



PONTIFICIA **UNIVERSIDAD CATÓLICA** DEL PERÚ

Esta obra ha sido publicada bajo la licencia Creative Commons
Reconocimiento-No comercial-Compartir bajo la misma licencia 2.5 Perú.

Para ver una copia de dicha licencia, visite
<http://creativecommons.org/licenses/by-nc-sa/2.5/pe/>



PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ

MODELACIÓN Y SIMULACIÓN DINÁMICA DE UN BRAZO ROBÓTICO DE 4 GRADOS DE LIBERTAD PARA TAREAS SOBRE UN PLANO HORIZONTAL

Tesis para optar el Título de Ingeniero Mecánico, que presenta el bachiller:

Luis Felipe López Apostolovich

ASESOR: Ing. Dante Ángel Elías Giordano

Lima, agosto del 2009

RESUMEN

En el presente trabajo se realizó la modelación y simulación de la dinámica inversa de un robot articular de cuatro grados de libertad que usa láser para realizar corte de precisión de madera y tiene definida su superficie de trabajo en un plano horizontal. Este trabajo es una parte de un proyecto multidisciplinario desarrollado por tesis de Ingeniería Mecánica e Ingeniería Electrónica que busca desarrollar máquinas automáticas industriales para uso en nuestro país, y obtener las herramientas y el conocimiento para poder mejorar nuestros propios procesos.

En el proyecto se realizó el diseño preliminar del mecanismo base correspondiente al brazo robótico, lo que incluyó el dimensionamiento previo de los eslabones y la definición de los pares cinemáticos. Estas características fueron determinadas de manera que el robot sea capaz de ubicarse sin problemas en toda el área de trabajo.

Usando la información del diseño preliminar se estudiaron la Cinemática y la Dinámica del mecanismo. El desarrollo de la Cinemática se enfocó principalmente en el estudio geométrico del movimiento del brazo con respecto a un sistema de referencia fijo y tomó como fundamento teórico las matrices de transformación del algoritmo de Denavit Hartenberg. Este estudio se complementó con el análisis de la Cinética del manipulador aplicando el método de Uicker, basado en la mecánica lagrangiana, para estudiar los efectos que las fuerzas y momentos tienen sobre el movimiento del brazo y obtener la ecuación de movimiento.

Finalmente con el modelo obtenido se procedió con la simulación numérica de la dinámica del brazo robot en MATLAB en donde se calcularon los valores que deberían tener los torques a producirse en los actuadores eléctricos a fin de obtener el movimiento deseado. Estos resultados permiten no sólo seleccionar actuadores adecuados para el robot sino también generar señales de referencia para un sistema de control que haga que el robot se desplace según un comportamiento deseado.



A toda mi familia, por su amor y apoyo brindado siempre.

A mis amigos, por su apoyo e incondicional amistad.

A mis profesores, por todos sus consejos y enseñanzas.

A todos ... Gracias!

INDICE DE CONTENIDO

	Pág.
RESUMEN.....	i
APROBACIÓN DE TEMARIO DE TESIS	ii
DEDICATORIA	iv
INDICE DE CUADROS	vii
INDICE DE FIGURAS.....	viii
INTRODUCCIÓN	1
I. ANTECEDENTES.....	4
1.1 Reseña histórica	4
1.2 Definición de un Robot Industrial	7
1.3 Clasificación de los Robots Industriales	8
1.4 Robots para corte con láser	13
II. DISEÑO PRELIMINAR DEL MECANISMO	15
2.1 Consideraciones o requerimientos	15
2.2 Mecanismo propuesto	16
2.3 Cargas en el mecanismo	19
2.3.1 Eslabones	19
2.3.2 Motores	20
2.3.3 Implementación de contrapesos	21
III. MODELACIÓN DEL MECANISMO	22
3.1 Análisis de la cinemática	22

3.1.1 Cinemática directa	24
3.1.2 Cinemática inversa	31
3.1.3 Modelo diferencial	37
3.2 Análisis de la dinámica	40
3.2.1 Obtención del modelo dinámico mediante la formulación de Lagrange	41
IV. SIMULACIÓN Y RESULTADOS	47
4.1 Simulación cinemática	47
4.2 Simulación dinámica	52
4.2.1 Simulación de movimiento paralelo al lado mayor	53
4.2.2 Simulación de movimiento diagonal con el peso a favor del movimiento	56
4.2.3 Simulación de movimiento diagonal con el peso en contra del movimiento	61
4.2.4 Simulación de movimiento más exigido para selección de actuadores	66
4.2 Rutinas para simulación	69
OBSERVACIONES Y RECOMENDACIONES	76
CONCLUSIONES	78
BIBLIOGRAFIA	79

INDICE DE CUADROS

	Pág.
Cuadro 3.1 – Parámetros de Denavit – Hartenberg	28
Cuadro 3.2 – Rutina del método de Newton-Raphson	33
Cuadro 3.3 – Resultados de correr la rutina de Newton-Raphson	34
Cuadro 4.1 – Código para simulación de ubicación	48
Cuadro 4.2 – Código para cinemática directa	50
Cuadro 4.3 – Código para cinemática inversa	51
Cuadro 4.4 – Rutina para evaluar coordenadas generalizadas	52



INDICE DE FIGURAS

	Pág.
<u>Capítulo 1</u>	
Figura 1.1 – ASIMO	6
Figura 1.2 – Robot para soldadura de arco	7
Figura 1.3 – Robot cartesiano de tres ejes	9
Figura 1.4 – Robot cilíndrico	9
Figura 1.5 – a) Robot esférico b) Robot articular	10
Figura 1.6 – Robot SCARA	11
Figura 1.7 – Robot paralelo para levantar objetos	12
Figura 1.8 – Robot para competencia Robo Soccer	13
Figura 1.9 – Robot de corte con láser	14
<u>Capítulo 2</u>	
Figura 2.1 – Modelo preliminar	17
Figura 2.2 – Eslabones del mecanismo	19
Figura 2.3 – Fuerzas y momentos producidos por los pesos	20
Figura 2.4 – Representación de los contrapesos para el efecto torsor	21
<u>Capítulo 3</u>	
Figura 3.1 – Diagrama de relación entre cinemática directa e inversa	23
Figura 3.2 – Esquema cinemático	24
Figura 3.3 – Cadena cinemática abierta en el plano	25
Figura 3.4 – Gráfica de tendencia	34
Figura 3.5 – Cadena cinemática con extremo perpendicular a la superficie	35

Figura 3.6 – Cadena cinemática equivalente de dos eslabones 35

Capítulo 4

Figura 4.1 – Simulación de ubicación variando las coordenadas articulares 49

Figura 4.2 – Relación entre las coordenadas para un caso dado 51

Figura 4.3 – Diagrama de flujo de la simulación dinámica 53

Primera simulación

Figura 4.4 – Trayectoria deseada paralela al lado mayor 54

Figura 4.5 – Primera coordenada generalizada (d) [mm] en función al tiempo [s] 55

Figura 4.6 – Fuerza generalizada (F) [kg×mm/s²] del primer efector en función al tiempo[s] ...55

Segunda simulación

Figura 4.7 – Trayectoria deseada en la simulación con el peso a favor del movimiento 56

Figura 4.8 – Primera coordenada generalizada (d) [mm] en función al tiempo [s] 57

Figura 4.9 – Segunda coordenada generalizada (θ_1) [rad] en función al tiempo [s] 57

Figura 4.10 – Tercera coordenada generalizada (θ_2) [rad] en función al tiempo [s] 58

Figura 4.11 – Cuarta coordenada generalizada (θ_3) [rad] en función al tiempo [s] 58

Figura 4.12 –Fuerza generalizada (F) [kg×mm/s²] del primer efector en función al tiempo [s] .59

Figura 4.13 –Torque generalizado(T_1)[kg×mm²/s²]del segundo efector en función al tiempo[s]59

Figura 4.14 –Torque generalizado(T_2)[kg×mm²/s²]del tercer efector en función al tiempo[s]...60

Figura 4.15 –Torque generalizado(T_3)[kg×mm²/s²]del cuarto efector en función al tiempo[s].. 60

Tercera simulación

Figura 4.16 – Trayectoria deseada en la simulación con el peso en contra del movimiento 61

Figura 4.17 – Primera coordenada generalizada (d) [mm] en función al tiempo [s] 62

Figura 4.18 – Segunda coordenada generalizada (θ_1) [rad] en función al tiempo [s] 62

Figura 4.19 – Tercera coordenada generalizada (θ_2) [rad] en función al tiempo [s] 63

Figura 4.20 – Cuarta coordenada generalizada (θ_3) [rad] en función al tiempo [s] 63

Figura 4.21 –Fuerza generalizada (F) [$\text{kg}\times\text{mm}^2/\text{s}^2$] del primer efector en función al tiempo [s] .64

Figura 4.22 –Torque generalizado(T_1)[$\text{kg}\times\text{mm}^2/\text{s}^2$]del segundo efector en función al tiempo[s]64

Figura 4.23 –Torque generalizado(T_2)[$\text{kg}\times\text{mm}^2/\text{s}^2$]del tercer efector en función al tiempo[s]... 65

Figura 4.24 –Torque generalizado(T_3)[$\text{kg}\times\text{mm}^2/\text{s}^2$]del cuarto efector en función al tiempo[s].. 65

Cuarta simulación

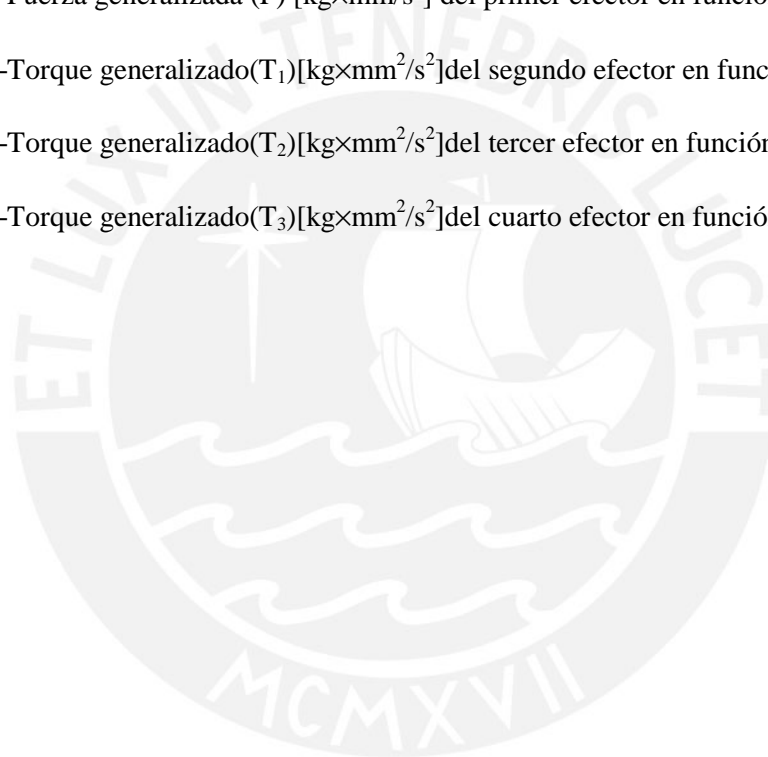
Figura 4.25 – Velocidad deseada en función del tiempo 66

Figura 4.26 –Fuerza generalizada (F) [$\text{kg}\times\text{mm}^2/\text{s}^2$] del primer efector en función al tiempo [s] .67

Figura 4.27 –Torque generalizado(T_1)[$\text{kg}\times\text{mm}^2/\text{s}^2$]del segundo efector en función al tiempo[s]67

Figura 4.28 –Torque generalizado(T_2)[$\text{kg}\times\text{mm}^2/\text{s}^2$]del tercer efector en función al tiempo[s]...68

Figura 4.29 –Torque generalizado(T_3)[$\text{kg}\times\text{mm}^2/\text{s}^2$]del cuarto efector en función al tiempo[s].. 68



INTRODUCCIÓN

La búsqueda de un incremento en la productividad, la obtención de una calidad uniforme en los productos finales así como la disminución de los costos de mano de obra han impulsado a la industria al uso de máquinas automáticas. Dependiendo de los lotes de producción muchas de las tareas dentro de un proceso de manufactura cualquiera pueden ser llevadas a cabo por máquinas diseñadas para un proceso específico, cuya inflexibilidad y alto costo se justifica sólo en el caso de lotes muy grandes, o por máquinas programables capaces de realizar labores múltiples en un ambiente de trabajo más flexible y a menores costos de producción.

Un ejemplo de esta última clasificación es el robot industrial tipo manipulador de uso general, controlado generalmente por computadora, que consiste en algunos elementos rígidos conectados en serie mediante articulaciones prismáticas o de revolución. El inicio de la cadena está fijo a una base soporte, mientras el extremo final está libre y equipado con una herramienta para manipular objetos o realizar otras tareas. El movimiento de las articulaciones resulta en, o produce, un movimiento relativo de los distintos elementos y por lo tanto en el desplazamiento de la herramienta.

En la actualidad se ve a la robótica como un campo de trabajo muy amplio desarrollando nuevas tecnologías en una serie de áreas interdisciplinarias como la cinemática, la dinámica, la planificación de sistemas, el control, sensores, lenguajes de programación e inteligencia artificial.

Una de las aplicaciones en las cuales se aplica la robótica para mejorar la productividad y la calidad del trabajo es en la carpintería y la industria de muebles, en

los cuales se usan tecnologías de corte por láser o por agua para mejorar el acabado en los trabajos realizados. Es pensando en estas aplicaciones que se desarrolla este trabajo, que trata sobre el análisis de la dinámica de un robot industrial capaz de realizar corte con láser.

El diseño de un robot empieza con la cinemática del mismo, la cual trata del estudio analítico del movimiento del brazo con respecto a un sistema de coordenadas de referencia fijo sin considerar las fuerzas o momentos que lo originan.

Hay dos problemas fundamentales en la cinemática del robot. El primer problema se suele conocer como el problema cinemático directo, mientras que el segundo es el problema cinemático inverso.

En la cinemática directa se tiene como datos de entrada los parámetros de los elementos y los desplazamientos en las articulaciones para obtener la posición y orientación del efector final. Mientras en la cinemática inversa los datos son los parámetros de los elementos, la posición y orientación del efector final con lo que se determinan los desplazamientos de las articulaciones.

La dinámica del robot, por otra parte, trata no sólo de la geometría del movimiento sino de las causas que lo originan: fuerzas y momentos. El modelo dinámico real de un brazo de robot se puede obtener de leyes físicas como las leyes de Newton y la mecánica lagrangiana. Esto conduce al desarrollo de las ecuaciones dinámicas de movimiento para las distintas articulaciones del manipulador en base de los parámetros geométricos e inerciales especificados para los distintos elementos.

Con el conocimiento de la cinemática y la dinámica de un manipulador se realiza la planificación de la trayectoria, es decir, decidir el movimiento antes de actuar. Se conoce como trayectoria a la curva espacial que el extremo del manipulador sigue desde una posición inicial hasta una posición final. Para definir correctamente una trayectoria ésta debe ser rápida y evitar todos los obstáculos que puedan haber en el camino.

Luego, el problema del control del robot consiste en utilizar los modelos dinámicos obtenidos del manipulador para determinadas leyes o estrategias de control para conseguir una respuesta y funcionamiento deseados. Se controla el movimiento final en el cual el efector interacciona dinámicamente con el objeto.

En los robots avanzados incluso se habla de inteligencia propia, es decir, que no necesitan una programación específica para cada acción o trayectoria sino que pueden decidir cuál es la mejor acción a tomar simplemente con datos del ambiente que los rodea.

Con el fin de continuar con la innovación tecnológica se desarrolla este tema de tesis, para así desarrollar el conocimiento necesario para el diseño y fabricación de un brazo robótico manipulador para corte por láser de maderas de aplicación industrial.

El objetivo general de este trabajo es la resolución de la dinámica inversa para un robot articular de cuatro grados de libertad para corte de madera en un plano horizontal. Los objetivos específicos a trabajar son el diseño preliminar del mecanismo del manipulador, la modelación del sistema, el cálculo de la cinemática directa e inversa del modelo preliminar definido, el análisis de la dinámica del modelo basándose en la mecánica de Lagrange y la simulación numérica del sistema usando MATLAB.

A través de este trabajo se espera contribuir al desarrollo e investigación de máquinas automáticas de uso industrial para uso en nuestro país a fin de acortar la brecha existente con la innovación tecnológica mundial así como poseer las herramientas y el conocimiento para poder mejorar nuestros propios procesos, además se espera proveer de información a personas interesadas en continuar trabajando esta rama de la automatización.

CAPÍTULO 1

ANTECEDENTES

1.1 Reseña histórica

La automatización industrial es el uso de elementos mecánicos, eléctricos o electrónicos para controlar procesos industriales substituyendo el trabajo del ser humano. Provee a los operadores humanos de mecanismos autónomos o semi autónomos para ayudarlos a extender sus capacidades físicas al realizar tareas conocidas por él de una manera más eficiente y segura reduciendo ampliamente la necesidad sensorial y mental del hombre.

Una de las ramas más visible de esta automatización es la robótica. Según El “*Robot Institute of America*” (RIA), un robot es “un manipulador reprogramable, multifuncional, diseñado para mover materiales, partes, herramientas y objetos especiales a través de movimientos programados variables para el cumplimiento de una variedad de tareas” [SPONG, 1993]

Según esta definición el robot debe tener la capacidad de ser reprogramado sin mayores modificaciones físicas. Esto le da un grado de flexibilidad del cual carecen las máquinas diseñadas para una tarea específica, ya que con un pequeño cambio en la

programación, y en la herramienta de trabajo, un mismo robot podría desarrollar labores tan diferentes como la soldadura, el grabado de placas o el corte de metales.

El inicio de la robótica actual puede fijarse en la industria textil del siglo XVIII, cuando Joseph Jacquard inventa en 1801 una máquina textil programable mediante tarjetas perforadas. La revolución industrial impulsó el desarrollo de estos agentes mecánicos, entre los cuales se destacaron el torno mecánico motorizado de Babbitt (1892) y el mecanismo programable para pintar con spray de Pollard y Roselund (1939). Además durante los siglos XVII y XVIII fueron construidos en Europa ingeniosos muñecos mecánicos que tenían algunas características de robots. Esencialmente se trataba de robots mecánicos diseñados para un propósito específico: la diversión. Estas creaciones mecánicas de forma humana deben considerarse como invenciones aisladas que reflejan el genio de hombres que se anticiparon a su época.

La palabra robot se empleó por primera vez en 1920 en una obra de teatro llamada "R.U.R." o "Los Robots Universales de Rossum" escrita por el dramaturgo checo Karel Capek. La trama era sencilla: el hombre fabrica un robot, luego el robot mata al hombre. Muchas películas han seguido mostrando a los robots como máquinas dañinas y amenazadoras. La palabra eslovaca '*Robota*' significa trabajo forzado, y cuando se tradujo al inglés se convirtió en el término robot, término que se continúa usando en la actualidad aunque el concepto haya evolucionado.

Son varios los factores que intervienen para que se desarrollaran los primeros robots en la década de 1950. La investigación en inteligencia artificial desarrolló maneras de emular el procesamiento de información humana con computadoras electrónicas e inventó una variedad de mecanismos para probar sus teorías.

Las primeras patentes aparecieron en 1946 con los muy primitivos robots para traslado de maquinaria de Devol. En 1954, Devol diseña el primer robot programable y acuña el término "autómata universal", que posteriormente recorta a Unimation. Así llamaría Engleberger a la primera compañía de robótica que iniciaría la comercialización de robots en 1959.

En 1964 se abren laboratorios de investigación en inteligencia artificial en el MIT (*Massachusetts Institute of Technology*), el SRI (*Stanford Research Institute*) y en la universidad de Edimburgo. Poco después los japoneses que anteriormente importaban su tecnología robótica, se sitúan como pioneros del mercado, puesto que conservan hasta la actualidad.

Actualmente, el concepto de robótica ha evolucionado hacia los sistemas móviles autónomos, que son aquellos que son capaces de desenvolverse por sí mismos en entornos desconocidos y parcialmente cambiantes sin necesidad de supervisión.

La robótica se debate entre modelos sumamente ambiciosos, como es el caso del *Kobian*, robot diseñado por la Universidad de Waseda para expresar emociones, el *Cog*, también conocido como el robot de cuatro sentidos, el famoso *Sojourner* o el *Lunar Rover*, vehículo de turismo con control remoto, y otros muchos más específicos como el *Cypher*, un helicóptero robot de uso militar, el guardia de tráfico japonés *Anzen Taro* o los robots mascotas de Sony.

En el campo de los robots antropomorfos (androides) se debe mencionar el ASIMO (acrónimo de “*Advanced Step in Innovative Mobility*”) de Honda (Figura 1.1) que es capaz de moverse, interactuar con los seres humanos y ayudarles; es sin duda, una de las mayores proezas tecnológicas del siglo XXI.



Figura 1.1. ASIMO de Honda¹

¹ HONDA WORLDWIDE, “ASIMO at YES 2008”. Disponible en: [http://world.honda.com/ASIMO/event/2008/080625/photo/pages/01.html], 2008

En general la historia de la robótica la podemos clasificar en cinco generaciones (división hecha por Michael Cancel, director del Centro de Aplicaciones Robóticas de Science Application Inc. en 1984). Las dos primeras, ya alcanzadas en los ochenta, incluían la gestión de tareas repetitivas con autonomía muy limitada. La tercera generación incluiría visión artificial, en lo cual se ha avanzado mucho en los ochenta y noventas. La cuarta incluye movilidad avanzada en exteriores e interiores y la quinta entraría en el dominio de la inteligencia artificial en lo cual se está trabajando actualmente.

1.2 Definición de un Robot Industrial

Un robot industrial, como el de la Figura 1.2, es un manipulador de uso general controlado generalmente por computadora que consiste en algunos elementos rígidos conectados en serie mediante articulaciones prismáticas o de revolución. El inicio de la cadena está fijo a una base soporte, mientras el extremo final está libre y equipado con una herramienta para manipular objetos o realizar otras tareas. El movimiento de las articulaciones resulta en, o produce, un movimiento relativo de los distintos elementos y por lo tanto en el desplazamiento de la herramienta o efector final.



Figura 1.2. Robot para soldadura de arco ²[ABB, 2008]

² ABB, “Arc welding robots”. Disponible en: [http://www02.abb.com/global/seitp/seitp202.nsf/0/505f16d72b44118bc1256fee004a3c7a/\$file/ABB+Welding+Robot.195jpg.jpg], 2008

Mecánicamente, un robot se compone de un brazo una muñeca más una herramienta. Se diseña para trabajar una pieza localizada dentro de su volumen de trabajo. El volumen de trabajo es el rango de influencia de un robot compuesta por el conjunto de puntos que el brazo puede alcanzar.

Los movimientos posibles para estos robots dependen de los grados de libertad que posean. El grado de libertad es el número de desplazamientos longitudinales o rotacionales independientes que puede tener un mecanismo. La cantidad de grados de libertad, y por lo tanto la complejidad del robot, dependen de la labor o labores que se desean realizar con él.

La mayoría de robots industriales de la actualidad, aunque están controlados por mini y microcomputadores, son básicamente simples máquinas posicionales. Ejecutan una tarea dada mediante la grabación de secuencias pre programadas de movimiento previamente enseñadas por el usuario. Más aún, la mayoría de estos robots industriales poseen pocos o ningún sensor externo para poder obtener información de su entorno de trabajo. Como resultado de esto, los robots se utilizan principalmente en tareas repetitivas simples.

En la actualidad se ve a la robótica como un campo de trabajo muy amplio desarrollando nuevas tecnologías en una serie de áreas interdisciplinarias como la cinemática, la dinámica, la planificación de sistemas, el control, sensores, lenguajes de programación e inteligencia artificial.

1.3 Clasificación de los Robots Industriales

Los robots industriales que están disponibles comercialmente se pueden clasificar de acuerdo a distintos criterios como el área de aplicación o su propia esencia, siendo la clasificación basada en su configuración la más común.

De acuerdo a su configuración un robot se puede clasificar en:

a) Robot cartesiano: Este robot tiene capacidad de desplazamiento paralelo a los ejes coordenados, como se muestra en la Figura 1.3. Estos robots presentan un volumen de trabajo cúbico; los actuadores de posición y orientación son accionados por fuentes hidráulicas, neumáticas, o eléctricas.



Figura 1.3. Robot cartesiano de tres ejes³

b) Robot cilíndrico: Este robot tiene capacidad de girar con respecto a un eje de referencia y además cuenta con dos desplazamientos lineales, como se muestra en la Figura 1.4. Estos robots tienen un volumen de trabajo parecido a un cilindro, aunque casi nunca completan los 360° de una vuelta completa.



Figura 1.4. Robot cilíndrico⁴

³ AIRLINE HYDRAULICS, “Three-Axis Cartesian Robot”. Disponible en: [http://www.airlinehyd.com/Images/Sub_Assemblies/sub1(Cartesian).jpg], 2008

⁴ INTERNATIONAL FEDERATION OF ROBOTICS, “Photograph of a Cylindrical Robot”. Disponible en: [http://www.ifr.org/picture/robots/struct1/jpeg/4c.jpg], 2004

c) Robot esférico: A este robot a veces se le llama polar, requiere del movimiento coordinado en cada eje de colocación (dos rotaciones y uno lineal) para obtener un movimiento en las direcciones X, Y o Z, como se observa en la Figura 1.5a. Por lo general son impulsados eléctrica o hidráulicamente.

d) Robot articular: Este robot está formado por una serie de articulaciones como se muestra en la Figura 1.5b. Varían la posición y orientación, gracias a su elevado número de grados de libertad (usualmente 5 o 6), son idóneos en un amplio y variado abanico de aplicaciones industriales, desde la soldadura por puntos, hasta aplicaciones de pintado y sellado.



(a)



(b)

Figura 1.5. (a) Robot esférico⁵, (b) Robot articular⁶

e) Robot SCARA: Los robots SCARA (*“Selective Compliant Assembly Robot Arm”*), como el de la Figura 1.6, son robots equipados de libertad total de movimientos en los ejes X e Y pero limitados severamente en sus desplazamientos en el eje Z.

⁵ INTERNATIONAL FEDERATION OF ROBOTICS, “Photograph of a Spherical Robot”. Disponible en: [<http://www.ifr.org/picture/robots/struct1/jpeg/fanupol.jpg>], 2004

⁶ INTERNATIONAL FEDERATION OF ROBOTICS, “Photograph of an Articulated Robot”. Disponible en: [<http://www.ifr.org/picture/robots/struct1/jpeg/Bild 8 c.jpg>], 2004

Es decir, se comportan de forma parecida al brazo humano, permitiendo ubicar el extremo de la mano en cualquier ubicación pero siempre sobre el plano. En el eje vertical solo realizan manipulaciones simples que habitualmente consisten en presionar y desplazarse unos pocos centímetros.

Debido a estas características se usan principalmente en la fabricación de electrónica de consumo y en la clasificación de artículos para su empaquetado.



Figura 1.6. Robot SCARA ⁷

f) Robot paralelo: Un robot paralelo, como el de la Figura 1.7, es aquel cuya estructura mecánica está formada por un mecanismo de cadena cerrada en el que el efector final se une a la base por al menos dos cadenas cinemáticas independientes.

Puesto que para definir totalmente la posición y orientación de un cuerpo en el espacio son necesarios seis parámetros, si se pretende que un robot posicione y oriente su extremo de cualquier modo en el espacio será necesario que el robot tenga por lo menos seis grados de libertad.

⁷ INTERNATIONAL FEDERATION OF ROBOTICS, "Photograph of a SCARA Robot". Disponible en: [http://www.ifr.org/picture/robots/struct1/jpeg/Bild_6_c.jpg], 2004d



Figura 1.7. Robot paralelo para levantar objetos ⁸

En la práctica es muy común encontrar robots que aunque no tienen seis grados de libertad el contar con cuatro o cinco grados de libertad les resulta suficiente para realizar las labores para las cuales fueron diseñados. Existen también casos en los cuales seis grados de libertad no resultan suficientes, como los robots que trabajan en un entorno con obstáculos. Un ejemplo sería un robot usado en competencias de *soccer* como el de la Figura 1.8, estos robots suelen tener más de veinte grados de libertad.

En muchas ocasiones los robots han de desarrollar operaciones que no consisten en manipular objetos sino que incluyen el uso de una herramienta. En general, esta herramienta debe ser construida o adaptada de manera específica para el robot, pero como hay aplicaciones que son comúnmente robotizadas se comercializan herramientas específicas para el uso de robots. Un ejemplo de estas actividades es el corte usando tecnología láser, aplicación definida para el robot a diseñar.

⁸ ABB, “ABB Flex Picker”. Disponible en: [http://www02.abb.com/global/seitp/seitp202.nsf/0/5dbbffa53149f395482574340009e6a5/\$file/Robot+FlexPicker_195.jpg], 2007



Figura 1.8. Robot para competencia Robo Soccer

1.4 Robots para corte con láser

La utilización de los láseres en la industria se presenta en dos modalidades: en aplicaciones metrológicas, formando parte de instrumentos de medición e inspección, y en el tratamiento de los materiales, usándose láseres de media y alta potencia para cortar, perforar, soldar, marcar, grabar y realizar tratamientos térmicos en los más diversos materiales como metales, plásticos, vidrios y cerámicos, papel, madera, etc.

Un sistema láser para tratamiento de materiales está compuesto principalmente por el láser, la óptica para focalizar la radiación láser sobre el material, el sistema para presentar y manipular la pieza a tratar y los servicios conexos. La óptica es de vital importancia para poder focalizar el láser sobre la pieza a tratar, como se muestra en la Figura 1.9. Esta óptica requiere de un preciso montaje optomecánico para poder posicionar adecuadamente el haz láser sobre la pieza a tratar y de un sistema CAM (“*Computer Aided Manufacturing*”) para controlar el movimiento.

El desarrollo de nuevos materiales, tales como nuevos polímeros y materiales compuestos, así como especificaciones de calidad más exigentes suponen nuevas demandas a la tecnología de corte. Una alternativa para estas nuevas exigencias es el

corte con láser, que es flexible y rápido porque usa una focalización integrada del haz en el brazo del robot, presenta una alta velocidad de corte, así como precisión al cortar plásticos y metales, sellar cantos de piezas textiles y de plástico. Además, estos robots no generan rebabas o virutas, ni ensucian la pieza y tampoco generan fuerzas reactivas sobre la pieza y el útil de fijación.

En el presente caso, los robots de corte con láser, se usan generalmente los robots del tipo articular, debido a que pueden ser impulsados directamente por motores eléctricos sin necesidad de un mecanismo para cambiar el tipo de movimiento, o del tipo pórtico.



Figura 1.9. Efecto de corte con láser⁹

⁹ NIBBLER, “Corte láser”. Disponible en:
[<http://www.nibbler.com.ar/UserFiles/Image/cortelaser.JPG>], 2008

CAPÍTULO 2

DISEÑO PRELIMINAR DEL MECANISMO

El proyecto presentado pertenece a un trabajo multidisciplinario desarrollado por tesisistas y asesores de las secciones de Ingeniería Mecánica e Ingeniería Electrónica. En esta tesis se necesita desarrollar solamente el diseño preliminar del mecanismo para estudiar la dinámica del mismo y así poder relacionar el movimiento descrito por la herramienta de trabajo del robot con el movimiento descrito por cada uno de los motores eléctricos.

Los resultados de este proyecto servirán como datos de referencia al sistema de control para asegurarse que la herramienta de trabajo siga la trayectoria deseada. En lo que se refiere al diseño mecánico del brazo robótico se hará también en un trabajo posterior.

2.1 Consideraciones o requerimientos

Se realizará el análisis para un brazo robótico capaz de realizar el corte con láser de maderas, además de otras labores industriales al cambiar el efector final, para este caso tenemos los siguientes requerimientos:

- a) Dimensiones: El alcance del brazo viene definido por un plano horizontal de tamaño A0, es decir, de 841 mm x 1189 mm. El movimiento del actuador en el eje perpendicular al plano no es requerido dado que todo el trabajo se hará en el plano.
- b) Geometría: Los movimientos a realizar por el brazo robótico harán que sea capaz de posicionarse en cada uno los puntos del plano de trabajo guardando siempre perpendicularidad con la superficie.
- c) Cinemática: Las velocidades del extremo final del mecanismo deben de ser regulables y dependen de la potencia de la boquilla para corte y del espesor de la madera a cortar. De acuerdo a lo ofrecido por algunos distribuidores de estas máquinas se observa que las velocidades de corte van de 25 a 80 mm/min. Según esta velocidad se deberían analizar las velocidades de cada uno de los motores, que es finalmente lo que se controlará.
- d) Pesos e inercias: Deben de ser diseñados de manera que la inercia que tengan con respecto al eje de giro sea pequeña, de otra manera se estaría sobre exigiendo a los motores. Según las herramientas a usar para las tareas deseadas se puede estimar el peso a soportar por el efector final entre 5 y 15 kg¹⁰.

2.2 Mecanismo propuesto

Tomando como base las necesidades de movimiento propuestas se requiere que el mecanismo tenga 4 grados de libertad. Ante tal caso se propone un sistema con tres eslabones en serie, siendo al final del último eslabón donde iría adaptada la herramienta de trabajo. El brazo sería del tipo TRRR (Traslacional-rotacional-rotacional-rotacional), como se muestra en la Figura 2.1. Es importante mencionar que se selecciona esta disposición frente a otras más simples porque es posible ser usada posteriormente en labores más complicadas, en las cuales la orientación del efector final no necesariamente sea perpendicular al plano y donde la superficie de trabajo no necesariamente sea horizontal.

¹⁰ PRAXAIR SOLDADURA, S.L. Catálogos de productos para soldadura y corte.

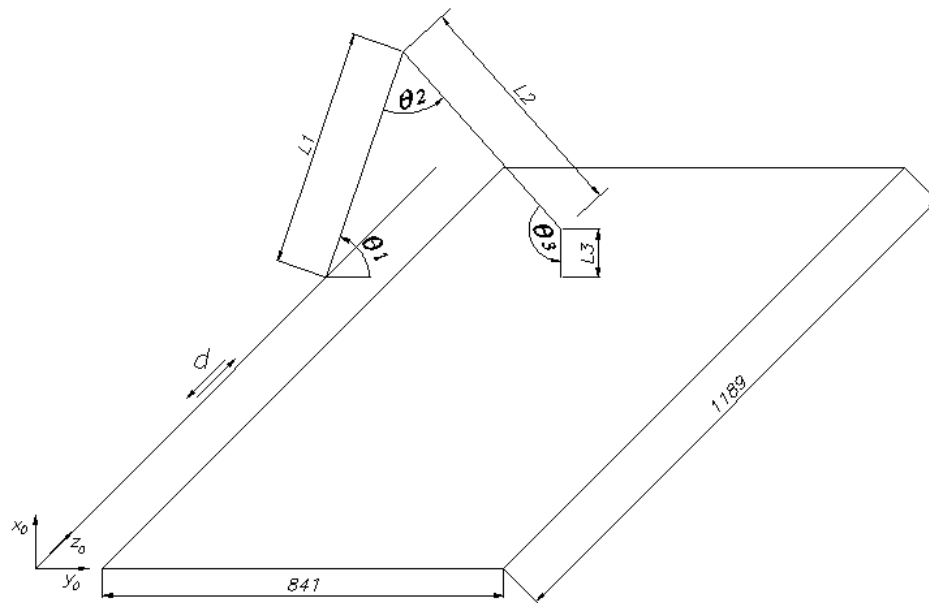


Figura 2.1. Modelo preliminar

Con este esquema se tienen las siguientes posibilidades de movimiento:

- i. Un par cinemático traslacional en un eje paralelo al lado mayor del plano. Este grado de libertad permite que la cadena cinemática abierta formada por los tres eslabones se pueda ubicar en cualquier posición en el eje z_0 .
- ii. Un par cinemático rotacional que permite el giro de los tres eslabones en el eje z_0 .
- iii. Un par cinemático rotacional que permite el giro de los dos últimos eslabones.
- iv. Un par cinemático rotacional que permite el giro del último eslabón.

Con ellas el extremo de la cadena cinemática abierta tiene la posibilidad de desplazarse en todo el plano de trabajo conservando la perpendicularidad con respecto a la superficie.

Al tener decididos los grados de libertad, sólo tendríamos que decidir las longitudes de cada uno de los eslabones sabiendo que se debe poder acceder a cada punto del área de trabajo.

En este caso se han tenido las siguientes consideraciones:

- a) La base del primer eslabón se desliza paralelamente al lado mayor del plano A0, conservando distancia de hasta 100 mm con respecto al plano de trabajo.
- b) El primer eslabón puede girar con respecto a su base, es decir, tiene un grado de libertad de revolución.
- c) La unión entre el primer y el segundo eslabón, así como la unión del segundo y el tercero permiten giros.
- d) El tercer eslabón debe conservar perpendicularidad con respecto al plano de trabajo.
- e) Como el tercer eslabón es sólo para conservar perpendicularidad no necesita tener mucha longitud, razón por la cual se fijó en 100 mm.
- f) El primer y el segundo eslabón tendrán la misma longitud para facilitar las labores en manufactura.

Según estas consideraciones se empezó a determinar longitudes que cumplieran con el plano de trabajo, es decir, que logren alcanzar los dos puntos extremos del plano conservando la perpendicularidad del tercer eslabón con respecto al plano.

Después de realizar pruebas con algunas longitudes se obtuvieron resultados positivos para 500 mm de longitud. Para este valor se accede a ambos puntos sin necesidad de ángulos muy exigidos, como se ve en la Figura 2.2.

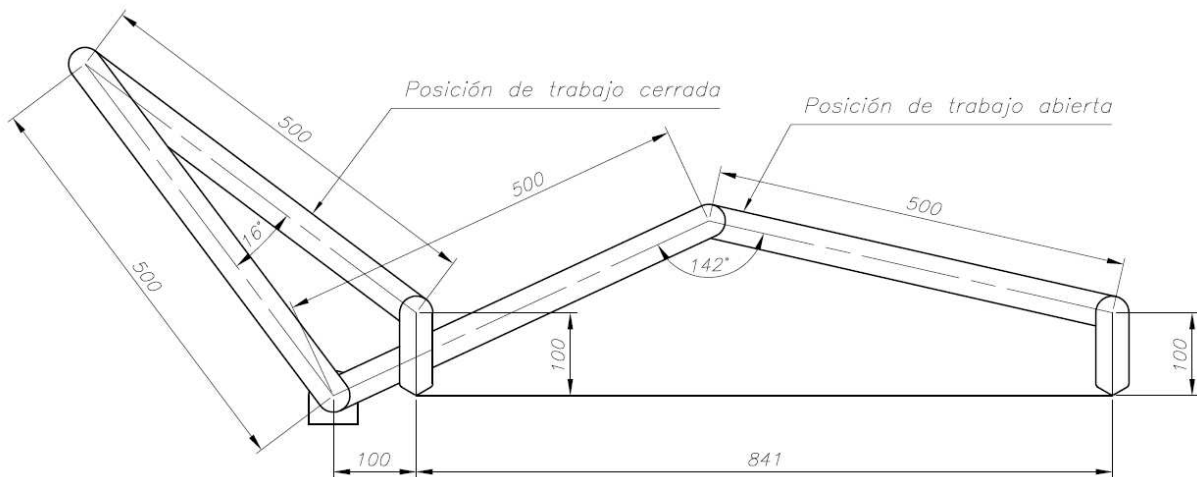


Figura 2.2. Eslabones del mecanismo

2.3 Cargas en el mecanismo

El conocimiento de las cargas presentes en el mecanismo es esencial para el estudio de la dinámica del brazo robótico. En el presente mecanismo se encuentran cargas en la forma de fuerzas y momentos (pares) ocasionados por el peso de los eslabones y el peso de los motores principalmente.

2.3.1 Eslabones

Los eslabones de los brazos robóticos generalmente se construyen de materiales ligeros, debido principalmente a que un sobrepeso en eslabones puede conducir a exigir demasiado torque del motor.

El momento (par) exigido a los motores depende no sólo del peso sino también de las dimensiones de los eslabones. Estas últimas están definidas por la cadena cinemática obtenida en la síntesis del mecanismo, entonces para poder disminuir la magnitud de las cargas se debe tratar de aligerar los eslabones sin afectar su longitud y preferentemente tampoco su rigidez para evitar excesivas deformaciones.

2.3.2 Motores

Los motores se ubican en los extremos de los eslabones, debido a esa configuración van a generar una fuerza y dos momentos: uno flector y uno torsor.

El efecto de la fuerza no puede compensarse de otra manera más que con el torque a suministrar por los motores eléctricos.

En el caso del momento flector a ocasionar por el peso, esto obliga a analizar la rigidez de los eslabones, ya que en el diseño mecánico probablemente se deba tener en cuenta la inercia de los eslabones con respecto al eje de flexión.

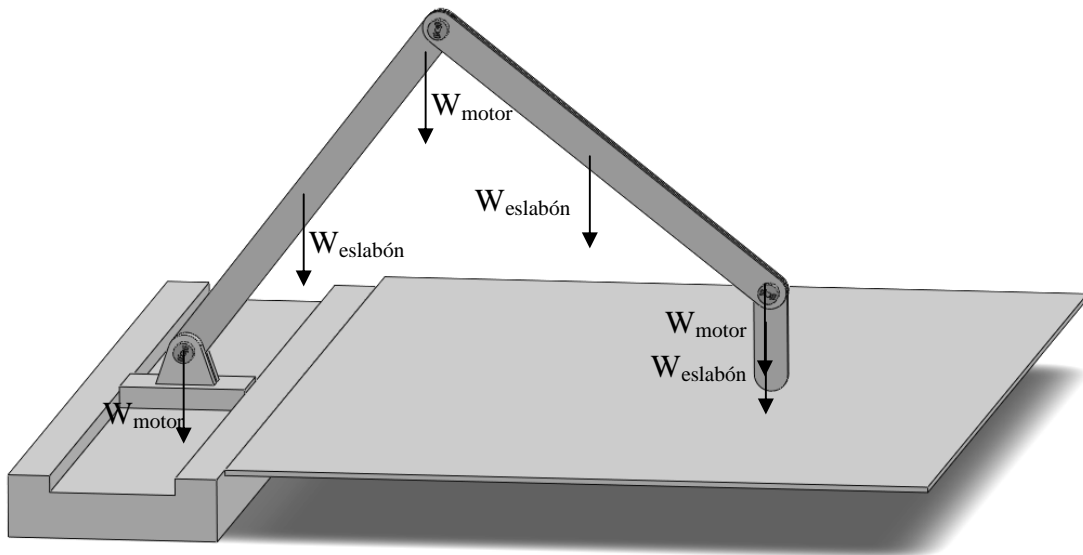


Figura 2.3. Fuerzas y momentos producidos por los pesos

Las fuerzas que influyen en la modelación de la dinámica del sistema son los pesos de los motores y eslabones, que se consideran como fuerzas externas. Estos pesos ocasionan en el sistema momentos flectores y torsores. Los momentos flectores se ocasionan por el brazo de palanca que se forma al abrirse el brazo, y los momentos torsores se ocasionarían porque el motor estaría ubicado a un solo lado de los eslabones.

2.3.3 Implementación de contrapesos

En el diseño del brazo robótico se podrían incluir ciertos contrapesos para balancear el movimiento del mismo:

- i. En cuanto al momento torsor es una carga descompensada porque aparece sólo en un lado del mecanismo, pero esta carga se podría evitar al hacer uso de un contrapeso del lado contrario de cada eslabón como se muestra en la Figura 2.4.
- ii. El peso de los eslabones genera un momento con respecto al centro de giro lo cual facilita el movimiento en un sentido pero lo dificulta en el otro. Siendo éste el caso se puede proponer el uso de un contrapeso que facilite el movimiento del brazo en sentido contrario al de la aceleración de la gravedad.

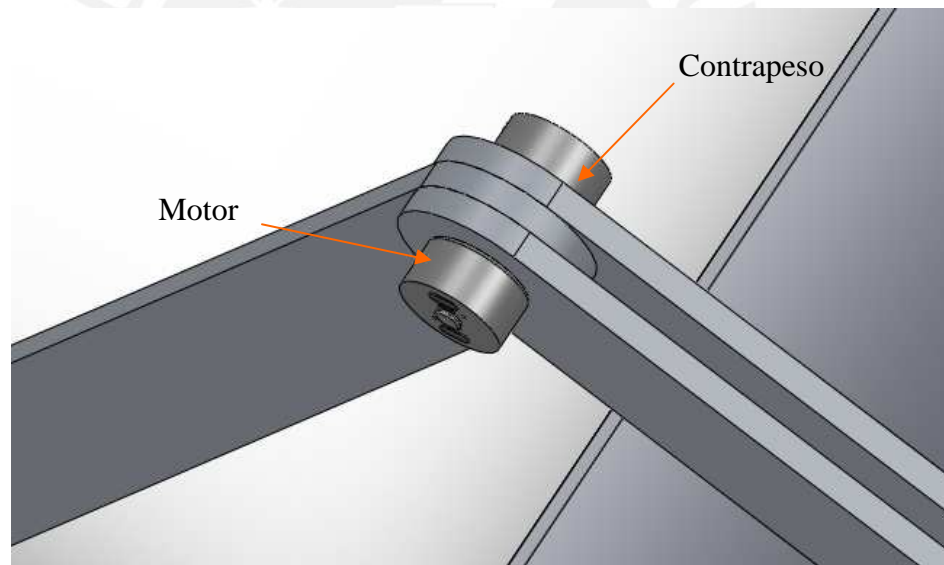


Figura 2.4. Representación de los contrapesos para el efecto torsor

CAPÍTULO 3

MODELACIÓN DEL MECANISMO

El objetivo de este capítulo es el estudio del movimiento de brazo robótico, y para ello se toman dos partes: la cinemática y la dinámica.

3.1 Análisis de la cinemática

La cinemática del robot estudia el movimiento del mismo con respecto a un sistema de referencia sin considerar las fuerzas que lo producen. Es decir, que se interesa por la descripción analítica de la trayectoria espacial del robot como una función del tiempo, y en particular por las relaciones entre la posición y orientación del extremo final del robot con los valores angulares de las articulaciones.

Dentro de la cinemática existen dos problemas fundamentales, que son la cinemática directa y la cinemática inversa.

El problema cinemático directo consiste en determinar la posición y orientación del efector final del robot, con respecto a un sistema de coordenadas tomado como referencia, al conocer los valores angulares de las articulaciones y los parámetros geométricos de los elementos del robot.

En cambio, el problema cinemático inverso trata sobre la configuración que el robot debe de adoptar para alcanzar unas determinadas posición y orientación del efector final, véase Figura 3.1.

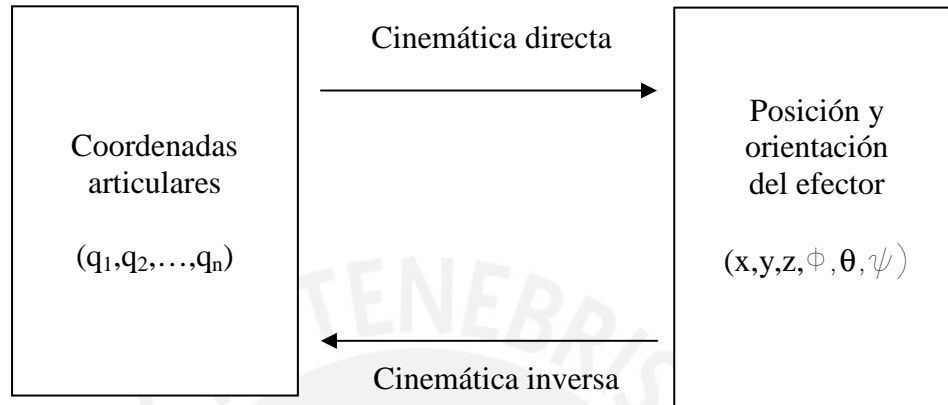


Figura 3.1. Diagrama de relación entre cinemática directa e inversa

Para el trabajo de la cinemática Denavit y Hartenberg [DENAVID, 1995] propusieron un método sistemático para describir y representar la geometría espacial de los elementos de la cadena cinemática, y en el caso particular de un robot, con respecto a un sistema de referencia fijo. Este método hace uso de la *matriz de transformación homogénea* para describir la relación espacial entre dos elementos rígidos adyacentes, reduciendo el problema a hallar una matriz de transformación 4 x 4 que relacione la localización espacial del extremo del robot con respecto al sistema de coordenadas de su base.

Una matriz de transformación homogénea es una matriz 4 x 4, como en la ecuación 3.1, que representa la transformación de un vector de coordenadas homogéneas de un sistema de coordenadas a otro.

$$T = \begin{bmatrix} \mathbf{R}_{3 \times 3} & \mathbf{p}_{3 \times 1} \\ \mathbf{f}_{1 \times 3} & w_{1 \times 1} \end{bmatrix} = \begin{bmatrix} \text{Rotación} & \text{Traslación} \\ \text{Perspectiva} & \text{Escalado} \end{bmatrix} \quad (3.1)$$

Se puede considerar que una matriz homogénea está compuesta por cuatro submatrices de distinto tamaño: una submatriz $\mathbf{R}_{3 \times 3}$ que simboliza a la rotación, una

submatriz $p_{3 \times 1}$ que corresponde a la traslación, una submatriz $f_{1 \times 3}$ que representa a una transformación de perspectiva y una submatriz $w_{1 \times 1}$ que representa a la escala.

3.1.1 Cinemática directa

Conocer la cinemática directa del modelo permite relacionar las coordenadas generalizadas con las coordenadas cartesianas en la punta del brazo usando como datos los movimientos realizados en cada uno de los pares cinemáticos. Para lograr realizar este cálculo se proponen dos métodos, los cuales son:

- a) Resolución por métodos geométricos:

Este es el método más sencillo y consiste en la resolución de la cinemática usando cálculos geométricos simples. El mecanismo del brazo robótico es equivalente al que se muestra en la Figura 3.2.

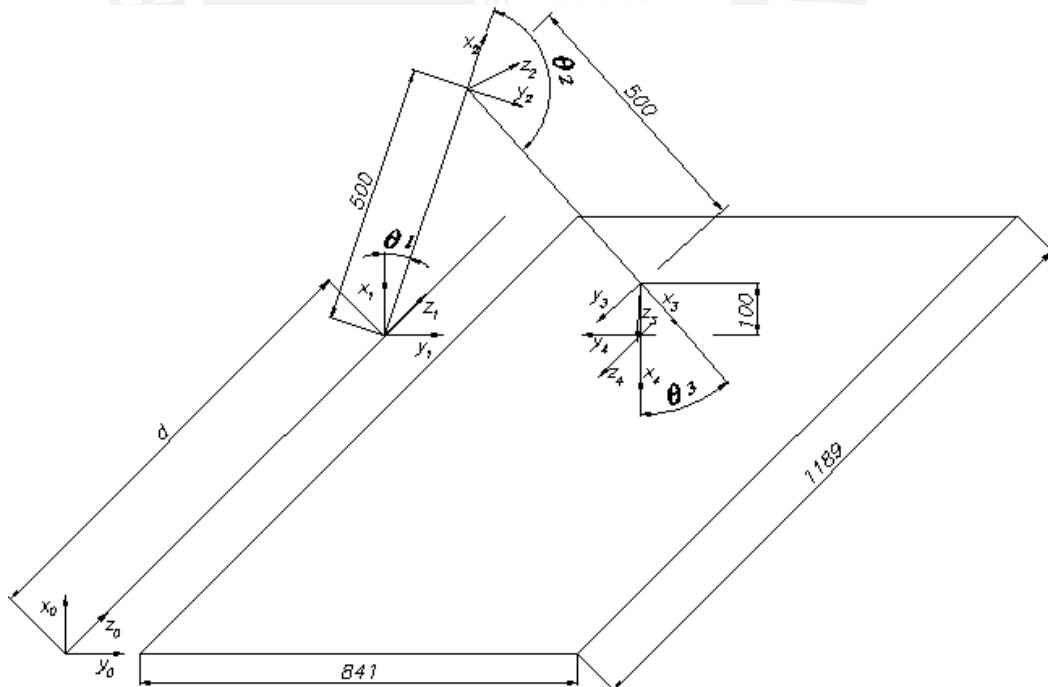


Figura 3.2. Esquema cinemático

El análisis de la cinemática de este mecanismo se puede simplificar al analizar sólo el movimiento en un plano en vez del movimiento en el espacio. Basándonos en el sistema de coordenadas de referencia $X_0 Y_0 Z_0$ entonces el efector final tendrá su posición definida por un vector del tipo (x, y, z) . Es importante mencionar que cuando el robot se encuentra en su espacio de trabajo el vector final será del tipo $(0, y, z)$ porque no habría desplazamiento en el eje X_0 perpendicular al plano.

Luego, la proyección de la posición en el eje Z_0 quedará definida por el primer grado de libertad del sistema, entonces es posible reducir el análisis cinemático directo a hallar el valor de la componente Y_0 para un sistema de tres grados de libertad.

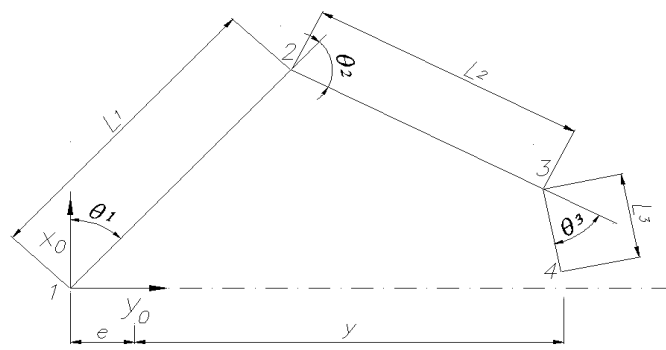


Figura 3.3. Cadena cinemática abierta en el plano

Definiendo la posición de los puntos 1, 2, 3 y 4 en función de los ángulos θ se obtiene:

Punto 1: $(0 ; 0)$

Punto 2: $(l_1 \times \cos(\theta_1) ; l_1 \times \sin(\theta_1))$

Punto 3: $(l_1 \times \cos(\theta_1) + l_2 \times \cos(\theta_2 + \theta_1) ; l_1 \times \sin(\theta_1) + l_2 \times \sin(\theta_2 + \theta_1))$

Punto 4: $(l_1 \times \cos(\theta_1) + l_2 \times \cos(\theta_2 + \theta_1) + l_3 \times \cos(\theta_3 + \theta_2 + \theta_1) ;$
 $l_1 \times \sin(\theta_1) + l_2 \times \sin(\theta_2 + \theta_1) + l_3 \times \sin(\theta_3 + \theta_2 + \theta_1))$

Entonces, cuando el mecanismo está trabajando a partir de las relaciones geométricas se obtienen las siguientes ecuaciones:

$$l_1 \times \cos(\theta_1) + l_2 \times \cos(\theta_2 + \theta_1) + l_3 \times \cos(\theta_3 + \theta_2 + \theta_1) = x = 0 \quad (3.2)$$

$$l_1 \times \sin(\theta_1) + l_2 \times \sin(\theta_2 + \theta_1) + l_3 \times \sin(\theta_3 + \theta_2 + \theta_1) = y \quad (3.3)$$

Además, para conseguir la perpendicularidad con respecto a la superficie de trabajo se trabaja con una restricción más, que sería $(90^\circ - \theta_1) + (180^\circ - \theta_2) + (180^\circ - \theta_3) + 90^\circ = 360^\circ$, lo que deriva en la ecuación 3.4.

$$\theta_1 + \theta_2 + \theta_3 = 180^\circ \quad (3.4)$$

Con estas cuatro incógnitas (θ_1 , θ_2 , θ_3 , y) y tres ecuaciones ((3.2), (3.3) y (3.4)) se puede definir la posición final “y” del efector final conociendo sólo uno de los ángulos θ .

Extendiendo los resultados para un análisis en el espacio:

Variables definidas por los grados de libertad: $d, \theta_1, \theta_2, \theta_3$

Variables a determinar: (x, y, z)

Al derivar este conjunto de ecuaciones se podrán analizar las relaciones dinámicas del sistema.

Ejemplo de cálculo

Definir la posición del efector final del mecanismo conociendo:

$$z = 300, \theta_2 = 45^\circ, l_1 = 500, l_2 = 500, l_3 = 100$$

Reemplazando en la ecuación (3.4): $\theta_3 = 160^\circ - \theta_1$

Luego, se reemplaza el resultado en la ecuación (3.2):

$$500 \times \cos(\theta_1) + 500 \times \cos(\theta_1 + 45^\circ) - 100 = 0$$

$$\theta_1 = 61.286^\circ$$

Finalmente, este valor se reemplaza en la ecuación (3.3) y se obtiene: $y = 918.45$

b) Resolución por métodos matriciales

El problema cinemático directo se basa en la búsqueda de una *matriz de transformación* que relacione la posición y orientación del extremo del manipulador representado por el sistema de coordenadas $X_4Y_4Z_4$ con respecto a un sistema de coordenadas de referencia $X_0Y_0Z_0$, mostrado en la Figura 3.2.

En general, un robot articular de n grados de libertad está formado por n eslabones unidos por n articulaciones, de manera que cada articulación – eslabón constituye un grado de libertad. A cada eslabón se le puede asociar con un sistema de referencia. La matriz de transformación homogénea que relacionan la posición y orientación de dos eslabones consecutivos se denomina matriz ${}^{i-1}A_i$.

Las *matrices de transformación* T son relaciones existentes entre la posición y orientación del extremo de un robot respecto del sistema de referencia fijo situado en la base del mismo. Para un sistema de n grados de libertad sería el producto de las n matrices de transformación de cada articulación – eslabón, como se muestra a continuación:

$$T = {}^0A_n = {}^0A_1 {}^1A_2 {}^2A_3 \dots {}^{n-1}A_n \quad (3.5)$$

Para describir la relación existente entre dos elementos continuos habitualmente se hace uso de la representación de Denavit-Hartenberg [DENAVID, 1995]. Este método matricial establece la localización que debe tomar cada sistema de coordenadas ligado a cada eslabón de una cadena articulada, para así sistematizar la obtención de las ecuaciones cinemáticas de la cadena completa.

Al escoger los sistemas de coordenadas asociados a cada eslabón según la representación propuesta por Denavit-Hartenberg será posible pasar de uno al siguiente mediante 4 transformaciones básicas que dependen exclusivamente de la geometría del eslabón.

Estas 4 transformaciones consisten en una sucesión de rotaciones y traslaciones que permiten relacionar el sistema de referencia de un eslabón i con el sistema de su elemento antecesor $i-1$. Las transformaciones en cuestión son:

1. Rotación alrededor del eje z_{i-1} un ángulo θ_i .
2. Traslación a lo largo de z_{i-1} una distancia d_i ; vector d_i $(0, 0, d_i)$.
3. Traslación a lo largo de x_i una distancia a_i ; vector a_i $(a_i, 0, 0)$.
4. Rotación alrededor del eje x_i un ángulo α_i .

Con estos valores se pueden determinar matrices de transformación que relacionarán los sistemas de coordenadas de un eslabón al sistema de coordenadas del precedente, como se muestra en la ecuación 3.6.

$${}^{i-1}A_i = \text{Rot}_z(\theta_i)T(0,0,d_i)\Gamma(a_i,0,0)\text{Rot}_x(\alpha_i) \quad (3.6)$$

En la ecuación (3.6) se muestra cómo calcular la matriz de transformación para relacionar dos elementos adyacentes haciendo uso de matrices de rotación y traslación.

Es decir, que obteniendo los parámetros de Denavit-Hartenberg, mostrados en la Tabla 3.1, fácilmente se pueden hallar las matrices de transformación homogénea.

	θ	d	a	α
1	0	d	0	0
2	θ_1	0	500	0
3	θ_2	0	500	0
4	θ_3	0	100	0

Cuadro 3.1. Parámetros de Denavit-Hartenberg

Reemplazando los valores en la ecuación (3.6) se obtendrían las siguientes matrices de transformación:

1) Para el primer elemento:

$${}^0A_1 = T_{(0,0,d)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.7)$$

2) Para el segundo elemento:

$${}^1A_2 = \text{Rotz}_{(\theta_1)} T_{(11,0,0)} = \begin{bmatrix} \cos(\theta_1) - \text{sen}(\theta_1) & 0 & 0 & 0 \\ \text{sen}(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & l_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.8)$$

$${}^1A_2 = \text{Rotz}_{(\theta_1)} T_{(11,0,0)} = \begin{bmatrix} \cos(\theta_1) - \text{sen}(\theta_1) & 0 & l_1 \cos(\theta_1) \\ \text{sen}(\theta_1) & \cos(\theta_1) & l_1 \text{sen}(\theta_1) \\ 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.9)$$

3) Para el tercer elemento:

$${}^2A_3 = \text{Rotz}_{(\theta_2)} T_{(12,0,0)} = \begin{bmatrix} \cos(\theta_2) - \text{sen}(\theta_2) & 0 & 0 & 0 \\ \text{sen}(\theta_2) & \cos(\theta_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.10)$$

$${}^2A_3 = \text{Rotz}_{(\theta_2)} T_{(12,0,0)} = \begin{bmatrix} \cos(\theta_2) - \text{sen}(\theta_2) & 0 & l_2 \cos(\theta_2) \\ \text{sen}(\theta_2) & \cos(\theta_2) & l_2 \text{sen}(\theta_2) \\ 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.11)$$

4) Para el cuarto elemento:

$${}^3A_4 = \text{Rotz}_{(\theta_3)} T_{(13,0,0)} = \begin{bmatrix} \cos(\theta_3) - \text{sen}(\theta_3) & 0 & 0 & 0 \\ \text{sen}(\theta_3) & \cos(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & l_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.12)$$

$${}^3A_4 = \text{Rotz}_{(03)}T_{(l_3,0,0)} = \begin{bmatrix} \cos(\theta_3) & -\text{sen}(\theta_3) & 0 & l_3\cos(\theta_3) \\ \text{sen}(\theta_3) & \cos(\theta_3) & 0 & l_3\text{sen}(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.13)$$

Entonces la matriz de transformación general sería el producto de las matrices de transformación de cada elemento como se mencionó anteriormente:

$$T = {}^0A_n = {}^0A_1 {}^1A_2 {}^2A_3 {}^3A_4 \quad (3.14)$$

Ejemplo de cálculo

Definir la posición del efector final del mecanismo para los siguientes datos:

$$d = 300, \theta_1 = 74.17^\circ, \theta_2 = 20^\circ, \theta_3 = 85.83^\circ, l_1 = 500, l_2 = 500, l_3 = 100$$

Reemplazando los valores en las ecuaciones (3.7), (3.9), (3.11) y (3.13) se obtiene:

$${}^0A_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 300 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.15)$$

$${}^1A_2 = \begin{bmatrix} 0.2728 & -0.9621 & 0 & 136.3920 \\ 0.9621 & 0.2728 & 0 & 481.0376 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.16)$$

$${}^2A_3 = \begin{bmatrix} 0.9397 & -0.3420 & 0 & 469.8463 \\ 0.3420 & 0.9397 & 0 & 171.0101 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.17)$$

$${}^3A_4 = \begin{bmatrix} 0.0727 & -0.9974 & 0 & 7.2716 \\ 0.9974 & 0.0727 & 0 & 99.7353 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.18)$$

Multiplicando los valores se obtiene:

$${}^0A_4 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 979.72 \\ 0 & 0 & 1 & 300 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.19)$$

Entonces la posición del efector final estará definida por el vector posición:

$$(0, 979.72, 300) \text{ mm}$$

El cual coincide con los valores hallados con el método geométrico, con lo cual se observa que con ambos métodos se puede llegar a la misma respuesta. A pesar que la obtención de las matrices de transformación no es más que la aplicación de un algoritmo, sus resultados son de vital importancia para los pasos siguientes en la simulación dinámica.

3.1.2 Cinemática inversa

El conocimiento de la cinemática inversa del brazo mecánico nos permite obtener, mediante el conocimiento de la posición y orientación deseadas para el efector final, el valor que deben de tener los ángulos entre los eslabones del mecanismo.

A diferencia del problema cinemático directo, en este caso no existe una manera sistemática de obtener una solución, siendo el procedimiento de obtención fuertemente dependiente de la configuración del robot.

Para resolver sistemas de pocos grados de libertad se recomienda el uso de:

a) Resolución por métodos geométricos

Este es el método más sencillo y consiste en la resolución de la cinemática usando solamente cálculos geométricos simples. El mecanismo del brazo robótico es igual al que se muestra en la Figura 3.2, usado para el cálculo de la cinemática directa.

Al igual que en la cinemática directa el cálculo se simplificará al analizar solamente el movimiento en un plano en vez de analizarlo en el espacio. Tomando como base en las variables definidas en la Figura 3.3 ahora lo que se buscará obtener son los valores de los ángulos θ para una posición deseada.

Tomando como base en las ecuaciones (3.2), (3.3) y (3.4) se llega a un sistema de ecuaciones no lineales, cuya respuesta son los ángulos θ .

Ejemplo de cálculo

Definir los valores de los ángulos θ para la configuración deseada si tenemos de datos:

$$z = 300, l_1 = 500, l_2 = 500, l_3 = 100, y = 918.45$$

Reemplazando los valores en las ecuaciones (3.2) y (3.3) se obtiene:

$$f_1(\theta_1, \theta_2) = 500 \times \cos(\theta_1) + 500 \times \cos(\theta_2 + \theta_1) - 100 = 0 \quad (3.20)$$

$$f_2(\theta_1, \theta_2) = 500 \times \sin(\theta_1) + 500 \times \sin(\theta_2 + \theta_1) - 918.45 = 0 \quad (3.21)$$

Por lo que ahora se cuenta con un sistema no lineal de dos incógnitas y dos ecuaciones, el cual se resolverá mediante el Método de Newton-Raphson. Se inicia definiendo la función $F(\theta_1, \theta_2)$:

$$F(\theta_1, \theta_2) = \begin{bmatrix} f_1(\theta_1, \theta_2) \\ f_2(\theta_1, \theta_2) \end{bmatrix} \quad (3.22)$$

Dicho método se ha implementado en la rutina presentada en el Cuadro 3.1.

```

% Newton Raphson solución de un sistema de dos ecuaciones no
% lineales por medio de iteración

error1 = 1.e8;
xx(1) = 1.0; % valores iniciales de iteración
xx(2) = 0.5;
iter=0;
itermax=50.
% inicia iteración
while error1>1.e-6
iter=iter+1;
x = xx(1);
y = xx(2);
% Ingresar las dos funciones iguales a cero
f(1) = 500*cos(x)+500*cos(y+x)-100;
f(2) = 500*sin(x)+500*sin(y+x)-918.45;
% Ingresar el Jacobiano del sistema
J(1,1) = -500*sin(x)-500*sin(y+x);
J(1,2) = -500*sin(y+x);
J(2,1) = 500*cos(x)+500*cos(y+x);
J(2,2) = 500*cos(y+x);
% resuelve las ecuaciones lineales
y = -J\f';
% mueve las soluciones, xx(k+1) - xx(k), a xx(k+1)
xx = xx + y';
% calcula norma
error1=sqrt(y(1)*y(1)+y(2)*y(2))
error(iter)=sqrt(f(1)*f(1)+f(2)*f(2));
ii(iter)=iter;
if (iter > itermax)
error1 = 0.;
s=sprintf('****No convergió en las iteraciones.****',itermax);
disp(s)
end
% verifica si error1 < 1.e-6
end
x = xx(1);
y = xx(2);
f(1) = 500*cos(x)+500*cos(y+x)-100;
f(2) = 500*sin(x)+500*sin(y+x)-918.45;
% imprime resultado
f
xx
iter
% imprime gráfica de tendencia
semilogy(ii,error)
xlabel('número de iteraciones')
ylabel('norma de funciones')
clear ii
clear error
  
```

Cuadro 3.2. Rutina del método de Newton-Raphson

Este método necesita de cierta información para empezar las iteraciones, como por ejemplo:

a) Los valores iniciales para iniciar las iteraciones, los cuales se definen como:

$$\theta_1 = 1,0 \quad \theta_2 = 0,5$$

b) El máximo valor de error admisible se define como: 10^{-6}

c) Máximo número de iteraciones: 50

Los resultados de correr la rutina se presentan en el Cuadro 3.2.

```

>> newraph

itermax =
    50

error1 =
    0.53575544067037

error1 =
    0.25027250326918

error1 =
    0.03004354098289

error1 =
    4.984181678665200e-004

error1 =
    1.329726878070880e-007

f =
    1.0e-011 *
    -0.03126388037344  -0.15916157281026

xx =
    1.06964113639084  0.78540661496581

iter =
    5
  
```

Cuadro 3.3. Resultados de correr la rutina de Newton-Raphson

Es decir, que luego de cinco iteraciones se obtiene: $\theta_1 = 1.070 \text{ rad} = 61.306^\circ$
 $\theta_2 = 0.785 \text{ rad} = 44.978^\circ$

En la Figura 3.4 se presenta la gráfica de tendencia resultado del programa.

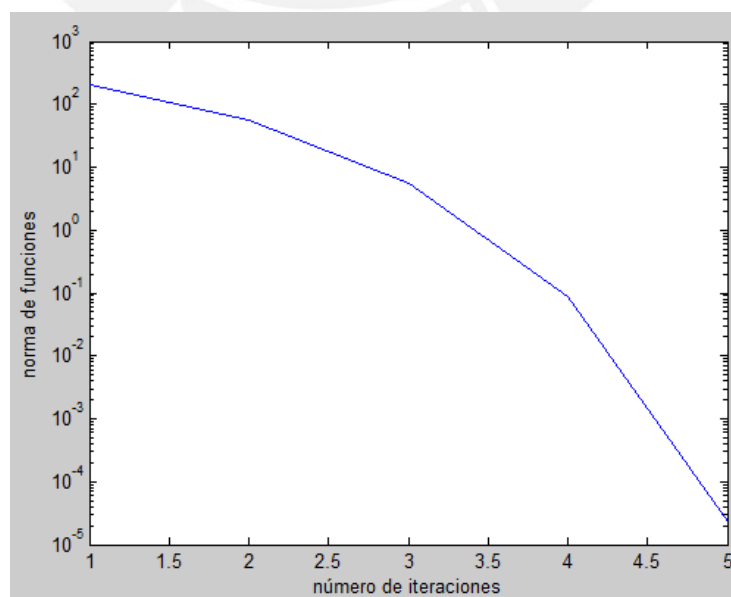


Figura 3.4. Gráfica de tendencia

En la Figura 3.4 se muestra lo rápido que converge este método para el caso planteado, pero es importante recordar que en algunas ocasiones ni siquiera llega a converger dependiendo esto del punto inicial elegido para las iteraciones.

Sería interesante considerar un caso en el cual se logre obtener una expresión analítica directa para relacionar las coordenadas articulares con la ubicación del efector final. Si bien es cierto que esto no se puede obtener para el caso general de este robot de 4 GDL, sí es posible en el caso en que se estudie el movimiento enteramente en el plano, es decir, el desplazamiento del efector final una vez que se ha ubicado en el plano y está listo para trabajar.

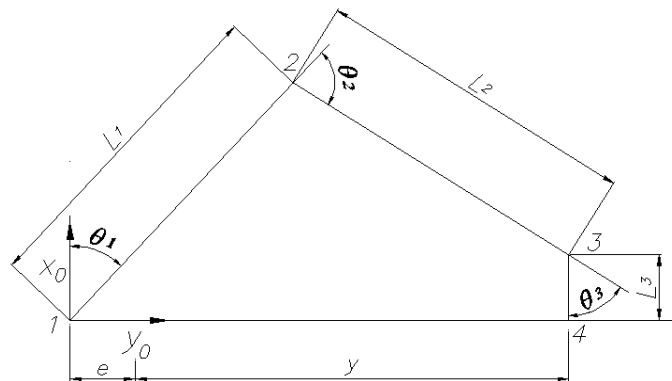


Figura 3.5. Cadena cinemática con extremo perpendicular a la superficie

Si consideramos que el tercer eslabón se va a conservar siempre perpendicular con respecto a la superficie de trabajo entonces el sistema se puede simplificar aún más.

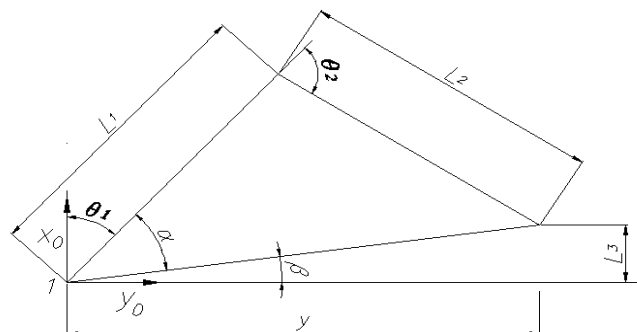


Figura 3.6. Cadena cinemática equivalente de dos eslabones

Con la disposición mostrada se puede notar que el extremo del segundo eslabón tendrá coordenadas de la forma (l_3, y) y que haciendo uso de trigonometría se pueden hallar expresiones para θ_1 y θ_2 . Sin embargo, al tener este sistema simplificado de dos grados de libertad tendremos dos posibles respuestas para los ángulos representando las dos posibles disposiciones de la articulación intermedia: con el codo hacia arriba o con el codo hacia abajo. Para el trabajo que se le planea dar al mecanismo lo conveniente será que trabaje con el codo hacia arriba, y definiendo eso se limita el rango de θ_2 entre 0 y π .

Usando el teorema del coseno se llega a la expresión:

$$\frac{l_3^2 + y^2 - l_1^2 - l_2^2}{2l_1l_2} = \cos\theta_2 \quad (3.23)$$

De donde se puede despejar el valor del ángulo θ_2 :

$$\arccos\left(\frac{l_3^2 + y^2 - l_1^2 - l_2^2}{2l_1l_2}\right) = \theta_2 \quad (3.24)$$

La función arco coseno está definida en MATLAB entre 0° y 180° , al igual que nuestra función θ_2 , por lo que obtendremos el valor deseado para nuestro caso a pesar de que la función coseno sea una función periódica.

Luego, el ángulo θ_1 también puede ser expresado en función de la posición del efector final como se muestra en la Figura 3.6.

$$\theta_1 = \frac{\pi}{2} - \alpha - \beta \quad (3.25)$$

$$\alpha = \arctan\left(\frac{l_2 \sin\theta_2}{l_1 + l_2 \cos\theta_2}\right) \quad (3.26)$$

$$\beta = \arctan\left(\frac{l_3}{y}\right) \quad (3.27)$$

Finalmente, tomando la expresión (3.4) se obtienen todas las coordenadas articulares en función de la posición del extremo final.

3.1.3 Modelo diferencial

El modelado cinemático, sin embargo, no termina al encontrar relaciones entre las variables articulares y la posición y orientación del efector final, sino que además se tiene que considerar la relación entre las velocidades de las coordenadas articulares y las de la posición y orientación del extremo. Esta relación queda definida por el modelo diferencial.

En este caso se hará uso de la llamada Jacobiana geométrica que permite relacionar las velocidades articulares y la velocidad lineal y angular del extremo del robot. Este método hace uso de la matriz de transformación homogénea que se haya según la ecuación (3.14).

$$\text{Rotación}_1 = \begin{bmatrix} \cos\theta_1(\cos\theta_2\cos\theta_3 - \text{sen}\theta_2\text{sen}\theta_3) - \text{sen}\theta_1(\text{sen}\theta_2\cos\theta_3 + \cos\theta_2\text{sen}\theta_3) \\ \text{sen}\theta_1(\cos\theta_2\cos\theta_3 - \text{sen}\theta_2\text{sen}\theta_3) + \cos\theta_1(\text{sen}\theta_2\cos\theta_3 + \cos\theta_2\text{sen}\theta_3) \\ 0 \end{bmatrix} \quad (3.28)$$

$$\text{Rotación}_2 = \begin{bmatrix} -\cos\theta_1(\cos\theta_2\text{sen}\theta_3 + \text{sen}\theta_2\cos\theta_3) - \text{sen}\theta_1(\cos\theta_2\cos\theta_3 - \text{sen}\theta_2\text{sen}\theta_3) \\ -\text{sen}\theta_1(\cos\theta_2\text{sen}\theta_3 + \text{sen}\theta_2\cos\theta_3) + \cos\theta_1(\cos\theta_2\cos\theta_3 - \text{sen}\theta_2\text{sen}\theta_3) \\ 0 \end{bmatrix} \quad (3.29)$$

$$\text{Rotación}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (3.30)$$

$$\text{Traslación} = \begin{bmatrix} c\theta_1(l_3(c\theta_3c\theta_2 - s\theta_3s\theta_2) + l_2c\theta_2) - s\theta_1(l_3(c\theta_3s\theta_2 - s\theta_3c\theta_2) + l_2s\theta_2) + l_1c\theta_1 \\ s\theta_1(l_3(c\theta_3c\theta_2 - s\theta_3s\theta_2) + l_2c\theta_2) + c\theta_1(l_3(c\theta_3s\theta_2 - s\theta_3c\theta_2) + l_2s\theta_2) + l_1s\theta_1 \\ d \end{bmatrix} \quad (3.31)^{11}$$

¹¹ Las letras c y s representan la notación compacta usada para representar a las funciones trigonométricas coseno y seno cuando se busca ahorrar espacio.

La velocidad lineal del efector se haya simplemente obteniendo la derivada de la velocidades articulares $\dot{d}, \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3$.

$$l_1 \times \cos(\theta_1) + l_2 \times \cos(\theta_2 + \theta_1) + l_3 \times \cos(\theta_3 + \theta_2 + \theta_1) = x \quad (3.32)$$

$$l_1 \times \sin(\theta_1) + l_2 \times \sin(\theta_2 + \theta_1) + l_3 \times \sin(\theta_3 + \theta_2 + \theta_1) = y \quad (3.33)$$

$$d = z \quad (3.34)$$

Al derivar las ecuaciones anteriores se obtiene:

$$\begin{aligned} \dot{x} = & -\dot{\theta}_1(l_1 \sin(\theta_1) + l_2 \sin(\theta_2 + \theta_1) + l_3 \sin(\theta_3 + \theta_2 + \theta_1)) \\ & -\dot{\theta}_2(l_2 \sin(\theta_2 + \theta_1) + l_3 \sin(\theta_3 + \theta_2 + \theta_1)) - \dot{\theta}_3(l_3 \sin(\theta_3 + \theta_2 + \theta_1)) \end{aligned} \quad (3.35)$$

$$\begin{aligned} \dot{y} = & \dot{\theta}_1(l_1 \cos(\theta_1) + l_2 \cos(\theta_2 + \theta_1) + l_3 \cos(\theta_3 + \theta_2 + \theta_1)) \\ & + \dot{\theta}_2(l_2 \cos(\theta_2 + \theta_1) + l_3 \cos(\theta_3 + \theta_2 + \theta_1)) + \dot{\theta}_3(l_3 \cos(\theta_3 + \theta_2 + \theta_1)) \end{aligned} \quad (3.36)$$

$$\dot{z} = \dot{d} \quad (3.37)$$

Para las velocidades angulares también se puede simplificar la expresión obtenida usando trigonometría, resultando:

$$R = \begin{bmatrix} \cos(\theta_1 + \theta_2 + \theta_3) & -\sin(\theta_1 + \theta_2 + \theta_3) & 0 \\ \sin(\theta_1 + \theta_2 + \theta_3) & \cos(\theta_1 + \theta_2 + \theta_3) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.38)$$

Esta matriz de rotación es ortonormal y se usa para definir una matriz antisimétrica

$\Omega = \dot{R} \cdot R^T$. Desarrollando \dot{R} y R^T se obtiene:

$$\dot{R} = (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) \begin{bmatrix} -\sin(\theta_1 + \theta_2 + \theta_3) & -\cos(\theta_1 + \theta_2 + \theta_3) & 0 \\ \cos(\theta_1 + \theta_2 + \theta_3) & -\sin(\theta_1 + \theta_2 + \theta_3) & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$R^T = \begin{bmatrix} \cos(\theta_1 + \theta_2 + \theta_3) & \text{sen}(\theta_1 + \theta_2 + \theta_3) & 0 \\ -\text{sen}(\theta_1 + \theta_2 + \theta_3) & \cos(\theta_1 + \theta_2 + \theta_3) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Reemplazando se obtiene la ecuación 3.39,

$$\Omega = (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (3.39)$$

De la expresión (3.34) se pueden obtener las velocidades angulares en función de las derivadas de las variables articulares, resultando:

$$\omega_x = 0 \quad (3.40)$$

$$\omega_y = 0 \quad (3.41)$$

$$\omega_z = \dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 \quad (3.42)$$

Luego, se construye una matriz de transformación que relaciona las velocidades articulares con las velocidades lineales y angulares del efector final. Como además esta matriz es invertible sirve también para hallar la cinemática inversa.

$$\begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} -1_1s(\theta_1) - 1_2s(\theta_2 + \theta_1) - 1_3s(\theta_3 + \theta_2 + \theta_1) & -1_2s(\theta_2 + \theta_1) - 1_3s(\theta_3 + \theta_2 + \theta_1) & -1_3s(\theta_3 + \theta_2 + \theta_1) & 0 \\ 1_1c(\theta_1) + 1_2c(\theta_2 + \theta_1) + 1_3c(\theta_3 + \theta_2 + \theta_1) & 1_2c(\theta_2 + \theta_1) + 1_3c(\theta_3 + \theta_2 + \theta_1) & 1_3c(\theta_3 + \theta_2 + \theta_1) & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ d \end{bmatrix} \quad (3.43)$$

3.2 Análisis de la dinámica

La dinámica se ocupa de la relación entre fuerzas que actúan en un cuerpo y el movimiento que en él se origina. Por lo tanto en el denominado *modelo dinámico* se establece la relación matemática entre:

- i. La ubicación del robot definida por sus variables articulares o por las coordenadas de localización de su extremo, y sus derivadas: velocidad y aceleración.
- ii. Las fuerzas y pares aplicados en las articulaciones y en el extremo del robot.
- iii. Los parámetros dimensionales del robot, como longitud, masas e inercias de sus elementos.

La obtención de este modelo para mecanismos de uno o dos grados de libertad no es compleja, pero se complica para sistemas con más de dos grados de libertad como es el caso de estudio. En estos casos el modelo dinámico debe ser resuelto entonces de manera iterativa mediante procedimientos numéricos.

El problema de la obtención del modelo dinámico de un robot es uno de los aspectos más importantes de la robótica porque permite conseguir los siguientes fines:

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

Es importante mencionar que el modelo dinámico completo de un robot debe incluir no sólo la dinámica de sus eslabones sino también la propia de sus sistemas de transmisión, de los actuadores y sus equipos electrónicos de mando. Estos elementos incorporan al sistema nuevas cargas que se verán reflejadas en pesos, inercias, rozamientos aumentando la complejidad del sistema.

La resolución de la dinámica de robots trata acerca de dos modelos principales:

- i. Modelo dinámico directo: Expresa la evolución temporal de las coordenadas articulares del robot en función de las fuerzas y pares que intervienen.
- ii. Modelo dinámico inverso: Determina las fuerzas y pares necesarios para conseguir una evolución deseada en las coordenadas articulares.

3.2.1 Obtención del modelo dinámico mediante la formulación de Lagrange

Las Ecuaciones de Lagrange (también conocidas como Ecuaciones de Euler-Lagrange) permiten contar con un sistema analítico para llegar a las ecuaciones que describen el comportamiento físico de las partículas, pero no se trata, de ningún modo, de una nueva teoría independiente de la teoría Newtoniana.

La ventaja de usar los principios de la mecánica de Lagrange, en realidad de todos los métodos de Mecánica Analítica, es que se tendrán tantas ecuaciones como coordenadas generalizadas, mientras que con las ecuaciones de Newton se obtendrían 6 ecuaciones por cada cuerpo. Además, se tiene un sistema de ecuaciones escalares que es más simple que un sistema de ecuaciones vectoriales para trabajar computacionalmente.

La notación de Lagrange hace uso de un término, al cual se define como lagrangiano, que es la diferencia entre la co-energía cinética del sistema y la energía potencial del mismo expresadas en términos de las coordenadas generalizadas del sistema, que son un conjunto de variables que permiten describir el estado del sistema en todo momento. En los casos de sistemas holonómicos, sistemas que como el estudiado tienen restricciones que pueden ser descritas cada una por una ecuación de superficie, se tienen tantos grados de libertad como coordenadas generalizadas (q), que en este caso son los ángulos θ_i y el desplazamiento horizontal de la base del primer eslabón d .

Para el caso de los sistemas holonómicos las ecuaciones de Lagrange se pueden expresar de la siguiente manera:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_k} \right) - \frac{\partial L}{\partial q_k} = Q_k \quad (3.44)$$

En donde Q_k representa a la fuerza no conservativa expresada para cada coordenada generalizada q_k para k de 1 a n , donde n es el número de coordenadas generalizadas usadas, t representa al tiempo y L al lagrangiano del sistema.

En 1965 Uicker utilizó la representación de Denavit Hartenberg, basada en las matrices de transformación homogénea, para formular el modelo dinámico de un robot mediante la ecuación de Lagrange.

En este caso se aplicará la formulación lagrangiana dada por Uicker para hallar las ecuaciones diferenciales de la dinámica del sistema:

- a) Para iniciar se debe asignar a cada eslabón un sistema de referencia de acuerdo a Denavit Hartenberg, lo cual se ha obtenido en el estudio de la Cinemática.
- b) Obtener las matrices de transformación homogénea para cada elemento.
- c) Para calcular la co-energía cinética primero se deben obtener las velocidades expresadas como función de las matrices U_{ij} como se muestra en la ecuación (3.45). La velocidad 0v_i se puede calcular como la derivada con respecto al tiempo de la matriz de transformación 0A_i que define la posición. A continuación se define esta derivada en función de las coordenadas generalizadas q_j .

$${}^0v_i = \frac{d}{dt} ({}^0A_i) = \sum_{j=1}^i \frac{\partial {}^0A_i}{\partial q_j} \dot{q}_j$$

Luego se definen las matrices U_{ij} como:

$$U_{ij} = \frac{\partial {}^0A_i}{\partial q_j} \quad (3.45)$$

d) Luego se obtienen las matrices U_{ijk} definidas por:

$$U_{ijk} = \frac{\partial U_{ij}}{\partial q_k} \quad (3.46)$$

e) Obtener las matrices de pseudo inercia J_i para cada elemento, que vienen definidas por:

$$J_i = \begin{bmatrix} \int_i x_i^2 dm & \int_i x_i y_i dm & \int_i x_i z_i dm & \int_i x_i dm \\ \int_i y_i x_i dm & \int_i y_i^2 dm & \int_i y_i z_i dm & \int_i y_i dm \\ \int_i z_i x_i dm & \int_i z_i y_i dm & \int_i z_i^2 dm & \int_i z_i dm \\ \int_i x_i dm & \int_i y_i dm & \int_i z_i dm & \int_i dm \end{bmatrix}$$

$$J_i = \begin{bmatrix} \frac{-I_{xx} + I_{yy} + I_{zz}}{2} & I_{xy} & I_{xz} & m_i \bar{x}_i \\ I_{xy} & \frac{I_{xx} - I_{yy} + I_{zz}}{2} & I_{yz} & m_i \bar{y}_i \\ I_{xz} & I_{yz} & \frac{I_{xx} + I_{yy} - I_{zz}}{2} & m_i \bar{z}_i \\ m_i \bar{x}_i & m_i \bar{y}_i & m_i \bar{z}_i & m_i \end{bmatrix} \quad (3.47)$$

Estas matrices de pseudo inercia deben describir las propiedades de los eslabones incluyendo actuadores y contrapesos debido a que en la rutina elaborada no se han considerado de manera separada los pesos e inercias de estos elementos.

f) Después de obtener la velocidad de cada articulación se necesita una expresión para la co-energía cinética del elemento i . La co-energía cinética de una partícula con masa diferencial dm se puede expresar como:

$$dK_i = \frac{1}{2} \left(\dot{x}_i^2 + \dot{y}_i^2 + \dot{z}_i^2 \right) dm = \frac{1}{2} Tr \left(v_i v_i^T \right) dm$$

$$dK_i = \frac{1}{2} Tr \left[\sum_{j=1}^i U_{ij} \dot{q}_j \left(\sum_{k=1}^i U_{ik} \dot{q}_k \right)^T \right] dm$$

Luego, se expresa la energía cinética para un cuerpo i .

$$K_i = \frac{1}{2} Tr \left[\sum_{j=1}^i \sum_{k=1}^i U_{ij} J_i U_{ik}^T \dot{q}_j \dot{q}_k \right]$$

Con lo que se define la matriz inercial de co-energía cinética como $D = [d_{ij}]$, donde:

$$d_{ij} = \sum_{k=\max(i,j)}^n Tr (U_{kj} J_k U_{ki}^T) \quad (3.48)$$

g) Se deben obtener los términos h_{ikm} definidos por:

$$h_{ikm} = \sum_{j=\max(i,k,m)}^n Tr (U_{jkm} J_j U_{ji}^T) \quad (3.49)$$

h) Sabiendo que la energía potencial del manipulador se puede definir como:

$$U = \sum_{i=1}^n -m_i g^0 A_i$$

El lagrangiano del sistema se expresaría como:

$$L = K - U = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^i \sum_{k=1}^i \left[Tr (U_{ij} J_i U_{ik}^T) \dot{q}_j \dot{q}_k \right] + \sum_{i=1}^n m_i g^0 A_i$$

$$Q_i = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i}$$

$$Q_i = \sum_{j=ik=1}^n \sum_{j=1}^j Tr (U_{jk} J_j U_{ji}^T) \ddot{q}_k + \sum_{j=ik=1}^n \sum_{j=1}^j \sum_{m=1}^j Tr (U_{jkm} J_j U_{ji}^T) \dot{q}_k \dot{q}_m - \sum_{j=i}^n m_j g U_{ji}$$

Usando la notación matricial se obtiene:

$$Q_i = \sum_{k=1}^n D_{ik} \ddot{q}_k + \sum_{k=1}^n \sum_{m=1}^n h_{ikm} \dot{q}_k \dot{q}_m + c_i \quad (3.50)$$

Simplificando aún más en forma matricial resulta:

$$Q(t) = D(q(t))\ddot{q}(t) + h\left(q(t), \dot{q}(t)\right) + c(q(t)) \quad (3.51)$$

$$h_i = \sum_{k=1}^n \sum_{m=1}^n h_{ikm} \dot{q}_k \dot{q}_m \quad (3.52)$$

$$c_i = \sum_{j=1}^n (-m_j g U_{ji}^j r_j) \quad (3.53)$$

Donde:

$Q(t)$: vector $n \times 1$ del par generalizado aplicado a las articulaciones

$q(t)$: vector $n \times 1$ de las coordenadas generalizadas

$\dot{q}(t)$: vector $n \times 1$ de la velocidad de las coordenadas generalizadas

$\ddot{q}(t)$: vector $n \times 1$ de la aceleración de las coordenadas generalizadas

$D(q)$: matriz simétrica inercial relacionada con la aceleración $n \times n$

$h\left(\dot{q}, q\right)$: vector de fuerzas de Coriolis y centrífuga no lineal $n \times 1$

$c(q)$: vector de las fuerzas de gravedad $n \times 1$

La implementación de este método se encuentra expresada en el código MATLAB del Anexo 1. El objeto del código desarrollado es el hallar las fuerzas generalizadas que son necesarias para que el sistema se mueva según una trayectoria definida para sus coordenadas generalizadas. En este caso el comportamiento deseado de las coordenadas generalizadas es expresado explícitamente en función del tiempo.

Para poder representar el valor de estas coordenadas en función del tiempo se define inicialmente la velocidad deseada en el efector. Por ejemplo, si se desea que el efector tenga una velocidad de 20mm/s en sentido paralelo al eje de traslación y una velocidad de 30mm/s en sentido perpendicular al mismo considerando que se empieza desde la esquina del plano de trabajo del robot, entonces la posición del efector final se puede definir como:

$$d = 20 \times t \quad (3.54)$$

$$y = 100 + 30 \times t \quad (3.55)$$

Es válido mencionar que las velocidades que se han elegido están dentro del promedio de lo usado en los procesos de corte de plancha metálica según lo expresado en las exigencias del Capítulo 2.

Luego, se reemplazan estas expresiones en las ecuaciones (3.24) a (3.27) y se obtienen expresiones explícitas de las coordenadas generalizadas en función del tiempo. Con estas expresiones se pueden calcular fácilmente las velocidades y aceleraciones al ser simples derivadas con respecto al tiempo. Conociendo esta información se necesitaría sólo conocer las matrices de inercia para cada eslabón para estar listos a simular la dinámica inversa.

Las matrices de inercia son propiedades físicas que dependen del diseño mecánico del robot, el cual como se mencionó se hará posteriormente. Entonces por el momento se terminará el modelado dinámico ingresando matrices de inercia con valores aproximados que no representan con exactitud las verdaderas propiedades físicas que el robot tendrá, pero sirven para tener una idea del comportamiento dinámico del mismo. Además, una vez terminado el diseño mecánico bastará ingresar las nuevas matrices de inercia para tener un modelo dinámico completo que verdaderamente describa la dinámica del sistema estudiado.

CAPÍTULO 4

SIMULACIÓN Y RESULTADOS

La simulación es el proceso de diseñar un modelo de un sistema real y llevar a término experiencias con él, con la finalidad de comprender el comportamiento del sistema o evaluar nuevas estrategias -dentro de los límites impuestos por un cierto criterio o un conjunto de ellos- para el funcionamiento del sistema. El objetivo de la simulación es la mejor comprensión del movimiento del mecanismo y para la predicción del movimiento del sistema sin necesidad de usar un prototipo físico.

Para poder lograr este objetivo se hizo uso de un Toolbox de Robótica implementado en MATLAB [CORKE, 1996] que fue desarrollado con el fin de trabajar fácilmente con la notación usada en el algoritmo dado por Denavit Hartenberg. En este caso se analizará la respuesta del mecanismo seleccionado ante velocidades articulares definidas, para la simulación cinemática, y ante fuerzas y momentos aplicados en la simulación dinámica.

4.1 Simulación cinemática

Mediante la simulación de la cinemática se busca conocer la relación existente entre las coordenadas articulares y la ubicación del robot mediante resultados numéricos y

gráficos, para lo cual se hará uso de MATLAB y de un *Toolbox* [CORKE, 1996] desarrollado para estos propósitos.

Una vez en el entorno de MATLAB se usa el comando *link* para definir los eslabones que definen nuestro robot usando la notación de Denavit Hartenberg. Los eslabones se definen según $link([alpha \ a \ theta \ d \ sigma], CONVENTION)^{12}$ donde sigma es 0 si se trata de una unión rotacional y es diferente de cero si se trata de una prismática. En el ejemplo mostrado en el Cuadro 4.1 se ha tomado el valor de 10. Además, se asignan valores arbitrarios también para los desplazamientos articulares d y theta.

```

>> L1=link([0 0 0 0.700 10], 'standard')
L1 =
    0.000000    0.000000    0.000000    0.700000    P    (std)

>> L2=link([0 0.500 0 0 0], 'standard')
L2 =
    0.000000    0.500000    0.000000    0.000000    R    (std)

>> L3=link([0 0.500 0 0 0], 'standard')
L3 =
    0.000000    0.500000    0.000000    0.000000    R    (std)

>> L4=link([0 0.100 0 0 0], 'standard')
L4 =
    0.000000    0.100000    0.000000    0.000000    R    (std)

>> Robot_1=robot({L1, L2, L3, L4}, 'Robot')
Robot_1 =
Robot (4 axis, PRRR)
    grav = [0.00 0.00 9.81]          standard D&H parameters

alpha    A    theta    D    R/P
0.000000 0.000000 0.000000 0.700000 P (std)
0.000000 0.500000 0.000000 0.000000 R (std)
0.000000 0.500000 0.000000 0.000000 R (std)
0.000000 0.100000 0.000000 0.000000 R (std)

>> drivebot(Robot_1)
  
```

Cuadro 4.1. Código para simulación de ubicación

En la parte final del Cuadro 4.1 se muestra la instrucción *drivebot* que se encarga de imprimir resultados gráficos de la ubicación del robot permitiendo además modificar

¹² Alpha, a, theta y d son los parámetros del método de Denavit-Hartenberg. Convention se refiere a la convención usada para asignar los parámetros de Denavit – Hartenberg: si es de acuerdo al algoritmo estándar o según la notación modificada por Craig.

los valores de las variables articulares y ver su influencia en la posición y orientación del robot como se muestra en la Figura 4.1.

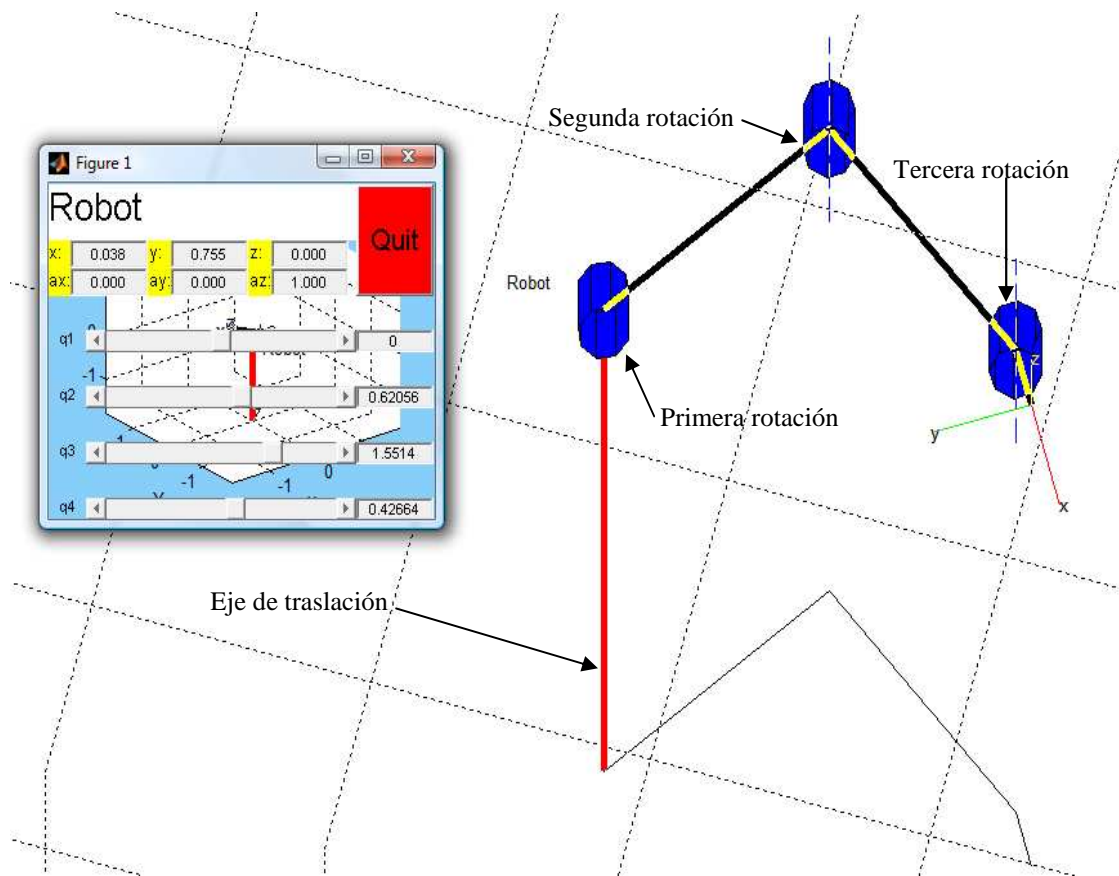


Figura 4.1. Simulación de ubicación variando las coordenadas articulares

Con la simulación anterior se ha obtenido una interesante respuesta gráfica, pero si lo que se desea es conocer un dato numérico de la cinemática directa o inversa del mecanismo entonces se cuenta con los comandos *fkine* e *ikine*.

Con el comando *fkine* (*forward kinematics*) se obtiene la matriz de transformación del efector final con respecto a un inicio de coordenadas que se fija en un punto determinado. En el Cuadro 4.2 se muestra el uso de este comando teniendo como datos las coordenadas articulares usadas en el ejemplo de cálculo del Capítulo 3.

De los resultados obtenidos en el Cuadro 4.2 se puede notar que la matriz de transformación es aproximadamente la misma obtenida en la ecuación (3.19). El motivo del error yace en la poca aproximación que se ha usado para ingresar los datos de las coordenadas articulares.

```

>> L1=link([0 0 0 300 10], 'standard')
L1 =
    0.000000    0.000000    0.000000    300.000000    P
    (std)

>> L2=link([0 500 1.2945 0 0], 'standard')
L2 =
    0.000000    500.000000    1.294500    0.000000    R
    (std)

>> L3=link([0 500 0.3491 0 0], 'standard')
L3 =
    0.000000    500.000000    0.349100    0.000000    R
    (std)

>> L4=link([0 100 1.4980 0 0], 'standard')
L4 =
    0.000000    100.000000    1.498000    0.000000    R
    (std)

>> drivebot(Robot_1)

>> fkine(Robot_1, [300 1.2945 0.349100 1.498000])
ans =
    -1.0000    0.0000     0         0.0275
    -0.0000   -1.0000     0         979.7109
     0         0         1.0000    300.0000
     0         0         0         1.0000
  
```

Cuadro 4.2. Códigos para cinemática directa

Luego, haciendo uso del comando *ikine* (*inverse kinematics*) se puede hacer el cálculo cinemático inverso. En este caso se tiene el inconveniente de que como se usan métodos iterativos existen ocasiones en que el método demora en hallar la respuesta y otros en los cuales ni siquiera converge. La convergencia o no convergencia del método depende únicamente del punto inicial elegido para las iteraciones.

En la rutina MATLAB presentada en el Cuadro 4.3 se presenta el uso del comando *ikine*, para el cual debe de indicarse el nombre con el cual se definió a los eslabones del robot, la matriz de transformación para el efector final deseada y las coordenadas desde las cuales se iniciará el proceso iterativo.

De los resultados mostrados en el Cuadro 4.3 se observa que en el primer caso no convergía mientras que en el segundo, teniendo un punto inicial más cercano a la respuesta final, converge sin problemas.

```

>> ikine(Robot_1, T, [0 0 0 0])
For a manipulator with fewer than 6DOF a mask matrix argument should be specified
??? Error using ==> ikine
Solution wouldn't converge

>> ikine(Robot_1, T, [200 1 0 1])
For a manipulator with fewer than 6DOF a mask matrix argument should be specified
ans =
    300.0000    1.2945    0.3491    1.4980
  
```

Cuadro 4.3. Códigos para cinemática inversa

Además, se conoce que el sistema es un sistema no lineal debido a las funciones trigonométricas existentes entre las coordenadas articulares y la ubicación del robot. Estas relaciones además tienen complejidad debido a la presencia de ángulos múltiples. Siendo así, es interesante observar cómo es que las coordenadas articulares varían unas con respecto de otras, para este caso se ha tomado el escenario en el cual la primera coordenada articular (θ_1) varía con velocidad constante.

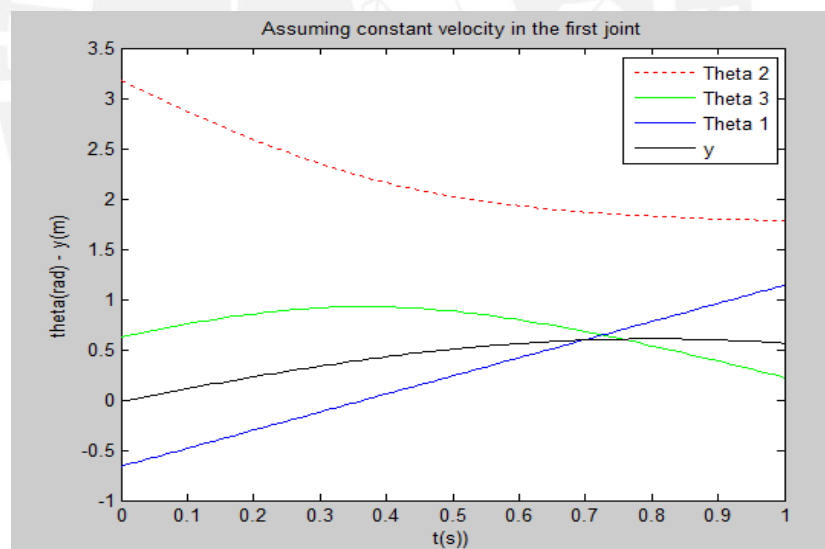


Figura 4.2. Relación entre las coordenadas para un caso dado

Para la simulación presentada en la Figura 4.2 se ha asumido que el efector final del mecanismo empieza en su extremo izquierdo y luego se va desplazando a la derecha como se indica en la rutina presentada en el Cuadro 4.4.

```

% Forward kinematic simulation working on a horizontal plane
% according to a desired behavior

close all;
clear all;
clc;

% Initial parameters assumed as in the initial given
% position

l1=0.500; % in m
l2=0.500; % in m
l3=0.100; % in m
y_0 = 0; % in m
theta1_0=-0.662059566615; % in radians
theta2_0=2.86233997327; % in radians
theta3_0=0.941312246933; % in radians

% Give a value to the velocity of the very first motor

d_theta_1=1.8; % in rad/s
time=0; % in sec

% Define the time vector in sample interval of 50ms

for k= 1:1:101
    theta1(k)=theta1_0+(d_theta_1*0.01*(k-1));
    alpha(k)=asin((l1*cos(theta1(k))-l3)/l2);
    theta2(k)=3.1415926-theta1(k)-alpha(k);
    theta3(k)=3.1415926-theta1(k)-theta2(k);

y(k)=l1*sin(theta1(k))+l2*sin(theta1(k)+theta2(k))+l3*sin(theta1(k)+
theta2(k)+theta3(k));
    end

vtime=[0:0.01:1];
plot(vtime,theta2,'r'); hold on;
xlabel('t(s)');title('Assuming constant velocity in the first
joint');
ylabel('theta(rad) - y(m)');

plot(vtime,theta3,'g'); hold on;
plot(vtime,theta1,'b'); hold on;
plot(vtime,y,'k'); hold off;legend('Theta 2','Theta 3', 'Theta 1',
'y');

```

Cuadro 4.4. Rutina para evaluar coordenadas generalizadas

4.2 Simulación dinámica

El objetivo de esta simulación es obtener los valores de fuerzas y momentos requeridos en los actuadores con la finalidad de conseguir un comportamiento deseado en el movimiento del robot. Para modelar el comportamiento de este sistema mecánico se usó en el Capítulo 3.2 la formulación de Uicker basada en mecánica Lagrangiana.

La simulación realizada sigue la secuencia mostrada en el Diagrama de Flujo que se presenta en la Figura 4.3.

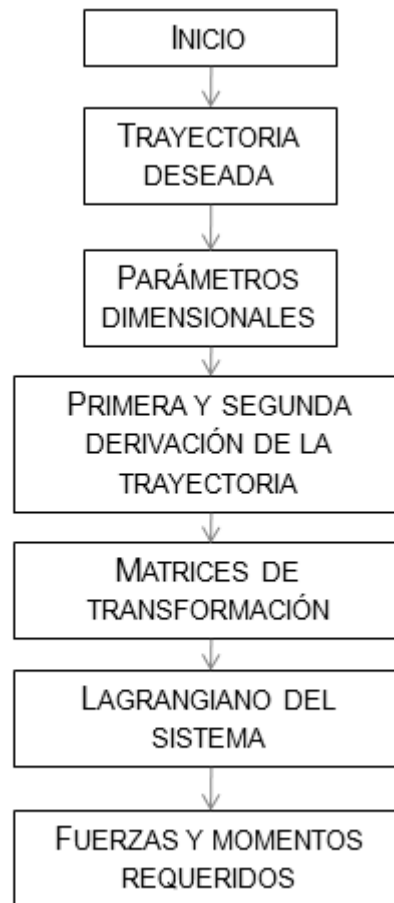


Figura 4.3. Diagrama de flujo de la simulación dinámica

4.2.1 Simulación de movimiento paralelo al lado mayor

La primera simulación a realizar será la del movimiento más simple, en el cual el robot se mueve sólo de manera paralela al lado mayor de la superficie de trabajo como se muestra en la Figura 4.4. Con este movimiento el brazo no se cierra ni se abre porque sólo se desplaza.

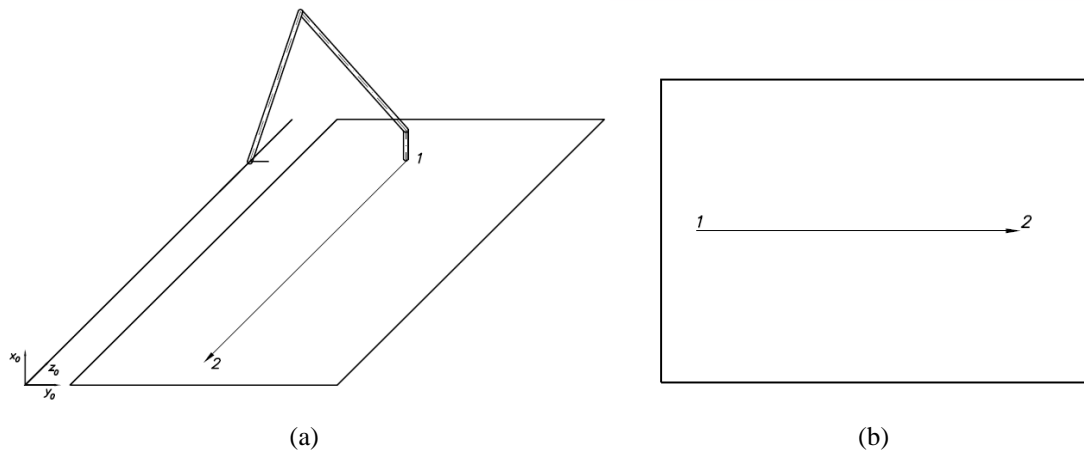


Figura 4.4. Trayectoria deseada paralela al lado mayor (a)Vista 3D, (b) Vista 2D

En esta primera simulación sólo será necesario observar la variación de la primera coordenada generalizada y de la primera fuerza generalizada, debido a que las demás coordenadas generalizadas no variarán, y el torque requerido en los demás actuadores será de cero.

Consideraremos el caso en que los valores de las coordenadas generalizadas sean de:

$$d = 1000 + 60 \times t - 10 \times t^2$$

$$\theta_1 = 74.17^\circ = 1.294560 \text{ rad}$$

$$\theta_2 = 20^\circ = 0.349038 \text{ rad}$$

$$\theta_3 = 85.53^\circ = 1.492780 \text{ rad}$$

Se ha elegido esta trayectoria porque su movimiento tiene un valor de aceleración constante diferente de cero en la primera coordenada generalizada, por lo cual esperaríamos ver una fuerza constante requerida para obtener este movimiento.

Se ha elaborado una rutina en MATLAB que se presenta en la sección 4.3 denominada Rutina 4.1. Los resultados se muestran en las Figuras 4.5 y 4.6.

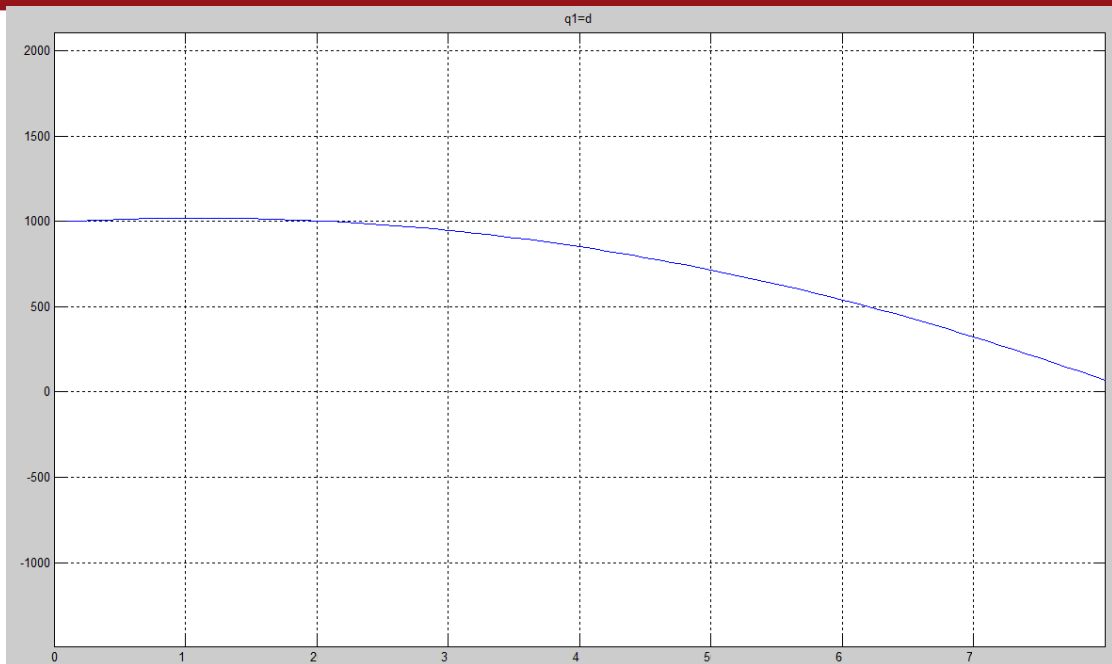


Figura 4.5. Primera coordenada generalizada (d) [mm] en función al tiempo [s]

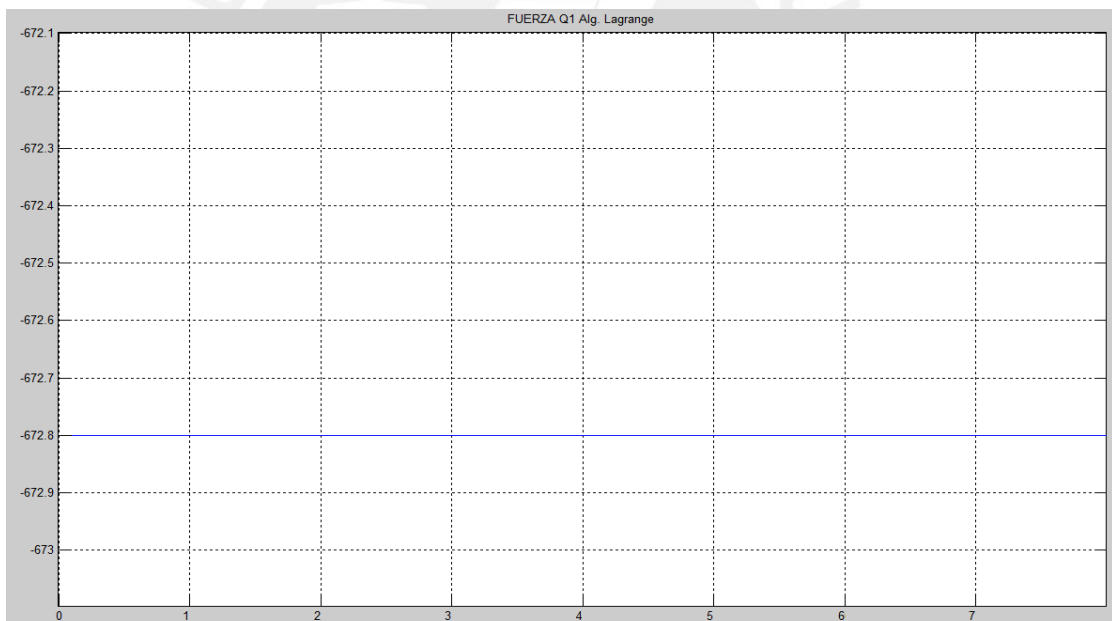


Figura 4.6. Fuerza generalizada (F) [kg × mm/s²] del primer efector en función al tiempo [s]

Los resultados obtenidos en la simulación están dentro de lo esperado ya que se puede apreciar que para la trayectoria requerida se necesita aplicar una fuerza constante. Por lo tanto, y habiendo realizado esta primera simulación, se pasará a analizar casos algo más complejos.

4.2.2 Simulación de movimiento diagonal con el peso a favor del movimiento

En la sección 4.3 aparece la Rutina 4.2 que muestra el cambio que se hace en los Datos de Entrada de la Rutina 4.1 para cuando el robot empieza a moverse desde el inicio del plano de trabajo según la ley de movimiento descrita en el Capítulo 3.2 que es un movimiento a velocidad constante en forma diagonal del efector final, teniendo al efecto del peso a favor del movimiento deseado. Dicha trayectoria se muestra en la Figura 4.7.

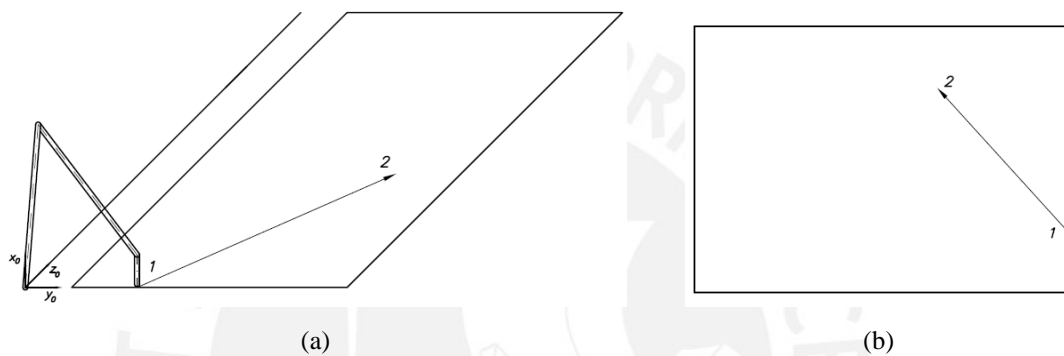


Figura 4.7. Trayectoria deseada con el peso a favor del movimiento (a) Vista 3D, (b) Vista 2D

Los datos que se obtienen del programa implementado son los valores de las coordenadas generalizadas deseadas en función del tiempo y los torques requeridos en los actuadores a fin de obtener el comportamiento deseado. Dichos resultados se presentan en las Figuras 4.8 a 4.15.

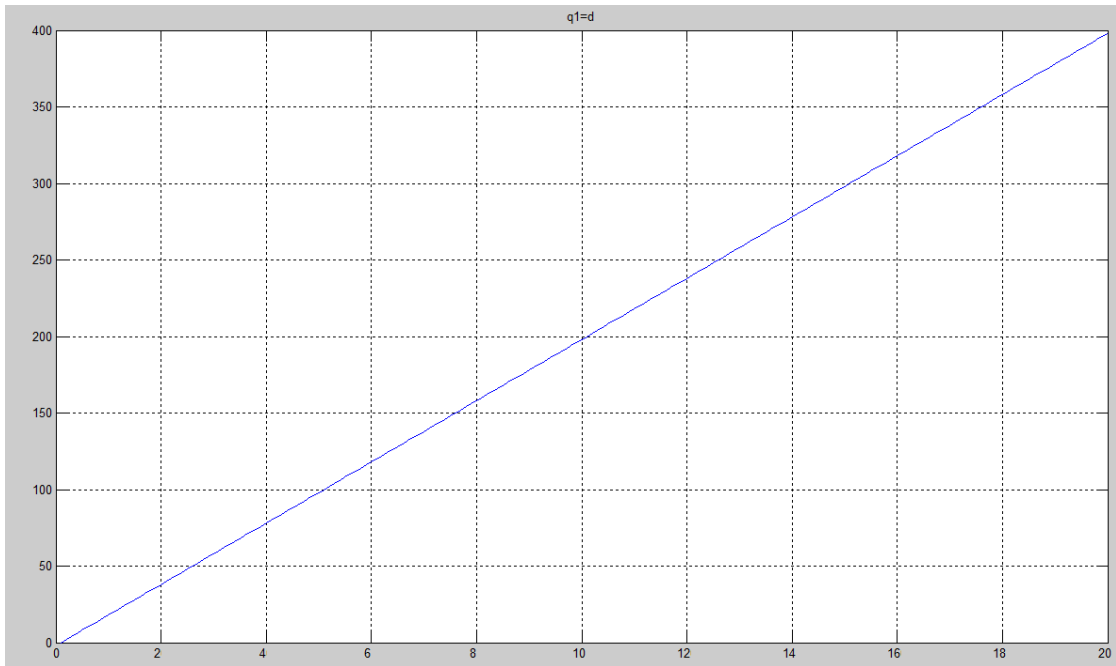


Figura 4.8. Primera coordenada generalizada (d) [mm] en función al tiempo [s]

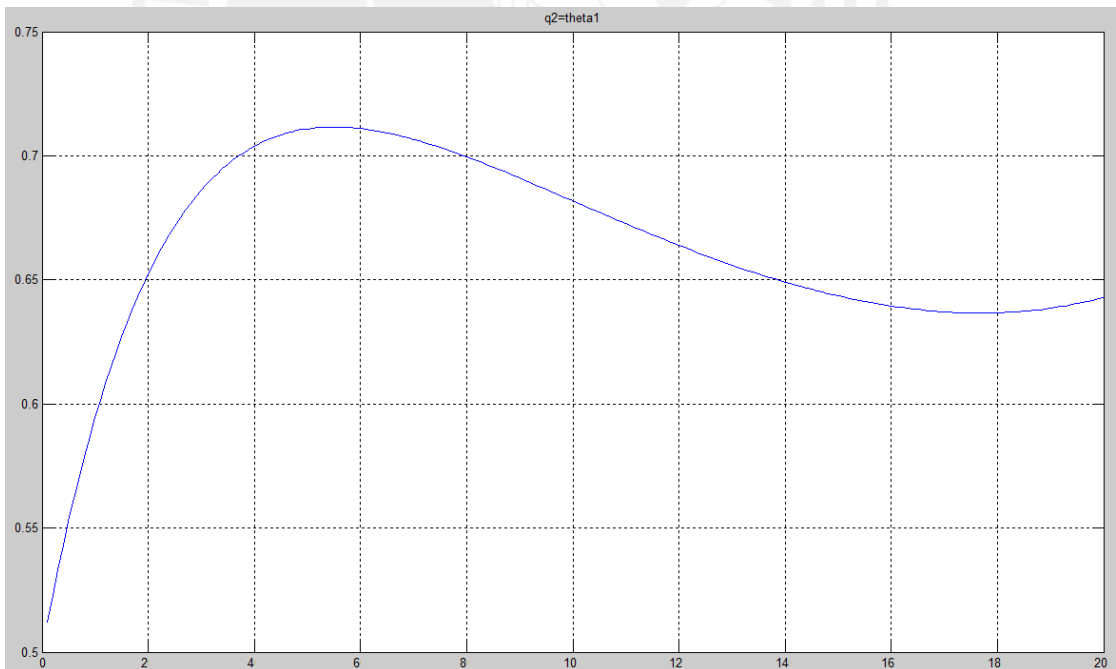


Figura 4.9. Segunda coordenada generalizada (θ_1) [rad] en función al tiempo [s]

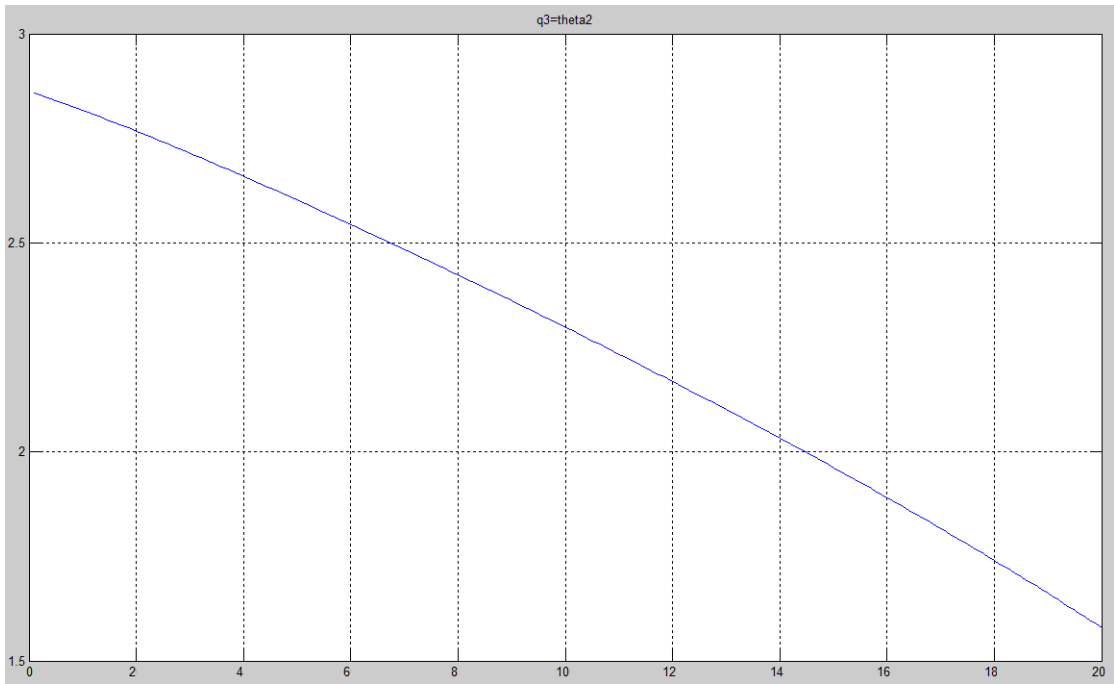


Figura 4.10. Tercera coordenada generalizada (θ_2) [rad] en función al tiempo [s]

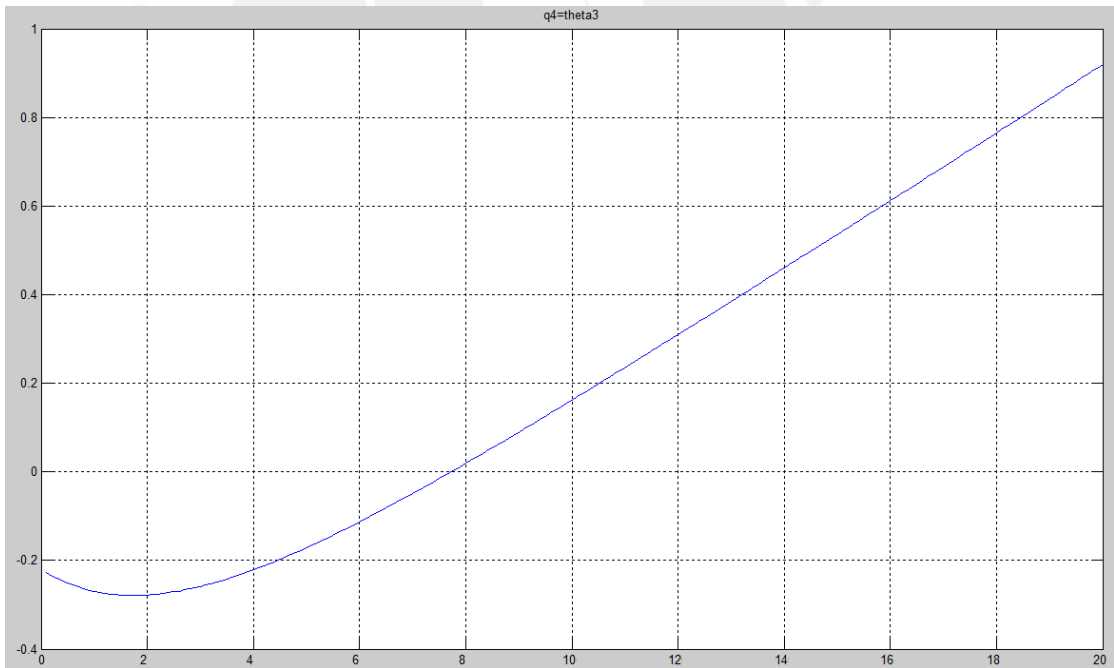


Figura 4.11. Cuarta coordenada generalizada (θ_3) [rad] en función al tiempo [s]

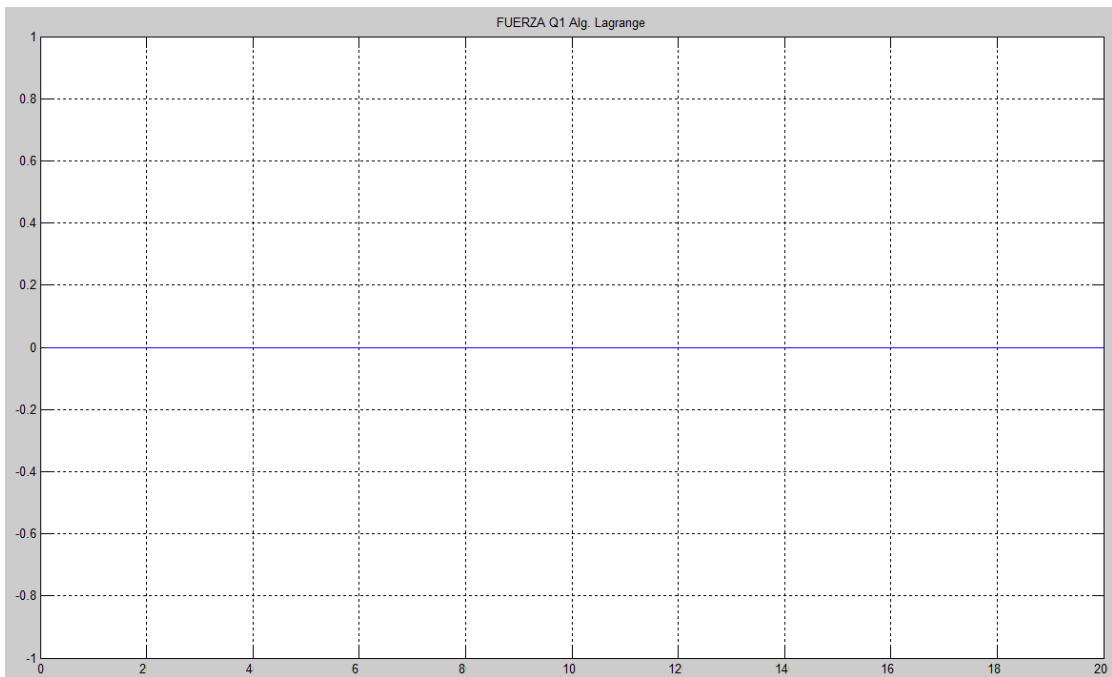


Figura 4.12. Fuerza generalizada (F) [kg × mm/s²] del primer efector en función al tiempo [s]

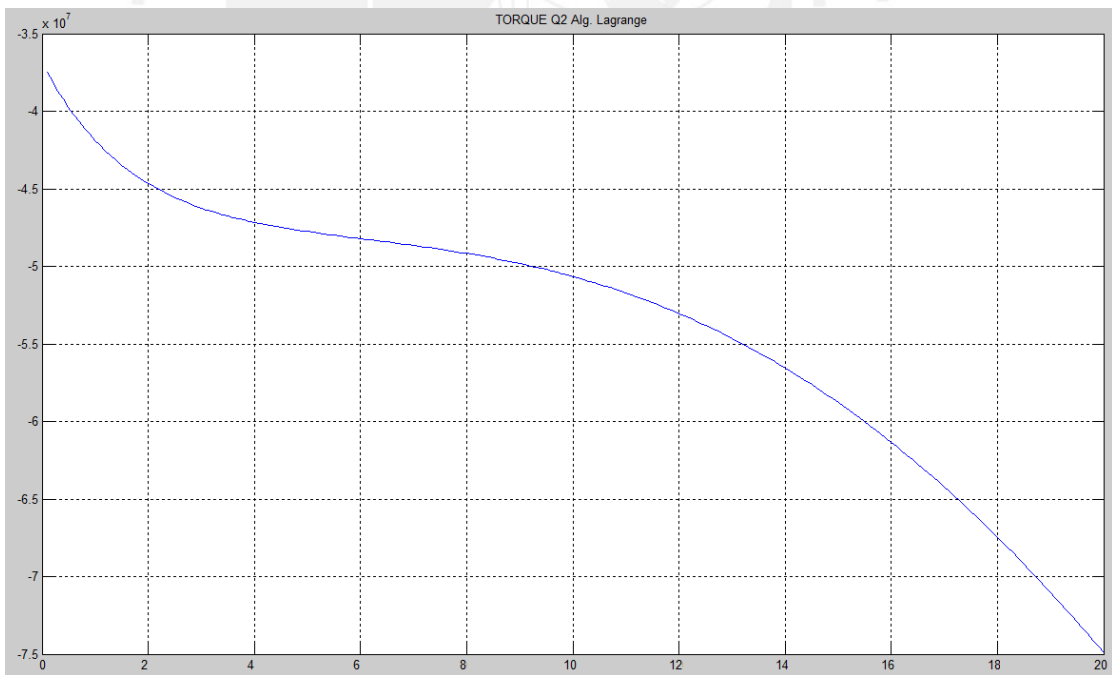


Figura 4.13. Torque generalizado (T₁) [kg × mm²/s²] del segundo efector en función al tiempo [s]

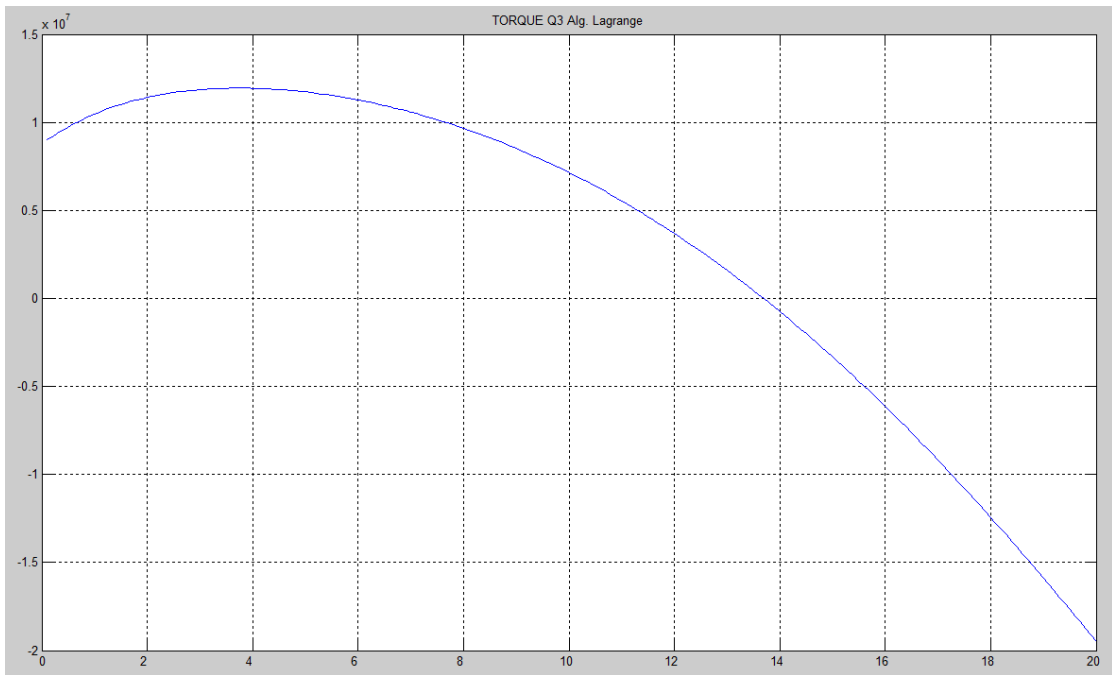


Figura 4.14. Torque generalizado (T_2) [$\text{kg} \times \text{mm}^2/\text{s}^2$] del tercer efector en función al tiempo [s]

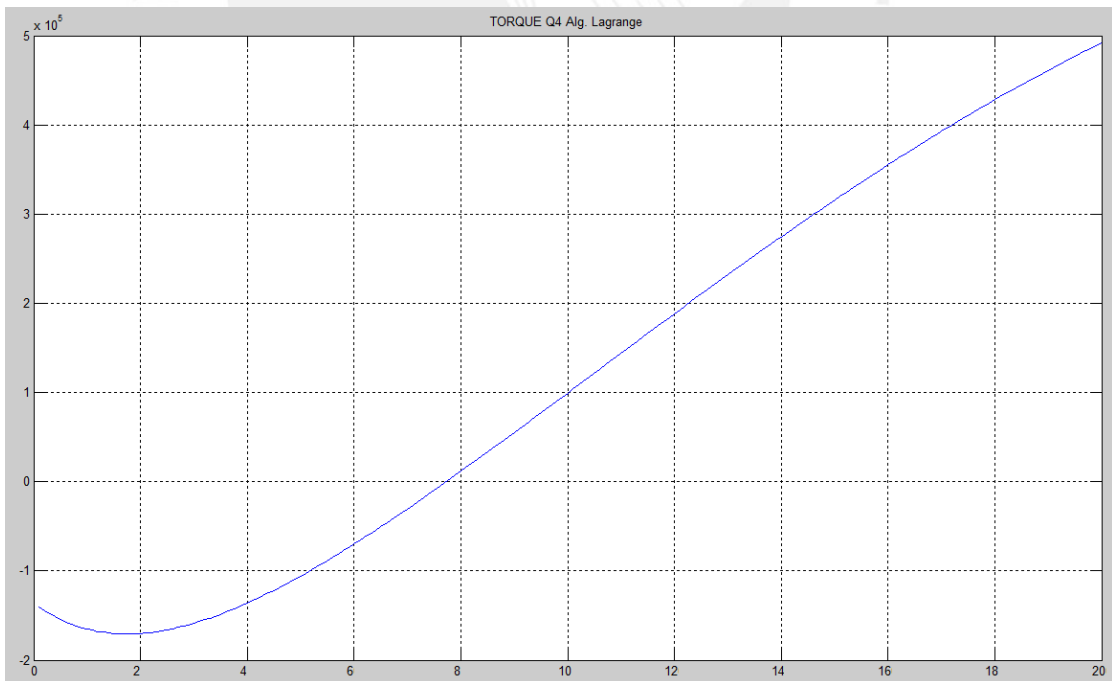


Figura 4.15. Torque generalizado (T_3) [$\text{kg} \times \text{mm}^2/\text{s}^2$] del cuarto efector en función al tiempo [s]

Los resultados de esta simulación muestran un comportamiento que se podía esperar porque al tener una velocidad lineal constante en el sentido paralelo al lado mayor de la superficie de trabajo indica que no es necesario aplicar fuerza sobre la base del mecanismo, lo cual era un resultado esperado.

Además, el torque requerido en los motores eléctricos disminuye en valor al extenderse más el brazo, lo cual tiene sentido porque cuando se abre el brazo aumenta el momento ocasionado por los pesos de los eslabones, lo que favorece al movimiento. También está el hecho que cuando se abre el brazo se requiere un menor giro en el motor para desplazar el efector final a la misma velocidad.

4.2.3 Simulación de movimiento diagonal con el peso en contra del movimiento

En este caso se analiza el movimiento para una trayectoria opuesta a la del caso anterior como se muestra en la Figura 4.16, en la cual el peso ya no favorezca al movimiento sino que esta vez se oponga al mismo.

Los valores requeridos para las coordenadas generalizadas y fuerzas generalizadas se muestran a continuación en las Figuras 4.17 a 4.24.

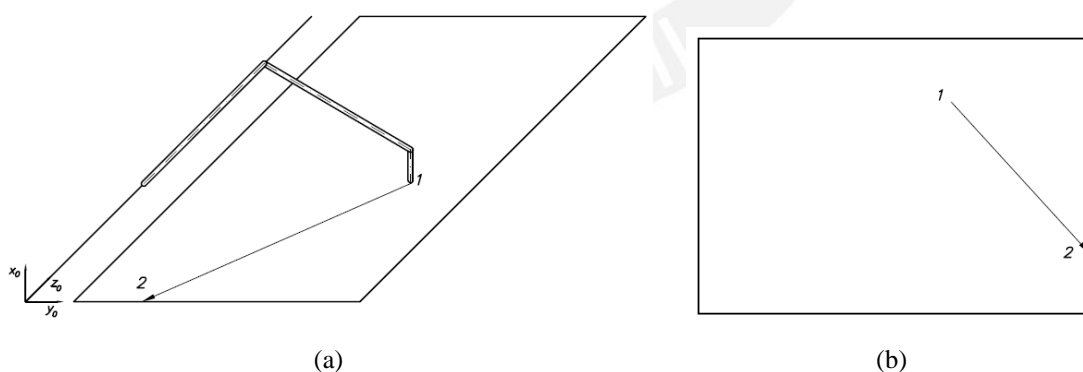


Figura 4.16. Trayectoria deseada con el peso en contra del movimiento (a) Vista 3D, (b) Vista 2D

Para esta nueva trayectoria deseada se hace un cambio en los Datos de Entrada del programa MATLAB, tal y como se muestra en la Rutina 4.3 en la sección 4.3.

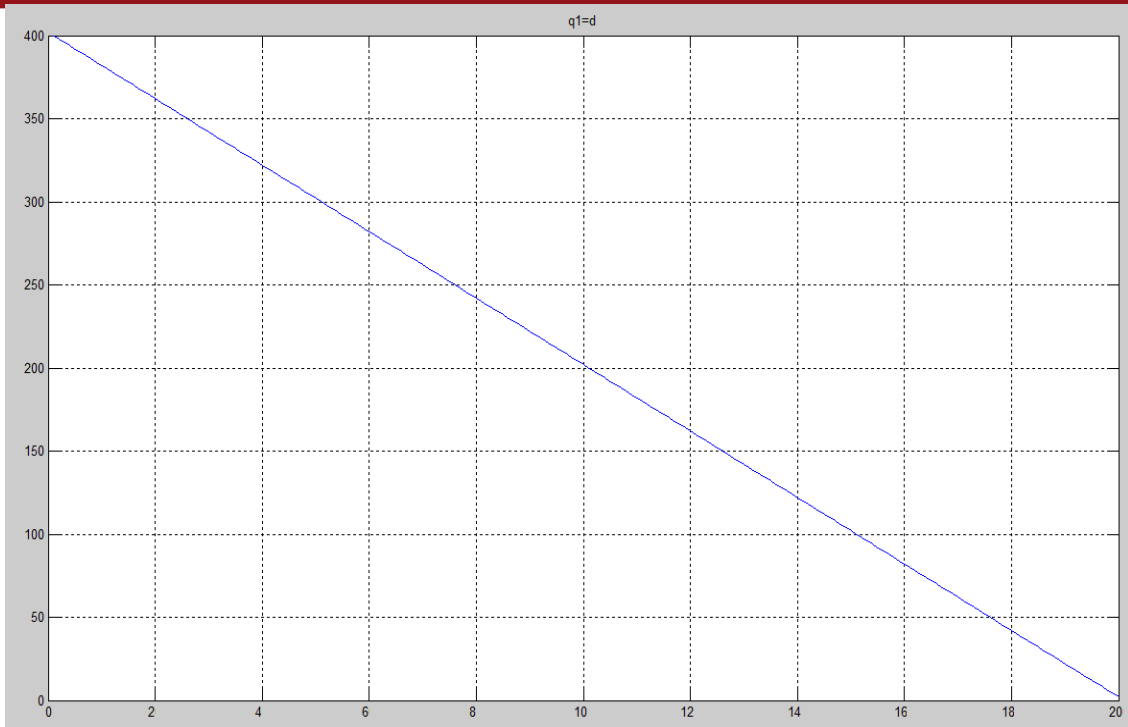


Figura 4.17. Primera coordenada generalizada (d) [mm] en función al tiempo [s]

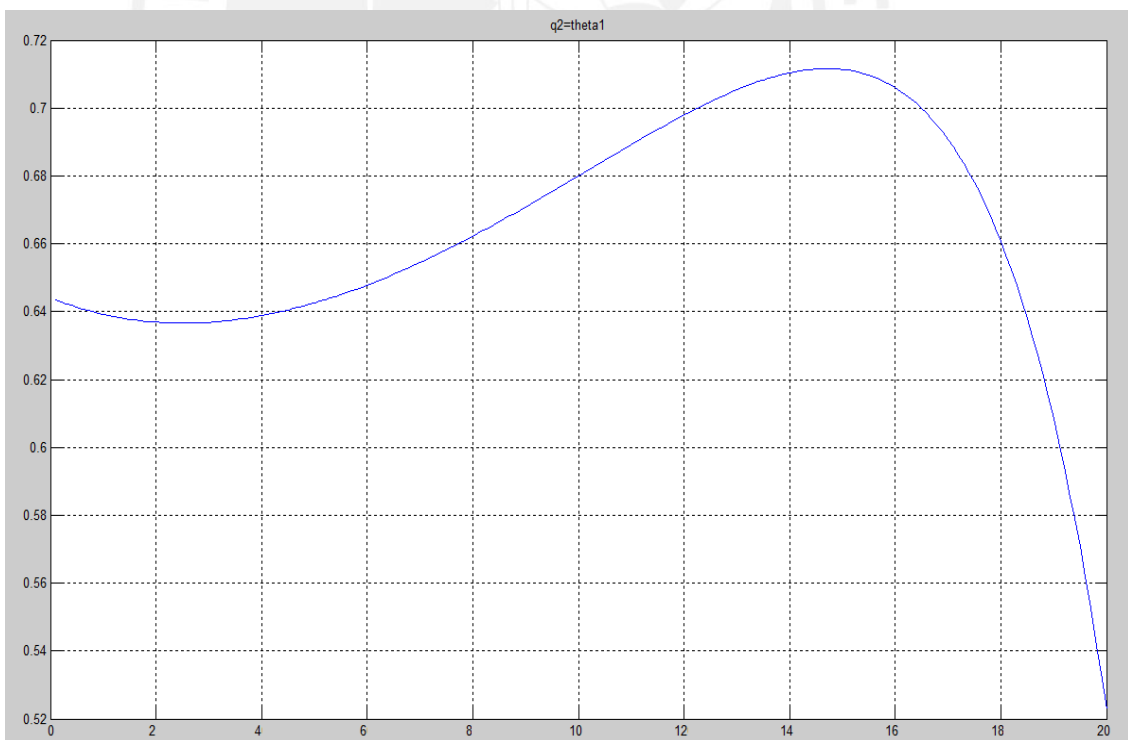


Figura 4.18. Segunda coordenada generalizada (θ_1) [rad] en función al tiempo [s]

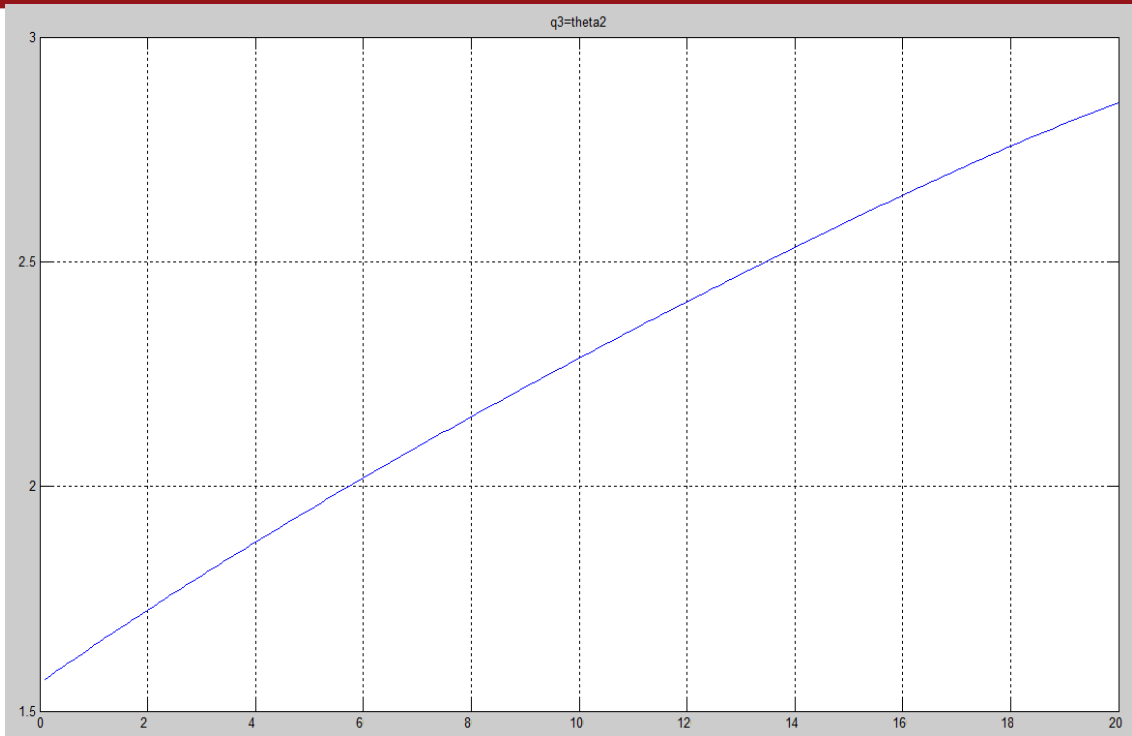


Figura 4.19. Tercera coordenada generalizada (θ_2) [rad] en función al tiempo [s]

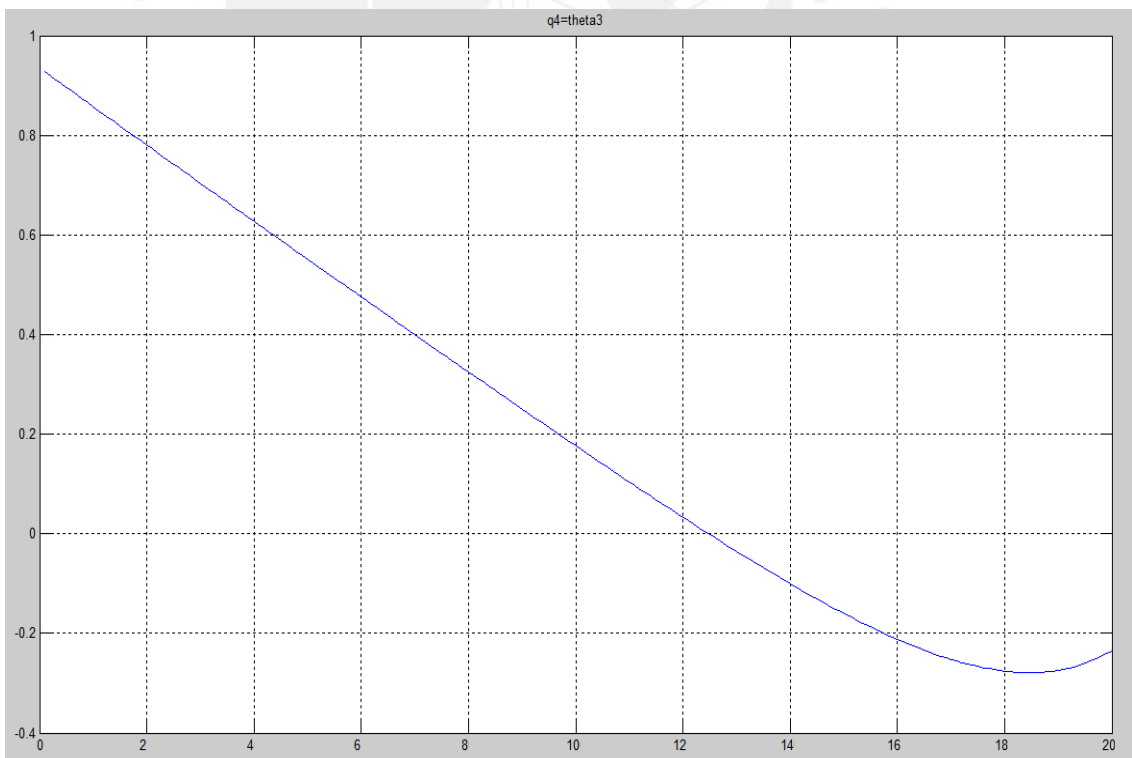


Figura 4.20. Cuarta coordenada generalizada (θ_3) [rad] en función al tiempo [s]

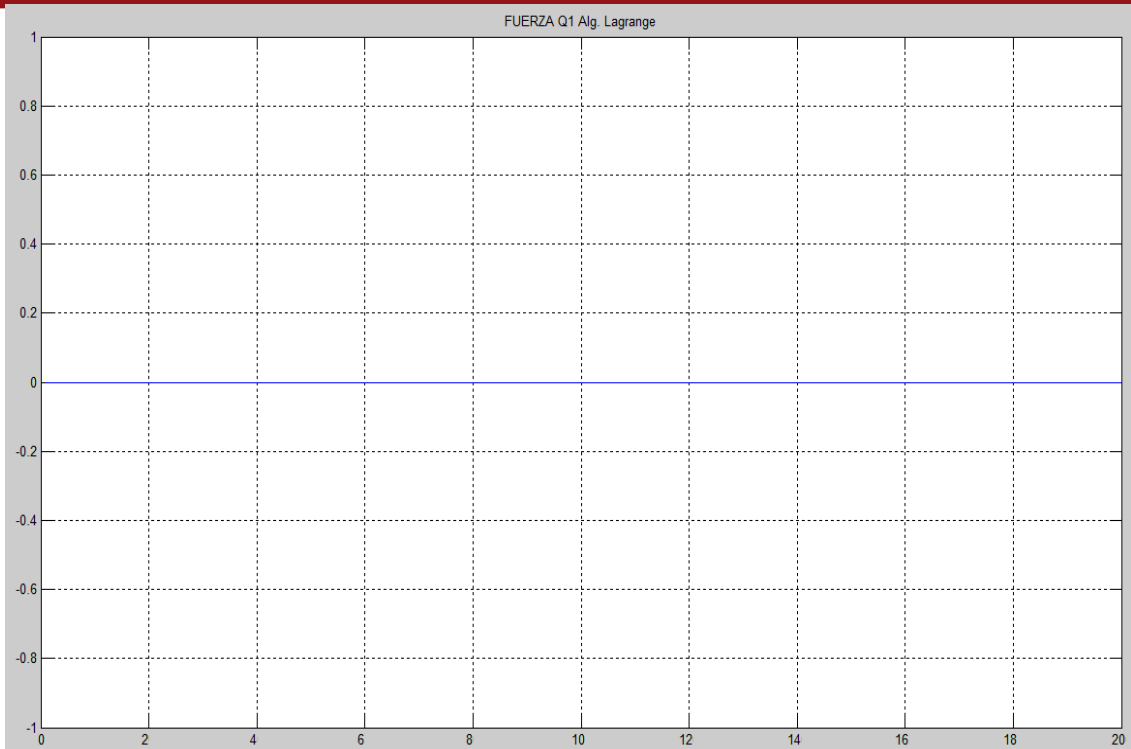


Figura 4.21. Fuerza generalizada (F) [$\text{kg} \times \text{mm}/\text{s}^2$] del primer efector en función al tiempo [s]

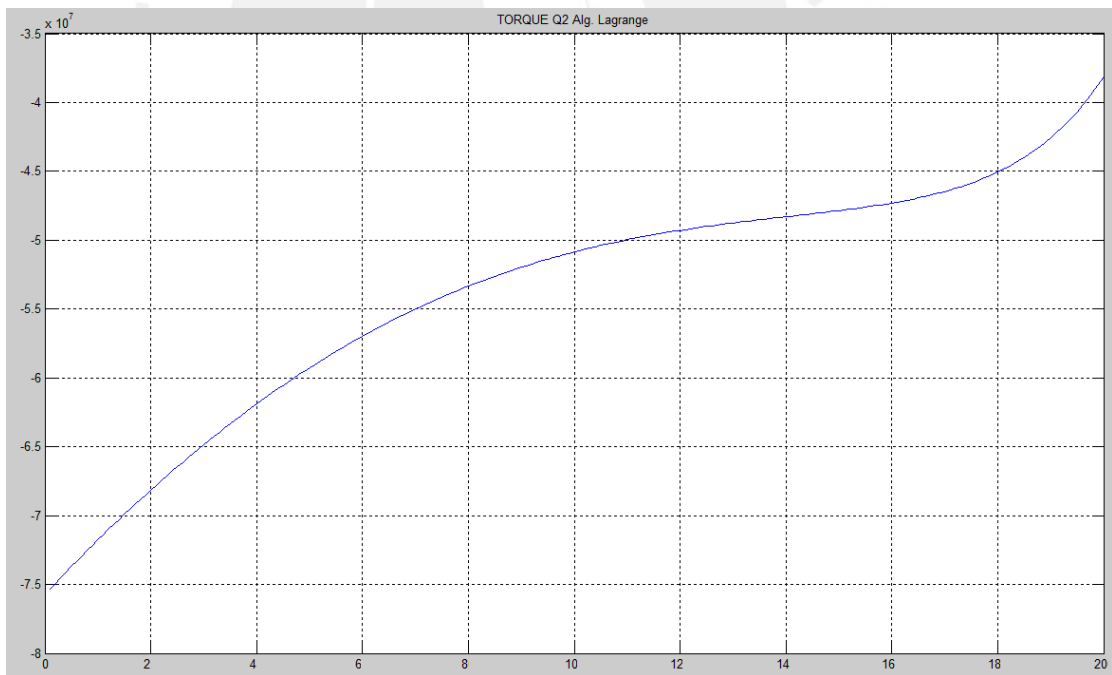


Figura 4.22. Torque generalizado (T_1) [$\text{kg} \times \text{mm}^2/\text{s}^2$] del segundo efector en función al tiempo [s]

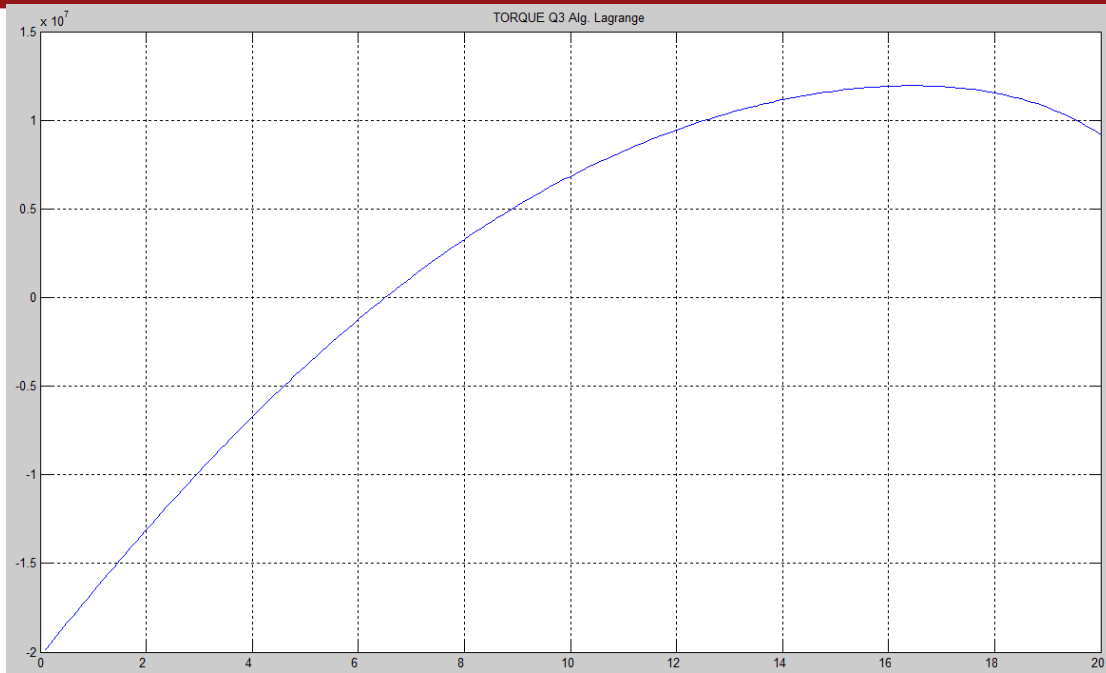


Figura 4.23. Torque generalizado (T_2) [$\text{kg} \times \text{mm}^2/\text{s}^2$] del tercer efector en función al tiempo [s]

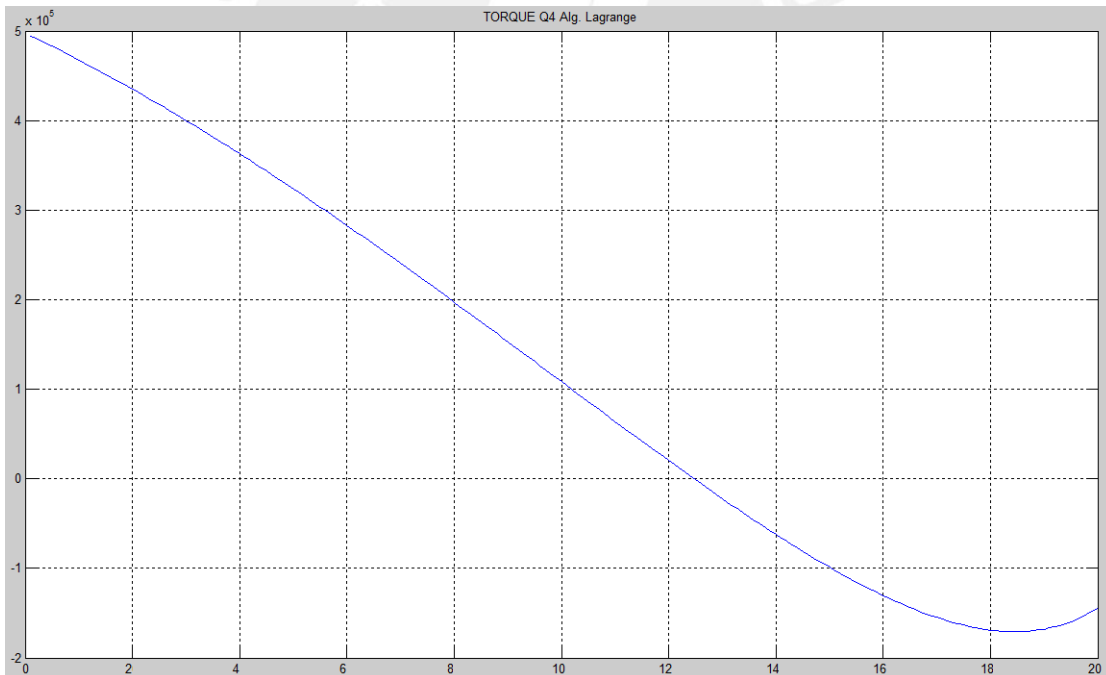


Figura 4.24. Torque generalizado (T_3) [$\text{kg} \times \text{mm}^2/\text{s}^2$] del cuarto efector en función al tiempo [s]

Comparando los resultados de las dos simulaciones realizadas encontramos que los comportamientos son opuestos siendo los torques requeridos cuando el efecto de peso es contrario al movimiento ligeramente mayores, lo que significa que el peso es poco significativo a comparación de las fuerzas de inercia requeridas para esta clase de mecanismos en donde los eslabones son largos y livianos.

4.2.4 Simulación de movimiento más exigido para selección de actuadores

La información que brinda el estudio del movimiento más exigido es fundamental para la selección de los actuadores, por ello consideraremos un caso en el que se necesite llegar a la velocidad máxima de trabajo definida en el Capítulo 2 como 80 mm/min en un tiempo razonable, que en este caso se ha considerado en 0.5 segundos, valor que puede cambiar según la aplicación. En este caso, como no se tiene un tiempo de puesta en marcha muy exigido este valor se ajusta a las necesidades.

En este caso se está considerando que se desplace en una línea recta llegando a la velocidad máxima de trabajo. Esta disposición de la velocidad es elegida porque el robot empezará su movimiento desde el reposo y luego acelerará hasta alcanzar la velocidad requerida, la cual se deseará mantener constante durante el trabajo.

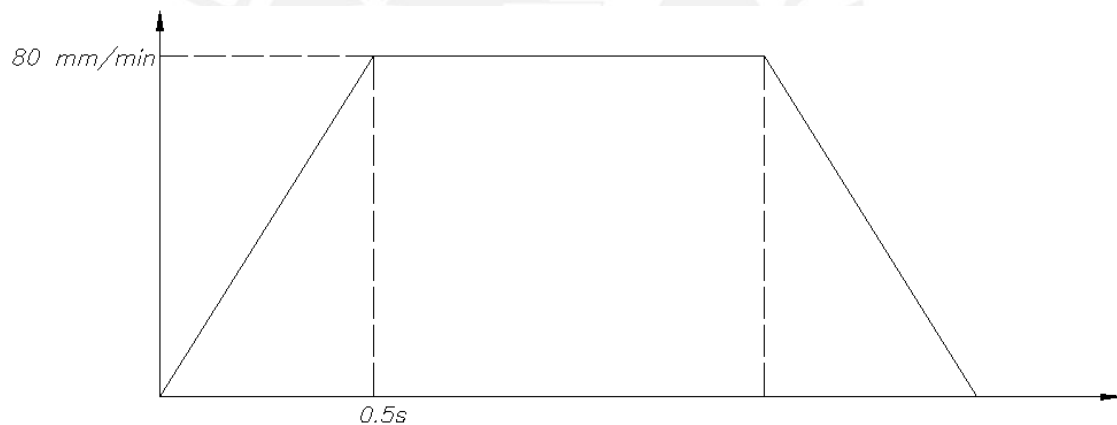


Figura 4.25. Velocidad deseada en función del tiempo

Para analizar los dos casos más críticos para los actuadores vamos a considerar que la velocidad en ambos ejes debe llegar a ser máxima en 0.5 segundos, con lo cual ambas trayectorias se podrían definir como:

$$y = y_0 - 1.3333 \times t^2 \text{ [mm]}$$

$$z = z_0 - 1.3333 \times t^2 \text{ [mm]}$$

Los valores presentados se han reemplazado en la Rutina 4.4 en la sección 4.3 para realizar una nueva simulación. En dicha simulación se puede notar que la velocidad es

negativa lo cual significa que trata del movimiento de retorno, en el cual el valor de las coordenadas generalizadas angulares está disminuyendo.

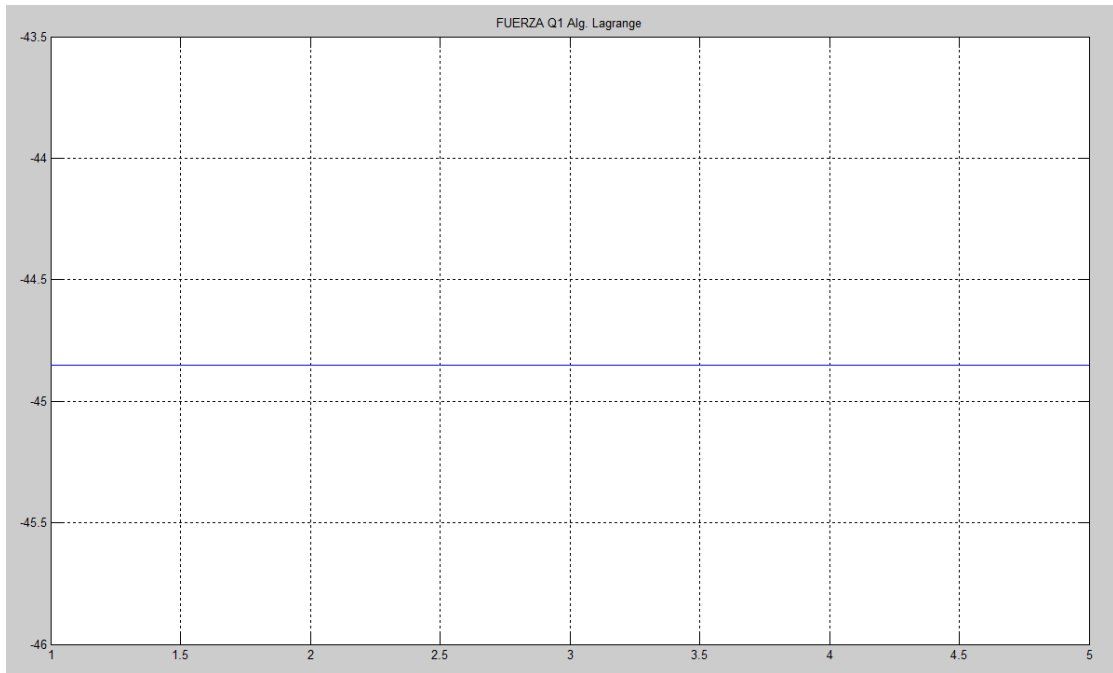


Figura 4.26. Fuerza generalizada (F) [$\text{kg} \times \text{mm}/\text{s}^2$] del primer efector en función al tiempo [0.1 x s]

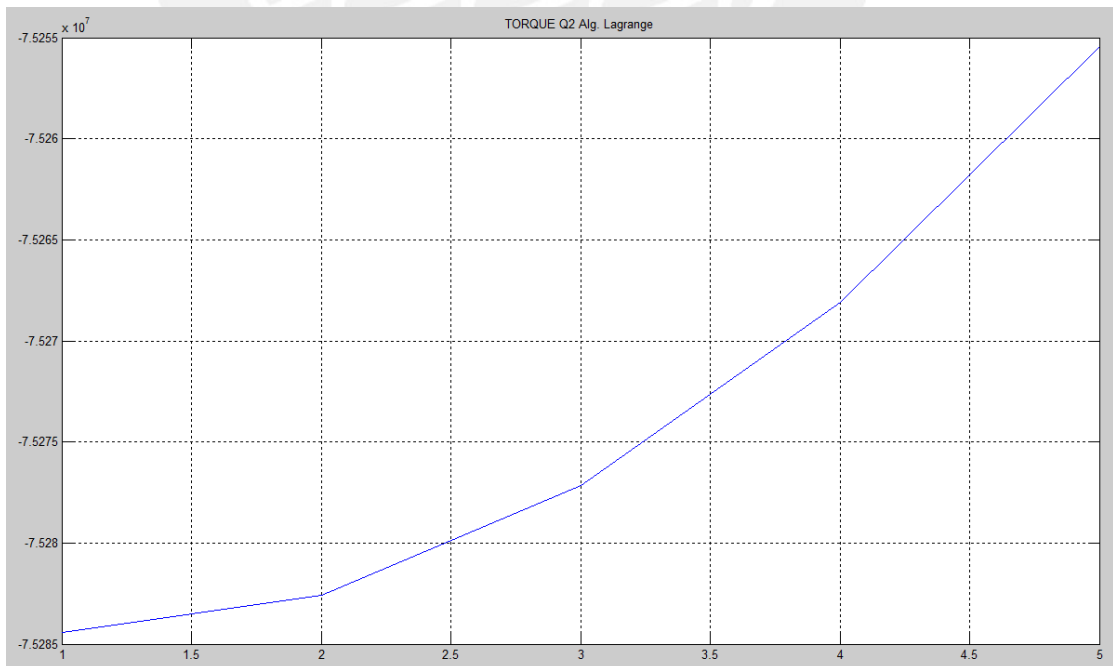


Figura 4.27. Torque generalizado (T_1) [$\text{kg} \times \text{mm}^2/\text{s}^2$] del segundo efector en función al tiempo [0.1s]

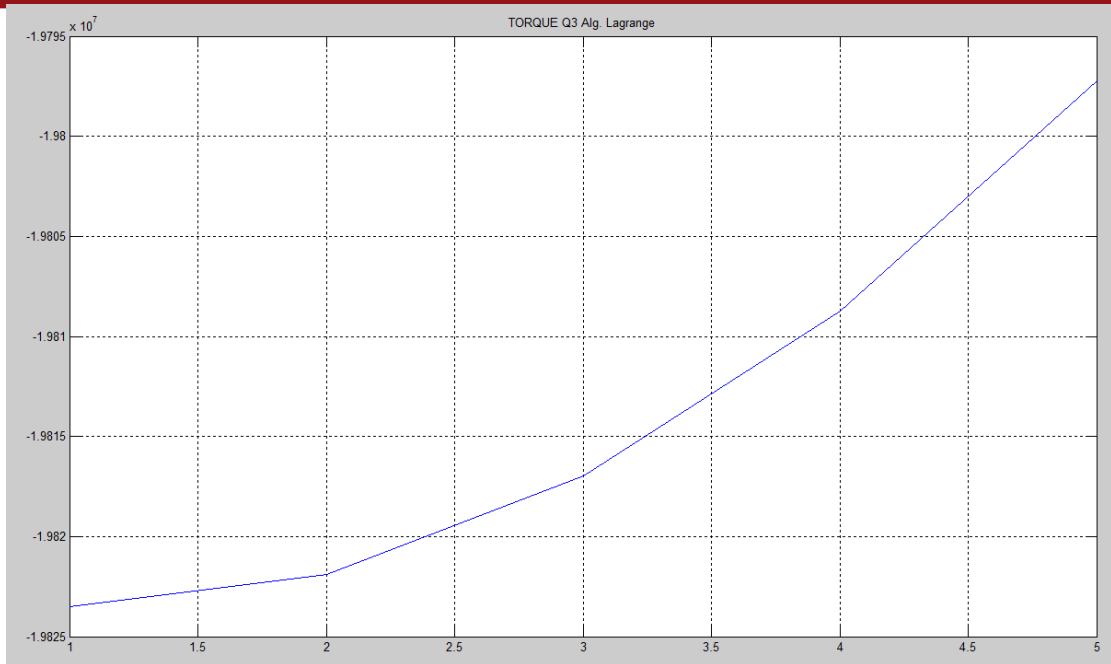


Figura 4.28. Torque generalizado (T_2) [$\text{kg} \times \text{mm}^2/\text{s}^2$] del tercer efector en función al tiempo [0.1s]

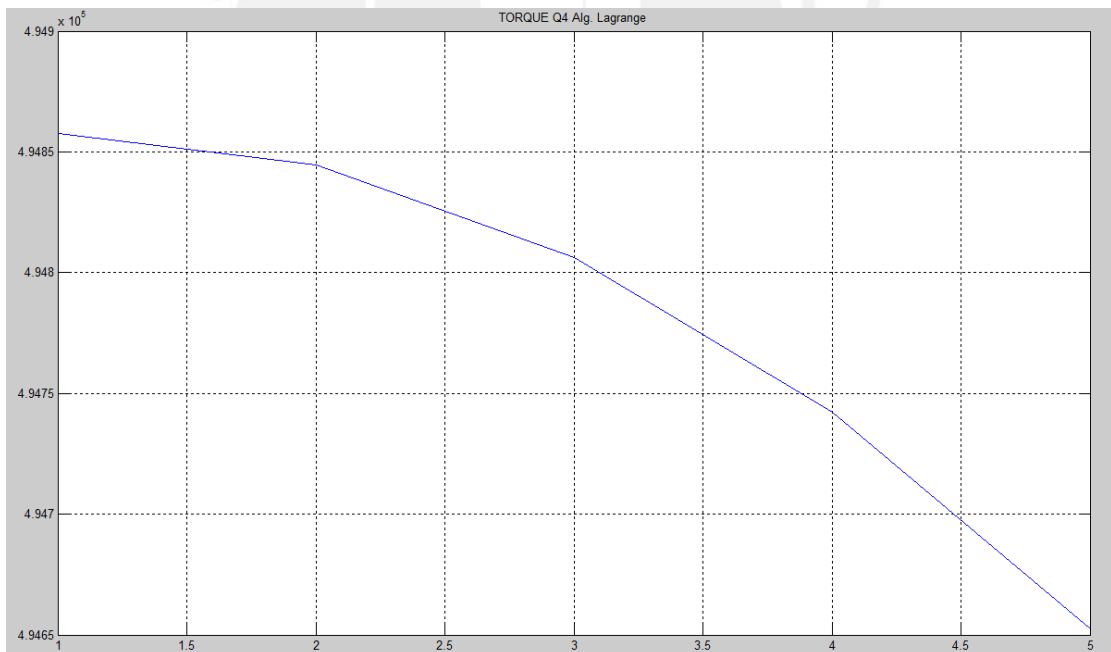


Figura 4.29. Torque generalizado (T_3) [$\text{kg} \times \text{mm}^2/\text{s}^2$] del cuarto efector en función al tiempo [0.1s]

El resultado de esta simulación cobrará importancia cuando se realice la selección de actuadores en el proceso del diseño mecánico en donde se tendrá que comprobar que ellos sean capaces de alcanzar el torque y velocidad máximos requeridos para esta aplicación. Ambos datos pueden obtenerse de los resultados de estas simulaciones.

4.3 Rutinas para Simulación

Rutina 4.1: Código para simulación dinámica para movimiento paralelo al lado mayor

```

% Aplicación del algoritmo computacional de Lagrange para obtener
% el modelo dinámico del robot de 4 grados de libertad

%
%=====
% DATOS DE ENTRADA
%=====

% Parámetros dimensionales del mecanismo

l1=500; % in mm
l2=500; % in mm
l3=100; % in mm
g=9810; % in mm/s^2
G=[-g 0 0 0];

% Ahora se necesita definir la matriz de pseudo inercia tomando como
% base las propiedades físicas de los eslabones, actuadores, juntas y
% todos los cuerpos cuya inercia influye en la ecuación de movimiento
% (En este ejemplo se están usando valores aproximados que no
% necesariamente representan el valor verdadero de inercia del robot
% real. Estos valores se han calculado tomando paralelepípedos metálicos
% de las longitudes determinadas)

m1=3; % in kg
m2=6.28; % in kg
m3=6.28; % in kg
m4=1.26; % in kg

% Inercias en kg*mm^2/s^2 de los eslabones incluyendo actuadores y
contrapesos

J1=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 m1];
J2=[523333.33 0 0 1570; 0 837.33 0 0; 0 0 837.33 0; 1570 0 0 m2];
J3=[523333.33 0 0 1570; 0 837.33 0 0; 0 0 837.33 0; 1570 0 0 m3];
J4=[4200 0 0 63; 0 168 0 0; 0 0 168 0; 63 0 0 m4];

% Definición simbólica de la trayectoria cuando el actuador tiene
% movimiento paralelo al lado mayor, d=1000+40*t-20*t^2

sq1='1000+40*t-20*t^2';
sq2='1.2945602';
sq3='0.349038';
sq4='1.492780';

%
% RUTINA PARA SIMULACION NUMERICA
%=====

% Cálculo de velocidades y aceleraciones

sqd1=diff(sq1, 't'); sqdd1=diff(sqd1, 't');
sqd2=diff(sq2, 't'); sqdd2=diff(sqd2, 't');
sqd3=diff(sq3, 't'); sqdd3=diff(sqd3, 't');
sqd4=diff(sq4, 't'); sqdd4=diff(sqd4, 't');

% Bucle de tiempo

for tk=1:1:50;
t=(tk-1)/10;

```

```

U21=[0 0 0 0; 0 0 0 0; 0 0 0 1; 0 0 0 0];
U22=[-sin(q2) -cos(q2) 0 -l1*sin(q2); cos(q2) -sin(q2) 0 l1*cos(q2); 0 0
0 0; 0 0 0 0];
U23=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U24=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];

U31=[0 0 0 0; 0 0 0 0; 0 0 0 1; 0 0 0 0];
U32=[-sin(q2+q3) -cos(q2+q3) 0 -l1*sin(q2)-l2*sin(q2+q3); cos(q2+q3) -
sin(q2+q3) 0 l1*cos(q2)+l2*cos(q2+q3); 0 0 0 0; 0 0 0 0];
U33=[-sin(q2+q3) -cos(q2+q3) 0 -l2*sin(q2+q3); cos(q2+q3) -sin(q2+q3) 0
l2*cos(q2+q3); 0 0 0 0; 0 0 0 0];
U34=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];

U41=[0 0 0 0; 0 0 0 0; 0 0 0 1; 0 0 0 0];
U42=[-sin(q2+q3+q4) -cos(q2+q3+q4) 0 -l1*sin(q2)-l2*sin(q2+q3)-
l3*sin(q2+q3+q4); cos(q2+q3+q4) -sin(q2+q3+q4) 0
l1*cos(q2)+l2*cos(q2+q3)+l3*cos(q2+q3+q4); 0 0 0 0; 0 0 0 0];
U43=[-sin(q2+q3+q4) -cos(q2+q3+q4) 0 -l2*sin(q2+q3)-l3*sin(q2+q3+q4);
cos(q2+q3+q4) -sin(q2+q3+q4) 0 l2*cos(q2+q3)+l3*cos(q2+q3+q4); 0 0 0 0; 0
0 0 0];
U44=[-sin(q2+q3+q4) -cos(q2+q3+q4) 0 -l3*sin(q2+q3+q4); cos(q2+q3+q4) -
sin(q2+q3+q4) 0 l3*cos(q2+q3+q4); 0 0 0 0; 0 0 0 0];

U111=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U112=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U113=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U114=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];

U121=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U122=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U123=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U124=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];

U131=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U132=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U133=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U134=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];

U141=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U142=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U143=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U144=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];

U211=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U212=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U213=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U214=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];

U221=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U222=[-cos(q2) sin(q2) 0 -l1*cos(q2); -sin(q2) -cos(q2) 0 -l1*sin(q2); 0
0 0 0; 0 0 0 0];
U223=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U224=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];

U231=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U232=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U233=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U234=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];

U241=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U242=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U243=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U244=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
    
```

```

U311=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U312=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U313=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U314=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];

U321=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U322=[-cos(q2+q3) sin(q2+q3) 0 -l1*cos(q2)-l2*cos(q2+q3); -sin(q2+q3) -
cos(q2+q3) 0 -l1*sin(q2)-l2*sin(q2+q3); 0 0 0 0; 0 0 0 0];
U323=[-cos(q2+q3) sin(q2+q3) 0 -l2*cos(q2+q3); -sin(q2+q3) -cos(q2+q3) 0
-l2*sin(q2+q3); 0 0 0 0; 0 0 0 0];
U324=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];

U331=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U332=[-cos(q2+q3) sin(q2+q3) 0 -l2*cos(q2+q3); -sin(q2+q3) -cos(q2+q3) 0
-l2*sin(q2+q3); 0 0 0 0; 0 0 0 0];
U333=[-cos(q2+q3) sin(q2+q3) 0 -l2*cos(q2+q3); -sin(q2+q3) -cos(q2+q3) 0
-l2*sin(q2+q3); 0 0 0 0; 0 0 0 0];
U334=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];

U341=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U342=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U343=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U344=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];

U411=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U412=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U413=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U414=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];

U421=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U422=[-cos(q2+q3+q4) sin(q2+q3+q4) 0 -l1*cos(q2)-l2*cos(q2+q3)-
l3*cos(q2+q3+q4); -sin(q2+q3+q4) -cos(q2+q3+q4) 0 -l1*sin(q2)-
l2*sin(q2+q3)-l3*sin(q2+q3+q4); 0 0 0 0; 0 0 0 0];
U423=[-cos(q2+q3+q4) sin(q2+q3+q4) 0 -l2*cos(q2+q3)-l3*cos(q2+q3+q4); -
sin(q2+q3+q4) -cos(q2+q3+q4) 0 -l2*sin(q2+q3)-l3*sin(q2+q3+q4); 0 0 0 0;
0 0 0 0];
U424=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];

U431=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U432=[-cos(q2+q3+q4) sin(q2+q3+q4) 0 -l2*cos(q2+q3)-l3*cos(q2+q3+q4); -
sin(q2+q3+q4) -cos(q2+q3+q4) 0 -l2*sin(q2+q3)-l3*sin(q2+q3+q4); 0 0 0 0;
0 0 0 0];
U433=[-cos(q2+q3+q4) sin(q2+q3+q4) 0 -l2*cos(q2+q3)-l3*cos(q2+q3+q4); -
sin(q2+q3+q4) -cos(q2+q3+q4) 0 -l2*sin(q2+q3)-l3*sin(q2+q3+q4); 0 0 0 0;
0 0 0 0];
U434=[-cos(q2+q3+q4) sin(q2+q3+q4) 0 -l3*cos(q2+q3+q4); -sin(q2+q3+q4) -
cos(q2+q3+q4) 0 -l3*sin(q2+q3+q4); 0 0 0 0; 0 0 0 0];

U441=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
U442=[-cos(q2+q3+q4) sin(q2+q3+q4) 0 -l3*cos(q2+q3+q4); -sin(q2+q3+q4) -
cos(q2+q3+q4) 0 -l3*sin(q2+q3+q4); 0 0 0 0; 0 0 0 0];
U443=[-cos(q2+q3+q4) sin(q2+q3+q4) 0 -l3*cos(q2+q3+q4); -sin(q2+q3+q4) -
cos(q2+q3+q4) 0 -l3*sin(q2+q3+q4); 0 0 0 0; 0 0 0 0];
U444=[-cos(q2+q3+q4) sin(q2+q2+q4) 0 -l3*cos(q2+q3+q4); -sin(q2+q3+q4) -
cos(q2+q3+q4) 0 -l3*sin(q2+q3+q4); 0 0 0 0; 0 0 0 0];
    
```

```

% Matriz de inercia D

D(1,1)=trace(U11*J1*(U11'))+trace(U21*J2*(U21'))+trace(U31*J3*(U31'))+tra
ce(U41*J4*(U41'));
D(1,2)=trace(U22*J2*(U21'))+trace(U32*J3*(U31'))+trace(U42*J4*(U41'));
D(1,3)=trace(U33*J3*(U31'))+trace(U43*J4*(U41'));
D(1,4)=trace(U44*J4*(U41'));

D(2,1)=trace(U21*J2*(U22'))+trace(U31*J3*(U32'))+trace(U41*J4*(U42'));
D(2,2)=trace(U22*J2*(U22'))+trace(U32*J3*(U32'))+trace(U42*J4*(U42'));
D(2,3)=trace(U33*J3*(U32'))+trace(U43*J4*(U42'));
D(2,4)=trace(U44*J4*(U42'));

D(3,1)=trace(U31*J3*(U33'))+trace(U41*J4*(U43'));
D(3,2)=trace(U32*J3*(U33'))+trace(U42*J4*(U43'));
D(3,3)=trace(U33*J3*(U33'))+trace(U43*J4*(U43'));
D(3,4)=trace(U44*J4*(U43'));

D(4,1)=trace(U41*J4*(U44'));
D(4,2)=trace(U42*J4*(U44'));
D(4,3)=trace(U43*J4*(U44'));
D(4,4)=trace(U44*J4*(U44'));

% Términos hikm

h111=trace(U111*J1*(U11'))+trace(U211*J2*(U21'))+trace(U311*J3*(U31'))+tr
ace(U411*J4*(U41'));
h112=trace(U212*J2*(U21'))+trace(U312*J3*(U31'))+trace(U412*J4*(U41'));
h113=trace(U313*J3*(U31'))+trace(U413*J4*(U41'));
h114=trace(U414*J4*(U41'));

h121=trace(U221*J2*(U21'))+trace(U321*J3*(U31'))+trace(U421*J4*(U41'));
h122=trace(U222*J2*(U21'))+trace(U322*J3*(U31'))+trace(U422*J4*(U41'));
h123=trace(U323*J3*(U31'))+trace(U423*J4*(U41'));
h124=trace(U424*J4*(U41'));

h131=trace(U331*J3*(U31'))+trace(U431*J4*(U41'));
h132=trace(U332*J3*(U31'))+trace(U432*J4*(U41'));
h133=trace(U333*J3*(U31'))+trace(U433*J4*(U41'));
h134=trace(U434*J4*(U41'));

h141=trace(U441*J4*(U41'));
h142=trace(U442*J4*(U41'));
h143=trace(U443*J4*(U41'));
h144=trace(U444*J4*(U41'));

h211=trace(U211*J2*(U22'))+trace(U311*J3*(U32'))+trace(U411*J4*(U42'));
h212=trace(U212*J2*(U22'))+trace(U412*J3*(U32'))+trace(U412*J4*(U42'));
h213=trace(U313*J3*(U22'))+trace(U413*J4*(U42'));
h214=trace(U414*J4*(U42'));

h221=trace(U221*J2*(U22'))+trace(U321*J3*(U32'))+trace(U421*J4*(U42'));
h222=trace(U222*J2*(U22'))+trace(U322*J3*(U32'))+trace(U422*J4*(U42'));
h223=trace(U323*J3*(U32'))+trace(U423*J4*(U42'));
h224=trace(U424*J4*(U42'));

h231=trace(U331*J3*(U32'))+trace(U431*J4*(U42'));
h232=trace(U332*J3*(U32'))+trace(U432*J4*(U42'));
h233=trace(U333*J3*(U32'))+trace(U433*J4*(U42'));
h234=trace(U434*J4*(U42'));

h241=trace(U441*J4*(U42'));
h242=trace(U442*J4*(U42'));
h243=trace(U443*J4*(U42'));
h244=trace(U444*J4*(U42'));

```

```
h311=trace(U311*J3*(U33'))+trace(U411*J4*(U43')) ;
h312=trace(U312*J3*(U33'))+trace(U412*J4*(U43')) ;
h313=trace(U313*J3*(U33'))+trace(U413*J4*(U43')) ;
h314=trace(U414*J4*(U43')) ;
```

```
h321=trace(U321*J3*(U33'))+trace(U421*J4*(U43')) ;
h322=trace(U322*J3*(U33'))+trace(U422*J4*(U43')) ;
h323=trace(U323*J3*(U33'))+trace(U423*J4*(U43')) ;
h324=trace(U424*J4*(U43')) ;
```

```
h331=trace(U331*J3*(U33'))+trace(U431*J4*(U43')) ;
h332=trace(U332*J3*(U33'))+trace(U432*J4*(U43')) ;
h333=trace(U333*J3*(U33'))+trace(U433*J4*(U43')) ;
h334=trace(U434*J4*(U43')) ;
```

```
h341=trace(U341*J3*(U33'))+trace(U441*J4*(U43')) ;
h342=trace(U342*J3*(U33'))+trace(U442*J4*(U43')) ;
h343=trace(U343*J3*(U33'))+trace(U443*J4*(U43')) ;
h344=trace(U444*J4*(U43')) ;
```

```
h411=trace(U411*J4*(U44')) ;
h412=trace(U412*J4*(U44')) ;
h413=trace(U413*J4*(U44')) ;
h414=trace(U414*J4*(U44')) ;
```

```
h421=trace(U421*J4*(U44')) ;
h422=trace(U422*J4*(U44')) ;
h423=trace(U423*J4*(U44')) ;
h424=trace(U424*J4*(U44')) ;
```

```
h431=trace(U431*J4*(U44')) ;
h432=trace(U432*J4*(U44')) ;
h433=trace(U433*J4*(U44')) ;
h434=trace(U434*J4*(U44')) ;
```

```
h441=trace(U441*J4*(U44')) ;
h442=trace(U442*J4*(U44')) ;
h443=trace(U443*J4*(U44')) ;
h444=trace(U444*J4*(U44')) ;
```

% Matriz de aceleraciones centrípeta y de Coriolis

```
H(1,1)=h111*qd1*qd1+h121*qd2*qd1+h131*qd3*qd1+h141*qd4*qd1+h112*qd1*qd2+
h122*qd2*qd2+h132*qd3*qd2+h142*qd4*qd2+h113*qd1*qd3+h123*qd2*qd3+h133*qd
3*qd3+h143*qd4*qd3+h114*qd1*qd4+h124*qd2*qd4+h134*qd3*qd4+h144*qd4*qd4 ;
H(2,1)=h211*qd1*qd1+h221*qd2*qd1+h231*qd3*qd1+h241*qd4*qd1+h212*qd1*qd2+
h222*qd2*qd2+h232*qd3*qd2+h242*qd4*qd2+h213*qd1*qd3+h223*qd2*qd3+h233*qd
3*qd3+h243*qd4*qd3+h214*qd1*qd4+h224*qd2*qd4+h234*qd3*qd4+h244*qd4*qd4 ;
H(3,1)=h311*qd1*qd1+h321*qd2*qd1+h331*qd3*qd1+h341*qd4*qd1+h312*qd1*qd2+
h322*qd2*qd2+h332*qd3*qd2+h342*qd4*qd2+h313*qd1*qd3+h323*qd2*qd3+h333*qd
3*qd3+h343*qd4*qd3+h314*qd1*qd4+h324*qd2*qd4+h334*qd3*qd4+h344*qd4*qd4 ;
H(4,1)=h411*qd1*qd1+h421*qd2*qd1+h431*qd3*qd1+h441*qd4*qd1+h412*qd1*qd2+
h422*qd2*qd2+h432*qd3*qd2+h442*qd4*qd2+h413*qd1*qd3+h423*qd2*qd3+h433*qd
3*qd3+h443*qd4*qd3+h414*qd1*qd4+h424*qd2*qd4+h434*qd3*qd4+h444*qd4*qd4 ;
```

% Cálculo de la matriz de gravedad C

```
r11=[0 0 0 1]';
r22=[l1*cos(q2)/2 l1*sin(q2)/2 0 1]';
r33=[l2*cos(q3)/2 l2*sin(q3)/2 0 1]';
r44=[l3*cos(q4)/2 l3*sin(q4)/2 0 1]';
```

```

C(1,1)=-m1*G*U11*r11-m2*G*U21*r22-m3*G*U31*r33-m4*G*U41*r44;
C(2,1)=-m1*G*U12*r11-m2*G*U22*r22-m3*G*U32*r33-m4*G*U42*r44;
C(3,1)=-m1*G*U13*r11-m2*G*U23*r22-m3*G*U33*r33-m4*G*U43*r44;
C(4,1)=-m1*G*U14*r11-m2*G*U24*r22-m3*G*U34*r33-m4*G*U44*r44;

PARES(:,tk)=D*[qdd1; qdd2; qdd3; qdd4]+H+C;

end

% fin del bucle de tiempo

% Ploteo de los resultados de la simulación

subplot(4,2,1), plot(Q1), grid, title('q1=d'), hold
subplot(4,2,2),
plot(PARES(1,:)), grid, title('FUERZA Q1 Alg. Lagrange'), hold
subplot(4,2,3), plot(Q2), grid, title('q2=theta1'), hold
subplot(4,2,4),
plot(PARES(2,:)), grid, title('TORQUE Q2 Alg. Lagrange'), hold
subplot(4,2,5), plot(Q3), grid, title('q3=theta2'), hold
subplot(4,2,6),
plot(PARES(3,:)), grid, title('TORQUE Q3 Alg. Lagrange'), hold
subplot(4,2,7), plot(Q4), grid, title('q4=theta3'), hold
subplot(4,2,8),
plot(PARES(4,:)), grid, title('TORQUE Q4 Alg. Lagrange'), hold

```



Rutina 4.2: Modificación a los datos de entrada cuando simulación cuando el peso está a favor del movimiento

```
% Definición simbólica de la trayectoria considerando el movimiento
% longitudinal como d=20*t y el transversal como 30*t+100

sq1='20*t';
sq2='1.57079632679490-atan(12*sin(acos((13^2+(30*t+100)^2-11^2-12^2)/(2*11*12)))/(11+((13^2+(30*t+100)^2-11^2-12^2)/(2*11*12))))-atan(13/(100+30*t))';
sq3='acos((13^2+(30*t+100)^2-11^2-12^2)/(2*11*12))';
sq4='1.57079632679490+atan(12*sin(acos((13^2+(30*t+100)^2-11^2-12^2)/(2*11*12)))/(11+((13^2+(30*t+100)^2-11^2-12^2)/(2*11*12))))+atan(13/(100+30*t))-acos((13^2+(30*t+100)^2-11^2-12^2)/(2*11*12))';
```

Rutina 4.3: Modificación a los datos de entrada para simulación cuando el peso se opone al movimiento

```
% Definición simbólica de la trayectoria considerando el movimiento
% longitudinal como d=400-20*t y el transversal como 700-30*t

sq1='400-20*t';
sq2='1.57079632679490-atan(12*sin(acos((13^2+(700-30*t)^2-11^2-12^2)/(2*11*12)))/(11+((13^2+(700-30*t)^2-11^2-12^2)/(2*11*12))))-atan(13/(700-30*t))';
sq3='acos((13^2+(700-30*t)^2-11^2-12^2)/(2*11*12))';
sq4='1.57079632679490+atan(12*sin(acos((13^2+(700-30*t)^2-11^2-12^2)/(2*11*12)))/(11+((13^2+(700-30*t)^2-11^2-12^2)/(2*11*12))))+atan(13/(700-30*t))-acos((13^2+(700-30*t)^2-11^2-12^2)/(2*11*12))';
```

Rutina 4.4: Modificación a los datos de entrada para analizar el caso más crítico

```
% Definición simbólica de la trayectoria considerando el movimiento
% longitudinal como d=400-1.3333*t^2 y el transversal como 700-1.3333*t^2

sq1='400-1.3333*t^2';
sq2='1.57079632679490-atan(12*sin(acos((13^2+(700-1.3333*t^2)^2-11^2-12^2)/(2*11*12)))/(11+((13^2+(700-1.3333*t^2)^2-11^2-12^2)/(2*11*12))))-atan(13/(700-1.3333*t^2))';
sq3='acos((13^2+(700-1.3333*t^2)^2-11^2-12^2)/(2*11*12))';
sq4='1.57079632679490+atan(12*sin(acos((13^2+(700-1.3333*t^2)^2-11^2-12^2)/(2*11*12)))/(11+((13^2+(700-1.3333*t^2)^2-11^2-12^2)/(2*11*12))))+atan(13/(700-1.3333*t^2))-acos((13^2+(700-1.3333*t^2)^2-11^2-12^2)/(2*11*12))';
```

OBERVACIONES Y RECOMENDACIONES

1. Las ecuaciones que gobiernan el movimiento de los robots pueden obtenerse tanto por ecuaciones vectoriales como por ecuaciones escalares; sin embargo, el usar ecuaciones vectoriales como las ecuaciones de Newton complica la programación de una rutina para poder simular el movimiento del robot. Esto no ocurre cuando se usan métodos como las ecuaciones de Lagrange en donde el trabajar con ecuaciones escalares facilita el trabajo con herramientas computacionales.
2. El uso de MATLAB para realizar el cálculo de la dinámica puede resultar un poco complicado debido al entorno poco amigable en el cual se trabaja con líneas de comando. Probablemente el uso de un programa de entorno más amigable, como es el caso del Mathcad por ejemplo, pudo haber simplificado el cálculo.
3. En la simulación dinámica realizada en el presente trabajo se analizó sólo el movimiento durante el trabajo del robot, es decir cuando el efector final se desplaza en un plano horizontal. Se recomienda para un futuro trabajo analizar también el movimiento del robot cuando el efector final aún no se ha ubicado en el plano de trabajo y, por lo tanto, tiene también movimiento en el eje vertical.
4. En el diseño de la cadena cinemática a usar se recomienda intentar el uso de un mecanismo cartesiano en vez de un robot articular, debido a las ventajas que presentan los robots cartesianos para el trabajo en plano siendo sus principales desventajas su mayor tamaño y su dificultad para trabajar con piezas de gran altura.
5. Se recomienda para un trabajo a futuro considerar también el efecto de la fricción en la dinámica de este tipo de mecanismos. Los efectos de fricción se verían reflejados como otra fuerza no conservativa que se incluiría en las ecuaciones de Lagrange.
6. En la búsqueda de un modelo matemático para describir el movimiento del robot se exploraron muchas posibilidades, dentro de las cuales estaban también los Bond Graphs, que permiten modelar no sólo la dinámica de mecanismo sino también a

los actuadores. Al obtener un modelo con este método se pudo notar cuánto se complicaba el modelo debido principalmente a la presencia de modulación en las relaciones de transformación de energía, por ese motivo se decidió desistir y continuar trabajando el modelo matemático tomando como base las formulaciones de Denavit Hartenberg y de Uicker. Sin embargo, se recomienda que se explore esta posibilidad para el estudio de robots cartesianos en los cuales el efecto de esta modulación no está presente.



CONCLUSIONES

1. En el presente trabajo se ha elaborado un programa en MATLAB para trabajar las ecuaciones de movimiento de un robot articular de cuatro grados de libertad como contribución a un proyecto multidisciplinario de diseño de un robot para corte de madera usando láser. El programa toma como datos de entrada los pesos e inercias de los eslabones incluyendo actuadores y la trayectoria deseada para el efector final para obtener como respuesta los valores de fuerzas y momentos requeridos en los motores para lograr que el robot cumpla con el comportamiento deseado.
2. El comportamiento del modelo en las simulaciones estuvo dentro de lo esperado para los distintos casos estudiados siguiendo las tendencias de crecimiento y decrecimiento esperadas en los casos mostrados en el Capítulo 4.2. Estos resultados de torques requeridos, en especial los de la simulación en el caso considerado más crítico, serán útiles para dimensionar los actuadores del robot cuando se empiece el proceso de diseño mecánico. El resultado de esta modelación matemática sirve también para el sistema de control del robot, ya que al definir la trayectoria deseada se obtiene información del torque requerido en los motores, lo cual serviría a su vez como una señal de referencia que describa el comportamiento deseado para el sistema de control.
3. El método aplicado en la obtención del modelo matemático está basado en los algoritmos de Denavit Hartenberg y de Uicker. Estos métodos permiten que la rutina MATLAB pueda ser utilizada para hacer simulaciones de movimiento de otros robots haciendo cambios las sub rutinas correspondientes a los parámetros de Denavit Hartenberg y las matrices de inercia.
4. El desarrollo del programa ha comprobado la facilidad que brinda el uso de métodos computacionales para el estudio de la mecánica de cuerpos rígidos, aplicado en este caso a la Robótica, para realizar cálculos a gran velocidad que es lo que nos permitió poder evaluar las fuerzas dinámicas en nuestro mecanismo.

BIBLIOGRAFÍA

- [BARRIENTOS, 2007] BARRIENTOS, Antonio. Fundamentos de Robótica. Madrid: McGraw-Hill / Interamericana, 2007. 624 p.
- [CESOL, 1995] CESOL, “Manual del Soldador”, 6ta Edición, 1995.
- [CORKE, 1995] CORKE, Peter I. A Robotics Toolbox for Matlab. IEEE Robotics and Automation Magazine. March 1996. Volume 3.
- [CRAIG, 2006] CRAIG, John J. Robótica. Pearson Education 3era Ed.
- [DENAVID, 1995] DENAVIT, J. HARTENBERG, R.S. A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices. Journal of Applied Mechanics. Junio 1995.
- [FU, 1988] FU, K.S. Robótica: control, detección, visión e inteligencia. Madrid: McGraw-Hill, 1988. 599 p. ISBN: 8476152140.
- [MARION, 1981] MARION, Jerry. Dinámica clásica de partículas y sistemas. Editorial Reverté. 1981.
- [SPONG, 1996] SPONG, M. W. Motion Control of Robot Manipulators. IEEE Press. 1993.