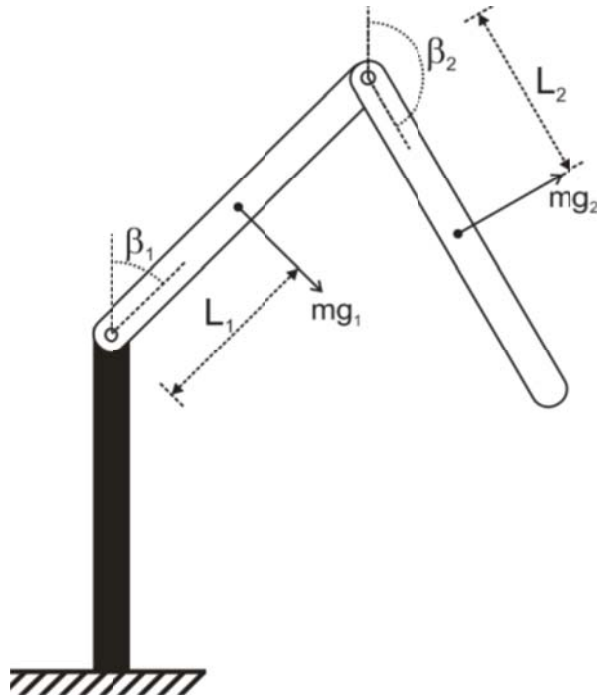


Figura 17. Estructura propuesta de un robot manipulador con tres grados de libertad [4].



Para el sistema propuesto de la figura No. 17, las variables generalizadas son  $\beta_1$  y  $\beta_2$ . La ecuación de velocidad para el centro de gravedad de cada unión es:

$$V_1^2 = L_1^2 \dot{\beta}_1^2 \quad (2.104)$$

$$V_2^2 = 4L_1^2 \dot{\beta}_1^2 + L_2^2 \dot{\beta}_2^2 + 4L_1L_2 \cos(\beta_1 - \beta_2) \dot{\beta}_1 \dot{\beta}_2 \quad (2.105)$$

La energía cinemática del sistema está dada por

$$V_1^2 = L_1^2 \dot{\beta}_1^2 \quad (2.106)$$

$$K = \left( \frac{1}{2} m_1 L_1^2 + 2m_2 L_1^2 + \frac{1}{2} J_1 \right) \dot{\beta}_1^2 + \frac{1}{2} (m_2 L_2^2 + J_2) \dot{\beta}_2^2 + 2m_2 L_1 L_2 \cos(\beta_2 - \beta_1) \dot{\beta}_1 \dot{\beta}_2 \quad (2.107)$$

La energía potencial es

$$U = gL_1 \left( \frac{1}{2} m_1 + m_2 \right) \cos \beta_1 + \frac{1}{2} m_2 g L_2 \cos \beta_2 \quad (2.108)$$

Finalmente, la ecuación Lagrangiana  $L = K - U$ , está dada por:

$$L = \left( \frac{1}{2} m_1 L_1^2 + 2m_2 L_1^2 + \frac{1}{2} J_1 \right) \dot{\beta}_1^2 + \frac{1}{2} (m_2 L_2^2 + J_2) \dot{\beta}_2^2 - gL_1 \left( \frac{1}{2} m_1 + m_2 \right) \cos \beta_1 - \frac{1}{2} m_2 g L_2 \cos \beta_2 + 2m_2 L_1 L_2 \cos(\beta_2 - \beta_1) \dot{\beta}_1 \dot{\beta}_2 \quad (2.109)$$

Las ecuaciones de movimiento, son derivadas como sigue:

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\beta}_1} \right) - \frac{\partial L}{\partial \beta_1} = 0 \text{ y } \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\beta}_2} \right) - \frac{\partial L}{\partial \beta_2} = 0 \quad (2.110)$$

$$2 \left( \frac{1}{2} m_1 L_1^2 + 2m_2 L_1^2 + \frac{1}{2} J_1 \right) \ddot{\beta}_1 + 2m_2 L_1 L_2 \cos(\beta_2 - \beta_1) \ddot{\beta}_2 - 2m_2 L_1 L_2 \sin(\beta_2 - \beta_1) \dot{\beta}_2^2 - gL_1 \left( \frac{1}{2} m_1 + m_2 \right) \sin \beta_1 = 0 \quad (2.111)$$

$$(m_2 L_2^2 + J_2) \ddot{\beta}_2 + 2m_2 L_1 L_2 \cos(\beta_2 - \beta_1) \ddot{\beta}_1 + 2m_2 L_1 L_2 \sin(\beta_2 - \beta_1) \dot{\beta}_1^2 - \frac{1}{2} m_2 g L_2 \sin \beta_2 = 0 \quad (2.112)$$

Para convertir las ecuaciones (2.111) y (2.112) en estados de espacio, las ecuaciones (2.113) son usadas.





$$\begin{aligned}
 X_1 &= \dot{\beta}_1; \dot{X}_1 = \ddot{\beta}_1 \\
 X_2 &= \beta_1; \dot{X}_2 = \dot{\beta}_1 = X_1 \\
 X_3 &= \dot{\beta}_2; \dot{X}_3 = \ddot{\beta}_2 \\
 X_4 &= \beta_2; \dot{X}_4 = \dot{\beta}_2 = X_3
 \end{aligned}
 \tag{2.113}$$

El estado de espacio para las ecuaciones dinámicas es:

$$\begin{aligned}
 &\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C_3 \cos(X_4 - X_2) & 0 & 2C_1 \\ 0 & 0 & 1 & 0 \\ 0 & 2C_2 & 0 & C_3 \cos(X_4 - X_2) \end{bmatrix} \begin{bmatrix} \dot{X}_4 \\ \dot{X}_3 \\ \dot{X}_2 \\ \dot{X}_1 \end{bmatrix} \\
 &= \begin{bmatrix} X_3 \\ C_4 \sin X_2 + C_3 \sin(X_4 - X_2) X_4^2 \\ X_1 \\ C_5 \sin X_4 - C_3 \sin(X_4 - X_2) X_1^2 \end{bmatrix}
 \end{aligned}
 \tag{2.114}$$

Donde:

$$\begin{aligned}
 C_1 &= \frac{1}{2} m_1 L_1^2 + 2m_2 L_1^2 + \frac{1}{2} J_1 \\
 C_2 &= \frac{1}{2} (m_2 L_2^2 + J_2) \\
 C_3 &= 2m_2 L_1 L_2 \\
 C_4 &= g L_1 \left( \frac{1}{2} m_1 + m_2 \right) \\
 C_5 &= \frac{1}{2} m_2 g L_2
 \end{aligned}$$

Considerando la fuerza de fricción, en cada una de las uniones, la ecuación (2.115) está dada por:

$$\begin{aligned}
 &\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C_3 \cos(X_4 - X_2) & 0 & 2C_1 \\ 0 & 0 & 1 & 0 \\ 0 & 2C_2 & 0 & C_3 \cos(X_4 - X_2) \end{bmatrix} \begin{bmatrix} \dot{X}_4 \\ \dot{X}_3 \\ \dot{X}_2 \\ \dot{X}_1 \end{bmatrix} = \begin{bmatrix} X_3 \\ C_4 \sin X_2 + C_3 \sin(X_4 - X_2) X_4^2 - K_2 X_2 \\ X_1 \\ C_5 \sin X_4 - C_3 \sin(X_4 - X_2) X_1^2 - K_1 X_1 \end{bmatrix}
 \end{aligned}
 \tag{2.115}$$

Donde K es la constante de fricción de cada unión. Esta ecuación será probada utilizando el programa Matlab posteriormente, y será parte importante del



proyecto, pues será la base para el trabajo con tres grados de libertad, que es el objetivo final de este proyecto.

# **CAPÍTULO 3**

## **SIMULACIÓN**

### 3. Simulación.

Una vez obtenidos los modelos cinemático directo, cinemático inverso y el dinámico del robot manipulador, el siguiente paso fue diseñar y desarrollar un sistema de software cuyo objetivo es realizar la simulación gráfica de sus movimientos.

#### 3.1 Programación interfaz multilenguaje.

Para esta simulación gráfica, se utilizó una metodología basada en una interfaz multilenguaje; [4], [6], [7] y [8] plantean básicamente la más actual, y fue adaptada para este proyecto, y es como la que se muestra en el diagrama de bloques de la figura No.18.

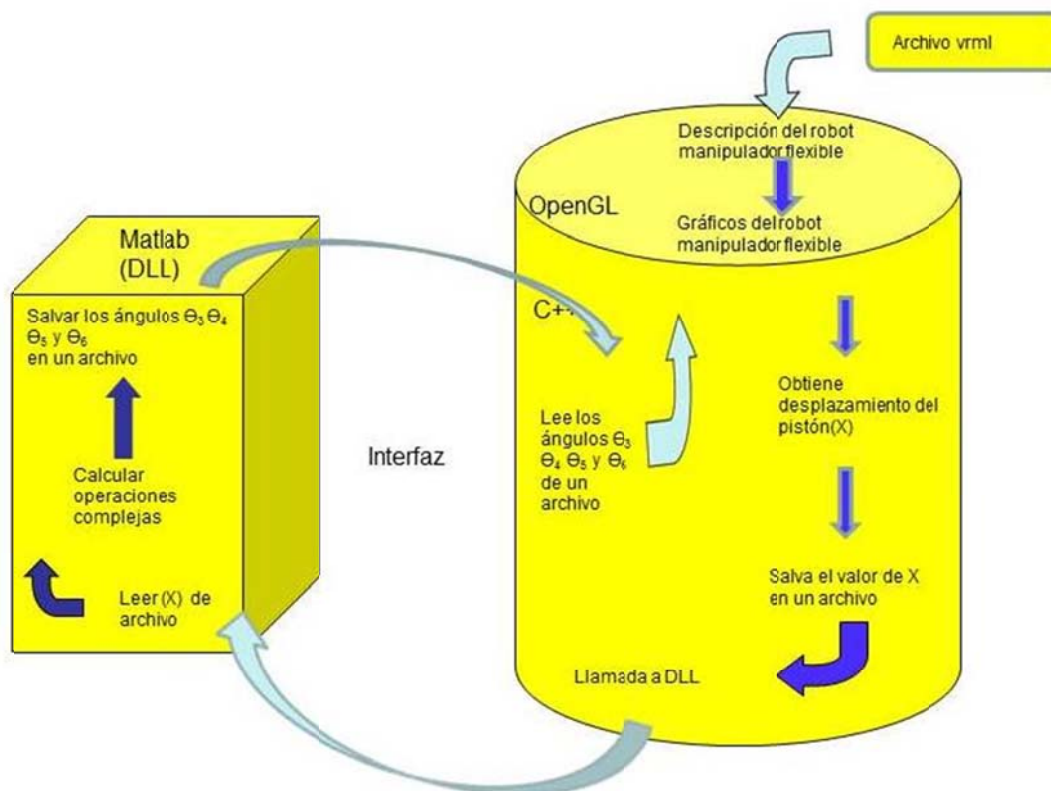


Figura 18. Diagrama de bloques para el desarrollo de una interfaz multilenguaje [4].



De acuerdo con Bertrand Meyer, “la habilidad para mezclar lenguajes ofrece grandes promesas para el futuro de los lenguajes de programación”. La programación multilenguajes presenta un amplio conjunto de beneficios a la ingeniería del software:

*a. Integración de aplicaciones legadas.*

Muchos sistemas o componentes pueden ser aprovechados realizando pequeñas modificaciones en su código. Esta categoría incluye a los sistemas legados o el catálogo de componentes extraídos de la experiencia acumulada en los proyectos que hemos participado.

La compatibilidad entre lenguajes facilita esta tarea, ya que las funcionalidades básicas migradas escritas en un lenguaje pueden luego ser heredadas y extendidas por clases escritas en otro.

*b. Facilidad de migración de bloques concretos de código.*

Determinados bloques de código heredado pueden ser reaprovechados utilizando la característica multilenguaje de la plataforma .NET. De todas formas, en todos los casos será necesario un proceso de migración. Sin embargo, en algunos casos podemos encontrarnos con la ventaja de que la traducción no requiere tanto esfuerzo

Existen casos donde la sintaxis es muy similar, incluso ciertos elementos se podrían automatizar de una manera muy sencilla. La diferencia radica principalmente en la utilización de objetos del framework en lugar de APIs específicas del sistema operativo.

*c. Poder aprovechar las ventajas de cada lenguaje.*

Aprovechar lo mejor de cada lenguaje para el desarrollo del software es también una idea atrayente.



Mediante de un diseño adecuado podemos *adaptar* componentes escritos en cada uno de estos lenguajes con grados de dificultad relativamente pequeños.

Es fundamental tener en cuenta que la información de la aplicación no reside sólo en el código sino en el análisis y en el diseño, los cuales deben estar a un nivel de abstracción suficiente para tener constantemente el control de lo que se está haciendo.

*d. Aprovechar las habilidades y preferencias de cada programador.*

Cada integrante del equipo de desarrollo puede potencialmente aportar más si no está limitado a un lenguaje de programación fijo. Es normal que cada desarrollador conozca más de un lenguaje de programación. Indudablemente será más experto en uno que en otro.

Adoptando la filosofía propuesta, podemos hacer que el desarrollador tenga flexibilidad a la hora de programar, siempre que cumpla con las premisas que especifica el diseño.

Cabe mencionar, que en este proyecto no se pudo realizar directamente la interfaz como originalmente se propone, ya que actualmente no existen herramientas para realizarla con las herramientas de desarrollo seleccionadas para este proyecto, las cuales son Visual Studio 2010 y Matlab 2011, dado que son versiones recientes y los fabricantes respectivos aún están trabajando en ellas. Por tal motivo, los cálculos matemáticos complejos de los modelos cinemático y dinámico se realizaron en MatLab y se pasaron a un archivo texto para que posteriormente con C++ de Visual Studio 2010 se pudieran leer y en conjunto con OpenGL realizar la animación gráfica de los movimientos del robot.

Elementos del esquema multilenguaje para este proyecto:

### **3.2 Lenguaje C++.**

El lenguaje C tiene una gran cantidad de ventajas sobre otros lenguajes y constituyen precisamente la razón fundamental de que después de casi dos



décadas de uso siga siendo uno de los lenguajes más populares, utilizados en empresas, organizaciones y fábricas de software de todo el mundo y por ello es que se seleccionó para este proyecto. Es un lenguaje de propósito general y posee las siguientes características principales:

- La ventaja más destacable es la transportabilidad o portabilidad, es decir, la posibilidad de utilizarlo tanto en macroordenadores como en mini y microordenadores.
- Es un lenguaje muy flexible
- Muy apropiado para controlar rutinas hechas en ensamblador
- Permite general programas de fácil modificación
- Muy veloz y potente, lo que permite un software efectivo
- Posibilita una programación estructurada o modular
- Produce programas de código compacto y eficiente
- Características de Alto nivel que conserva características de bajo nivel, por lo que se puede clasificar como lenguaje de bajo-medio nivel.
- Es un lenguaje compilado
- Se caracteriza por su velocidad de ejecución.
- Debido a que existen muchos programas escritos en el lenguaje C, se han creado numerosas bibliotecas C para programadores profesionales que soportan gran variedad de aplicaciones.
- Programación estructurada.
- Expresiones Simples.
- Variedad de operadores matemáticos, lógicos y comparativos.
- Variedad de tipos de datos.
- Codificación en alto y bajo nivel simultáneamente.
- No está orientado a un área en específico.
- Facilidad de aprendizaje.
- Los programas nuevos pueden ser desarrollados en menos tiempo porque se puede rehusar el código



- Crear y usar nuevos tipos de datos es más fácil que en otros lenguajes como
- El manejo de memoria en C++ es más fácil y transparente
- Los programas tendrán menos errores porque C++ usa una sintaxis y chequeo de tipos más estricto
- Orientación a objetos
- Sobrecarga de operadores y funciones
- Rapidez
- Genera programas compactos
- Argumentos de funciones por default
- Lenguaje muy eficiente puesto que es posible utilizar sus características de bajo nivel para realizar implementaciones óptimas.
- A pesar de su bajo nivel es el lenguaje más portado en existencia, habiendo compiladores para casi todos los sistemas conocidos.
- Este lenguaje de programación tiene la facilidad de utilizar las interfaces de aplicación gráfica (API), a través de las librerías gráficas abiertas (OpenGL)[8].

### **3.3 Matlab 2011.**

Matlab es un lenguaje de programación matemático, utilizado para implementar operaciones matemáticas computacionalmente complejas, de una manera más rápida que lenguajes tradicionales como C++, Basic y Fortran[8]. MATLAB es el nombre abreviado de "MATriz LABoratory". Es un programa para realizar cálculos numéricos con vectores y matrices, y por tanto se puede trabajar también con números escalares (tanto reales como complejos), con cadenas de caracteres y con otras estructuras de información más complejas. Matlab es un lenguaje de alto rendimiento para cálculos técnicos, es al mismo tiempo un entorno y un lenguaje de programación. Uno de sus puntos fuertes es que permite construir nuestras propias herramientas reutilizables.

Principales características:





- Podemos crear fácilmente nuestras propias funciones y programas especiales (conocidos como M-archivos) en código Matlab, los podemos agrupar en Toolbox (también llamadas librerías): colección especializada de M-archivos para trabajar en clases particulares de problemas.
- Matlab, a parte del cálculo matricial y álgebra lineal, también puede manejar polinomios, funciones, ecuaciones diferenciales ordinarias, gráficos ...
- MATLAB (Matrix Laboratory) es un programa interactivo de uso general en Ciencias e Ingeniería
- La interacción se realiza mediante instrucciones (comandos), y también mediante funciones y programas (scripts). Los objetos básicos con los cuales opera MATLAB son matrices. La especificación y asignación de cada variable la realiza MATLAB en forma dinámica y eficiente, por lo que no son necesarias las declaraciones de variables de algunos lenguajes de computación convencionales.
- Cálculo numérico rápido y con alta precisión
- Manejo simbólico
- Graficación y visualización avanzada
- Programación mediante un lenguaje de alto nivel
- Programación estructurada y orientada a objetos
- Soporte básico para diseño de interfaz gráfica
- Extensa biblioteca de funciones
- Aplicaciones especializadas para algunas ramas de ciencias e ingeniería
- Interacción con otros entornos
- No hay que preocuparse por las estructuras de datos, en particular, los arreglos no tienen que ser declarados antes de usarse.
- Los archivos de matlab son completamente portátiles en diferentes plataformas (Linux, Apple, Windows).

### 3.4 OpenGL.

OpenGL es utilizado para generar gráficos en la computadora en dos dimensiones (2D) y en tres dimensiones (3D). Este tipo de librerías son utilizadas por medio del



lenguaje C/C++, y las escenas tridimensionales pueden ser dibujadas desde primitivas simples. OpenGL es ampliamente utilizado en aplicaciones de software CAD, realidad virtual, visualización científica, visualización de datos, simuladores de vuelo, juegos de video, entre otras. OpenGL es un estándar sobre gráficos por computadora de los más conocidos del mundo. En 1982 nació en la Universidad de Standford el concepto de "graphics machine" y este fue utilizado por Silicon Graphics Corporation en su propia estación Silicon IRIS para crear un renderizador. Así nació la librería IRIS GL. A raíz de esto, allá por 1992 muchas empresas del hardware y software se pusieron de acuerdo para desarrollar conjuntamente una librería gráfica libre: OpenGL. Entre estas empresas destacaban Silicon Graphics Inc., Microsoft, IBM Corporation, Sun Microsystems, Digital Equipment Corporation (DEC), Hewlett-Packard Corporation, Intel e Intergraph Corporation. Así nació OpenGL (Open Graphics Library). La característica de ser "Abierta" significa que un programa escrito para una plataforma puede ser fácilmente convertible a prácticamente cualquier tipo de plataforma, obteniendo prácticamente los mismo resultados :). Esta era la principal novedad, ya que liberaba a los programadores de escribir programas para un hardware concreto: Si el hardware (gráfico) soporta una determinada función la ejecuta, y si no lo soporta pues le pasa el control al Micro y que le ejecute el.

Desde el punto de vista del programador OpenGL es una API para interactuar con dispositivos gráficos y aceleradoras 3D. Contiene cerca de 150 comandos que nos ayudan a definir objetos, aplicar transformaciones a esos objetos, cambiar sus propiedades (color, textura, luz...), posición de la cámara... entre otros. También hay que tener claro que OpenGL es una librería gráfica, no posee funciones para el control de Audio, Red o Control de Entrada.

### **3.5 Metodología multilenguaje propuesta.**

El uso de técnicas multilenguaje puede ser aplicado al diseño o análisis de sistemas mecatrónicos por medio de una simulación gráfica con animación. En este trabajo, los lenguajes de programación C++ con OpenGL, y Matlab se combinan para desarrollar la tarea mencionada. [8]

En la figura 19 se muestra la metodología propuesta en [8]. Esta metodología será adaptada para obtener como producto un sistema de software que permita la simulación gráfica del comportamiento de un manipulador flexible con actuador neumático con tres grados de libertad.

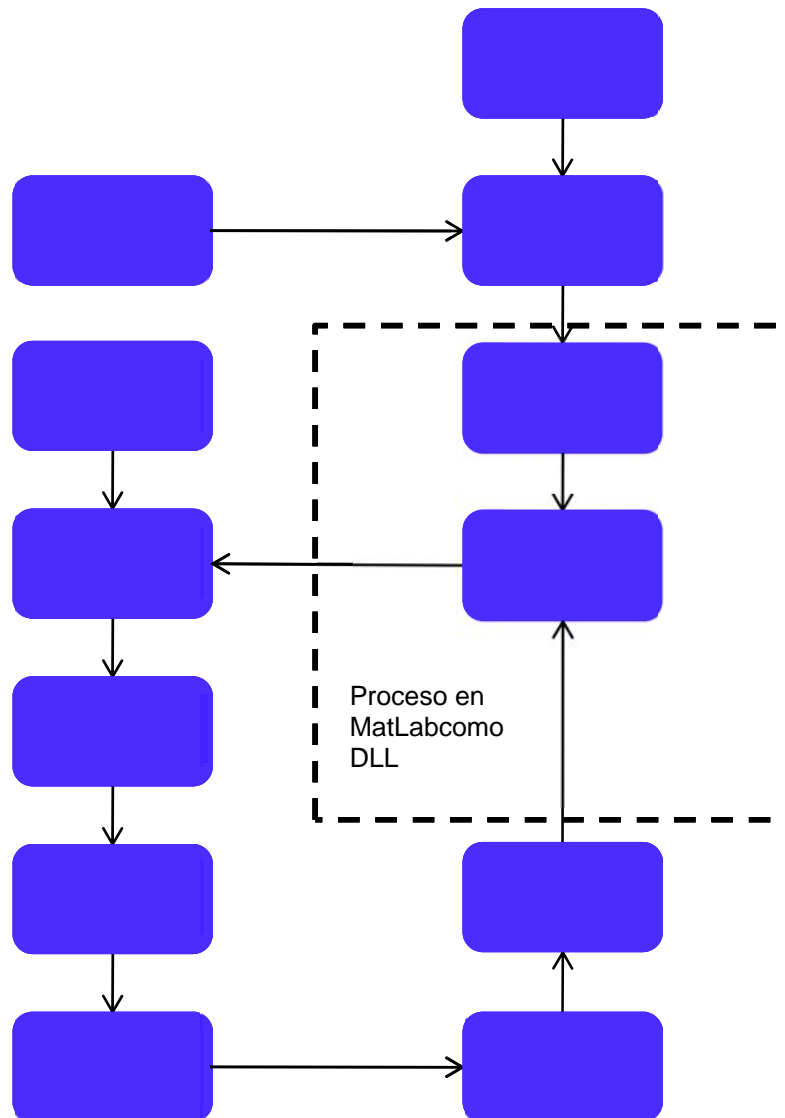


Figura 19. Metodología propuesta para la simulación gráfica de un proceso mecatrónico.

Primero, se requiere haber obtenido previamente los parámetros de los modelos cinemáticos y dinámicos. Una vez que se cuenta con estos valores, se requiere



realizar una simplificación del modelo para tener un modelo computacionalmente más sencillo de calcular. Segundo, el usuario establece el valor de referencia en la interfaz del lenguaje C, junto con los valores de los parámetros. Los valores de los parámetros son utilizados como entrada del modelo dinámico y cinemático. La interfaz del lenguaje C envía los valores a Matlab, donde se evalúa la ley de control. La salida de la ley de control es usada como entrada al proceso de simulación gráfica a través de la interfaz en lenguaje C, donde OpenGL toma dichos valores para generar la animación gráfica del sistema. Tercero, los resultados son analizados por el usuario, y en caso de que el comportamiento no corresponda al deseado, se realizan ajustes a los parámetros de control y se repite la evaluación de la ley de control. De esta manera, hasta que los resultados correspondan al comportamiento deseado [8].

## **CAPÍTULO 4**

### **RESULTADOS, CONCLUSIONES Y TRABAJOS A FUTURO**

## 4. Resultados, conclusiones y trabajos a futuro.

### 4.1 Resultados.

Se obtuvo como resultado un sistema de software que reproduce los movimientos del robot manipulador con tres grados de libertad, en la figura 20 se muestra el esquema del robot, con cada uno de sus componentes que fueron diseñados en un sistema CAD y posteriormente convertidos a un formato legible para C++ y OpenGL.

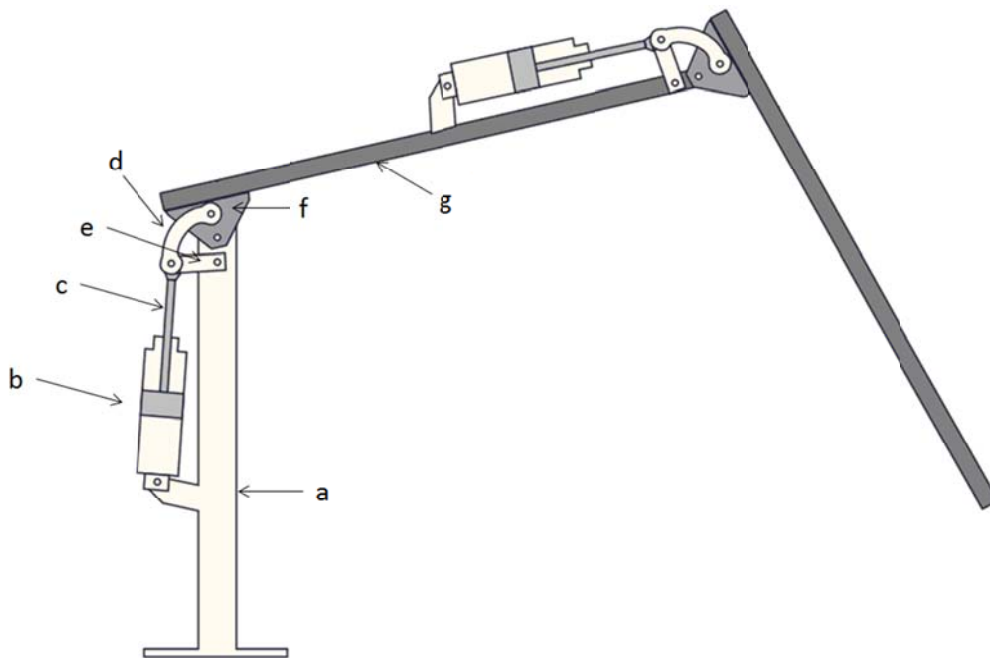


Figura 20. Figura 20. Objetos formados por primitivas OpenGL para el manipulador flexible. (a) Base del manipulador. (b) cilindro neumático. (c) barra (d) mecanismo impulsor. (e) Barra de manivela. (f) base del brazo. (g) Brazo.

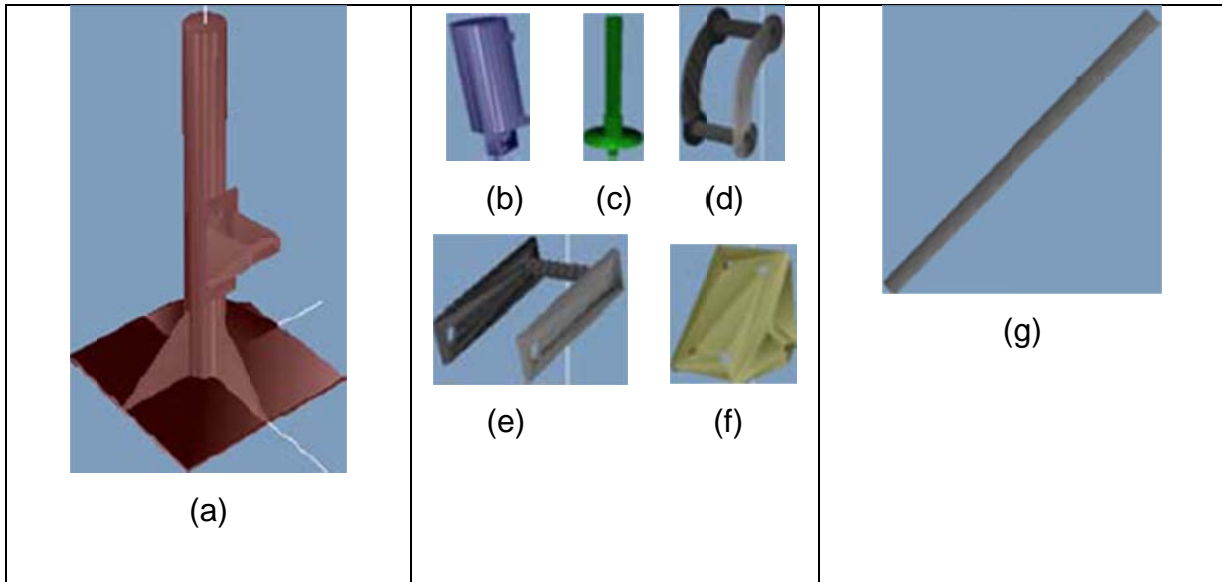


Figura 20 (Continuación). Objetos formados por primitivas OpenGL para el manipulador flexible. (a) Base del manipulador. (b) cilindro neumático. (c) barra (d) mecanismo impulsor. (e) Barra de manivela. (f) base del brazo. (g) Brazo.

En la figura 21 se muestra ya completamente ensamblado el robot manipulador flexible de tres grados de libertad, obtenido en OpenGL.

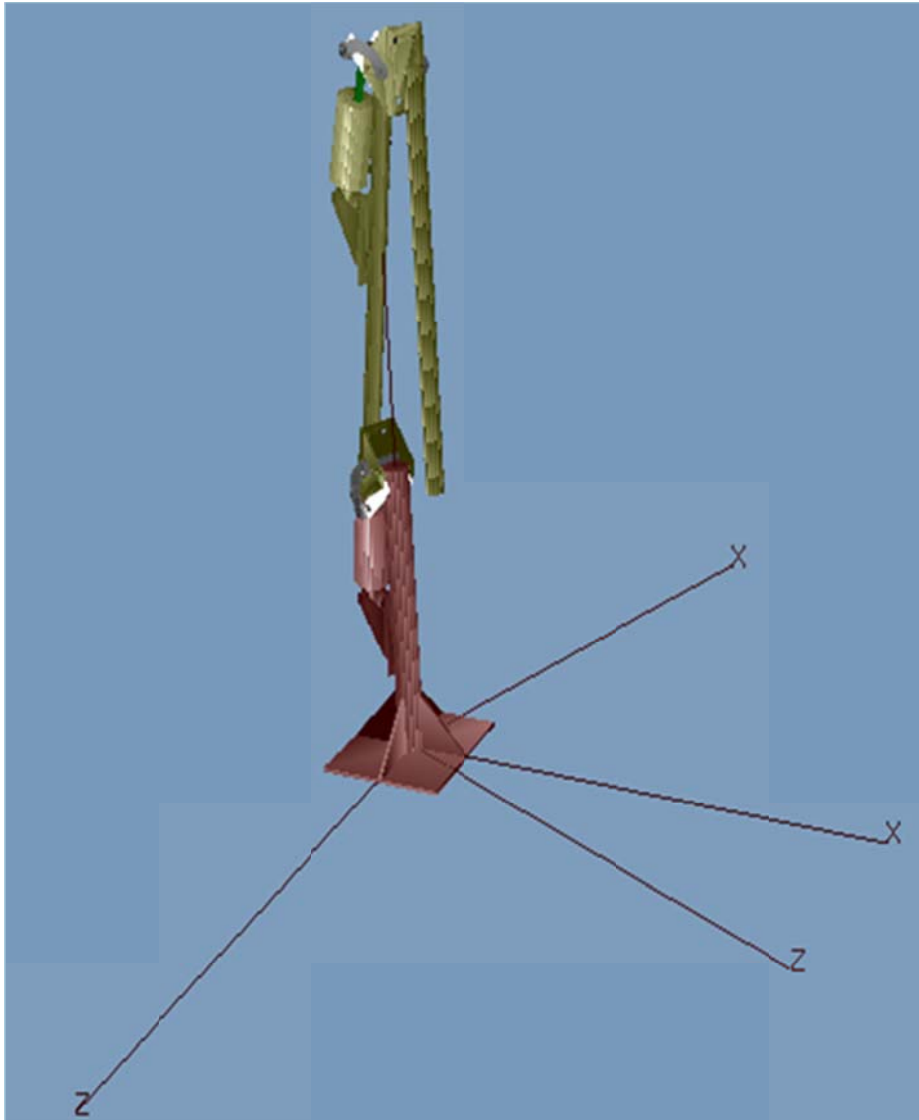


Figura 21. Robot manipulador flexible de tres grados de libertad.

En la siguiente figura 22 se observa la posición del brazo así como los ángulos y la posición del efector final (x, y, z) de cada uno de los eslabones y el valor de los ángulos así como la rotación.





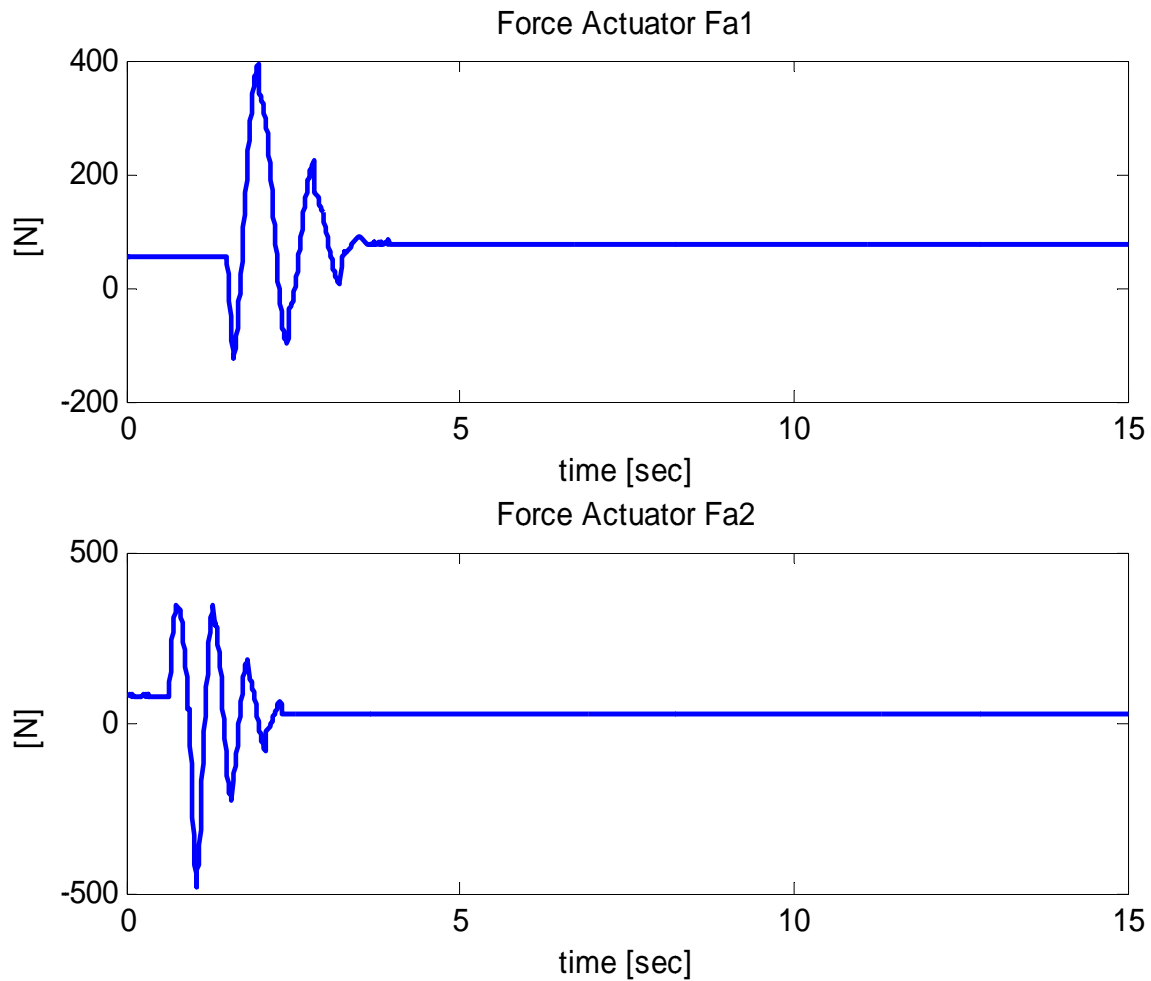


Figura No. 23. Gráficas de la dinámica de los actuadores del robot neumático.

En la gráfica 24, se muestra el diagrama de flujo que ilustra de manera general el funcionamiento del programa principal de la simulación del robot manipulador flexible de tres grados de libertad.

Posteriormente, en la gráfica 25, se muestra el diagrama de flujo general para dibujar completamente el robot manipulador, y en la gráfica 26 se muestra el diagrama para dibujar el movimiento del robot manipulador en tres dimensiones, utilizando un



archivo texto que proviene de Matlab 2010, con los valores que definen el movimiento de los componentes del robot.

Programa principal del robot manipulador

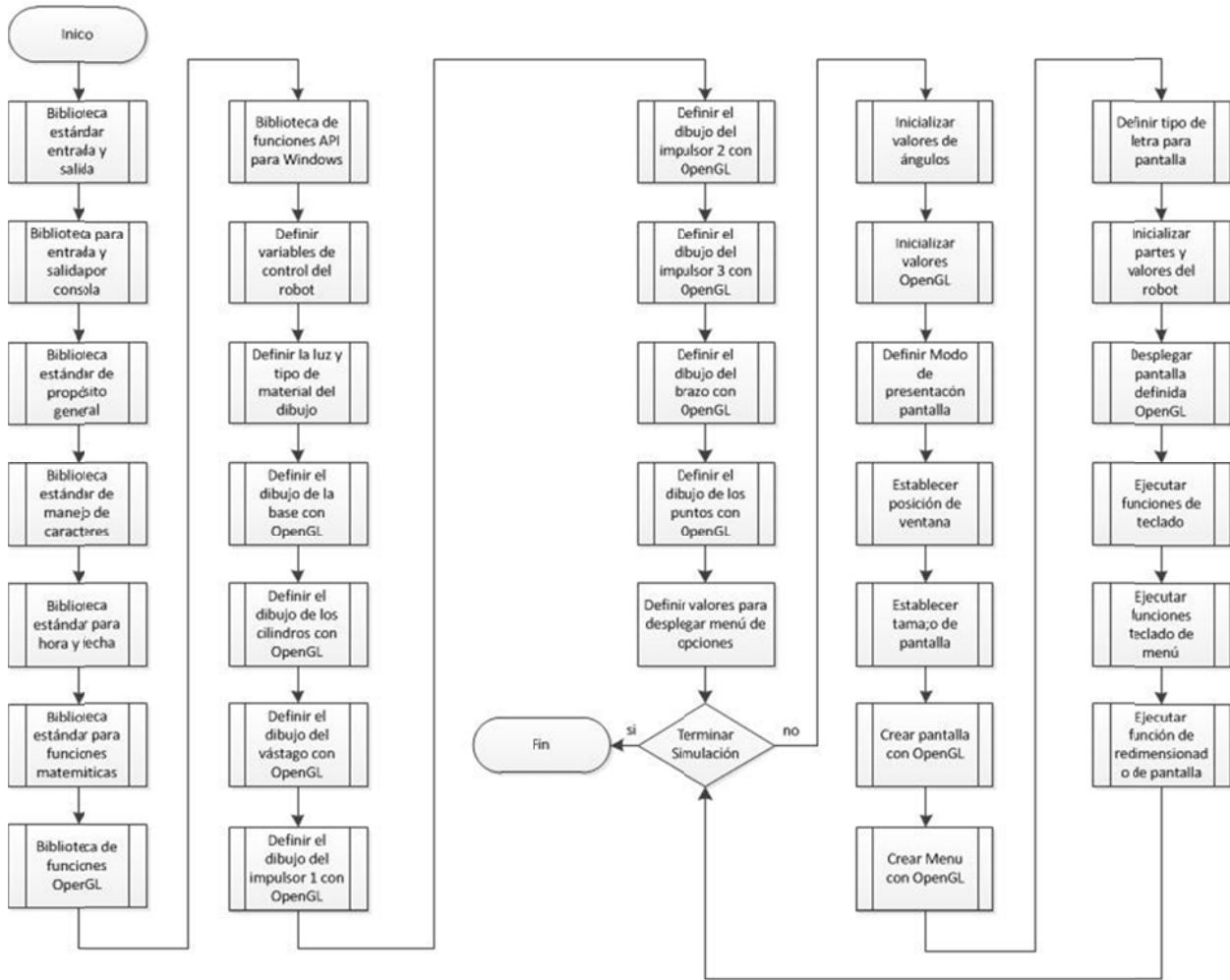


Figura No. 24. Diagrama de flujo del programa principal del robot neumático.



### Rutina general para dibujar el robot manipulador

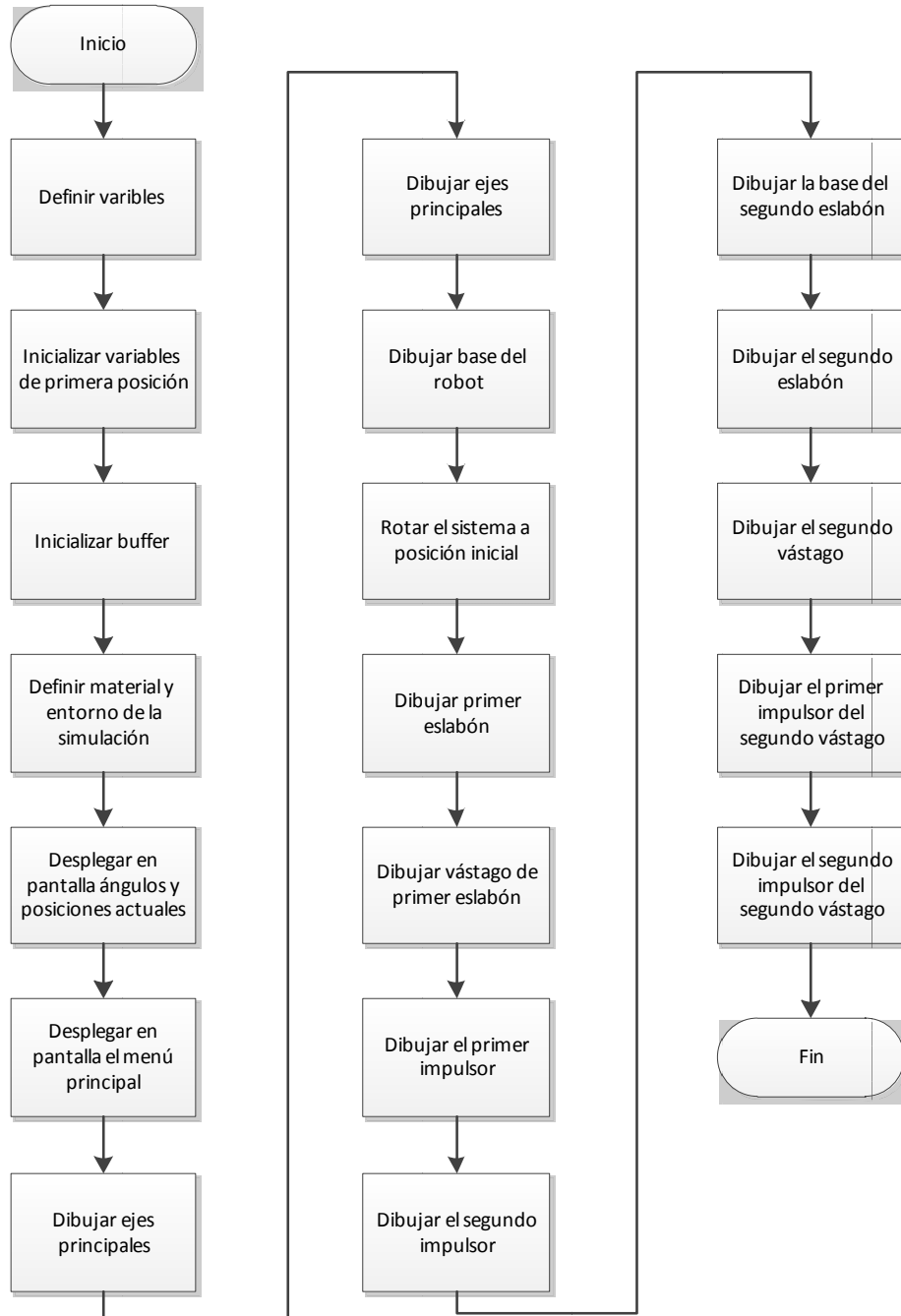


Figura No. 25. Diagrama de flujo de la rutina para dibujar el robot manipulador neumático.



Rutina para dibujar el movimiento del robot manipulador desde un archivo de Matlab

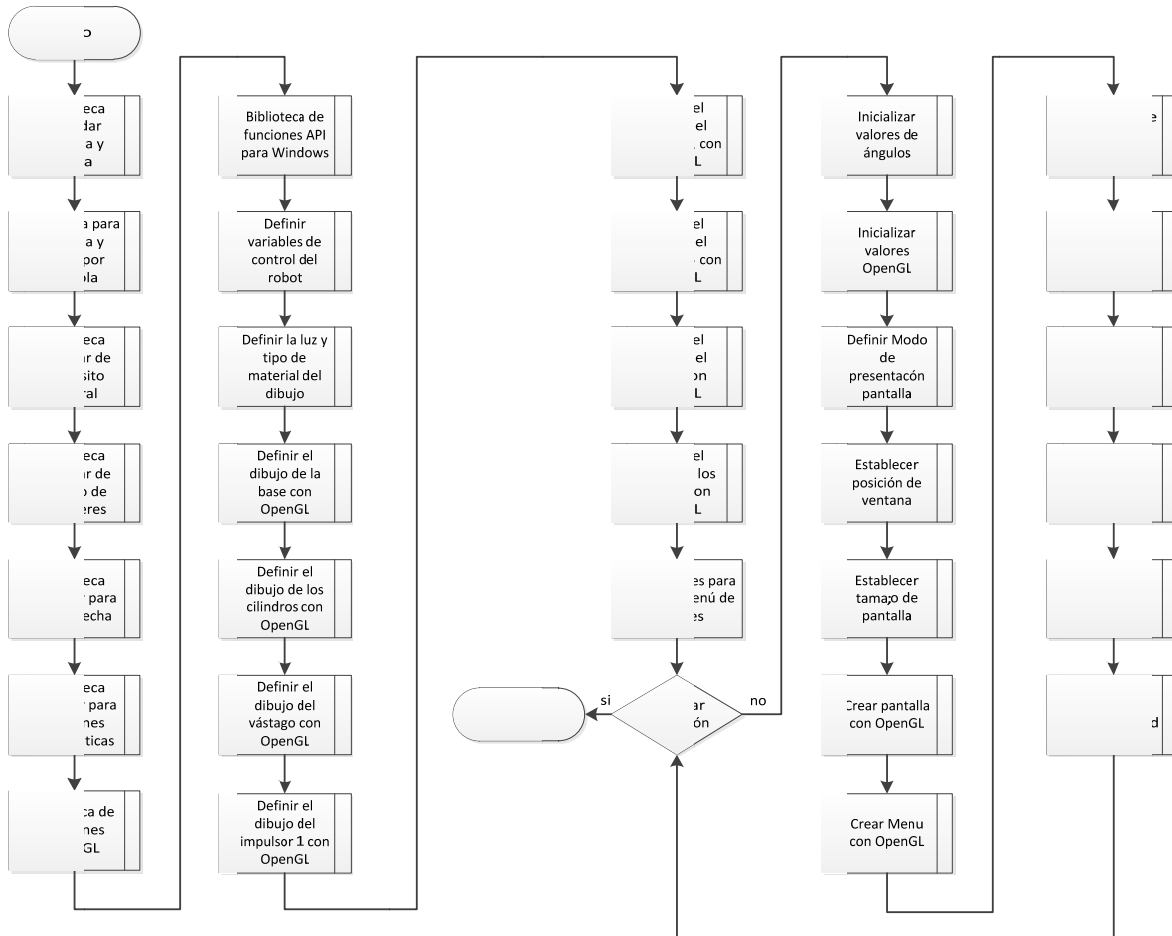


Figura No. 26. Diagrama de flujo de la rutina para dibujar el movimiento del robot manipulador neumático.

#### 4.2. Programas y Librerías del software.

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



Al instalar el software de Microsoft Visual Studio 2010 se deben de instalar las librerías de OpenGL de la siguiente manera:

d. En C:/Archivos de programa/Microsoft Visual Studio 10.0/VC/lib



..\lib\opengl32.lib

..\lib\glu32.lib

..\lib\glaux.lib

e. En C:/Archivos de programa/Microsoft Visual Studio 10.0/VC/include



O bien de la siguiente manera

f. En C:/Archivos de programa/Microsoft Visual Studio 10.0/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

g. 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

### 4.3 Conclusiones.

- Se utilizaron como base trabajos importantes previos realizados con uno y dos grados de libertad del robot neumático con un solo eslabón y en este trabajo se contemplan 3 grados, el segundo eslabón así como sus respectivos movimientos.
- Se hizo una adaptación de la modelación de la cinemática directa e inversa de la arquitectura del robot, basados en trabajos anteriores.
- Se ha estudiado, comprendido y adaptado para tres grados de libertad un modelo termo mecánico del comportamiento de los elementos neumáticos del manipulador.
- Se diseñó y desarrolló un sistema computacional que permite la simulación cinemática y dinámica del manipulador utilizando C++ de Microsoft Visual Studio 2010, Open GL y Matlab 2011.
- En este trabajo se presenta el desarrollo de una metodología de programación multilenguaje, la cual fue adaptada también de una metodología diseñada anteriormente para dos grados de libertad.

### 4.4 Trabajos a futuro.

- Elaborar artículos relacionados con esta investigación para ser propuestos para su publicación en revistas especializadas.
- Presentar el trabajo realizado en congresos.



- Servirá como base para realizar otros trabajos similares, así como para en un futuro realizar la implementación física del robot manipulador neumático con tres grados de libertad.
- Interesar a programadores e investigadores relacionados con el área de conocimiento, para que sigan desarrollando la metodología de programación multilenguaje.



# **BIBLIOGRAFÍA**



## Bibliografía.

- [1]. Spong, Mark W., et. al. Robot Modeling and Control. John Wiley and Sons Inc., 2008. USA.
- [2]. [http://www.articulosinformativos.com.mx/Auge\\_de\\_la\\_Robotica\\_Mexico-r1073093-Mexico.html](http://www.articulosinformativos.com.mx/Auge_de_la_Robotica_Mexico-r1073093-Mexico.html). Auge de la robótica en México. Consultada el 10-Oct-2009.
- [3]. Kelly, R., et. al. Control of Robot Manipulators in Joint Space. Springer, 2005. USA.
- [4]. 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.
- [5]. Bajd, T., et. al. Robotics. Springer, 2010. USA.
- [6]. 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.
- [7]. 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.
- [8]. 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 multilinguaje. VII Congreso Internacional en Innovación y Desarrollo Tecnológico, 2009. Autorizado y pendiente de imprimir.
- [9]. Sciavicco, L., et. al. Modelling and Control of Robot Manipulators. Springer, 2005. USA.
- [10]. Ramos, J.M. Algoritmos de Control para un Actuador Neumático de un Manipulador Neumático. Tesis de Doctorado. Centro de Ingeniería y Desarrollo Industrial, Febrero 2007.



- [11]. <http://www.cnnexpansion.com/tecnologia/2009/08/18/la-robotica-podria-hacer-crecer-a-mexico>. La robótica podría hacer crecer a México. Consultada el 10-Oct-2009.
- [12]. Blilie, Charles. The Promise and Limits of Computer Modeling. World Scientific, 2007. USA.
- [13]. IEEE Standard for Software Project Management Plans IEEE Std 1058-1998.
- [14]. ISO/IEC 9126-1:2001, Software engineering — Product quality — Part 1: Quality model.
- [15]. ISO/IEC TR 9126-3), Software engineering — Product quality — Part 2: External metrics.
- [16]. ISO/IEC 9126-4), Software engineering — Product quality — Part 4: Quality in use metrics.
- [17]. Lyshevski, Sergey E. Engineering and Scientific Computations Using MATLAB. Wiley, 2003. USA.
- [18]. Lever, Nick. Real-time 3D Character Animation with Visual C++. Wiley Interscience, 2005. USA.
- [19]. Holzbecher, Ekkehard. Environmental Modeling with MATLAB. Springer, 2007. USA.
- [20]. Munshi, Aaftab, et. al. OpenGL Programming Guide. Addison Wesley, 2009. USA.
- [21]. Barrientos, Antonio, et. al. Fundamentos de Robótica. McGraw Hill, 2007. España.
- [22]. Barrientos, Antonio, et. al. Fundamentos de Robótica. McGraw Hill, 2007. España.
- [23]. <http://proton.ucting.udg.mx/robotica>. Programa interactivo tutorial sobre robótica de la Universidad de Guadalajara. Consultada el 25-Febrero-2011.
- [24]. [http://platea.pntic.mec.es/vgonzale/cyr\\_0708/index.htm](http://platea.pntic.mec.es/vgonzale/cyr_0708/index.htm). Curso en línea de Control y Robótica del Centro de Formación del Profesorado e Innovación Educativa. Consultada el 28-Febrero-2011.



- [25]. Hernández, J.M. Modelación de un robot neumático de dos grados de libertad. Tesis de maestría. Instituto Tecnológico de Morelia., Febrero 2007.
  
- [26]. <http://www26.brinkster.com/lwelicki/articles/PatronesMultilenguaje.pdf>
  
- [27]. Fernando F. Kiyama and Dr. Emilio Vargas Soto Modelling of a single degree of freedom flexible arm pneumatically operated. Fernando F. Kiyama and Dr. Emilio Vargas Soto.

# **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 el estudio riguroso del espacio y de las formas.

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

**Manipulador robótico.** Mecanismo reprogramable.

**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.** Línea descrita por un punto material en movimiento.

## **ANEXO A.**

# **CÓDIGO FUENTE DEL PROGRAMA PRINCIPAL DE LA ANIMACIÓN DEL ROBOT**



## Anexo A. Código fuente del programa principal de la animación.

A continuación se muestra el código fuente realizado en lenguaje C++ de Visual Studio 2010, del programa principal de la animación gráfica del robot manipulador de tres grados de libertad.

```

//*****
//*
//*          UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
//*          FACULTAD DE INFORMÁTICA
//*          MAESTRÍA EN INGENIERÍA DE SOFTWARE DISTRIBUIDO
//*
//*
//* Proyecto de tesis: Modelación y simulación de un robot neumático
//* de tres grados de libertad.
//*
//* Nombre : RobotManipulador3GL.cpp
//* Fecha  : 7 de febrero de 2012.
//* Función: programa principal de la animación gráfica del robot
//*
//* Director de tesis: Dr. Juan Manuel Ramos Arreguín
//*
//*
//*****

#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(        // Select The Current DC
                    hDC, // Starting Character
                    0, // 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  
    {  
        Rotacion=dato[0];  
        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("desplazamiento.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
    glBeginList(startList+7, GL_COMPILE);
        base0();
        base5();
        base6();
        base8();
    glEndList();

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

    //////////// 10. SE DIBUJA EL DESPLAZAMIENTO DESEADO
    glBeginList(startList+9, GL_COMPILE);
        puntos_tray();
    glEndList();

    glBeginList(startList+10, 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("Robot Manipulador Flexible con tres grados de libertad",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=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, "z3=%.4f", z3);renderBitmapCharacher(169,160,0, (void *)font[2],s);
    sprintf(s, "y2=%.4f", y3);renderBitmapCharacher(168,170,0, (void *)font[2],s);
    sprintf(s, "x3=%.4f", x3);renderBitmapCharacher(167,180,0, (void *)font[2],s);
    sprintf(s, "Rotacion=%.1f", Rotacion);renderBitmapCharacher(166,190,0, (void
*)font[2],s);
    sprintf(s, "theta3=%.1f", theta[7]);renderBitmapCharacher(165,200,0, (void
*)font[2],s);
    sprintf(s, "theta2=%.1f", theta[3]);renderBitmapCharacher(164,210,0, (void
*)font[2],s);
    sprintf(s, "theta1=%.1f", Rotacion);renderBitmapCharacher(163,220,0, (void
*)font[2],s);
    sprintf(s, "z2=%.4f", z2);renderBitmapCharacher(162,230,0, (void *)font[2],s);
    sprintf(s, "y2=%.4f", y2);renderBitmapCharacher(161,240,0, (void *)font[2],s);
    sprintf(s, "x2=%.4f", x2);renderBitmapCharacher(160,250,0, (void *)font[2],s);

    glPushMatrix();
        glCallList(startList+10);
    glPopMatrix();

    glPushMatrix();
        glTranslatef(POSX, POSY, POSZ);
        glScalef(Escala, Escala, Escala);
        glRotatef(anglex, 1.0, 0.0, 0.0);
        glRotatef(angley, 0.0, 1.0, 0.0);
        glRotatef(anglez, 0.0, 0.0, 1.0);

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



```

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

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

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

    // Se dibujan los ejes principales
    glBegin(GL_LINES);
        glVertex3f(0,0,0);
        glVertex3f(1,0,0);
        glVertex3f(0,0,0);
        glVertex3f(0,1,0);
        glVertex3f(0,0,0);
        glVertex3f(0,0,0);
        glVertex3f(0,0,1);
    glEnd();
    sprintf(s,"X");renderBitmapCharacher(1,0,0,(void *)font[2],s);
    sprintf(s,"Y");renderBitmapCharacher(0,1,0,(void *)font[2],s);
    sprintf(s,"Z");renderBitmapCharacher(0,0,1,(void *)font[2],s);

    // Se dibuja el primer eslabon
    glPushMatrix();
        SetMaterial(mat_specularRED, mat_ambientRED, mat_diffuseRED,
mat_shinnessRED);

        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_shinnessGREEN);

            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_shinnessWHITE);

        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_shinnessGRAY);

        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();
    glPopMatrix();
    glutSwapBuffers();
}

void keyboard(unsigned char key, int x, int y)
{
    switch(key)
    {
        case '1':
            animation=fopen("desplazamiento.txt","r");
            glutIdleFunc(spinDisplay);
            break;
        case 'A':
            Rotacion-=5;
            glutPostRedisplay();
            break;
        case 'a':
            Rotacion+=5;
            glutPostRedisplay();
            break;
        case 'B':
            POSX-=10;
            glutPostRedisplay();
            break;
        case 'b':
            POSX+=10;
            glutPostRedisplay();
            break;
        case 'C':
            POSY+=10;
            glutPostRedisplay();
            break;
        case 'c':
            POSY-=10;
            glutPostRedisplay();
            break;
        case 'D':
            POSZ-=10;
            glutPostRedisplay();
            break;
        case 'd':
            POSZ+=10;
            glutPostRedisplay();
            break;
        case 'E':
            Escala+=20;
            glutPostRedisplay();
            break;
        case 'e':
            Escala-=20;
            glutPostRedisplay();
            break;
        case 'X':
            anglex-=2;
        case 'x':
            anglex++;
            glutPostRedisplay();
            break;
        case 'Y':
            angley-=2;
        case 'y':
            angley++;
            glutPostRedisplay();
            break;
    }
}
```





```
case 'Z':
    anglez-=2;
case 'z':
    anglez++;
    glutPostRedisplay();
    break;
case 'L':
case 'l':
    Tipo=GL_LINE_LOOP;
    init();
    glutPostRedisplay();
    break;
case 'S':
case 's':
    Tipo=GL_POLYGON;
    init();
    glutPostRedisplay();
    break;
case 'P':
case 'p':
    Tipo=GL_POINTS;
    init();
    glutPostRedisplay();
    break;
case 'Q':
case 'q':
    glutIdleFunc(spinDisplay);
    break;
case 27:
    exit(0);
    break;
default:
    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("Robot Manipulador Flexible");
    hDC = wglGetCurrentDC();
    BuildFont();
    init();
    glutDisplayFunc(renderScene);
    glutKeyboardFunc(keyboard);
    glutReshapeFunc(changeSize);
    glutMainLoop();
}
```