

**Tecnológico Nacional de México**  
**Instituto Tecnológico de Nuevo León**  
**División de Estudios de Posgrado e Investigación**



Adaptación de trayectorias de un robot planar en un sistema de neuro rehabilitación motriz de pacientes con enfermedad cerebro vascular

**T E S I S**

Para obtener el grado de:  
**MAESTRO EN INGENIERÍA**

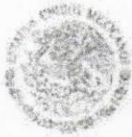
Presenta:

**LUIS FERNANDO MONRREAL ARELLANO**

Director de tesis:

**DR. JUAN ANTONIO ROJAS ESTRADA**

GUADALUPE, N.L. MAYO, 2020



"2020. Año de Leonora Vicario, Benemérita Madre de la Patria"

**Aceptación de documento de Tesis**

Guadalupe Nuevo León, **3/ Junio 2020**


**DR. MARIO CÉSAR OSORIO ABRAHAM**  
**JEFE DE LA DIVISIÓN DE ESTUDIOS**  
**DE POSGRADO E INVESTIGACIÓN**  
**PRESENTE:**

La Comisión de Revisión de Tesis nos es grato comunicarle que, conforme a los lineamientos para la obtención del grado de Maestría en Ingeniería de este Instituto y después de haber sometido a revisión académica el proyecto de Tesis titulado: "Adaptación de trayectorias de un robot planar en un sistema de neuro rehabilitación motriz de pacientes con enfermedad cerebro vascular", realizado por el Ing. Luis Fernando Monrreal Arellano , No. De Control: G09050725, dirigida por el Dr. Juan Antonio Rojas Estrada, y habiendose realizado las correcciones que le fueron indicadas, acordamos **ACEPTAR** el documento final de proyecto de Tesis, por lo que le solicitamos tenga a bien extender el correspondiente oficio de autorización de impresión.

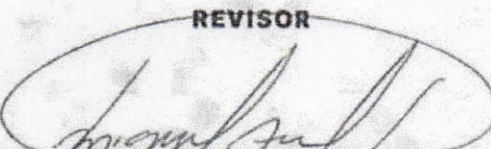
Sin otro particular, agradecemos la atención.

**ATENTAMENTE**  
Excelencia en Educación Tecnológica.  
"CIENCIA Y TECNOLOGÍA AL SERVICIO DEL HOMBRE"

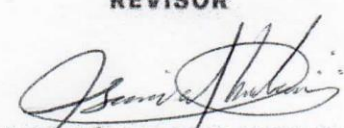
**DIRECTOR DE TESIS**

  
**DR. JUAN ANTONIO ROJAS ESTRADA**  
**DOCTOR EN INGENIERÍA ELÉCTRICA**  
**CÉDULA: 3482434**

**REVISOR**

  
**DR. MIGUEL ÁNGEL OCHOA VILLEGAS**  
**DOCTOR EN TECNOLOGÍAS DE INFORMACIÓN**  
**Y COMUNICACIÓN**  
**CÉDULA: 9386150**

**REVISOR**

  
**DR. MARIO CÉSAR OSORIO ABRAHAM**  
**DOCTOR EN INGENIERÍA FÍSICA**  
**INDUSTRIAL**  
**CÉDULA: 7576591**

C.c.p.- Expediente







"2020, Año de Leona Vicario, Benemérita Madre de la Patria"

Guadalupe Nuevo León, 8/Junio /2020

ING. LUIS FERNANDO MONRREAL ARELLANO  
CANDIDATO A OBTENER EL GRADO DE  
MAESTRÍA EN INGENIERÍA EN INGENIERÍA  
PRESENTE:

Después de haber atendido las recomendaciones sugeridas por la Comisión del Consejo de Posgrado de Ingeniería, en relación a su trabajo de Proyecto de Tesis, cuyo título es: **"Adaptación de trayectorias de un robot planar en un sistema de neuro rehabilitación motriz de pacientes con enfermedad cerebro vascular"**, me permito comunicarle que, conforme a los Lineamientos para la Operación de los Estudios de Posgrado, se le concede la autorización para que proceda con la impresión de su proyecto de Tesis.

Sin otro particular, agradecemos la atención.

**A T E N T A M E N T E**

Excelencia en Educación Tecnológica.  
"CIENCIA Y TECNOLOGÍA AL SERVICIO DEL HOMBRE"

DR. MARIO CÉSAR OSORIO ABRAHAM  
JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO.



SEP  
T N M                      S E S  
INSTITUTO TECNOLÓGICO  
DE NUEVO LEÓN  
DIV. DE ESTUDIOS DE  
POSGRADO E INVESTIGACIÓN

C.c.p.- Dr. René Sanjuan Galindo -Coordinador de la Maestría en Ingeniería  
C.c.p.- Expediente





“2020, Año de Leona Vicario, Benemérita Madre de la Patria”

**AUTORIZACIÓN PARA PRESENTAR EXAMEN DE GRADO DE MAESTRÍA**

Guadalupe Nuevo León, **8/Junio/2020**

**ING. LUIS FERNANDO MONRREAL ARELLANO  
PRESENTE:**

Por medio del presente me es grato comunicarle que **SE AUTORIZA** la presentación del examen de **Maestría en Ingeniería**, toda vez que ha cubiertos los requisitos necesarios.

Aprovecho la ocasión para desearle el mejor de los éxitos en su examen, así como en su vida profesional, y agradecerle la confianza depositada en nuestra institución para la realización de sus estudios de Posgrado.

**ATENTAMENTE**  
Excelencia en Educación Tecnológica  
“CIENCIA Y TECNOLOGÍA AL SERVICIO DEL HOMBRE”

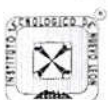


**DR. MIGUEL ÁNGEL OCHOA VILLEGAS  
SUBDIRECTOR ACADÉMICO**



SEP  
T N M      S E S  
INSTITUTO TECNOLÓGICO  
DE NUEVO LEÓN  
SUBDIRECCIÓN ACADÉMICA

C.c.p.- Dr. René Sanjuan Galindo. -Coordinador de la Maestría en Ingeniería  
C.c.p.- Expediente







Instituto Tecnológico de Nuevo León

"2020, Año de Leona Vicario, Benemérita Madre de la Patria"

Guadalupe Nuevo León, **21/enero/2020**

Asunto: Cesión de derechos

**DR. MARIO ALBERTO MARTÍNEZ HERNÁNDEZ**  
INSTITUTO TECNOLÓGICO DE NUEVO LEÓN  
DIRECTOR  
PRESENTE

Hago manifiesta la

**CESIÓN DE DERECHOS**

De la tesis Adaptación de trayectorias de un robot planar en un sistema de neuro rehabilitación motriz de pacientes con enfermedad cerebro vascular que fue dirigida por el Dr. Juan Antonio Rojas Estrada.

En cumplimiento a los requisitos de obtención del grado que señala el numeral 2.15.5 de los Lineamientos para la Operación de los Estudios de Posgrado en el Tecnológico Nacional de México.

**A T E N T A M E N T E**  
*Excelencia en Educación Tecnológica®*

**Luis Fernando Monrreal Arellano**  
Número de Control G09050725  
CANDIDATO AL GRADO DE MAESTRO DEL PROGRAMA MAESTRÍA EN INGENIERÍA

ccp. Archivo



## **Resumen**

En este proyecto se propone la simulación y aplicación de un robot de dos grados de libertad, el cual se usará en un sistema para la rehabilitación de pacientes que han sufrido un accidente cerebro vascular. El sistema consiste en la generación y aplicación de trayectorias que puedan adaptarse a los requerimientos establecidos. Mediante la aplicación de trayectorias, y por medio del robot, se realizan sesiones de terapias para el brazo del paciente que serán de utilidad para su rehabilitación. Comercialmente existen algunas plataformas las cuales por ser de alto costo no están disponibles en los centros de salud. En este trabajo se toma en cuenta el par de reacción del paciente y se refleja en las simulaciones del modelo. Se estudió el modelo dinámico del robot tomando en cuenta la fuerza que ejercía el paciente expresándolo en el modelo global. Se realizaron simulaciones utilizando un controlador clásico PID y en los resultados obtenidos, se observó el seguimiento de la trayectoria mostrando un cierto grado de error por el efecto de la reacción del paciente alrededor del 6% en comparación de lo previsto en la simulación mientras que la estimación de los pares se obtuvo un diferencia significativa de cerca del 15% de lo esperado.

## **Abstract**

This work proposes the simulation and application of a robot with two degrees of freedom, which will be used in a system for the rehabilitation of patients who have had a cerebral brain accident. The system consists of the generation and application of trajectories that can be adapted to the established requirements. Through the application of trajectories and by means of the robot, therapy sessions are carried out for the patient's arm that will be useful for his rehabilitation. Commercially there are some platforms which, due to their high cost, they aren't available at the health centers. In this work, the reaction torque of the patient that generate is taken into account and is reflected in the model. The dynamic model of the robot was studied taking into account the force exerted by the patient expressing in the global model. Simulations were carried out using a PID controller and in the results a tracking of the trajectory was achieved for a certain degree of error by the effect of the patient's reaction, around 6% compared to that predicted in the simulation, while the checking of the pairs obtained a significant difference of about to 15% than expected.

## **Agradecimientos**

Quiero agradecer primeramente a Dios por darme la oportunidad de realizar estos estudios.

A mi director de tesis el Dr. Juan Antonio Rojas Estrada por los conocimientos y guía que fueron invaluable en el desarrollo de este proyecto.

A mi comité por su apoyo durante la realización de este proyecto.

A mis padres y a mis hermanas por el apoyo durante esta etapa de mi vida.

A mis compañeros de la maestría por sus consejos y su amistad.

A todos los maestros y miembros del departamento de posgrado.

Y al consejo Nacional Ciencias y Tecnología (CONACYT) por su apoyo por medio de la beca No.889446 proporcionada para poder realizar este proyecto.

## Índice

Resumen.....	6
Abstract.....	6
Agradecimientos .....	7
<b>CAPÍTULO 1 .....</b>	<b>12</b>
1.1 Introducción .....	12
1.2. Planteamiento del problema .....	14
1.3. Objetivo General .....	15
1.3.1 Objetivos específicos.....	15
1.4. Justificación.....	16
1.5. Aportación del proyecto .....	17
<b>CAPÍTULO 2 .....</b>	<b>18</b>
2. Marco teórico .....	18
2.1. Robótica.....	18
2.1.1. Brazo robótico .....	20
2.1.1. Historia de los brazos robóticos.....	21
2.2 Cinemática .....	22
2.2.1. Cinemática directa .....	23
2.2.2. Cinemática Inversa.....	24
2.3 Dinámica .....	27
2.3.1 Modelo dinámico del manipulador .....	28
2.3.2 Modelo matemático para un motor DC .....	32
2.4 Controladores.....	33
2.4.2 Controlador del robot manipulador .....	36
2.4.1 Control de proporcional derivativo con retro de velocidad .....	38
2.4.3. Sintonización de PID .....	38
2.5 Generación de Trayectorias .....	39
2.5.2 Estabilidad del sistema .....	40
2.6 Rehabilitación con robots.....	41
2.7 Modelo completo dinámico manipulador-paciente .....	46
<b>CAPÍTULO 3 .....</b>	<b>47</b>
3. Método y materiales.....	47
3.1 Desarrollo de la plataforma experimental.....	47
3.1.1 Plataforma para el manipulador.....	48
3.1.2 Controlador.....	51
3.2 Cinemática del robot .....	52
3.2.1 Solución de la Cinemática directa .....	52
3.2.2 Solución de la cinemática inversa .....	53
3.3 Generación de trayectorias .....	53
3.3.1 Conversión de punto en el espacio a radianes.....	54



3.4 Cálculo de los momentos de Inercia .....	55
3.4.1 Centro de masas .....	55
3.4.1 Cálculo del momento de Inercia del eslabón .....	57
3.4.2 Estimación de Inercias del motor y carga .....	59
3.5 Estimación del coeficiente de fricción .....	59
3.6 Valores PID para motor DC .....	59
3.7 Datos para la Simulación. ....	61
3.7.1 Modelo Simulink®.....	61
3.8 Medición de Par mecánico.....	63
3.8.1 Sensor efecto hall.....	64
<b>CAPÍTULO 4 .....</b>	<b>65</b>
4. Resultados .....	65
4.1 Aplicación de PID a un motor de DC.....	65
4.2 Resultados de la simulación del manipulador .....	67
4.3 Resultados obtenidos de plataforma.....	72
<b>CAPÍTULO 5 .....</b>	<b>76</b>
9. Conclusiones.....	76
10. Recomendaciones .....	77
<b>REFERENCIAS.....</b>	<b>79</b>
<b>APÉNDICES.....</b>	<b>84</b>
Apéndice A. Programación.....	84
6.1.1 Programa principal .....	84
6.1.2 Programa para calibración del sensor ACS712.....	92
6.1.3 Programa Simulink® para función transferencia del motor .....	93
6.1.4 Programa Matlab® para generación de trayectoria polinómica.....	94
6.1.5 Programa Matlab® para solución de la cinemática directa.....	94
Apéndice B. Anexos de documentos .....	96
6.2.1 Valores de Eslabón en Solidworks®.....	96
6.2.2 ficha técnica GM9236S019.....	97
6.2.3 Distribución de terminales del microcontrolador .....	98
6.2.4 Constancia de participación en congreso.....	99

## Índice de figuras y tablas

Figura 1. Clasificación de los robots según su morfología _____	20
a) Móviles, b) Poli articulados, c) Androides, d) Zoomórficos, e) Híbridos _____	20
Figura 2. Esquema con los parámetros del robot de 2 GDL _____	26
Figura 3 Robot planar de 2 GDL (vista superior) _____	29
Figura 4. Circuito del motor de corriente directa. _____	32
Figura 5. Diagrama de bloques para control PID _____	36
Figura 6. Diagrama control de movimiento para robot manipulador _____	38
Figura 7. Tipos de robot para rehabilitación del miembro superior a) robot efector final y b) exoesqueleto _____	42
Figura 8. Esquema superior paciente/robot de 2GDL _____	46
Figura 9. Esquema de la plataforma experimental general _____	48
Figura 10. Dimensiones físicas de Motor DC _____	49
Figura 11. Puente H modelo IBT_2 _____	50
Figura 12. Sistema embebido Arduino Mega 2560 _____	51
Figura 13. Modulo SD Arduino _____	52
Figura 14. Matlab® desarrollo de cinemática directa _____	53
Figura 15. Esquemas en Matlab® para la conversión de la cinemática inversa _	55
Figura 16. Dimensiones de eslabón _____	56
Figura 17. Esquema en Matlab® para calcular valores PID _____	61
Figura 18. Esquemas de control en Matlab® para el robot de 2GDL _____	62
Figura 19. Datos del bloque función Matlab® para robot de 2 GDL _____	63
Figura 20. Conexiones para el sensor ACS712 _____	64
Figura 21. Respuesta del sistema al aplicar control PID utilizando los parámetros Ziegler-Nichols. _____	65
Figura 22. Respuesta al aplicar control PID utilizando los parámetros Ziegler-Nichols ajustados _____	66
Figura 23. Respuesta al aplicar control PID utilizando el sintonizador de Simulink® _____	66
Figura 24. Gráfica de trayectoria en el plano XY en metros _____	67
a) Trayectoria polinómica, b) línea con origen (0,0.3) y c) círculo de radio 0.1m _	67
Figura 25. Gráfica de posiciones en radianes _____	68
a) Trayectoria polinómica, b) línea con origen (0,0.3) y c) círculo de radio 0.1m _	68
Figura 26. Gráfica del error de posición _____	69
a) Trayectoria polinómica, b) línea con origen (0,0.3) y c) círculo de radio 0.1m _	69
Figura 27. Gráfica (de) pares aplicados por el manipulador $\tau_i$ _____	70
a) Trayectoria polinómica, b) línea con origen (0,0.3) y c) círculo de radio 0.1m _	70
Figura 28. Gráfica (de) pares aplicados por el manipulador $\tau_i$ y los pares del paciente $\tau_{pi}$ _____	71
a) Trayectoria polinómica, b) línea con origen (0,0.3) y c) círculo de radio 0.1m _	71
Figura 29. Gráfica de trayectoria obtenida de robot en plano XY _____	72
Figura 30. Gráfica de seguimiento de trayectoria $q_i$ en comparación de robot _	73
a) Trayectoria polinómica, b) línea con origen (0,0.3) y c) círculo de radio 0.1m _	73

Figura 31. Gráfica estimada de pares aplicados por el manipulador $\tau_{pi}$ _____	74
a) Trayectoria polinómica, b) línea con origen (0,0.3) y c) círculo de radio 0.1m _	74
Figura 32. Gráfica estimada de pares aplicados por el paciente $\tau_{pi}$ _____	75
a) Trayectoria polinómica, b) línea con origen (0,0.3) y c) círculo de radio 0.1m _	75

<i>Tabla 1 Criterios de sintonización de Ziegler y Nichols para el método de la última ganancia.</i> _____	39
<i>Tabla 2 Parámetros Denavit-Hartenberg</i> _____	53
<i>Tabla 3 Valores dimensionales de secciones</i> _____	56
<i>Tabla 4 Cálculos de inercia para complementos</i> _____	57
<i>Tabla 5 Datos usados para el cálculo del momento de inercia</i> _____	58
<i>Tabla 6 Valores del motor de DC</i> _____	60
<i>Tabla 7 Datos para simulación de un robot de 2GDL</i> _____	61

## Capítulo 1

### 1.1 Introducción

Actualmente en México ocurren problemas de enfermedad cerebro vascular a una razón de 118 por cada 100,000 habitantes al año. Se conoce como enfermedad vascular cerebral (EVC) a una alteración en las neuronas, que provoca disminución de flujo sanguíneo en el cerebro, acompañada de alteraciones cerebrales de manera momentánea o permanente. Una de cada 3 personas presenta algún grado de discapacidad después de sufrir una EVC [1].

Se ha demostrado por medio de varios estudios que los pacientes afectados por las EVC pueden mejorar sus habilidades realizando una apropiada secuencia de rehabilitación [2]. Una sesión de terapia consiste en repeticiones de algunos movimientos que permiten al usuario fortalecer el área afectada. Estos ejercicios deben realizarse diariamente; sin embargo, esto no siempre es posible debido a la demanda en aumento y la poca capacidad dentro de las instituciones, lo que ocasiona que el paciente recibe dos o tres sesiones por semana como máximo [3].

Para llevar a cabo estas sesiones de terapia es necesario la asistencia del fisioterapeuta, quien es el especialista que se encarga de la recuperación óptima de las secuelas que puede obtenerse de algún accidente; el especialista puede ir dejando que el paciente haga los movimientos completamente solo, ayudar a realizar el ejercicio o incluso asistirlo completamente. Existen diferentes tipos de ejercicios de rehabilitación, uno de estos tipos es, los ejercicios de fortalecimiento, estos incrementan la cantidad de fuerza que el músculo puede generar; los cuales son el ejercicio isométrico en el cual la tensión muscular es constante y no hay necesidad de movimiento articular, el ejercicio isotónico es realizado con una carga constante pero sin controlar la velocidad y por último el ejercicio isocinético en el cual se mantiene el movimiento constante pero se varía la fuerza en respuesta a la aplicada [4].

Desde hace varios años, diversos tipos de robots se han comenzado a utilizar en el área de medicina como apoyo en la rehabilitación para pacientes con este tipo de



accidentes, estos replican los movimientos que usan los fisioterapeutas. Los robots en la rehabilitación, presentan la ventaja de que pueden generar un número elevado de repeticiones de manera sencilla, además de la capacidad de almacenamiento de información objetiva sobre el estado del paciente [5].

Existen dos tipos de robots o dispositivos que se usan en el apoyo de rehabilitaciones, los primeros dispositivos están hechos para alinear el efector final o herramienta, con el usuario (como el que se maneja en este proyecto). El otro tipo de robot es el que alinea mecánicamente sus articulaciones con las articulaciones del usuario (exoesqueletos), estas son externas al cuerpo y proporcionan fuerza para el movimiento del usuario a una posición dada [6].

Todos los robots usados en procesos de neuro rehabilitación además de manejar el mismo principio de estrategias de control para la posición, deben considerar la interacción entre el robot y el paciente. Para ello son necesarios modelos matemáticos con el fin de entender la dinámica del sistema, así como una solución para la cinemática del manipulador. Es también necesario determinar la fuerza con la que interactúan el robot y el paciente durante la secuencia de la terapia. Es necesario tomar en cuenta estos indicadores dentro del algoritmo de control para una correcta ejecución de la fuerza y su localización [7].

Existen actualmente plataformas comerciales que manejan el control para estos algoritmos. Entre los sistemas robóticos comerciales para rehabilitación se encuentran el Arm-guide es un dispositivo el cual ayuda al movimiento del brazo del usuario por una guía asistiéndolo con un servomotor, MIT manus es uno de los más estudiado en rehabilitación el cual esta enfocado al brazo y hombro este posee una interfaz gráfica que permite al usuario moverse de acuerdo a la ayuda visual proporcionada, además de poseer un algoritmo capaz de adaptarse de acuerdo a la información recolectada, Bi-ManuTrack el cual está más enfocado a la muñeca y el antebrazo tiene un sistema capaz de ajustarse a las necesidades del paciente [8]. Sin embargo, estas plataformas solamente dan el seguimiento al tratamiento posterior, no cuando el usuario esta interactuando con el sistema. Otro punto destacable de dichas plataformas es el precio elevado para su adquisición.

## 1.2. Planteamiento del problema

Hoy en día existen plataformas en neuro rehabilitación motriz del tipo cerradas, esto quiere decir que el diseño no puede ser examinado por programadores ajenos al equipo de desarrollo, aunque en la gran mayoría de sus casos, no se sabe qué tipo de controlador tienen para poder establecer trayectorias y fuerza recomendadas por los terapeutas. En este sentido, la rehabilitación se hace primeramente a nivel de movimiento (cinemático) y posteriormente a nivel fuerza (dinámico).

Una ventaja importante de estos sistemas robóticos es que pueden usar estos dos parámetros para calificar el progreso del paciente de forma más objetiva [9], sin embargo, no todas las plataformas pueden aportar la parte dinámica, esto debido a que los actuadores no están controlados en modo par, es decir, no se puede pedir que el sistema robótico ejecute las tareas a diferentes niveles de fuerza [10].

Hasta ahora existen algunas plataformas desarrolladas por investigadores como Mani [11], Elliott [12] o Schmidt [13], no están disponibles de forma comercial, sin embargo para la mayoría de estas no se toma en cuenta la fuerza de interacción entre el sistema robótico y el usuario por lo que se pueden considerar como modelos incompletos. Se realizan los modelos matemáticos (cinemático y en pocas ocasiones incluyen el dinámico) pensando en que son cadenas robóticas abiertas sin tomar en cuenta las fuerzas que se oponen al sistema robótico [14]. Algunos autores como Hasegawa [15], Wehner [16] y Ulrey [17], que tratan este tema de interacción de fuerzas, tienen en cuenta que la fuerza que el sujeto impone, puede ser considerada como una perturbación del sistema robótico, por lo que el control automático del sistema robótico debe de cumplir los requerimientos que se le pidan a pesar de la perturbación (fuerza del sujeto) [18].

Por todas estas razones, la propuesta de este trabajo aborda un sistema que complementa las cuestiones que no están tomadas en cuenta.

### **1.3. Objetivo General**

Adaptar y simular un sistema de trayectorias en un robot planar para la neuro rehabilitación motriz de pacientes con enfermedad cerebro vascular y con capacidad de definir el nivel de mejora del paciente.

#### **1.3.1 Objetivos específicos**

-Obtener la cinemática directa e inversa del robot planar a través de los parámetros de Denavit-Hartenberg.

-Implementar esquemas de control en el robot manipulador para seleccionar el mejor desempeño dinámico.

-Implementar el algoritmo de generación de trayectorias para el robot.

-Realizar pruebas experimentales para la evaluación del prototipo de plataforma que trabajará con el robot.

#### **1.4. Justificación**

Todas las terapias físicas pueden apoyar beneficiosamente la recuperación de las personas que han sufrido un accidente cerebro vascular, pero a su vez significa una inversión de recursos, como el tiempo en supervisar toda la terapia, y normalmente es necesaria su asistencia médica a largo plazo. En México el proceso de rehabilitación se lleva a cabo en las instituciones de salud de cada comunidad y, por lo general, estas se ven rebasadas en su capacidad; algunos estudios muestran que el uso de estos tratamientos iniciados tempranamente, por lo general las primeras semanas, obtienen mejores resultados para la rehabilitación del paciente además que es necesario un seguimiento constante durante un número significativo de semanas posteriores. Por lo que es indispensable el apoyo a los terapeutas para de esta forma tener un mayor alcance del que se tiene actualmente; debido a esto el uso de robot dentro de esta área ha estado tomando relevancia durante los últimos años con la idea de aumentar el número de beneficiados además que se pretende contribuir de manera sustancial a la reducción del coste económico en la rehabilitación.

De esto parte la búsqueda de algunas alternativas con el análisis de manipuladores para el apoyo en rehabilitación debido al incremento que se tiene estimado de casos en los siguientes años, además de que los manipuladores que actualmente se tienen en el mercado pueden ser difíciles de adquirir debido al alto precio.

Por lo tanto se pretende desarrollar una plataforma experimental donde se contemplen ambas partes; la cinemática y la dinámica, tanto la que involucra al manipulador como también las fuerzas externas que provienen del esfuerzo por parte del paciente para la aplicación en neuro rehabilitación motriz mediante el sistema propuesto. Lo cual tendría como un beneficio una retroalimentación en el momento de la sección de terapia; Otro punto a considerar como beneficioso es la capacidad de recolección de información de acuerdo con el seguimiento de la trayectoria, ya que de esta forma se puede observar como el paciente se comporta durante el avance de las secciones de rehabilitación la cual puede ser examinada posteriormente por el especialista.



## **1.5. Aportación del proyecto**

Este proyecto tiene como finalidad atacar dos problemáticas principales, primeramente el desarrollo de un modelo matemático para el manipulador de dos grados y la posterior implementación del control de este. La segunda problemática es la de la lectura de los pares de las diferentes fuerzas que se ejercen sobre nuestro manipulador al momento de realizar el seguimiento de una trayectoria.

La primera problemática se pretende resolver mediante el desarrollo de ecuaciones matemáticas que describan el sistema, así como de diferentes soluciones aplicables. Una vez obtenidas las ecuaciones se hará uso del software Matlab®. De manera que con esta información se resolverá a través del diseño, simulación y experimentación, un control clásico Proporcional-Integral-Derivativo (PID) aplicado a un sistema físico.

El segundo problema se plantea resolver mediante la medición y obtención de los datos referentes de la corriente de los motores. Estos se utilizarán en el cálculo de los pares ejercidos tanto por el manipulador como del paciente. Los pares de los motores del manipulador en vacío y con el paciente, se obtienen de manera separada para después mediante una diferencia entre ellos, es posible determinar los pares que ejerce el paciente. Estos pueden ser útiles en trabajos posteriores en la adaptación de las trayectorias.

## **Capítulo 2**

### **2. Marco teórico**

En este capítulo se analizarán varios aspectos que involucran el desarrollo de un robot manipulador partiendo de estudios previos que se tienen. Se analizaron distintas técnicas para realizar la formulación matemática de la descripción general del robot manipulador, así mismo se realizó el análisis de algunos conceptos que se estarán utilizando en capítulos posteriores.

Para el caso particular del robot manipulador con el que se plantea realizar la experimentación, este está compuesto de dos eslabones rígidos. Se puede utilizar el término planar para este robot debido a que los eslabones que se manejan solo son movidos dentro del eje  $X$  y eje  $Y$ , por lo que las ubicaciones quedan dentro del plano  $XY$ . El término rígido se puede definir como que no cambian el tamaño de sus dimensiones y no tienen flexión.

Dentro de este capítulo también se examinará la cinemática del robot manipulador, que puede ser definida como el estudio del movimiento. Dicho estudio abarcará el codo y el hombro en un solo plano, también se verificará el estudio del modelo matemático para las fuerzas que se están aplicando para el manipulador así como también se revisará el enfoque planteado en la descripción, simulando la fuerza del paciente, analizando la teoría de control y también los conceptos referentes a la estabilidad.

#### **2.1. Robótica**

La Robótica es la ciencia y la tecnología de los robots. Se ocupa del diseño, manufactura y aplicaciones de los robots. La robótica combina diversas disciplinas como son: la mecánica, la electrónica, la informática, la inteligencia artificial y la ingeniería de control. Otras áreas importantes en robótica son el álgebra, los autómatas programables y las máquinas de estados [19].

La definición de robot dada por el Instituto de Robot de América (por sus siglas en inglés RIA) es:

Un manipulador reprogramable y multifuncional concebido para transportar materiales, piezas, herramientas o sistemas especializados: con movimientos variados y programados, con la finalidad de ejecutar tareas diversas [20].

Anteriormente los robots no eran más que herramientas para la automatización. Estaban teóricamente destinados a ejecutar una tarea específica: transportar, cargar, descargar, soldar, atornillar, pintar, pegar, pulir, limpiar, etcétera. Estas tareas no implicaban de ninguna manera que los robots fueran “inteligentes”.

Además de los robots que permiten la manipulación automática y programable, existen los robots llamados “inteligentes” que efectúan funciones tales como la detección de cualquier modificación de su medio ambiente. Estos robots “inteligentes” actúan en consecuencia considerando las nuevas modificaciones ya sea cambiando la secuencia de operaciones o descubriendo una nueva [21].

Los dispositivos y mecanismos pueden agruparse bajo una denominación genérica del robot y son muy diversos. Establecer una clasificación coherente de los mismos que resista un análisis crítico y riguroso, es muy complicado. La subdivisión de los robots, desde su aspecto morfológico, se puede apreciar en la Figura 1:

- a) Móviles. Los robots que son capaces de moverse de manera autónoma sobre una superficie y poseen una gran capacidad de desplazamiento. Están basados en sistemas locomotores del tipo rodante, suelen guiarse por la información que reciben de su entorno.
- b) Poli articulados. Los robots con una amplia capacidad de movimiento en sus extremidades para tomar o mover objetos.
- c) Androides. Se distinguen por su apariencia similar a la humana y que por esta misma configuración puede imitar ciertas acciones que realiza el hombre.
- d) Zoomórficos. Robots que imitan la locomoción de diversos seres vivos, y a su vez estos se dividen en caminadores y no caminadores.
- e) Híbridos. Aquellos robots que presentan una mezcla de dos o más de las morfologías anteriores por lo que se hace difícil su clasificación [22].

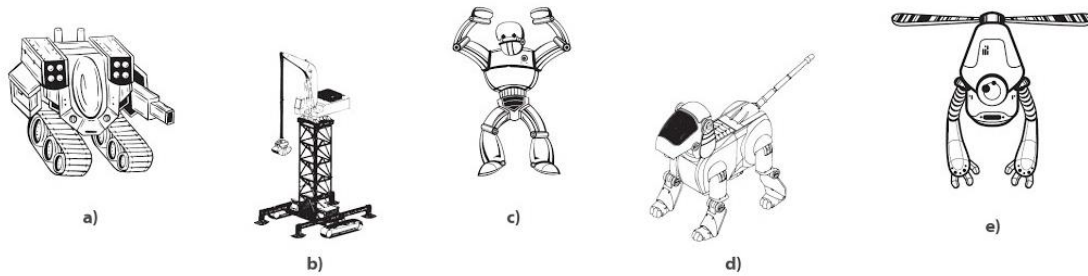


Figura 1. Clasificación de los robots según su morfología  
 a) Móviles, b) Poli articulados, c) Androides, d) Zoomórficos, e) Híbridos  
 (Programación de Sistemas Digitales con VHDL, David G. Maxiez)

Los robots poli articulados: son un grupo de robots de muy diversa forma y configuración, cuya característica común es la de ser básicamente sedentarios (aunque excepcionalmente pueden ser guiados para efectuar desplazamientos limitados) [23]. Se caracterizan por estar estructurados para mover sus elementos terminales en un determinado espacio de trabajo según uno o más sistemas de coordenadas, y con un número limitado de grados de libertad.

### 2.1.1. Brazo robótico

Un brazo robótico se puede definir como un conjunto de elementos rígidos llamados eslabones, que propician el movimiento de un elemento terminal (efector final), los cuales están conectados por articulaciones que permiten el movimiento de los eslabones adyacentes.

También se puede definir como un brazo que intenta simular los movimientos humanos. Está constituido por elementos estáticos (eslabones), los cuales se encuentran conectados por articulaciones que permiten el movimiento entre ellos de forma continua.

El número de grados de libertad (GDL) es un concepto importante relacionado con la movilidad de los robots. Se define como el número de parámetros independientes que fijan la situación del órgano terminal, aunque este suele coincidir con el número de eslabones de la cadena cinemática, esto puede entenderse como un número de



variables independientes que se especifican para localizar todas las piezas del mecanismo [24].

Un robot requiere establecer la posición y la orientación del efector final para ejecutar cualquier tarea, esto es necesario para resolver problemas de posicionamiento.

Los cálculos para realizar los movimientos de acuerdo a una posición final deseada son realizados por medio de una computadora que previamente analiza los puntos necesarios que tienen que alcanzar las articulaciones para generar las trayectorias. Para ello es necesario entender la forma de indicar al robot planar las ubicaciones que debe tomar teniendo en cuenta dos puntos principales: la cinemática directa y la cinemática inversa.

Cabe destacar que es necesario el uso de sensores anexados al robot para su apoyo en validar la ubicación actual, dada su localización de forma continua. El incremento en la precisión con el uso de sensores rotacionales, puede aumentar excesivamente el costo del sistema. En todo caso el uso de estos dispositivos hace necesario un controlador.

En cualquier caso, se debe de tener en cuenta la norma ISO / TS 15066 ya que ésta especifica los requisitos para los sistemas de seguridad del robot que están en contacto con operadores.

### **2.1.1. Historia de los brazos robóticos.**

A continuación, se anexa una breve historia de los brazos robóticos:

En 1954, George Devol y Joe Engleberger desarrollaron el primer brazo robótico programable y usaron el término “automatización universal” (Universal Automation) por primera vez, que daría el nombre a la empresa que fundaron en 1954 Unimation (Universal Automation), siendo esta la primera compañía de robots[25].

En los laboratorios del Instituto Tecnológico de Massachusetts (MIT), en 1956 se desarrolló el lenguaje de programación llamado “Automatically Programmed Tools” (APT) que actualmente es utilizado en la manufactura asistida por computadora [26].

El primer brazo puesto al mercado fue lanzado por la American Machine and Foundry Company, el cual fue llamado Versatran y diseñado en 1960 por Harry Johnson y Veljko Milenkovic.

Heinrich Emst desarrolla en 1961, un brazo robótico que era controlado por una computadora para reconocer bloques y apilarlos sin ayuda de ningún operario. Poco tiempo después Tomovic y Bon diseñaron un brazo capaz de detectar el peso y tamaño de un objeto. Y en 1962 se instala por primera vez un robot industrial en la compañía General Motors que se encargaba de las tareas repetitivas o peligrosas [27].

El ingeniero mecánico estadounidense Victor Scheinman, en 1975 desarrolló un manipulador polivalente realmente flexible conocido como Brazo Manipulador universal Programable (PUMA, por sus siglas en ingles). El PUMA fue capaz de mover un objeto y colocarlo en cualquier orientación en un lugar deseado que estuviera a su alcance. Este concepto ha pasado a ser ampliamente utilizado en los robots industriales [28].

Takeo Kanade construye el brazo de accionamiento directo en 1981. Es el primero en tener motores instalados directamente en las articulaciones del brazo. Este cambio hace que sea más rápido y mucho más precisos que los anteriores brazos robóticos [29].

En 1991, un nuevo robot llamado MIT-MANUS fue presentado como un banco de pruebas para estudiar el potencial de usar robots para ayudar y cuantificar la neuro-rehabilitación de función motora. MIT-MANUS inició de mano de Neville Hogan junto con Igor Krebs demostrando un ajuste excelente para la rehabilitación de hombro y codo de pacientes con accidente cerebrovascular con resultados en ensayos clínicos que muestran una reducción del deterioro en estas articulaciones [30].

## **2.2 Cinemática**

La cinemática es la rama de la física que estudia el movimiento de sólidos rígidos o flexibles, sin tener en cuenta las fuerzas que lo provocan. Esta disciplina juega un

rol fundamental en robótica, en donde se plantean dos problemáticas a resolver: la cinemática directa y la inversa [31].

### **2.2.1. Cinemática directa.**

Se puede definir la cinemática directa como las coordenadas articulares con base en las cuales quedará determinada la posición del extremo del robot. Existen algunas metodologías para su solución siendo la más conocida aquella que ocupa los parámetros Denavit-Hartenberg.

El método Denavit-Hartenberg es un método de procedimiento sencillo para conseguir la cinemática directa, el cual consiste en una estructura en términos de las transformaciones homogéneas. Fue implementado por Jacques Denavit y Richard S. Hartenberg en 1955 con la intención de representar la traslación y orientación de los robots; Esto se hace por medio de una tabla de parámetros que relaciona los eslabones del robot [32].

La cinemática directa parte de la conversión de la ubicación de la posición angular de los eslabones de acuerdo a los actuadores, a la ubicación del efector final en un punto en el espacio en el plano XYZ.

Para el desarrollo de la cinemática directa del robot manipulador se utilizaron los parámetros Denavit-Hartenberg para su posterior desarrollo. Los cuales corresponde  $\alpha_{i-1}$  como el ángulo del vector  $Z_{i-1}$  a  $Z_i$  alrededor de  $X_{i-1}$ ,  $a_{i-1}$  que es la distancia de  $Z_{i-1}$  a  $Z_i$  a lo largo de  $X_{i-1}$ ,  $d_i$  es la distancia  $X_{i-1}$  a  $X_i$  a lo largo de  $Z_i$  y por último  $\theta_i$  que es el Angulo de  $X_{i-1}$  a  $X_i$  alrededor de  $Z_i$ . Cada uno de estos parámetros es calculado al comparar los marcos de cada eslabón con referencia al anterior.

Posteriormente estas variables son suplantadas dentro de la matriz expresada en la ecuación 1, para expresar la rotación entre marcos, y se da el valor constante de los ángulos del eslabón según se indique.

$$A_i^{i-1} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \cos\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Cada una de estas matrices de rotación es multiplicada para conseguir la matriz final, para el caso particular de este robot son dos marcos y el del origen. Se multiplican ambas matrices, quedando como en la ecuación 2.

$$T_2^0 = A_1^0 A_2^1 \quad (2)$$

De esta matriz final se obtendrá la ubicación del efector final ingresando los ángulos correspondientes por lo que estos se resolverán como puntos en el espacio para cada uno de los vectores siendo estos  $P_x$ ,  $P_y$  y  $P_z$ , quedando como en la ecuación 3.

$$T_2^0 = \begin{bmatrix} r_{11} & r_{21} & r_{31} & P_x \\ r_{12} & r_{22} & r_{32} & P_y \\ r_{13} & r_{23} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

### 2.2.2. Cinemática Inversa.

Se calcula el valor que tienen que adoptar las coordenadas articulares para alcanzar una posición y orientación determinada en el extremo del robot. Este se considera un problema más complicado que la cinemática directa, ya que existen diversas soluciones o puede ocurrir el caso de que no exista una solución única.

Se considera que la cinemática inversa es un problema no lineal ya que relaciona la orientación del efector final del robot y las coordenadas articulares en función de las coordenadas cartesianas.

Está basada en el principio de que un movimiento ejerce una influencia sobre otro movimiento, en el punto de articulación, de esta forma permite crear movimientos de forma sencilla que sin la cinemática inversa serían muy complejos [33].

Existen dos métodos tradicionales en la obtención de las soluciones:

**Método Geométrico:** Este método puede ser denominado de forma gráfica, ya que consiste en la descomposición de la geometría espacial del robot en diferentes de

problemas de geometría plana, resueltos por aplicación de los conceptos como el del teorema de Pitágoras e identidades trigonométricas, este método tiende a complicarse su visualización entre mayor sea el número de grados de libertad[34].

Método Analítico: Está basado en la estructura de la matriz homogénea que toma desde el punto inicial hasta el deseado  ${}^0T_n = T_d$ . Es ideal para robots con más de tres grados de libertad, los ángulos de las uniones dependen de otros ángulos, sin embargo, su mayor ventaja es que entrega todas las soluciones para las articulaciones [35].

Al utilizar cualquiera de estos dos métodos, es necesario conocer la posición de los ángulos de las articulaciones del robot manipulador. Todos ellos en las diferentes posiciones que se desean alcanzar durante una trayectoria, si es que deseamos conocer la ubicación del efector final en el plano XYZ.

El desarrollo de la cinemática inversa fue realizado mediante el método geométrico. Se consideró este método como la mejor opción debido a que el robot con el que se trabajó es uno de efector final, el cual solo posee dos grados de libertad por lo que no es necesario un desarrollo complejo para obtener la solución. En la Figura 2 se muestran los parámetros del robot de 2 GDL.

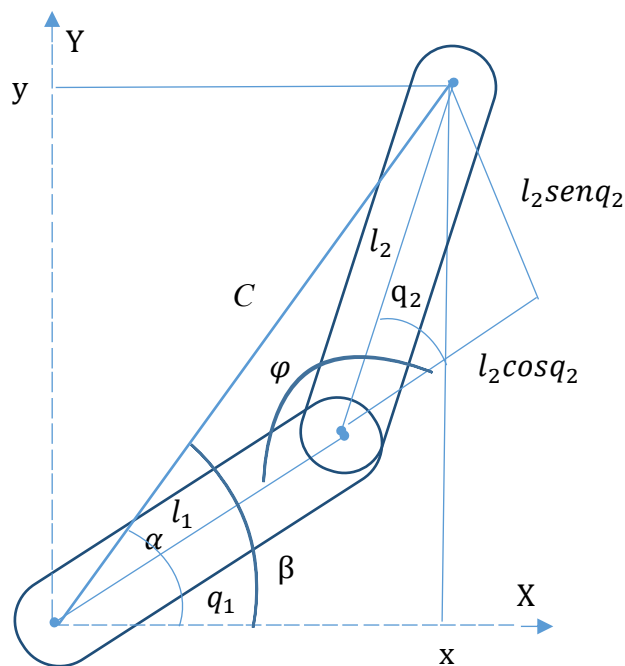


Figura 2. Esquema con los parámetros del robot de 2 GDL

Para su análisis, [32] se aplicó primeramente la ley de los cosenos, partiendo de la fórmula expresada en la ecuación (4).

$$C^2 = l_1^2 + l_2^2 - 2l_1l_2\cos\varphi \quad (4)$$

Siendo  $\varphi$  expresado por la ecuación (5).

$$\varphi = 180 - \theta_2 \quad (5)$$

Esta puede quedar expresada como (6).

$$\cos\varphi = \cos(180 - \theta_2) \quad (6)$$

El cual se puede reducir aplicando la identidad trigonométrica  $\cos(x + y) = \cos x \cos y + \sin x \sin y$  quedando expresada en la ecuación (7).

$$\cos\varphi = -\cos\theta_2 \quad (7)$$

Sustituyéndolo en la ecuación queda expresado en la forma

$$C^2 = l_1^2 + l_2^2 - 2l_1l_2(-\cos\theta_2) \quad (8)$$

Siendo  $C^2 = x^2 + y^2$  se sustituye en la ecuación anterior, por lo que la nueva ecuación queda reescrita en (9)

$$x^2 + y^2 = l_1^2 + l_2^2 + 2l_1l_2\cos\theta_2 \quad (9)$$

Sustituyendo  $D = \cos\theta_2$  dentro de la ecuación (9) y despejando este valor se obtiene la ecuación (10).

$$D = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1l_2} \quad (10)$$

Después partiendo de la identidad trigonométrica en (11)

$$\sin^2\theta_2 + \cos^2\theta_2 = 1 \quad (11)$$

Y tomando en cuenta (10) podemos llegar a la ecuación para el segundo ángulo quedando expresada en la ecuación (12).

$$\theta_2 = \operatorname{tg}^{-1}\left(\frac{\sqrt{1-D^2}}{D}\right) \quad (12)$$

Mientras que para la solución del primer ángulo partimos de las ecuaciones (13) y (14).

$$X' = l_2 \cos \theta_2 \quad (13)$$

$$Y' = l_2 \sin \theta_2 \quad (14)$$

Tomando en cuenta la expresión (15).

$$\operatorname{tg} \beta = \frac{Y'}{l_1 + X'} = \frac{l_2 \sin \theta_2}{l_1 + l_2 \cos \theta_2} \quad (15)$$

Al despejar (15) se obtiene la expresión (16)

$$\beta = \operatorname{tg}^{-1} \left( \frac{l_2 \sin \theta_2}{l_1 + l_2 \cos \theta_2} \right) \quad (16)$$

Y como el valor del ángulo  $\alpha$  se expresa en la ecuación (17).

$$\alpha = \operatorname{tg}^{-1} \left( \frac{y}{x} \right) \quad (17)$$

La formulación para el primer ángulo quedaría como

$$\theta_1 = \alpha - \beta \quad (18)$$

Sustituyendo ambos valores con (16) y (17) se obtiene la ecuación (19)

$$\theta_1 = \operatorname{tg}^{-1} \left( \frac{y}{x} \right) - \operatorname{tg}^{-1} \left( \frac{l_2 \sin \theta_2}{l_1 + l_2 \cos \theta_2} \right) \quad (19)$$

De esta forma al aplicar esta ecuación dando valores de posición en el plano  $XY$  podemos obtener la posición angular para ambos eslabones.

### 2.3 Dinámica

Se puede definir la dinámica como una rama de la mecánica que involucra los movimientos de los cuerpos por efecto de ciertas fuerzas. Estas pueden incluir el estudio de la cinemática, el estudio del movimiento sin tomar en cuenta las fuerzas que lo afectan, y la cinética que estudia la relación del movimiento resultante por efecto de la fuerza aplicada [36].

En cuanto a la dinámica del robot, está fundamentada en la formulación de un modelo matemático que intenta describir las fuerzas que actúan sobre el movimiento de sus partes móviles, el cual entre más intente apegarse a la realidad, se volverá más complejo debido a la mayor cantidad de variables que tiene que tomar en



cuenta. Una vez que se han desarrollado y probado dichos modelos, estos pueden ser útiles para actividades que van desde la simulación hasta el diseño de algoritmos para el control.

El modelo dinámico del robot se puede obtener por leyes físicas, como lo son la mecánica Newtoniana y Lagrangiana, de aquí se puede partir al desarrollo de ecuaciones de movimiento dinámico para las diversas articulaciones del manipulador, en términos de los parámetros geométricos e inerciales de los elementos.

La desventaja del primero de estos métodos es la complejidad del análisis, que se incrementa con el número de eslabones en el robot. En tal caso es mejor el uso de las ecuaciones de movimiento. Nombradas así por el matemático Francés Joseph Louis de Lagrange, quien fue el primero en realizar aportaciones en este ámbito, en su trabajo titulado: "Mecánica analítica" [37].

Esta metodología conocida como técnica Lagrangiana, se considera como una de las técnicas más comunes para la descripción de las ecuaciones de movimiento, la cual se basa en dos importantes conceptos; la energía potencial y la energía cinética. Estos conceptos se utilizan para describir el comportamiento de las fuerzas que se aplican directamente al sistema.

El modelo dinámico puede ser definido como una relación entre el movimiento y las fuerzas aplicadas al robot. Esto representa también el modelo matemático del robot.

### **2.3.1 Modelo dinámico del manipulador**

Como resultado del análisis del sistema se obtienen la velocidad, la posición y la aceleración con respecto a la aplicación de los pares de mecánicos.

Para la formulación matemática del modelo dinámico se requieren los parámetros del robot. En la Figura 3 se muestra el concepto del robot planar en una vista superior, del cual se obtendrá el modelo dinámico. Sean  $l_1$  y  $l_2$  las longitudes de los eslabones con  $m_1$  y  $m_2$  las respectivas masas de cada uno. Las distancias del centro de masas están dadas como  $l_{c1}$  para el primer eslabón y  $l_{c2}$  para el

segundo,  $I_1$  e  $I_2$  representan los momentos de inercia respectivamente para cada eslabón, estas se dan debido a la forma y material del que está hecho cada eslabón. Y por último se tienen las posiciones articulares  $q_1$  y  $q_2$ .

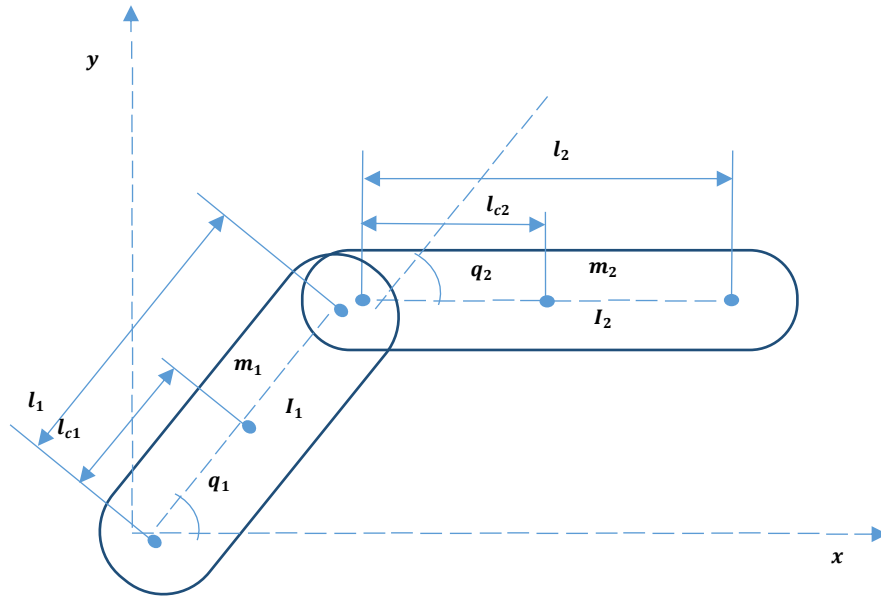


Figura 3 Robot planar de 2 GDL (vista superior)

Los otros parámetros son las matrices de inercia ( $I_i$ ) que están relacionadas al movimiento rotacional alrededor de un eje simple de un cuerpo rígido, esta se obtiene dependiendo del material y sus dimensiones y forma.

En cualquier instante cada eslabón de un robot en movimiento tiene algo de velocidad lineal y velocidad angular. Estas velocidades pueden ser sumadas cuando ambas son escritas en el mismo marco.

El modelo obtenido se puede utilizar para diferentes aplicaciones, como determinar el comportamiento del sistema por medio de simulación o para realizar un análisis general del sistema.

Para la obtención del modelo matemático en este caso, se usará la formulación Euler-Lagrange para el caso particular del robot de 2 GDL quedando como resultado el par mecánico generado por los actuadores, representado por la ecuación (20).

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}} \right) - \left( \frac{\partial L}{\partial \theta} \right) = \tau \quad (20)$$

Para este método se requiere el Lagrangiano que se define como la diferencia entre la energía cinética ( $K$ ) y la energía potencial ( $U$ ), descrita por la ecuación (21).

$$L = K - U \quad (21)$$

La energía cinética viene dada por la sumatoria de cada una de estas fuerzas de cada eslabón y se expresa por la ecuación (22).

$$K = \sum_{i=1}^n K_i \quad (22)$$

Para lo cual  $K_i$  esta expresado por la ecuación (23).

$$K_i = \frac{1}{2} m_i v_{ci}^T v_{ci} + \frac{1}{2} w_i^T I_i^i w_i \quad (23)$$

Siendo  $m_i$  la masa de cada eslabón y  $v_{ci}$  la velocidad lineal del centro de masa del eslabón perteneciente al primer término, mientras que el segundo término está dado por  $w_i$  que es el vector de la velocidad angular y por la matriz  $I_i$  que es la matriz simétrica de inercias y que se expresa en la ecuación (24).

$$I = \begin{bmatrix} I_{11} & I_{12} & I_{13} \\ I_{12} & I_{22} & I_{23} \\ I_{13} & I_{23} & I_{33} \end{bmatrix} \quad (24)$$

La otra parte, que sirve para el cálculo del Lagrangiano, es la energía potencial la cual se expresa por la ecuación (25).

$$U = \sum_{i=1}^n U_i \quad (25)$$

Para lo cual  $U_i$  se encuentra expresada en la ecuación (26).

$$U_i = -m_i^0 g^T P_{ci} + u_{ret} \quad (26)$$

Mientras  $m_i$  se expresa como la masa de cada eslabón,  $P_{ci}$  es el vector de localización del centro de masas. Por último  $g$  es el vector de la gravedad.

Aunque es necesario comprender las generalidades, para este caso en particular de robot tipo planar y de operación horizontal, la gravedad no interviene. Como

consecuencia la energía potencial es cero y el Lagrangiano consiste solamente del término de la energía cinética y que se expresa en la ecuación (27).

$$L = \frac{1}{2}\dot{q}_1^2(l_{c_1}^2 m + l_1^2 m_2) + \frac{1}{2}m_2 l_{c_2}^2 (\dot{q}_1 + \dot{q}_2)^2 + m_2 l_1 l_{c_2} (\dot{q}_1 + \dot{q}_2) q_1 \cos q_2 + \frac{1}{2}(I_1 \dot{q}_1^2) + I_2 (\dot{q}_1 + \dot{q}_2)^2 \quad (27)$$

Así que una vez aplicada la formulación por medio de la técnica Lagrangiana, que describe las energías del robot de 2 GDL, se pueden obtener las siguientes ecuaciones para los pares mecánicos aplicados por los actuadores expresados en (28) y (29), haciendo uso de la ecuación (20).

$$\begin{aligned} \tau_1 = & \ddot{q}_1 (l_{c_1}^2 m + l_1^2 m_2 + m_2 l_{c_2}^2) + m_2 l_{c_2}^2 \ddot{q}_2 \\ & + 2m_2 l_1 l_{c_2} (\ddot{q}_1 \cos q_2 - \dot{q}_1 \dot{q}_2 \sin q_2) \\ & + m_1 l_1 l_{c_2} (\ddot{q}_2 \cos q_2 - \dot{q}_2^2 \sin q_2) + I_1 \ddot{q}_1 + I_2 (\ddot{q}_1 + \ddot{q}_2) \end{aligned} \quad (28)$$

$$\begin{aligned} \tau_2 = & m_2 l_{c_2}^2 \ddot{q}_1 + m_2 l_{c_2}^2 \ddot{q}_2 + m_2 l_1 l_{c_2} (-\dot{q}_1 \dot{q}_2 \sin q_2 + \ddot{q}_1 \cos q_2) \\ & + I_2 (\ddot{q}_1 + \ddot{q}_2) + m_2 l_1 l_{c_2} (\dot{q}_1 + \dot{q}_2) q_1 \sin q_2 \end{aligned} \quad (29)$$

La ecuación dinámica general que modela este tipo de estructuras es la expresada por la ecuación (30) la cual servirá para escribir de forma compacta las ecuaciones (28) y (29) con propósitos de un control más práctico.

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \quad (30)$$

Los vectores de posición, velocidad, aceleración y pares aplicados están representados por (31)

$$q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}; \quad \dot{q} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}; \quad \ddot{q} = \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix}; \quad \tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (31)$$

Por lo que también pueden escribirse en su forma matricial y que está representada por la ecuación (32), donde el vector de gravedad es cero.

$$\begin{bmatrix} M_{11}(q) & M_{12}(q) \\ M_{21}(q) & M_{22}(q) \end{bmatrix} \ddot{q} + \begin{bmatrix} C_{11}(q, \dot{q}) & C_{12}(q, \dot{q}) \\ C_{21}(q, \dot{q}) & C_{22}(q, \dot{q}) \end{bmatrix} \dot{q} + \begin{bmatrix} g_1(q) \\ g_2(q) \end{bmatrix} = \tau \quad (32)$$

Donde las siguientes ecuaciones por separado completan la ecuación (32).

$$M_{11} = l_{c_1}^2 m + l_1^2 m_2 + m_2 l_{c_2} + 2m_2 l_1 l_{c_2} \cos q_2 + I_1 + I_2 \quad (33)$$

$$M_{12} = l_{c_2}^2 m_2 + m_2 l_1 l_{c_2} \cos q_2 + I_2 \quad (34)$$

$$M_{21} = l_{c_2}^2 m_2 + m_2 l_1 l_{c_2} \cos q_2 + I_2 \quad (35)$$

$$M_{22} = l_{c_2}^2 m_2 + I_2 \quad (36)$$

$$C_{11} = 2m_2 l_1 l_{c_2} \sin q_2 \dot{q}_2 \quad (37)$$

$$C_{12} = m_2 l_1 l_{c_2} \sin q_2 (\dot{q}_1 + \dot{q}_2) \quad (38)$$

$$C_{21} = m_2 l_1 l_{c_2} \sin q_2 \dot{q}_1 \quad (39)$$

$$C_{22} = 0 \quad (40)$$

$$g_1 = 0 \quad (41)$$

$$g_2 = 0 \quad (42)$$

### 2.3.2 Modelo matemático para un motor DC

Como se mencionó anteriormente, lo que hace posible el movimiento de los robots son los actuadores, que por lo general son eléctricos. En el caso de los manipuladores es regularmente común el ver que estos son movidos por motores de corriente directa.

En la Figura 4 se puede observar el diagrama de un motor de corriente directa (DC), el cual está compuesto por dos partes; un rotor y un estator. La corriente en la armadura es independiente a la otra y como resultado de ello, se puede tener un mejor manejo de control de posición para estos motores.

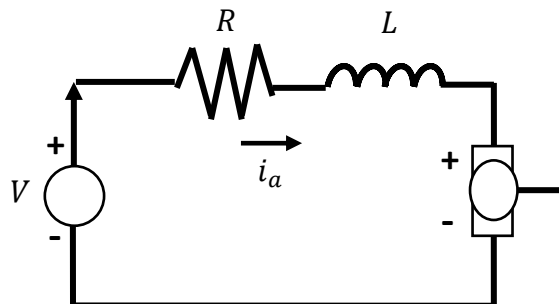


Figura 4. Circuito del motor de corriente directa.

La entrada de voltaje es aplicada al campo el cual tiene una inductancia  $L$  y una resistencia  $R$ . La corriente de armadura abastecida a la armadura se mantiene constante y por lo tanto el eje del motor es controlado por el voltaje de entrada.

Mientras que el coeficiente de fricción viscosa y la inercia del motor están dados por  $B$  y  $J$  respectivamente, las variables de salida que da el motor son la posición angular  $\theta_m$  y la velocidad angular  $\omega_m$ .

Las expresiones que describen el comportamiento de un motor DC están dadas por las siguientes ecuaciones que relacionan el par del motor  $K_t$  y la corriente de la armadura  $i_a$ , basados en la segunda ley de Newton y la ley de voltaje de Kirchhoff, resultan las ecuaciones (43) y (44)

$$J\ddot{\theta} + B\dot{\theta} = K_t i_a \quad (43)$$

$$L \frac{di}{dt} + Ri = V - K_t \dot{\theta} \quad (44)$$

Al aplicarle la transformada de Laplace a las ecuaciones anteriores, estas quedarán expresadas en términos de la variable Laplace  $s$  en (45) y (46).

$$(Js^2 + Bs)\theta(s) = KI(s) \quad (45)$$

$$(Ls + R)I(s) = V(s) - Ks\theta(s) \quad (46)$$

La cual al despejar (46) y sustituir el valor de  $I(s)$  en la ecuación (45), para posteriormente separar los valores  $V(s)$  y  $I(s)$ , y obtener la función de transferencia para el motor DC, la cual es una forma de representar el modelo matemático del sistema a través de un cociente que relaciona la respuesta de salida con una señal de entrada, quedando representada por la ecuación (47).

$$G(s) = \frac{\theta(s)}{V(s)} = \frac{K}{(Js+B)(Ls+R)+K^2} \quad (47)$$

## 2.4 Controladores

El control en cualquier sistema puede ser descrito como la aplicación de órdenes o el empleo de fuerzas para que el sistema trabaje o se mueva de acuerdo a las instrucciones que se le han ordenado. Para el control de un robot es necesario

conocer el modelo matemático del mismo, así como de un sistema de control que actué sobre él [38].

El control es un sistema lógico, responsable de determinar el flujo de ejecución, decide que tareas se desarrollan en cada instante, según la reacción de este sistema y el entorno que está siendo intervenido. Se encargará de determinar también las acciones que se realizarán a futuro de acuerdo con los parámetros que se obtuvieron con la finalidad de reducir el error que se está generado. De esta forma se controlan de manera general las actividades del robot para alcanzar la misión global del sistema lo más cercano al punto que se tiene establecido [39].

Muchos controladores industriales emplean fuentes de energía como la electricidad o un fluido presurizado [40]. Estos pueden clasificarse como hidráulicos, neumáticos, electrónicos o también la combinación de algunos de estos. Estos pueden ser elegidos de acuerdo a la naturaleza del sistema con el que se está trabajando así como otros factores como son: precisión, disponibilidad, costo y dimensiones.

Otra de sus clasificaciones puede ser de acuerdo a sus acciones de control [41]:

1. De dos posiciones. El controlador solo tiene dos posiciones las cuales son por lo general, encendido y apagado, por esto generalmente son sencillos y económicos.

2. Proporcionales. Un controlador con acción proporcional, posee una ganancia proporcional  $K_p$ , la cual es ajustable por medio de un amplificador en el mecanismo real, siendo  $u(t)$  la salida del controlador y  $e(t)$  el error. Expresado dentro de la ecuación (48).

$$u(t) = K_p e(t) \quad (48)$$

3. Integral. Para estos controladores la razón de cambio del valor de salida  $u(t)$  es sustituida por una razón proporcional al error  $e(t)$  o visto de otra manera el producto de la constante ajustable  $K_i$  y la integral del error. Expresado dentro de la ecuación (49).



$$\frac{du(t)}{dt} = K_i e(t) \quad (49)$$

4. Derivativo. Este tipo de controlador tiene como función el tratar de predecir el siguiente valor del error, y de esta manera anticipar el siguiente valor de la señal de control, lo cual resulta muy útil porque se va observando la tendencia y se corrige antes de que suceda. La desventaja es que esta acción es muy sensible al ruido. Su expresión está dada en la ecuación (50).

$$u(t) = k_d \frac{de(t)}{dt} \quad (50)$$

A partir de estos tres tipos de controladores, el Proporcional (P), Integral (I) y Derivativa (D), es posible tomar acciones de control que combinen alguna de estas partes o en su defecto el más comúnmente usado que integra las tres partes mencionadas (PID), por esto lo hace más viable como la primera opción de estudio.

La sumatoria de las tres partes se puede expresar en la ecuación (51).

$$u(t) = k_p e(t) + k_i \int_0^t e(t) dt + k_d \frac{de(t)}{dt} \quad (51)$$

El cual queda reflejado también en el esquema de la Figura 5, donde se puede observar que depende de una entrada de acuerdo con los valores que se desean que respete y a una retroalimentación de los valores reales provenientes del sistema. De la diferencia de estos dos valores sale una señal de error que pasa por las tres partes de acción y de la suma de las tres partes se obtiene una señal de control que va directo al sistema.

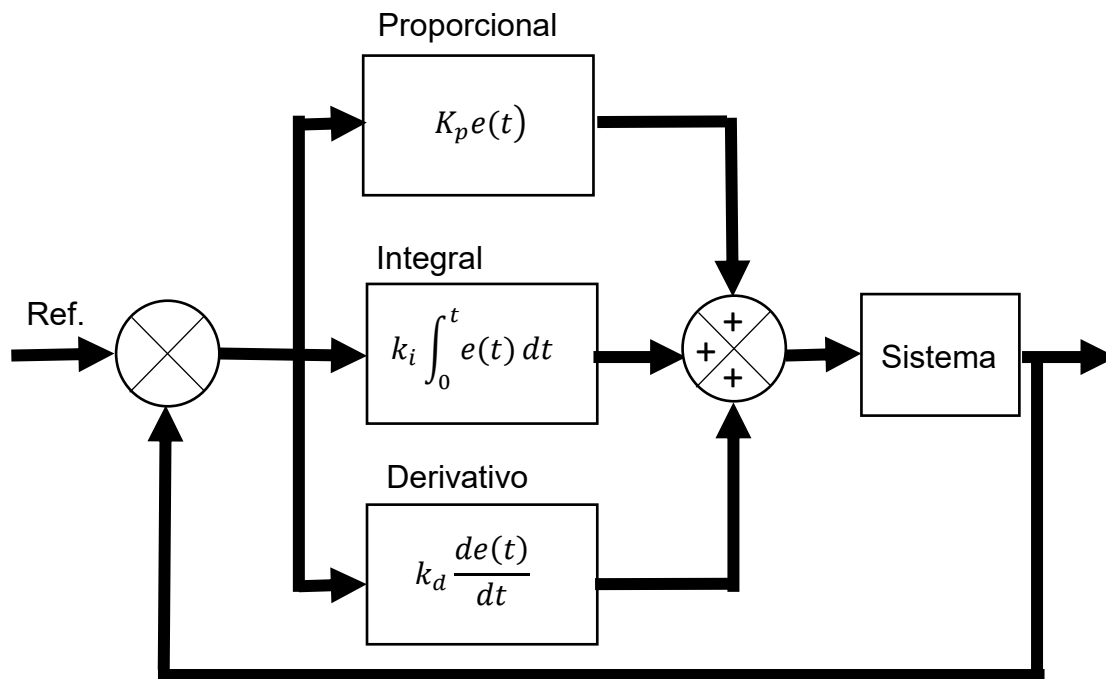


Figura 5. Diagrama de bloques para control PID

### 2.4.2 Controlador del robot manipulador

Una vez dada la señal de que se ha alcanzado la posición deseada por parte del sensor, el controlador marca la siguiente posición deseada hasta llegar al último punto, de esta forma genera una trayectoria.

Para implementar estos controladores avanzados, se requiere un modelo cinemático y dinámico apropiado del robot. Este modelo determina las fuerzas de interacción humano-robot y su movimiento, así como la fuerza de transmisión entre los actuadores y el punto de interacción.

Para el caso general de los robots manipuladores dentro de su algoritmo para el control del sistema se establece la obtención de información de su estado actual y se compara con el deseado, por lo general con el uso de distintos sensores, para de esta forma decidir la secuencia de acciones necesarias para reducir el error y alcanzar su objetivo siendo compensada por el uso de un actuador. Para estos manipuladores es necesario contemplar los pares mecánicos necesarios que

ejercerá el actuador a la plataforma para que este alcance la siguiente posición deseada por medio de un voltaje suministrado a dichos actuadores por parte del controlador con la finalidad de mantener la velocidad requerida.

El problema de control para sistemas dinámicos consiste en determinar la secuencia de entradas articulares necesarias para que el dispositivo ejecute un movimiento deseado [42]; en el caso particular del robot manipulador es determinar las trayectorias de entradas conjuntas para que el efector final alcance el objetivo.

Por lo general el control de los robots manipuladores es por medio de actuadores que suministran una fuerza o un par mecánico para ocasionar el movimiento de sus eslabones y que en la mayoría de los casos compuestos por motores de corriente continúa.

El control de posición de los robots manipuladores puede resumirse de la siguiente forma. Con una posición deseada  $q_d$  como punto de referencia, se debe obtener un vector  $\tau$  tal que la posición  $q$  asociada con el efector final del robot manipulador tienda a  $q_d$ .

El principal problema radica en la necesidad de que el sistema de control de posición compense automáticamente los errores por medio del conocimiento de los parámetros de un sistema. El error desvía al sistema de la trayectoria deseada.

En un sistema de control de lazo cerrado, el controlador se alimenta con la señal de error del actuador, correspondiente a la diferencia entre la señal de entrada y la señal de realimentación, con la finalidad de reducir el error y llevar la salida del sistema a un valor conveniente. En la Figura 6 se puede observar un diagrama del control para un robot 2 GDL.

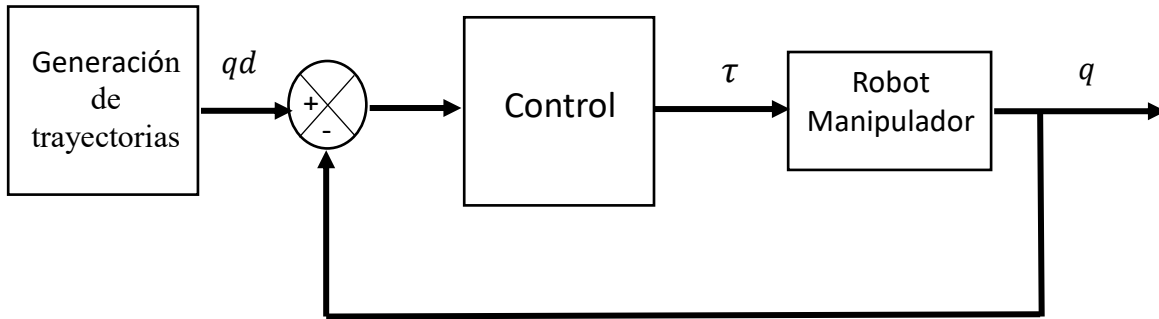


Figura 6. Diagrama control de movimiento para robot manipulador

### 2.4.1 Control de proporcional derivativo con retro de velocidad

Debido a que el control derivativo opera sobre la velocidad de cambio del error y no sobre el error mismo, este modo nunca se utiliza solo. Siempre se emplean junto con una acción de control proporcional o proporcional-integral.

Para un control con acción de control proporcional, la relación entre  $u(t)$  y la señal de error  $e(t)$  está dada por la ecuación (52).

$$U(t) = K_p e(t) \quad (52)$$

El control proporcional derivativo (PID) es una extensión inmediata del control proporcional más una retroalimentación de velocidad, y la ley de control está dada por la ecuación (53).

$$\tau = K_p \tilde{q} + K_v \dot{\tilde{q}} \quad (53)$$

Donde  $\tilde{q}$  es el error de posición y de velocidad, mientras que  $K_v$  y  $K_p$  son escogidas por el usuario y deberán ser matrices simétricas positivas definidas conocidas como de ganancia de posición y velocidad respectivamente.

$$\tilde{q} = q_d - q \quad (54)$$

### 2.4.3. Sintonización de PID

Para la sintonización de los parámetros proporcional, integral y derivativo para los actuadores del manipulador se utiliza el primer método de Ziegler y Nichols, de la última ganancia; este consiste en dejar la parte proporcional ( $K_p$ ) la cual comienza

en valores muy bajos y se aumenta gradualmente hasta conseguir un sistema oscilatorio, mientras la parte derivativa e integral se minimizan. Este valor proporcional se determina entonces como un valor proporcional crítico ( $K_{pc}$ ), y el otro valor que se registra es el periodo crítico ( $P_u$ ) que es el periodo que muestran las oscilaciones.

Después de haber obtenido ambos valores se utilizan diferentes métodos para calcular las diferentes constantes que puede utilizar el sistema dependiendo del esquema de control que se desea implementar ganancia que se está. Estos pueden ser: P, PI o PID. Los cuales pueden verse expresados en la Tabla 1.

Tabla 1 Criterios de sintonización de Ziegler y Nichols para el método de la última ganancia.

Control	$K_p$	$T_i$	$T_d$
Proporcional (P)	$0.5 * K_{pc}$	$\infty$	0
Proporcional Integral (PI)	$0.45 * K_{pc}$	$1/1.2 * P_u$	0
Proporcional Integral Derivativo (PID)	$0.6 * K_{pc}$	$0.5 P_u$	$P_u/8$

## 2.5 Generación de Trayectorias

Se puede definir la generación de trayectorias como una secuencia de diferentes configuraciones que tomará nuestro robot durante una ejecución, esto naturalmente marca dicha secuencia dentro de un espacio en el plano donde se puede trabajar sin restricciones, únicamente los límites que el sistema posee.

Dada una posición inicial  $q(t_i)$  y una posición final  $q(t_f)$  se establece que existe un periodo de tiempo en la ejecución de la trayectoria compuesto por  $t_f - t_i$ , durante el cual se adquieren valores de las variables como la posición la velocidad y aceleración.

Dependiendo de la tarea que el robot realiza, se genera una función con la cual se moverá, para ello se requiere un algoritmo de control para la ejecución de la trayectoria y calcula las entradas al controlador de movimiento del robot. La

planeación de estas trayectorias puede efectuarse en términos de las posiciones, del efector final, su orientación y su derivada respecto al tiempo.

El movimiento de un robot manipulador puede denominarse punto a punto, esto se entiende como una serie de posiciones en el espacio que se desea que alcance su efector final durante un cierto periodo de tiempo definido.

En este caso, se determina una curva o trayectoria trazada temporalmente en el espacio de trabajo y el problema de control consiste en hacer pasar el extremo por dicha trayectoria tan cerca como sea posible.

Cabe mencionar que estas posiciones a alcanzar deben de estar dentro de los límites físicamente alcanzables por el robot.

### **2.5.2 Estabilidad del sistema**

Se puede entender el estudio de la estabilidad dentro de los sistemas dinámicos como la respuesta del sistema ante perturbaciones pequeñas en los estados del sistema [43]. Se dice que un punto de equilibrio es estable si para valores pequeños de perturbaciones iniciales, el movimiento perturbado permanece en el espacio de estados.

La Teoría de Lyapunov es muy importante en el análisis de estabilidad de sistemas dinámicos lineales y no lineales, sin importar el orden del sistema, permite obtener información sobre la estabilidad del punto o puntos de equilibrio del sistema sin resolver la ecuación diferencial que caracteriza a dicho sistema dinámico [44].

Para el caso particular del robot manipulador se partirá del modelo que se toma de la comparación de los pares mecánicos tanto en el modelo matemático como en nuestra ley de control para el robot de dos grados de libertad sin tener en cuenta los pares gravitacionales, se toman de las ecuaciones (31) y (53) quedando descrita en la ecuación (55). Esto es, se aplica la ley de control al modelo de la estructura del robot.

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} = K_p \tilde{q} - K_v \dot{q} \quad (55)$$

Para el estudio del equilibrio del sistema, se usa el método directo de Lyapunov Considerando la función candidata de Lyapunov como [45]:

$$V(\tilde{q}, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q} + \frac{1}{2} \tilde{q}^T K_p \tilde{q} \quad (56)$$

La cual al derivarse con respecto al tiempo queda expresada en la ecuación (57).

$$\dot{V}(\tilde{q}, \dot{q}) = \dot{q}^T M(q) \ddot{q} + \frac{1}{2} \dot{q}^T \dot{M}(q) \dot{q} + \tilde{q}^T K_p \dot{\tilde{q}} \quad (57)$$

Por lo que al ser sustituido el valor de  $M(q)\ddot{q}$  y al reagrupar los términos se puede ver que se cumple con la forma  $\dot{q}^T \left[ \frac{1}{2} \dot{M}(q) - C(q, \dot{q}) \right] \dot{q} = 0$  [46] y por lo tanto la ecuación (57), se puede resumir en la expresión de la ecuación (58)

$$\dot{V}(\tilde{q}, \dot{q}) = -\dot{q}^T K_v \dot{q} \quad (58)$$

Como  $K_v$  una matriz positiva definida se entiende que el valor de  $\dot{V}(\tilde{q}, \dot{q}) \leq 0$ , por lo que se considera una función de Lyapunov y por lo tanto el origen es estable y las soluciones  $\tilde{q}$  y  $\dot{q}$  están acotadas [47].

## 2.6 Rehabilitación con robots

Actualmente existe un campo emergente en la robótica que está tomando importancia, la robótica rehabilitadora, la cual incluye dispositivos de apoyo a la rehabilitación y creación de prótesis, estas implican interacción entre robots y operadores humanos [48].

En general estos dispositivos robóticos, para soporte de extremidades superiores, pueden clasificarse: en prótesis y en ortesis. Las prótesis son un sustituto que puede colocarse para reemplazar partes perdidas del cuerpo mientras que las ortesis se pueden definir como aparatos que pueden utilizarse para mejorar la funcionalidad de las partes móviles del cuerpo. Las de tipo ortesis pueden subdividirse en los robots que alinean sus articulaciones con las del usuario (exo esqueletos), estos proporcionan fuerza externa a las extremidades del paciente para colocar los miembros en la posición deseada. El otro tipo son los que están hechos para alinear su efector final con el del usuario de esta forma apoyan al seguimiento de ciertas



trayectorias [49]. En la Figura 7 puede observarse los tipos de robot para rehabilitación del miembro superior a) un robot de efector final y b) un exoesqueleto.



Figura 7. Tipos de robot para rehabilitación del miembro superior a) robot efector final y b) exoesqueleto

Estos dispositivos robóticos para la rehabilitación de pacientes con problemas de movilidad, principalmente los miembros de la parte superior, que sufrieron algún tipo de accidente, son una alternativa óptima ya que pueden replicar los movimientos que se usan al momento de la terapia, incluso puede tener mejoras significativas como lo son una mayor duración, mayor número de secuencias y una mejor precisión. Se presenta a continuación trabajos relevantes específicos que se relacionan más directamente con el sistema que aquí se desarrolla.

Yali Liu et al desarrollaron una nueva plataforma de tipo EEULRebot (*Endeffector upper limb rehabilitation robot*) utilizando un robot de tipo efector final con dos grados para brindar capacitación de rehabilitación a pacientes hemipléjicos. La plataforma funciona teniendo en cuenta la fuerza de los pacientes mediante la implementación de un sensor de fuerza. Estableciendo tres modos de entrenamiento dentro del control para la plataforma, los cuales son planeados según los requerimientos del paciente y aplicados directamente a trayectorias consideradas como las básicas en la rehabilitación del miembro superior como el círculo y la línea. La desviación de la trayectoria del efector final y la requerida fue

inferior a 1 mm durante las tareas, lo que demostró la precisión de movimiento del robot; aunque no queda especificada la composición para su controlador [50].

Saadatzi M. et al presentaron un algoritmo no lineal para la estimación de los pares de perturbación en un robot de dos grados de libertad para la rehabilitación de la muñeca sin el uso de sensores. Estos autores partieron de un modelo dinámico de robot así como dos tipos de modelos de fricción y se le dio el seguimiento a una trayectoria senoidal por medio de un controlador PD más la compensación de un modelo de fricción. Con esto se pretende realizar una comparación más minuciosa en trabajos futuros mediante el uso de la dinámica y un sensor de fuerza [51].

Peng Liang et al, desarrollaron una metodología para establecer la interacción de un sistema dinámico entre el humano y el robot, donde ambas dinámicas se consideraron juntas. Este sistema dinámico se deriva de un espacio de trabajo de sus tres cadenas abiertas, por lo que ambas restricciones cinemáticas e interacción de las fuerzas se consideran durante la derivación; Además hizo una comparación con otros dos métodos (modelo reducido y el basado en la formulación Lagrange-D'Alembert) mostrando una mayor facilidad para derivarse y resultando computacionalmente más eficiente [52].

Naghavi N. et al, simularon y realizaron experimentos para una plataforma de un robot simple de un grado de libertad adaptado a una silla. La intención era la de realizar entrenamientos de rehabilitación de rodilla para pacientes en diferentes etapas de la enfermedad con movimientos simples de extensión y flexión. Para ello implementaron dos controladores, uno de los cuales limita la fuerza de interacción mediante la limitación de los parámetros PID, los cuales al avanzar la terapia son reducidos por el terapeuta. Para medir el par ejercido se usó el sensor ACS712 el cual medía la corriente retroalimentada al controlador y al ser proporcional ayudó al cálculo del par del motor de fuerza establecido y una vez cruzado el límite de fuerza el controlador cortaba la salida para bajar el esfuerzo del motor [53].

Nielsen et al, presentaron un enfoque novedoso para automatizar el entrenamiento de las extremidades superiores mediante la utilización de un brazo robótico industrial y del movimiento dinámico primitivo con fuerza de retroalimentación. Lo cual permitía un entrenamiento individualizado y adaptativo para personas con diferentes propiedades físicas y deficiencias. Aunque algunas partes de esto son principalmente simulaciones, se vio el uso de DMP como un paso en la dirección correcta para permitir una configuración fácil y flexible de la rehabilitación por medio de ejercicios que permiten a los terapeutas y pacientes, de manera simple, el control e interacción con una tecnología muy compleja [54].

En otro trabajo se presenta un algoritmo de control que provee la ayuda necesaria, proporcionando a los usuarios fuerza de retroalimentación solamente cuando es estrictamente necesaria. De esta forma se evita relajar completamente la actividad muscular. Obtuvieron información de las trayectorias generadas por el usuario de un brazo mecánico siguiendo una ruta predefinida y con el apoyo de estímulos vibrotáctiles, integrado por un micro controlador y un motor de vibración, que se colocó en una correa en la muñeca para llevar a cabo la trayectoria de una manera correcta, si el paciente está haciendo una trayectoria incorrecta. En cuanto a la cinemática utilizaron un modelo geométrico directo para obtener la matriz de transformación homogénea. Con lo que el 55.93% de los usuarios fueron capaz de corregir la trayectoria durante la terapia [55].

Xiann li et al, propusieron un controlador de impedancia iterativo en el uso de robots de rehabilitación impulsados por "SEAS" (*Series Elastic Actuators*). El cual garantizara el rendimiento transitorio del robot. La estabilidad del sistema global, que consiste tanto en el subsistema de enlace rígido y el subsistema actuador, se probó rigurosamente con Métodos de Lyapunov. Como resultado se obtuvo un error de impedancia tendiente a cero después de más interacciones con el aprovechamiento en el uso de tareas repetitivas [56].

Mancisidor et al, desarrollaron un robot de rehabilitación, pantógrafo háptico universal (UHP) compuesto de dos subsistemas, un actuador de serie elástico y una estructura de pantógrafo con la que se interactúa. Ambos sistemas están conectados a través del punto de transmisión, tal que el par ejercido por los actuadores del sistema de accionamiento, la fuerza y el movimiento aplicados al pantógrafo por el paciente se transmiten en la forma de fuerza y movimiento. Por esto se realizó un modelo matemático del robot necesario para implementar el controlador de interacción fuerza-robot-paciente. Dado que el modelo exige medidas de posición del sistema de accionamiento para la estimación del par, la fuerza y el movimiento. Para analizar sus resultados se colocaron sensores en el brazo con el fin de comprobar si los resultados que estimaban por medio del pantógrafo eran similares a los obtenidos por los sensores, demostrando que el error medio de movimiento en el punto de contacto es menos del 4% del área de movimiento, mientras que el error de fuerza de transmisión es más pequeño que el 10% del rango de fuerza máxima [57].

Un manipulador de dos grados de libertad, ver Figura 8, permite los movimientos alrededor de las articulaciones del codo y el hombro en el plano horizontal. Mientras la paciente sujeta con la mano un mango existente en el extremo del dispositivo de manipulación, los motores dirigen las articulaciones del hombro y el codo del brazo robótico al tiempo que varios sensores de fuerza y posición determinan el trayecto espacial de la mano y la cantidad de fuerza que realiza el paciente.

A medida que mejora el control muscular por parte del paciente, es posible reducir el grado de asistencia robótica. El tipo de ayuda robótica necesario se puede decidir individualmente en función de las capacidades de cada paciente.

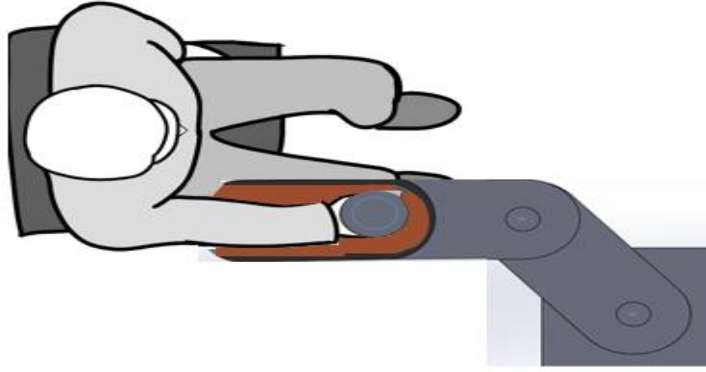


Figura 8. Esquema superior paciente/robot de 2GDL

## 2.7 Modelo completo dinámico manipulador-paciente

Una vez que se ha definido el modelo matemático para el robot de 2 GDL, se prosigue a anexar la parte principal del proyecto, ya que el robot manipulador estará interactuado con una fuerza externa proveniente del usuario el cual puede variar el nivel de fuerza que se está aplicando. Las ecuaciones de los pares mecánicos ejercidos por el manipulador se componen de la suma de dos pares mecánicos  $\tau_{pi}$  y  $\tau_{ri}$ , expresados en la ecuación (59).

$$\tau_i = \tau_{pi} + \tau_{ri} \quad (59)$$

Donde  $\tau_{p1}$  y  $\tau_{p2}$  son los pares generados por la fuerza del brazo humano mientras que  $\tau_{r1}$  y  $\tau_{r2}$  son los generados por cada uno de los actuadores para cada eslabón correspondiente; Donde el vector de los pares que representado por (60).

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} \tau_{p1} + \tau_{r1} \\ \tau_{p2} + \tau_{r2} \end{bmatrix} \quad (60)$$

Aplicando estos pares en la ecuación (32) quedaría expresada como en la ecuación (61).

$$\begin{bmatrix} M_{11}(q) & M_{12}(q) \\ M_{21}(q) & M_{22}(q) \end{bmatrix} \ddot{q} + \begin{bmatrix} C_{11}(q, \dot{q}) & C_{12}(q, \dot{q}) \\ C_{21}(q, \dot{q}) & C_{22}(q, \dot{q}) \end{bmatrix} \dot{q} + \begin{bmatrix} g_1(q) \\ g_2(q) \end{bmatrix} = \begin{bmatrix} \tau_{p1} + \tau_{r1} \\ \tau_{p2} + \tau_{r2} \end{bmatrix} \quad (61)$$

## **Capítulo 3**

### **3. Método y materiales**

En este capítulo se describen las características generales de la plataforma de dos grados de libertad con la que se trabajó durante el periodo de pruebas, se explica la instrumentación usada, así como el algoritmo que se aplicó al controlador.

También se presentan los datos pertenecientes a la plataforma para el desarrollo de la simulación mediante la ley de control y algunos parámetros específicos de los actuadores.

Partiendo de esta información se plantea la siguiente hipótesis sobre el proyecto:

Por medio de una plataforma que adapta un sistema de trayectorias para un robot planar, se puede mejorar la actividad motriz de un paciente con enfermedad cerebro vascular.

#### **3.1 Desarrollo de la plataforma experimental**

Para la prueba experimental y con la finalidad de corroborar los datos obtenidos por medio de la simulación fue necesario el uso de una plataforma o prototipo la cual consiste en dos partes, estas comprenden el robot manipulador mismo y la parte del controlador implementado en una tarjeta electrónica.

En la Figura 9 se puede observar un esquema general de los componentes utilizados para la plataforma experimental con lo que se generan y evalúan los resultados.

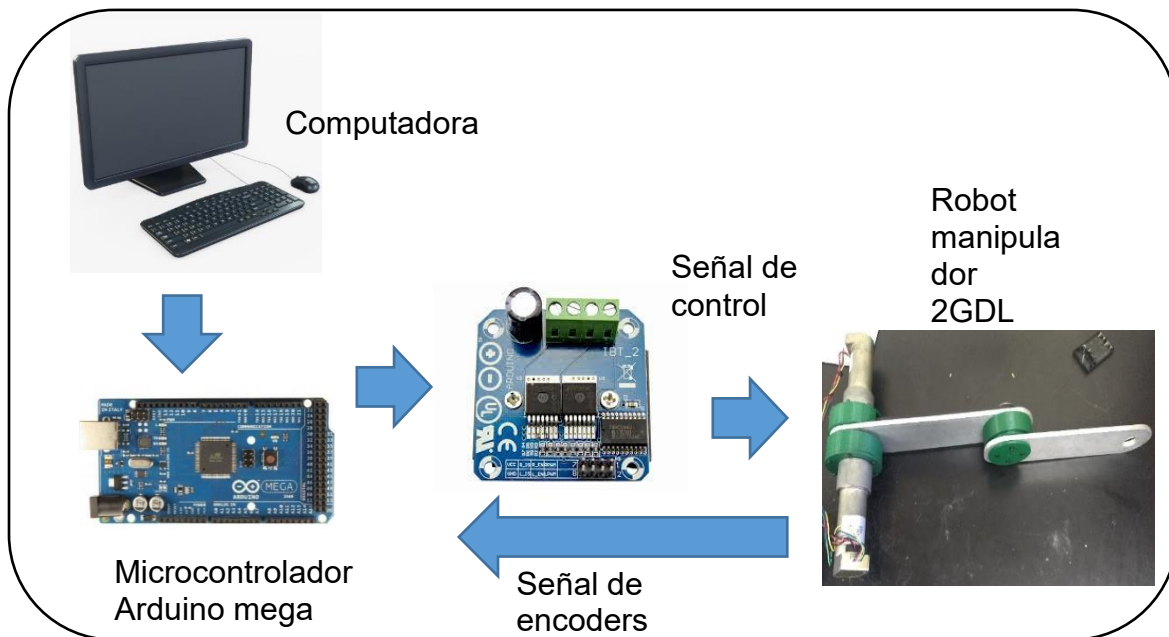


Figura 9. Esquema de la plataforma experimental general

### 3.1.1 Plataforma para el manipulador

En la construcción del prototipo de experimentación, se utilizaron diversos dispositivos como motores de CD, materiales para los eslabones y otros accesorios.

Para la parte de los actuadores se seleccionaron dos motores de corriente directa de 12 v modelo GM9236S019, uno para cada eslabón, de los cuales se pueden observar sus especificaciones en la Figura 10. Este actuador fue seleccionado debido a la variedad de velocidad que puede presentar, así como un control más sencillo en comparación con un motor de inducción, además de la flexibilidad que presenta en la relación par-velocidad lo que nos dará una mayor precisión en el control de este al momento de ejercer el movimiento de los eslabones.

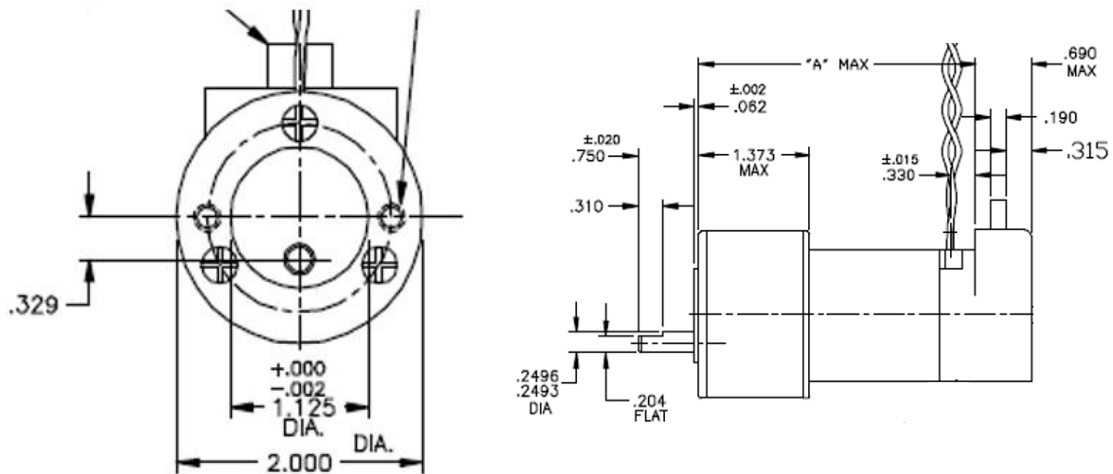


Figura 10. Dimensiones físicas de Motor DC

También cabe mencionar que este modelo de motor maneja un encoder incremental integrado en el cabezal del engranaje, el cual es un dispositivo de detección que convierte el movimiento en una señal, y puede ser leída por un dispositivo en un sistema de control de movimiento. Esta señal ayudará en cuanto a la interacción por medio de la comunicación con el controlador, marcando cuando se han alcanzado las posiciones deseadas, dicho encoder requiere una fuente de alimentación de 5 volts y la resolución del codificador es de 500 pulsos por revolución.

Los actuadores se encuentran conectados a unos reductores cuya finalidad es la de adaptar el movimiento del motor a uno más suave. Ya que para el propósito del proyecto se requieren movimientos lentos con la intención de llevar una óptima rehabilitación, siendo para nuestro caso una relación 736.78:1, la cual es la diferencia entre las revoluciones de los dos engranajes conectados para hacer girar al otro, por lo cual fue la menor relación que puede darse por medio del diseño del proveedor y que pueda adaptarse a nuestro tipo de motor.

Debido al peso que podría ejercer el segundo motor en el eslabón y que podría traer más complicaciones en un futuro, se optó por colocar el motor directamente en la base y hacer la transmisión del movimiento del motor por medio de una banda directamente hacia el segundo reductor.



Dichos actuadores fueron alimentados por una fuente de 12 volts con una corriente inicial de 2.3 A, la cual se conectaba directamente junto a ambos actuadores hacia un Puente H modelo IBT\_2, ver Figura 11, esto con la intención de evitar la conexión directa de nuestra fuente con el microcontrolador y de esta forma hacer posible la regulación de un voltaje más alto de forma segura.

El modelo IBT\_2 cuenta con terminales R\_EN y L\_EN, los cuales sirven para controlar el sentido de giro del motor. Funcionan de forma que, si IN1 está en HIGH e IN2 en LOW, el motor A gira en un sentido, y si está IN1 en LOW e IN2 en HIGH lo hace en el otro. También deben de estar conectados el VCC y GND a la fuente de 5 volts de la tarjeta y placa para el Arduino y la tierra respectivamente.



Figura 11. Puente H modelo IBT\_2

Para controlar la velocidad de giro de los motores tenemos que usar las terminales ENA y ENB. Se conectan a dos salidas PWM (*Pulse Width Modulation*) de Arduino mandando una señal de ciclo de forma que se envía un valor entre 0 y 5 que controla la velocidad de giro dependiendo del promedio de la señal que se está enviando.

En cuanto al apartado de la morfología del robot manipulador, está constituido por dos articulaciones rotacionales, de las cuales cada una de estas le otorga un grado de libertad (GDL). Cada una de ellas se encuentra conectada a un eslabón, el cual es de aluminio y tiene una longitud de 0.264 y 0.284 metros así como con un peso aproximado de .0379 y 0.329 kilogramos respectivamente para cada uno.

La primera articulación está localizada en el origen y rota el primer eslabón por un motor de CD, mientras que la segunda articulación está localizada al final del primer

eslabón y su movimiento es posible gracias a un motor DC el cual le trasmite el par por medio de una banda.

### 3.1.2 Controlador

En cuanto al controlador, se optó por utilizar una tarjeta conteniendo un microcontrolador debido a las facilidades que este presentaba. Esas facilidades son: bajo costo, fácil adquisición, vía de transmisión de las señales analógicas resultantes de los cálculos y también la lectura de información de los encoders de los actuadores del robot manipulador para su posterior procesamiento.

Se seleccionó la plataforma Arduino, cuyo modelo es el Arduino MEGA 2560 REV3, ver Figura 12, que cuenta con 54 pines digitales que pueden ser usados tanto de entrada como de salida y de los cuales 15 de estos pines pueden ser utilizadas para modulación PWM con una frecuencia de 490Hz.

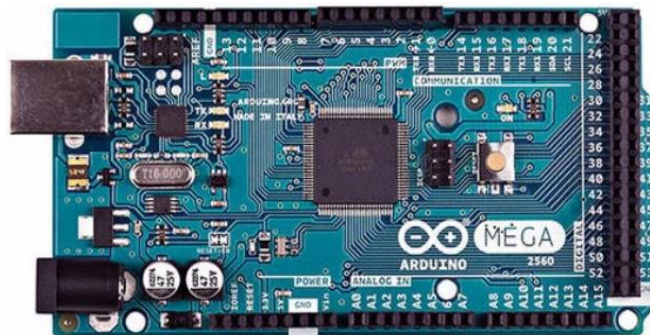


Figura 12. Sistema embebido Arduino Mega 2560

Según las especificaciones, se recomienda que el rango de alimentación debe encontrarse entre 7 a 12 VDC para trabajar de forma más estable ya que si se suministra un mayor VCD podría calentar el regulador y dañar la tarjeta.

Los algoritmos utilizados por el sistema son programados en lenguaje C++ y apoyadas por el uso de librerías que facilitan su elaboración, dichos algoritmos se transferirán hacia el controlador por vía serial desde una PC. El algoritmo utilizado se encuentra dentro de la sección del anexo.

Las señales de control necesarias son obtenidas a partir de las salidas digitales proporcionadas por el microcontrolador, tienen valores de 0 VDC y 5 VDC

suministrados en diferentes frecuencias. Es necesario una etapa de amplificación para los requerimientos de corriente que presentan los actuadores.

También cabe mencionar que se utilizó un módulo para tarjeta SD mostrado en la Figura 13 el cual es conectado al microcontrolador, esto para guardar las posiciones angulares en las que se encuentran ambos eslabones dentro de un periodo de tiempo especificado.



Figura 13. Modulo SD Arduino

Se optó por el uso de este módulo debido a la velocidad de almacenamiento de los datos obtenidos. Ya que al realizarlo vía serial con la computadora interfería con la ejecución de la trayectoria haciendo más lenta la ejecución.

### **3.2 Cinemática del robot**

En esta sección se abordan los conceptos de cinemática directa e inversa del robot así como su aplicación y solución de nuestro caso particular para un robot de 2 GDL.

Éste es un problema geométrico algo complicado, que se resuelve de manera rutinaria miles de veces, en ciertos casos, la solución de este problema es el elemento más importante en un sistema manipulador [58].

#### **3.2.1 Solución de la Cinemática directa**

Primero se toma la cinemática directa del manipulador el cual consiste en el uso de los ángulos, asignándoles un valor, la respuesta es el posicionamiento del efector final en un punto en el espacio.

Utilizando la convención Denavit-Hartenberg se colocaron los datos para el caso particular del robot y que se muestran en la Tabla 2.

Tabla 2 Parámetros Denavit-Hartenberg

$i$	$\alpha_{i-1}$	$a_{i-1}$	$d_i$	$\theta_i$
1	0	0.264	0	$\theta_1$
2	0	0.224	0	$\theta_2$

Esto se simplificó en Matlab® por medio de una función creada donde se colocaron los primeros seis valores de la tabla 2 y los valores variantes son los ángulos que ingresamos para obtener la ubicación del efector final en el plano XYZ. El programa se encuentra dentro de sección de anexos 6.1.5.

### 3.2.2 Solución de la cinemática inversa

En la Figura 14 se muestra el desarrollo de un programa de Matlab® para el cálculo de ambos ángulos partiendo de la introducción de la posición del eslabón final dejándolo de la siguiente forma:

```

1 function [teta1,teta2]=cinematicainversa2(x,y)
2     L1=.30;
3     L2=.30;
4     D=((x^2)+(y^2)-(L1^2)-(L2^2))/(2*(L1*L2));
5     teta2=atan((((1+((-D)^2)).^(1/2))/D));
6     teta1=(atan((y/x))-tan((L2)*sin((teta2)*(pi/180)))/((L1)+(L2*cos((teta2)*(pi/180)))));

```

Figura 14. Matlab® desarrollo de cinemática directa

### 3.3 Generación de trayectorias

El movimiento de un robot manipulador puede denominarse punto a punto, esto se entiende como una serie de posiciones en el espacio que se desea que alcance su efector final durante un cierto periodo de tiempo definido.

En este caso, se determina una curva o trayectoria trazada temporalmente en el espacio de trabajo y el problema de control consiste en hacer pasar el extremo por dicha trayectoria tan cerca como sea posible.

Cabe mencionar que estas posiciones a alcanzar deben de estar dentro del espacio de trabajo del robot.

Para el caso particular de este proyecto se tiene contemplado el seguimiento de una trayectoria polinómica, una recta y una circular; quedando representados las últimas dos por las siguientes ecuaciones para su generación en el plano XY.

Para la trayectoria de una recta, como se expresa en las ecuaciones (62) y (63).

$$x_r = t + a \quad (62)$$

$$y_r = mt + b \quad (63)$$

Donde  $a$  y  $b$  son el punto  $(x, y)$  de inicio de la recta y  $m$  es la pendiente de esta.

Mientras que para una trayectoria circular:

$$x_c = r \cos(t) + a \quad (64)$$

$$y_c = r \sin(t) + b \quad (65)$$

Donde  $a$  y  $b$  son el punto  $(x, y)$  del centro del círculo y  $r$  es el radio de este.

### 3.3.1 Conversión de punto en el espacio a radianes

Para el caso particular del robot de dos grados de libertad con el que se está trabajando, es necesario tener la sucesión de movimientos de la trayectoria en radianes. Esto se describe como las posiciones angulares que deben alcanzar ambos eslabones, en algunos de los casos solo se tiene la ubicación de puntos en el espacio XY del efector final del robot manipulador.

Para la posible aplicación, es necesaria su conversión a radianes por medio de la cinemática inversa que se mencionó en el capítulo anterior. Por lo que se elaboró un esquema en Simulink® en la Figura 15, Simulink® es un entorno de programación que funciona dentro del entorno de Matlab, el cual se puede considerar más visual ya que funciona mediante bloques, que puede ayudar en la simulación de sistema mediante la modelación en conjunto de las partes que lo componen.

En el bloque de función de Matlab® se agregó la solución de la cinemática inversa por medio del método geométrico, al cual se le dará como entrada la ubicación del

efector final en el plano XY. Una vez calculada la posición angular de ambos se prosigue a colocarlos en un archivo “\*.mat” para su posterior lectura.

De esta forma podemos procesar cualquier trayectoria en estos dos planos, si es que se desconocían las posiciones angulares requeridas. Para comprobar que nuestra trayectoria no ha sido alterada se pueden transformar las posiciones en radianes a la ubicación en el plano XY por medio de la cinemática directa tratada en el capítulo 2 y expresada en un bloque de función de Matlab®.

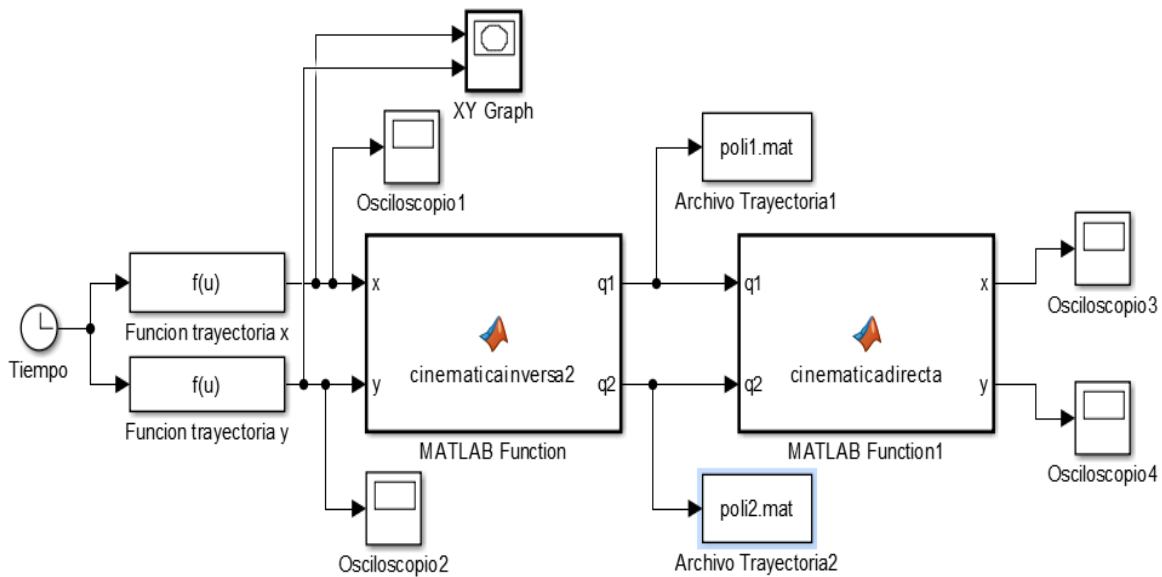


Figura 15. Esquemas en Matlab® para la conversión de la cinemática inversa

### 3.4 Cálculo de los momentos de Inercia

#### 3.4.1 Centro de masas

Para calcular el momento de inercia es necesario primero encontrar el centro de masa del eslabón, el cual, a simple vista, no se ve como una tarea complicada al tratarse de un cuerpo rígido homogénea. Sin embargo, se considera pertinente realizar los cálculos matemáticos ya que de aquí saldrán los valores que servirán posteriormente para la búsqueda de los momentos de inercia. En la Figura 16 se pueden observar las dimensiones del eslabón.

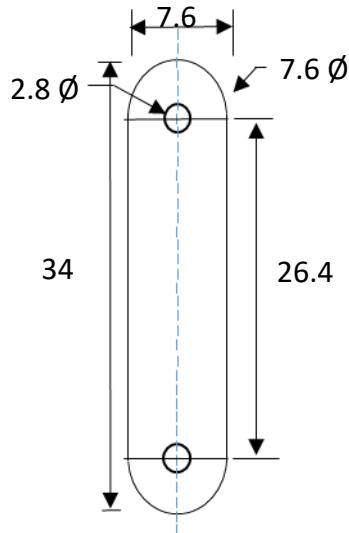


Figura 16. Dimensiones de eslabón

Observando la forma de la Figura 16 podemos partir de la división de esta en tres partes principales compuesta por dos medios círculos y un rectángulo, los cuales cuentan con una circunferencia menor que atraviesa la figura, abajo se puede colocaron los datos obtenidos de la división de las tres partes de la figura.

Tabla 3 Valores dimensionales de secciones

valor en y	valor en x	área cm <sup>2</sup>
30.2	3.8	19.603584
17	3.8	194.482464
3.8	3.8	19.603584

Siendo el área total  $A_T=233.689632$  cm<sup>2</sup> y quedando los valores del centro de masa representados como en la ecuaciones (66) y (67).

$$C_x = \sum_n A_n * x_n \quad (66)$$

$$C_y = \sum_n A_n * y_n \quad (67)$$

De aquí se tiene que los valores resultantes  $C_x =888.0206016$  y  $C_y =3972.72374$  los cuales al ser divididos por el área total resultan en las dimensiones del centro de masas expresando en las ecuaciones (68) y (69).

$$X_T = \frac{C_X}{A_T} \quad (68)$$

$$Y_T = \frac{C_Y}{A_T} \quad (69)$$

Siendo los valores resultantes  $X_T = 3.8$  y  $Y_T = 17$  los cuales indican que el centro de masas se ubica en estos puntos ya que la distribución del mismo material en el eslabón es uniforme.

### 3.4.1 Cálculo del momento de Inercia del eslabón

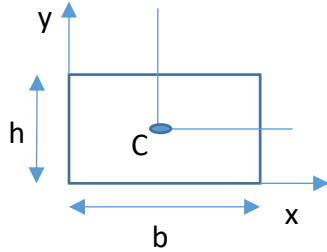
Para el cálculo del momento de inercia se tienen en cuenta las ecuaciones (70) y (71)

$$I_{xarea} = \sum_{i=1}^n (\overline{I_{xn}} + A_i d_{xi}^2) \quad (70)$$

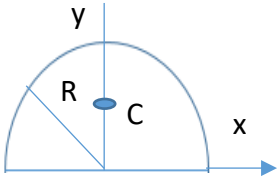
$$I_{yarea} = \sum_{i=1}^n (\overline{I_{yn}} + A_i d_{yi}^2) \quad (71)$$

Donde  $d_{xi}$  y  $d_{yi}$  son la diferencia entre el centro de masas de la sección y el centro de masas principal. El otro valor pertenece al momento de inercia del área de las distintas formas en las que se divide el eslabón, por lo que se calcula para ambos ejes  $\overline{I_{xn}}$  y  $\overline{I_{yn}}$  aplicando las ecuaciones de en la tabla 4.

Tabla 4 Cálculos de inercia para complementos

Forma	Figura	Momento de inercia del área
Rectángulo	 <p>El diagrama muestra un rectángulo con ejes de coordenadas x e y. El eje x es horizontal y el eje y es vertical. La altura del rectángulo es denotada como 'h' y la anchura como 'b'. El centro de masa, etiquetado como 'C', está ubicado en el centro del rectángulo. Se muestran líneas de proyección desde el centro de masa hacia los ejes x e y.</p>	$\overline{I_{xn}} = \frac{bh^3}{12}$ $\overline{I_{yn}} = \frac{b^3h}{12}$



Semicírculo		$\bar{I}_{xn} = \bar{I}_{yn} = \frac{\pi R^4}{8}$
-------------	---	---

Ya calculado el momento de inercia del área de cada una de las secciones, para aplicar el teorema de los ejes paralelos se calcula la distancia entre los centroides de cada una de estas secciones al centro de masa, la cual queda definida en las ecuaciones (72) y (73).

$$d_{x_i} = x_i - X_T \quad (72)$$

$$d_{y_i} = y_i - Y_T \quad (73)$$

Calculando los momentos de inercia de las áreas mediante secciones y la distancia a los centroides de las semiformas; estos quedan expresados como se muestra en la Tabla 5.

Tabla 5 Datos usados para el cálculo del momento de inercia

	$\bar{I}_{xn}$	$d_{xi}$	$\bar{I}_{yn}$	$d_{yi}$	área cm <sup>2</sup>
1	81.8833	0	81.8833	14.97	19.603584
2	965.7472	0	11653	0	194.482464
3	81.8833	0	81.8833	-14.97	19.603584

Sustituyendo estos valores en la ecuación (73) y (74), los valores que se obtienen son  $I_{xarea} = 0.01951$  y  $I_{yarea} = 0.00113$ , con los cuales obtendremos el valor de inercia al multiplicar por la densidad por área del aluminio, la cual es igual 0.162 g/cm<sup>2</sup>.

$$I_x = \rho_{area} \cdot I_{xarea} \quad (74)$$

$$I_y = \rho_{area} \cdot I_{yarea} \quad (75)$$

Además de tener en cuenta la ecuación (75) para el cálculo de la inercia con respecto al eje Z.

$$I_z = I_x + I_y \quad (76)$$

Los valores obtenidos para la inercia de eslabón quedarían definidos como  $I_x= 0.00316029$ ,  $I_y= 0.00018319$  y  $I_z = 0.003341158$ , que pueden representarse en la siguiente matriz.

$$I = \begin{bmatrix} 0.00316029 & 0 & 0 \\ 0 & 0.00018319 & 0 \\ 0 & 0 & 0.003341158 \end{bmatrix}$$

Los valores obtenidos tanto del centro de masas como de las inercias obtenidas para el eslabón pueden compararse con los obtenidos mediante el software Solidworks®, el cual se encuentra en el Anexo B.

### 3.4.2 Estimación de Inercias del motor y carga

Una vez obtenido el valor de inercia de la carga para los tres ejes, para el cálculo del momento de inercia del motor y carga se utiliza la ecuación (77).

$$J = J_m + n^2 J_L \quad (77)$$

Donde  $J_m$  es la inercia del motor,  $n$  es la inversa de la relación de engranaje, el cual se toma de los reductores y  $J_L$  corresponde a la inercia en la carga; donde el valor obtenido es igual  $J = 0.00000710615$ .

### 3.5 Estimación del coeficiente de fricción

Para la estimación del coeficiente de fricción viscosa ( $B_m$ ) se utilizó la ecuación (78).

$$B_m = \frac{k_m I_0}{\theta_0} \quad (78)$$

Donde  $k_m$  representa la constante de torque,  $I_0$  la corriente del motor sin carga y  $\theta_0$  la velocidad del motor sin carga, por lo que el valor obtenido es  $B_m=0.0003059$ , el cual fue similar al valor obtenido por medio del promedio al variar estos tres parámetros.

### 3.6 Valores PID para motor DC

Los valores mostrados en la Tabla 6 se tomaron de la ficha técnica del motor que se encuentra en la sección de anexos B y otros valores se estimaron de acuerdo con lo explicado en las secciones anteriores.

Tabla 6 Valores del motor de DC

Descripción	Notación	Valor	Unidad
<b>Momento de inercia del motor</b>	$J$	7.1E-06	Kg m <sup>2</sup>
<b>Constante de fricción viscosa del motor</b>	$b$	0.0003059	Nms
<b>Constata de fuerza electromotriz</b>	$K_e$	0.0229	V/rad/s
<b>Constante de torque del motor</b>	$K_t$	0.0229	Nm/Amp
<b>Resistencia eléctrica</b>	$R$	0.71	Ohm
<b>Inductancia eléctrica</b>	$L$	0.00066	H

Dichos valores se colocan en el programa en la sección del anexo 6.1.2 siendo los valores resultantes pertenecientes a la función de transferencia que describe el comportamiento del motor con el eslabón. Estos valores resultantes se colocan dentro del bloque función motor de Simulink®, así como un bloque para entrada y se anexa otro para simular una perturbación y de esta forma verificar que el sistema se siga manteniendo.

Dentro del bloque de Control PID se colocan los valores calculados para el sistema o se pueden sintonizar desde la misma planta, en la Figura 17. Se muestra el esquema de Matlab® del PID aplicado al motor.

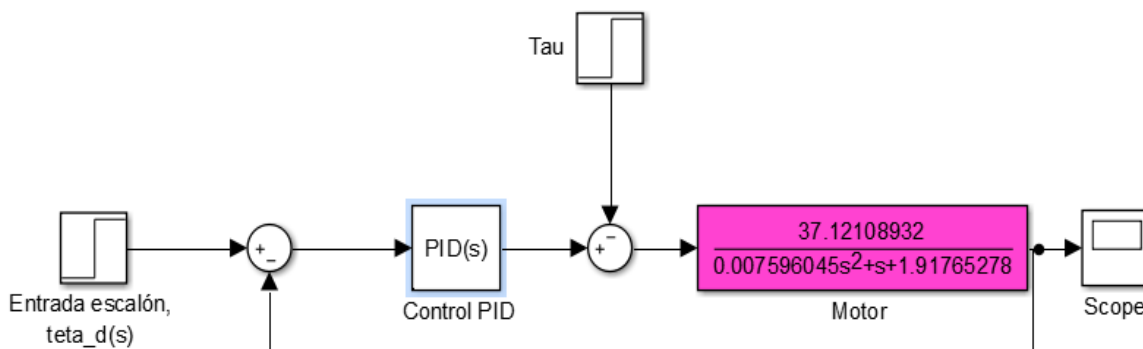


Figura 17. Esquema en Matlab® para calcular valores PID

### 3.7 Datos para la Simulación.

Para la simulación del seguimiento de las trayectorias se utilizaron los parámetros de la Tabla 7 pertenecientes al robot de dos grados de libertad que se base en la dimensiones del prototipo experimental.

Tabla 7 Datos para simulación de un robot de 2GDL

Descripción	Notación	Valor	Unidad
Longitud del eslabón 1	$l_1$	0.264	m
Longitud del eslabón 2	$l_2$	0.224	m
Distancia al centro de masas del eslabón 1	$l_{c1}$	0.132	m
Distancia al centro de masas del eslabón 2	$l_{c2}$	0.112	m
Masa del eslabón 1	$m_1$	0.379	kg
Masa del eslabón 2	$m_2$	0.329	kg
Inercia al centro de masa del eslabón 1	$I_1$	0.00334	Kgm <sup>2</sup>
Inercia al centro de masa del eslabón 2	$I_2$	0.00227	Kgm <sup>2</sup>

#### 3.7.1 Modelo Simulink®

El modelo del sistema se expresa por medio del software Matlab® Simulink®, para ayudar a la simulación de dicho modelo este puede observarse en la Figura 18. En el cual partimos de los archivos de la trayectoria que se desea que la plataforma intente seguir para ambos eslabones, los cuales deben ser presentados en forma de radianes.

La trayectoria se suma algebraicamente con la posición actual y el resultado se multiplica por la matriz de ganancia de posición. Por su parte la velocidad actual

también se multiplica por la matriz de ganancia de velocidad. Estos dos elementos forman el par mecánico aplicado y que se aplica como lo expresa la ecuación (53).

Matlab function es un bloque que permite la colocación de fórmulas de manera más directa, dentro de este bloque se introdujo el modelo matemático descrito en la ecuación (30) para la dinámica y que describe el comportamiento de la plataforma robótica, dentro también se colocan los datos para la plataforma de la tabla 7. Esto se puede observar en la Figura 19.

Para realizar la simulación del sistema el bloque de Matlab function es alimentado por el cálculo de los pares mecánicos definido por  $\tau$ , así como el valor de la velocidad definida por  $dq$  y  $q$  para la posición actual. Con estos se valores realizan los cálculos correspondientes aplicando un despeje de la ecuación (30) para la obtención de la aceleración. Esta aceleración es integrada para la obtención de los parámetros de velocidad y vuelta a integrar para la obtención de la posición, y posteriormente estos parámetros obtenidos volverán a retroalimentar al modelo durante del tiempo del ciclo del sistema, y de esta forma se simula el sistema.

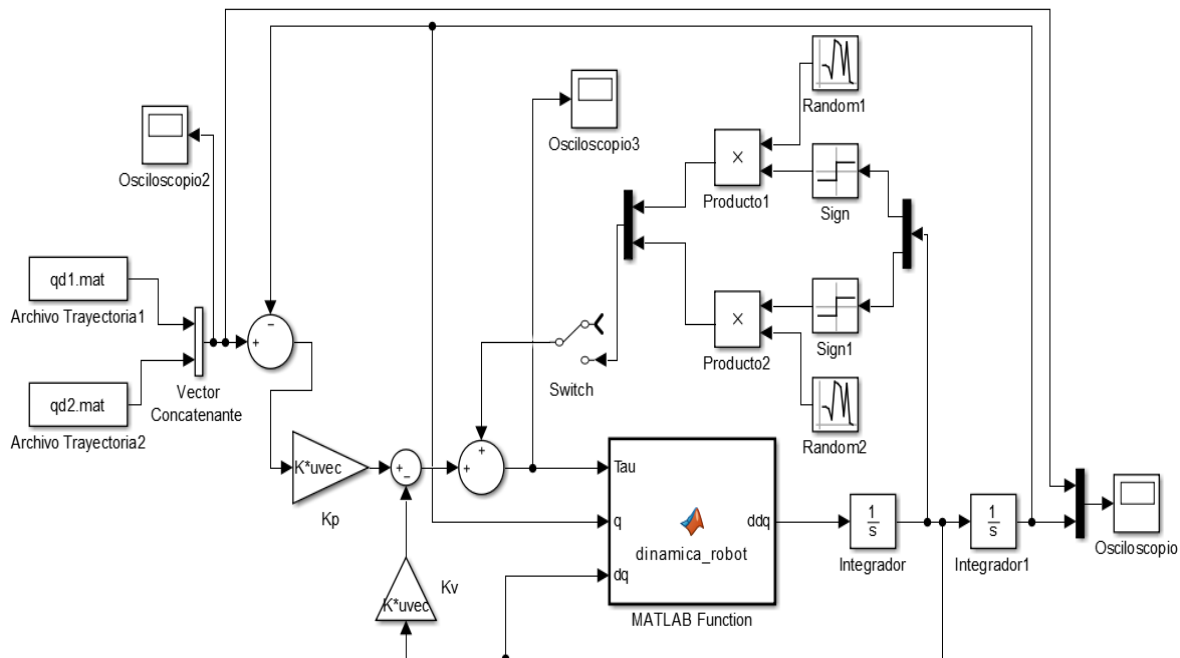


Figura 18. Esquemas de control en Matlab® para el robot de 2GDL

```

function ddq= dinamica_robot(Tau,q,dq)
l1=0.264;%longitud del eslabon 1 en metros
l2=0.224;%longitud del eslabon 2 en metros
lc1=0.132; %distancia al centro de masa del eslabon 1 en metros
lc2=0.112; %distancia al centro de masa del eslabon 2 en metros
m1=0.379; %masa del eslabon 1 en kg
m2=0.329; %masa del eslabon 2 en kg
I1=0.00334; %matriz de inercia relativa al eslabon 1 en kg*m^2
I2=0.00227; %matriz de inercia relativa al eslabon 2 en kg*m^2
g=9.81;% aceleracion de la gravedad en m/s^2
M11=(m1*(lc1^2)) + m2*((l1^2)+(lc2^2)+2*l1*lc2*cos(q(2)))+I1+I2;
M12=(m2*((lc2^2)+l1*lc2*cos(q(2))))+I2;
M21=(m2*((lc2^2)+l1*lc2*cos(q(2))))+I2;
M22=(m2*(lc2^2))+I2;
c11=(-m2*l1*lc2).*sin(q(2)).*dq(2));
c12=(-m2*l1*lc2).*sin(q(2)).*(dq(1)+dq(2));
c21=(m2*l1*lc2).*sin(q(2)).*dq(1);
c22=0;
G1=0;
G2=0;
M=[M11 M12;M21 M22];
C=[c11 c12;c21 c22];
G=[G1;G2];
ddq=M^-1*(Tau-C*dq-G);
end

```

Figura 19. Datos del bloque función Matlab® para robot de 2 GDL

Mientras que para simular el par generado por el brazo humano se tomó la velocidad del robot manipulador como referencia, para la entrada al bloque *signum* y combinado con una señal de una función aleatoria se simulan los diferentes pares que puede generar el paciente  $\tau_{pi}$ . Esto con la intención de simular las fuerzas externas que se ejercen sobre el manipulador.

### 3.8 Medición de Par mecánico

Para poder realizar la principal actividad del proyecto, que consiste en el par que opone el paciente mientras se rehabilita y se entiende que a medida que se mejora su capacidad motriz la oposición es cada vez menor. Para realizar esto es necesario medir el par mecánico y se calcula mediante la ecuación (79)

$$\tau_m = K_m \dot{\alpha} \quad (79)$$

Donde  $K_m$  es el valor de la constante del par motor, este viene del producto de  $K_1$  conocido como el torque del motor (N-m) y  $\phi$  que es el flujo magnético en webers.

Mientras que el valor  $i_a$  es a la corriente en el rotor (también llamado armadura), la cual está dada en Amperes.

Para el cálculo del par mecánico se tomó el valor de la constante de torque de la hoja de especificación del motor y la corriente de la armadura, la cual es el valor que se tiene que estar midiendo por lo que se propone el uso de un sensor de efecto hall.

### 3.8.1 Sensor efecto hall

Se propuso el uso de un sensor efecto hall conocido como ACS712 de 5ª generación mostrador en la Figura 20, debido al grado de exactitud y el relativo bajo costo que presenta, aunque este se puede considerar invasivo ya que es necesario que la corriente tenga que pasar por este.

Un sensor de Efecto Hall es un semiconductor que al estar expuesto a un campo magnético genera un voltaje de salida el cual puede ser leído [59].

El sensor ACS712 consta de tres terminales las cuales requieren una conexión de 5 VCD que va en VCC y GND para la tierra. Cuenta también con una terminal analógica que va directamente al microcontrolador, en esta se coloca la fuente a medir como una entrada y sale hacia la continuación del circuito; internamente trabaja con un sensor de efecto Hall que detecta el campo magnético que se produce por inducción de la corriente que circula por la línea que se está midiendo. El sensor entrega una salida de voltaje proporcional a la corriente.

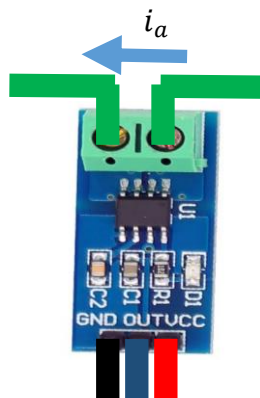


Figura 20. Conexiones para el sensor ACS712

## Capítulo 4

### 4. Resultados

En este capítulo se observarán los resultados tanto de la simulación de las diferentes sintonizaciones de control de la plataforma, como de los resultados obtenidos al simular el seguimiento de las trayectorias en la dinámica del manipulador. Dichos resultados son los pares mecánicos ejercidos por parte del robot planar y también la simulación de este mismo al compensar la reacción del paciente, y por último los datos que se obtuvieron mientras se realizaban las pruebas con la plataforma experimental sugerida.

#### 4.1 Aplicación de PID a un motor de DC

Por medio de la función de transferencia obtenida se simula una entrada de voltaje, al aplicar un control por medio de los valores conseguidos mediante la metodología Ziegler-Nichols de la última ganancia, los cuales son  $K_p = 3.177$ ,  $K_i = 0.0075$  y  $K_d = 0.001875$ , se obtiene una respuesta del comportamiento del sistema, el cual se puede observar en la gráfica de la Figura 21.

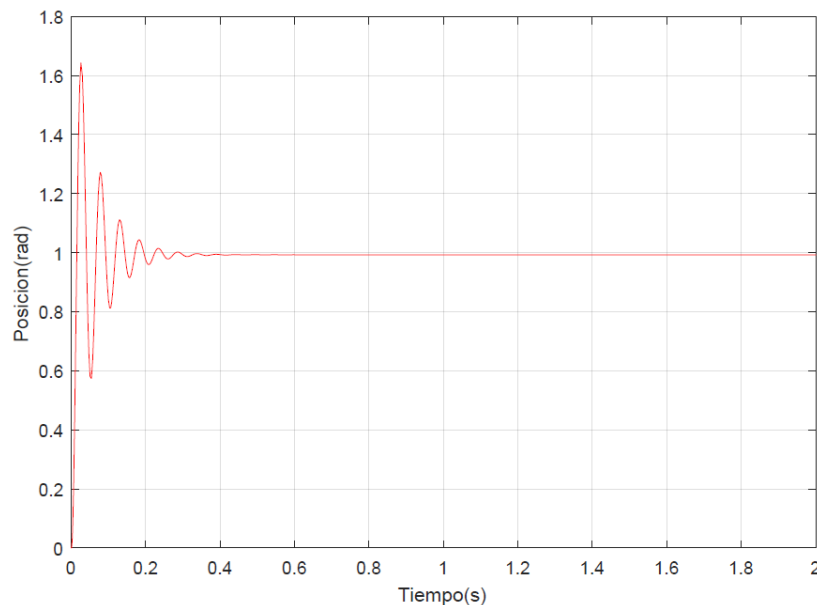


Figura 21. Respuesta del sistema al aplicar control PID utilizando los parámetros Ziegler-Nichols.



En la gráfica de la Figura 22 se puede observar la respuesta por parte del sistema a los parámetros Z-N ajustados, lo cuales son  $K_p = 0.1358$ ,  $K_i = 0.5277$  y  $K_d = 0.0087$

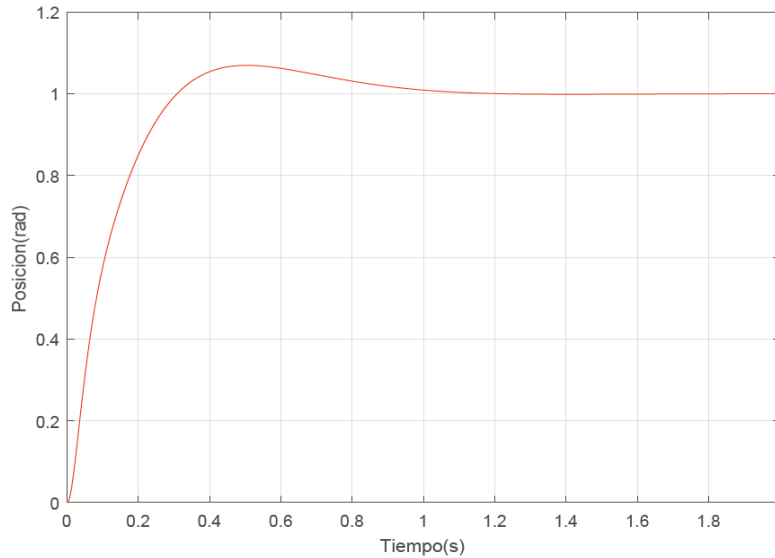


Figura 22. Respuesta al aplicar control PID utilizando los parámetros Ziegler-Nichols ajustados

Posteriormente se utiliza la herramienta de sintonización de PID en el software de Simulink® para obtener nuevos valores de las ganancias en una nueva sintonización, los cuales son  $K_p = 0.51270$ ,  $K_i = 3.7611$  y  $K_d = 0.00643$ . En la Figura 23 se puede ver la respuesta del sistema con estos valores.

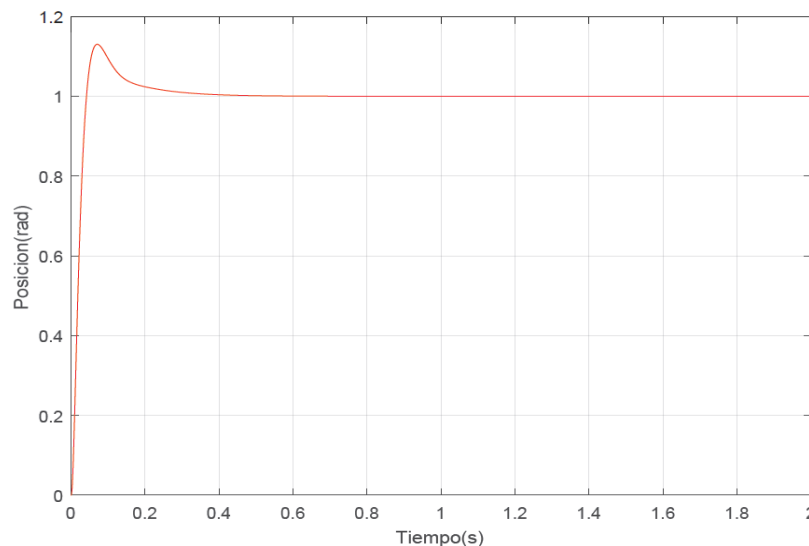


Figura 23. Respuesta al aplicar control PID utilizando el sintonizador de Simulink®

## 4.2 Resultados de la simulación del manipulador

En esta segunda sección se observan los resultados obtenidos por medio de simulación en Simulink® dentro del software Matlab®.

Como se puede ver en la gráfica de la Figura 24 la simulación de diferentes movimientos planteados para una rehabilitación siendo a) trayectoria polinómica, b) una línea desde el origen y c) una trayectoria de un círculo de radio 0.1m, las cuales estará siguiendo tanto el robot manipulador como el brazo del paciente.

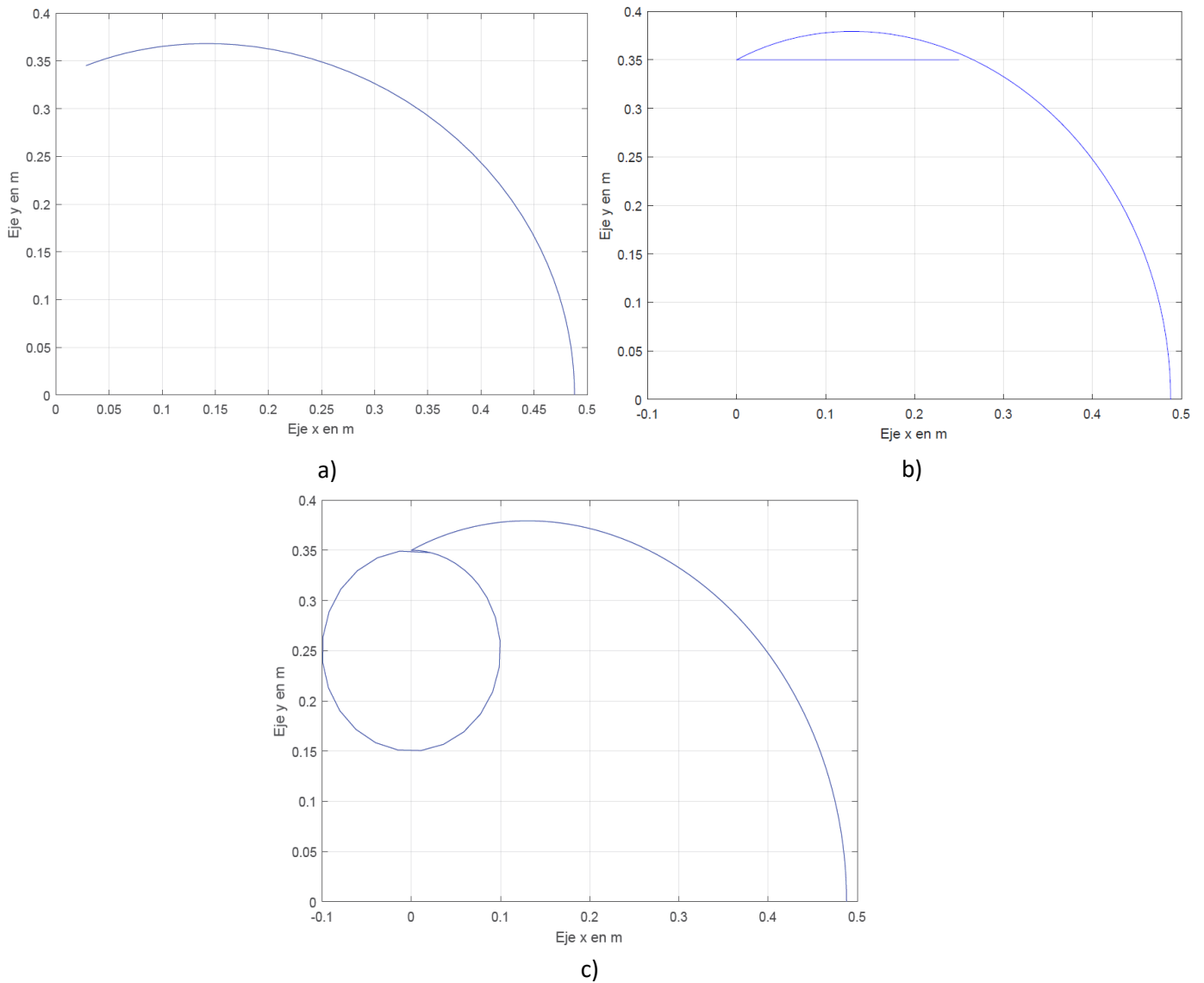


Figura 24. Gráfica de trayectoria en el plano XY en metros  
a) Trayectoria polinómica, b) línea con origen (0,0.3) y c) círculo de radio 0.1m

Las trayectorias deseadas tienen velocidades y aceleraciones angulares muy pequeñas que están determinadas dentro de un tiempo de ciclo en este caso de 5 y 10 segundos respectivamente. En la Figura 25 se aprecia la posición deseada y las posiciones generadas (o actuales) para cada una de las articulaciones dentro de las trayectorias mencionadas anteriormente.

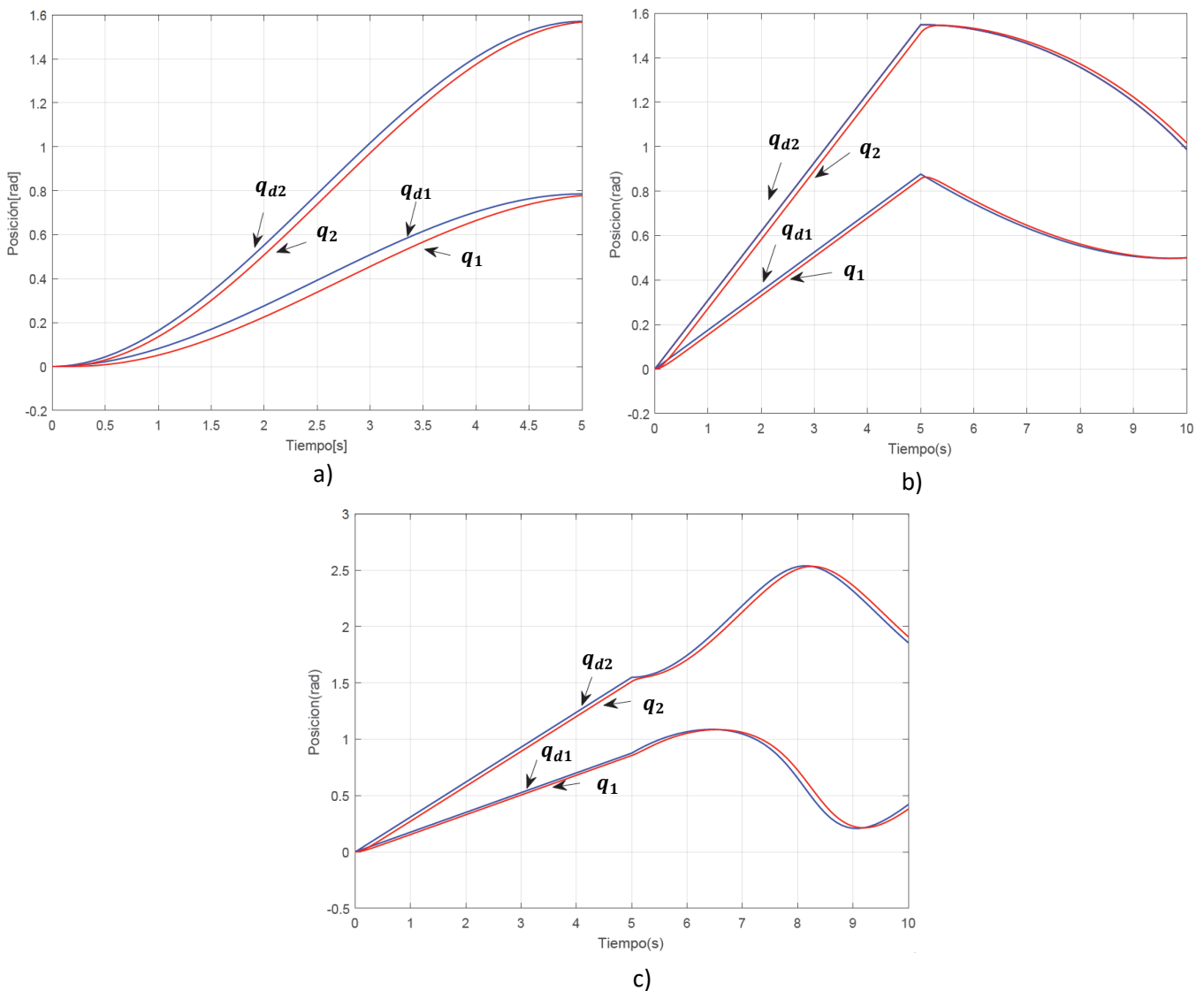


Figura 25. Gráfica de posiciones en radianes  
a) Trayectoria polinómica, b) línea con origen (0,0.3) y c) círculo de radio 0.1m

En las pruebas experimentales, una parte del valor de la posición angular actual se interpreta mediante los sensores como el deseo del paciente de extender o flexionar las articulaciones con una determinada fuerza.

El error de posición  $\tilde{q}_1$  se entiende como la diferencia entre la posición deseada y la actual, para esta sección la simulación de este caso en particular puede verse representado por las tres trayectorias mostradas en la Figura 26.

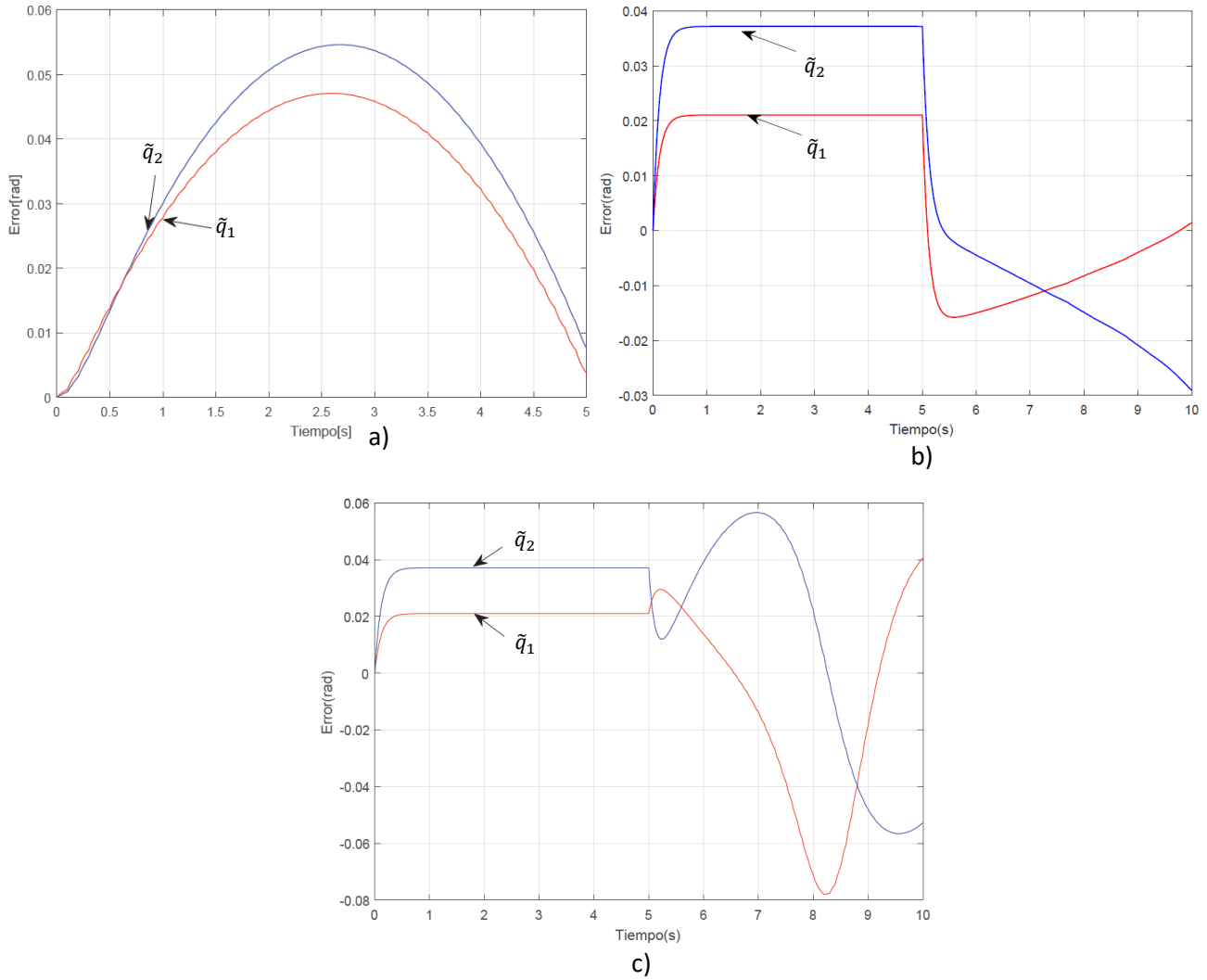


Figura 26. Gráfica del error de posición  
a) Trayectoria polinómica, b) línea con origen (0,0.3) y c) círculo de radio 0.1m

En la Figura 27 se muestran los resultados de la simulación de los pares aplicados por parte del manipulador  $\tau_{ri}$  durante el ciclo.

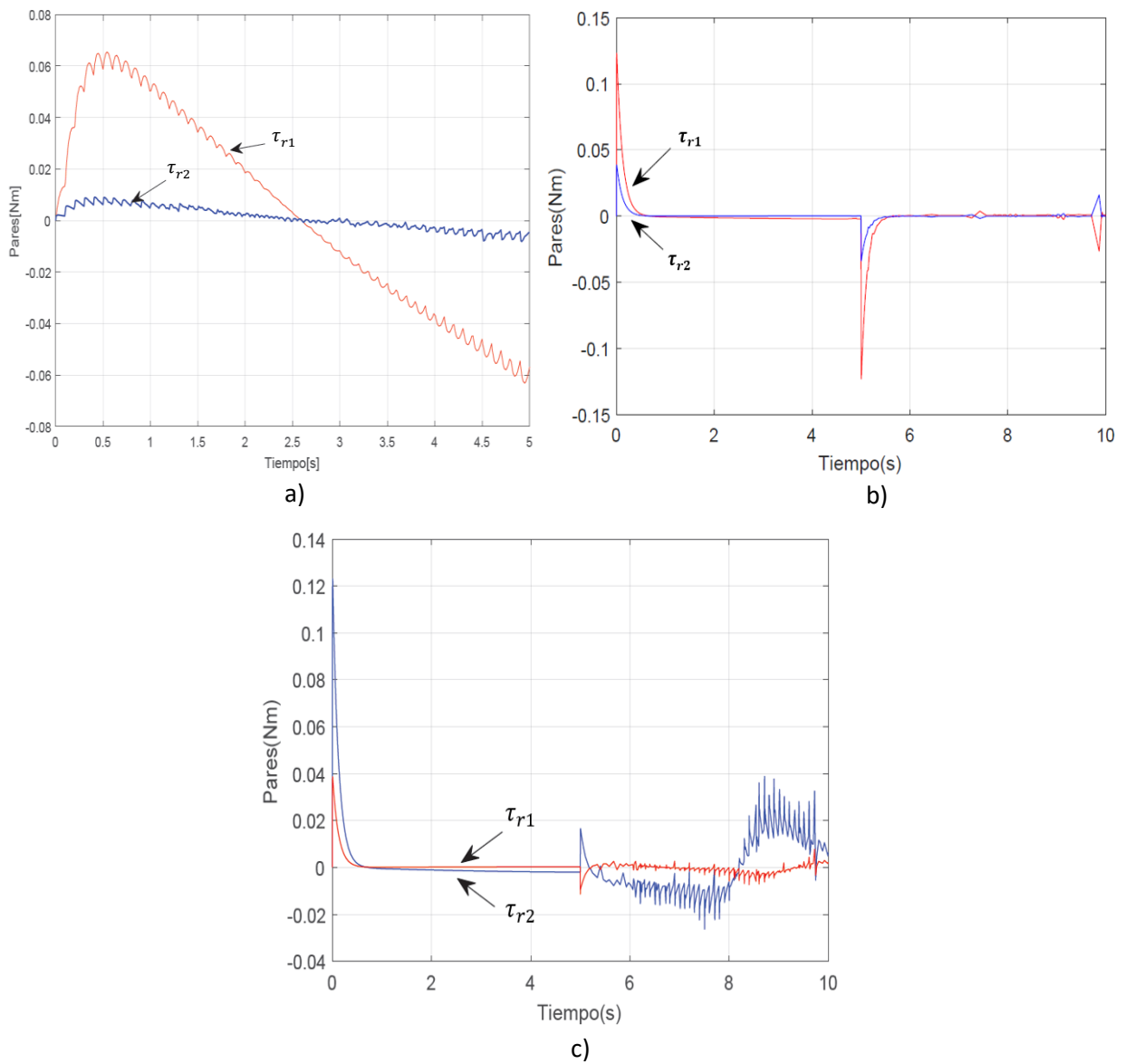
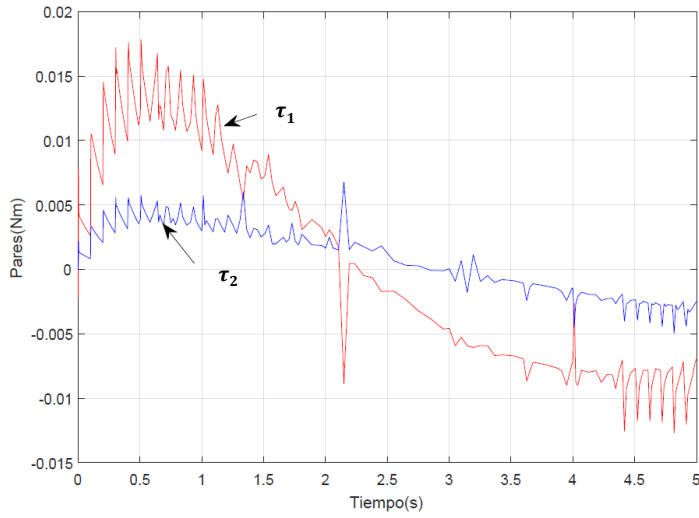
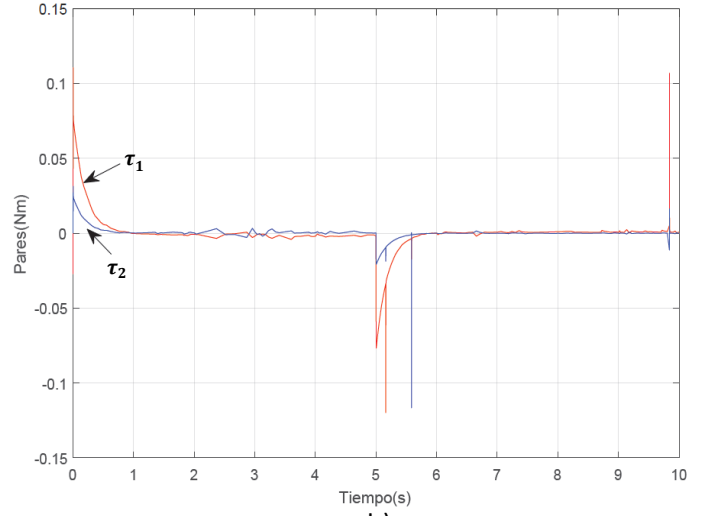


Figura 27. Gráfica (de) pares aplicados por el manipulador  $\tau_{ri}$   
a) Trayectoria polinómica, b) línea con origen (0,0.3) y c) círculo de radio 0.1m

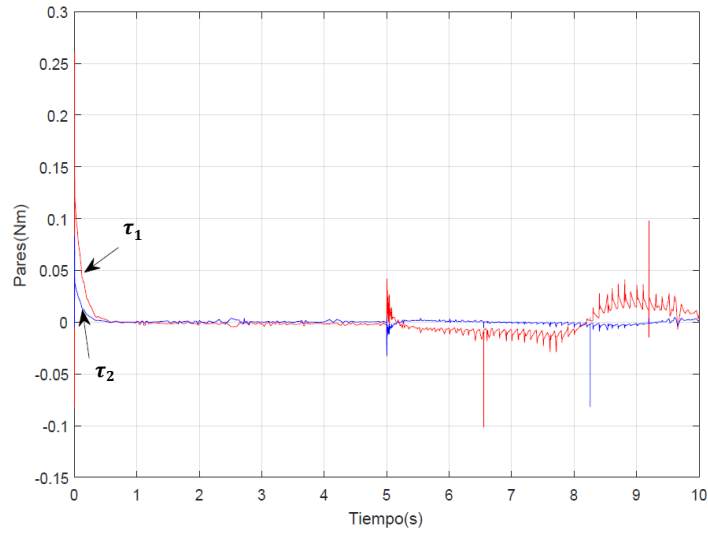
Mientras que en las gráficas de la Figura 28 se pueden ver los resultados de la simulación de los pares resultantes, esto es, pares generados por parte de los actuadores  $\tau_{ri}$  y el par de reacción aplicado por el paciente  $\tau_{pi}$ , que se describe como la reacción del paciente al seguimiento de la trayectoria.



a)



b)



c)

Figura 28. Gráfica (de) pares aplicados por el manipulador  $\tau_{ri}$  y los pares del paciente  $\tau_{pi}$   
a) Trayectoria polinómica, b) línea con origen (0,0.3) y c) círculo de radio 0.1m

### 4.3 Resultados obtenidos de plataforma

En esta sección se pueden observar las gráficas obtenidas a partir de los datos adquiridos durante las distintas pruebas que se realizaron por medio de la plataforma experimental propuesta. En la Figura 29 se pueden observar las posiciones alcanzadas dentro del plano  $XY$  por el robot manipulador según las trayectorias deseadas que se simularon anteriormente. Se pueden ver unas ligeras perturbaciones en el seguimiento de dicha trayectoria en comparación de las obtenidas por medio de la simulación.

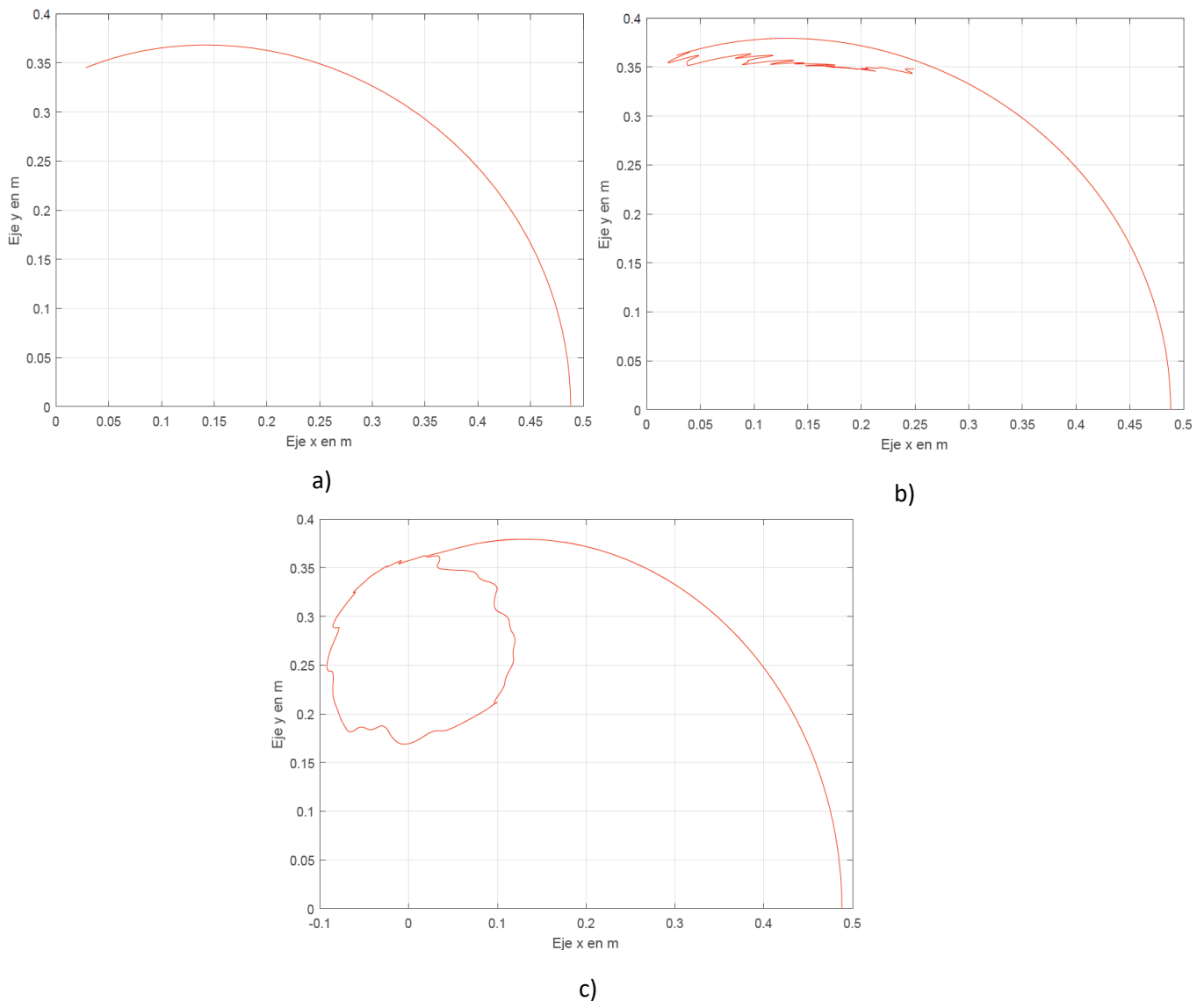


Figura 29. Gráfica de trayectoria obtenida de robot en plano  $XY$   
a) Trayectoria polinómica, b) línea con origen  $(0,0.3)$  y c) círculo de radio  $0.1\text{m}$

En las gráficas de la Figura 30 puede verse el error del seguimiento, por parte del manipulador, obtenido de la diferencia de la trayectoria deseada y la posición actual (posición adquirida por los sensores).

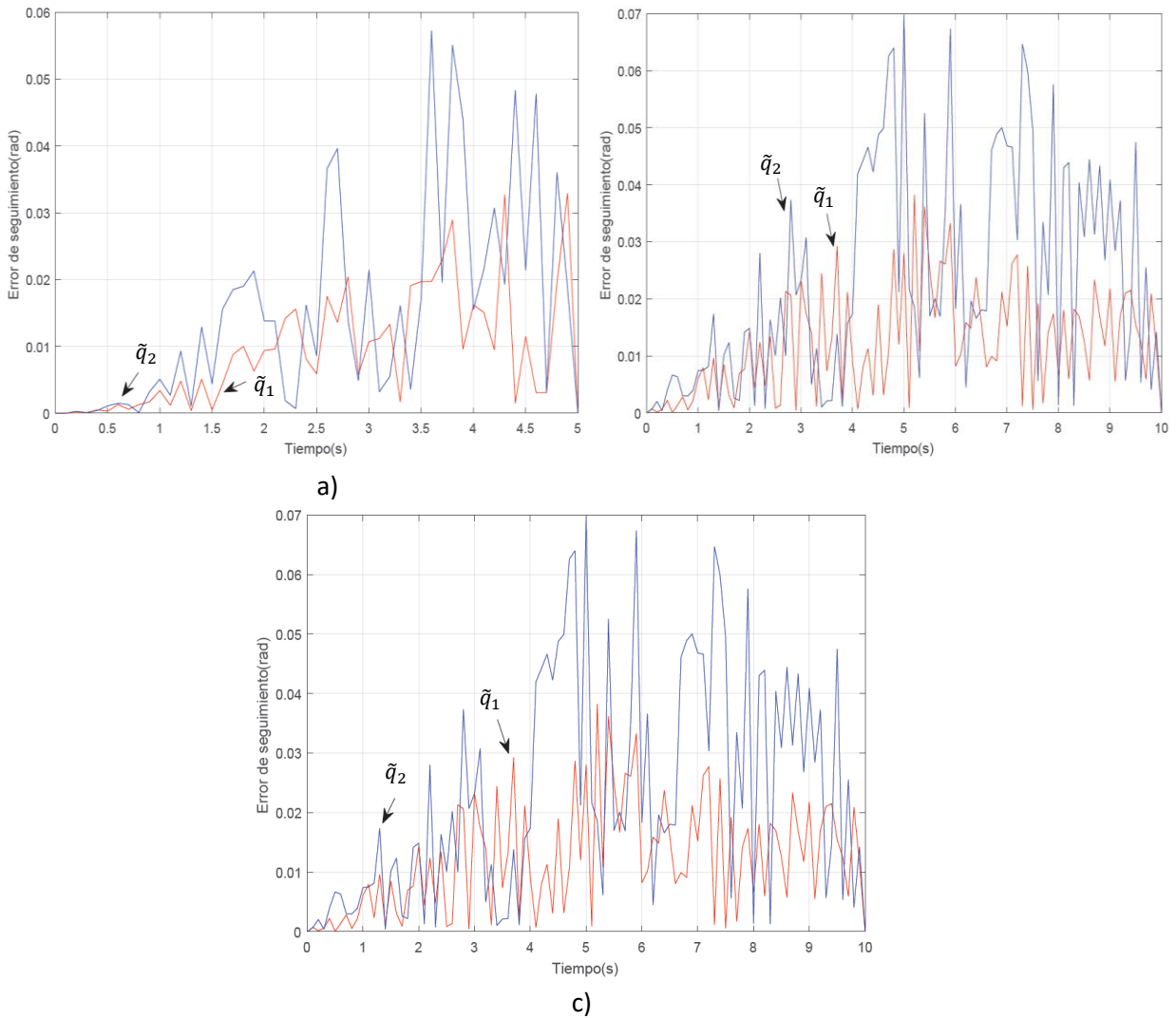


Figura 30. Gráfica de seguimiento de trayectoria  $\tilde{q}_i$  en comparación de robot  
a) Trayectoria polinómica, b) línea con origen (0,0.3) y c) círculo de radio 0.1m

En las gráficas de la Figura 31 pueden observarse los pares obtenidos después de obtener la adquisición de datos del sensor de corriente al aplicar las trayectorias. Los datos difieren significativamente de los obtenidos por la simulación en la Figura 27, ya que las señales adquiridas presentan ruido agregado, la inclusión del sensor de efecto para monitorear el par mecánico, también aporta parte de esta perturbación.



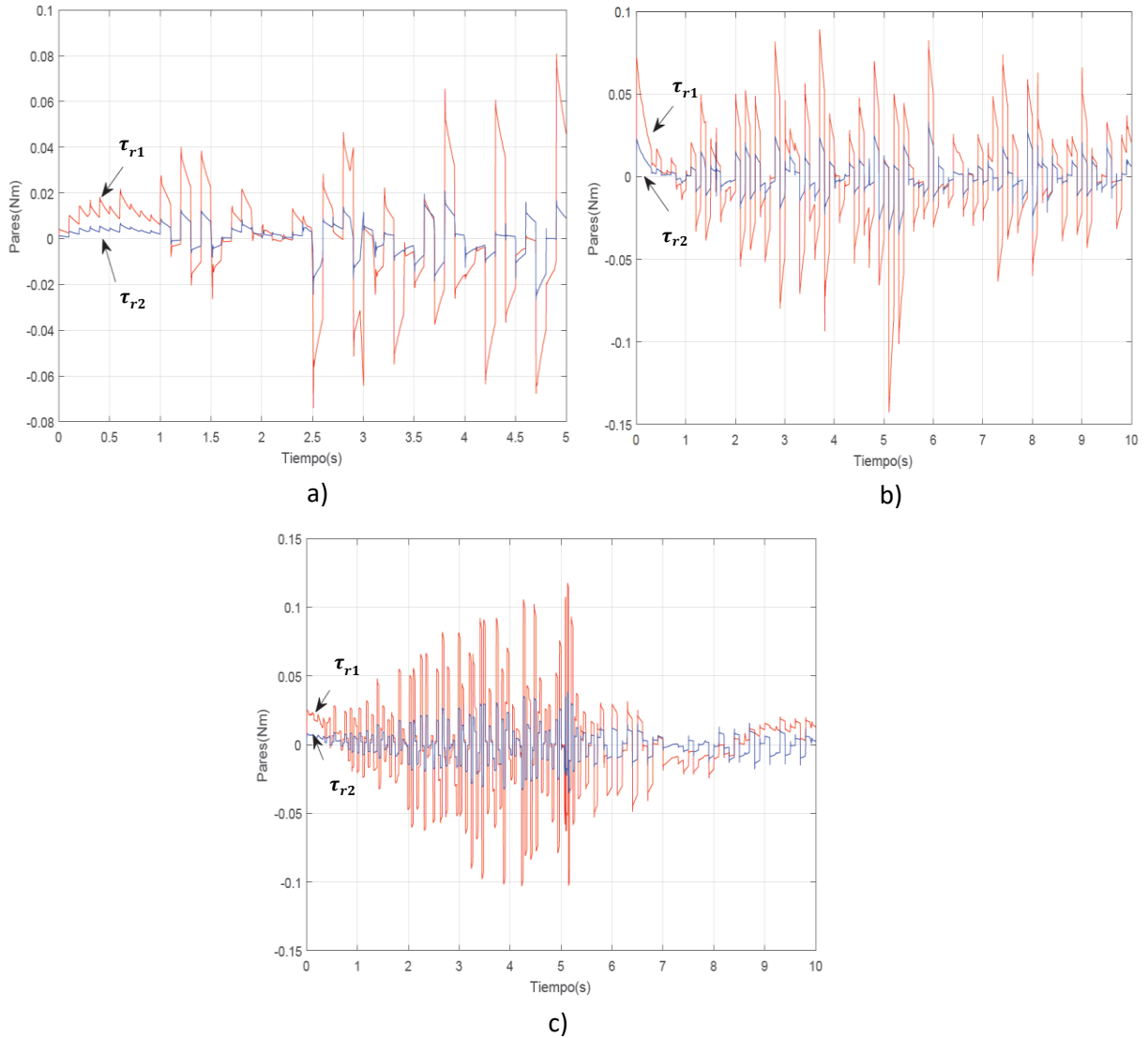


Figura 31. Gráfica estimada de pares aplicados por el manipulador  $\tau_{r_i}$   
a) Trayectoria polinómica, b) línea con origen (0,0.3) y c) círculo de radio 0.1m

En las gráficas de la Figura 32 pueden observarse la estimación de los pares ejercidos por medio del usuario a la plataforma, siendo la diferencia de los pares obtenidos del manipulador haciendo las trayectorias con el usuario, en este caso un paciente sano, y los adquiridos por la realización de la trayectoria normal sin fuerza externa que interactúe que se presentan en la Figura 30.

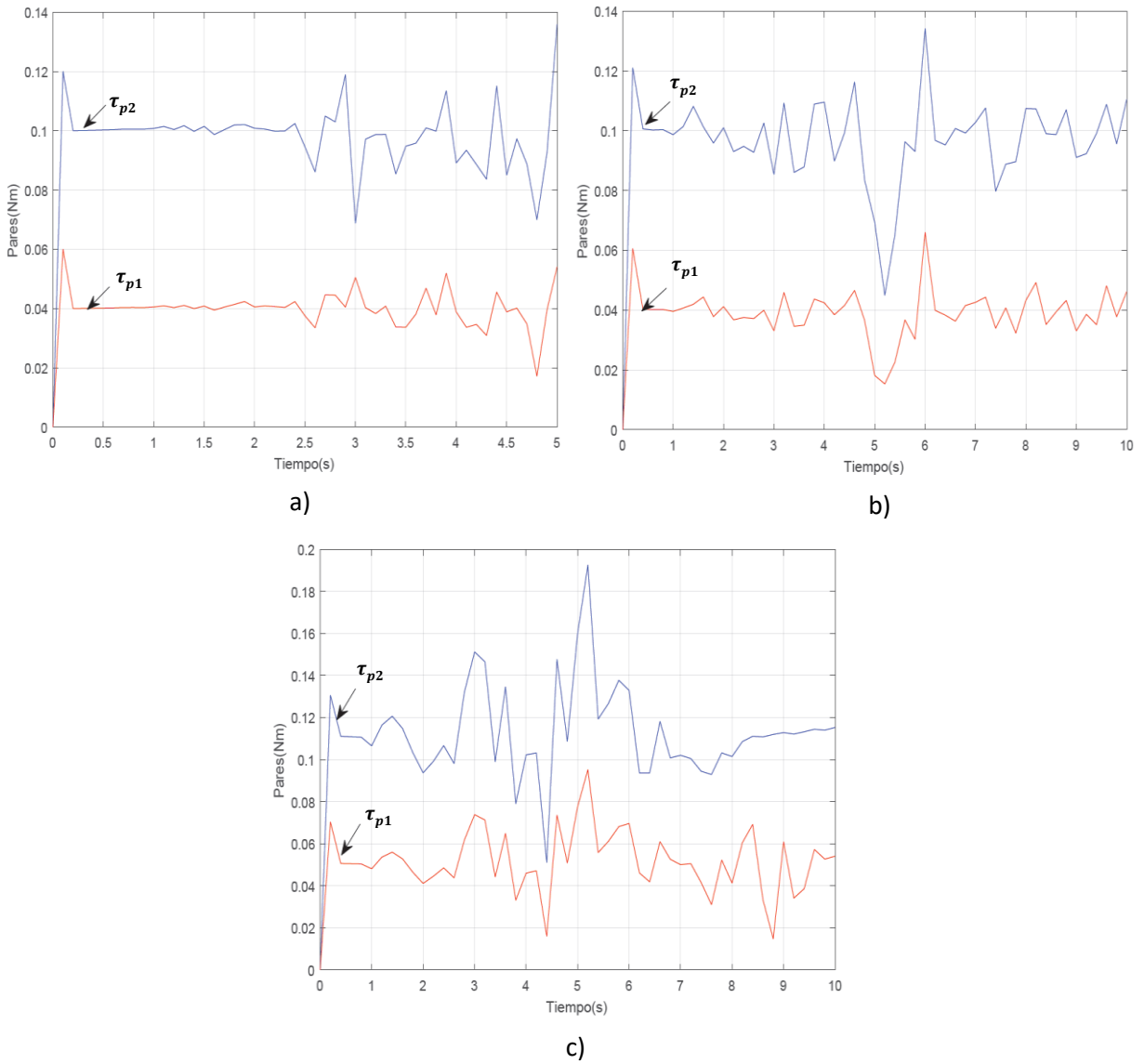


Figura 32. Gráfica estimada de pares aplicados por el paciente  $\tau_{pi}$   
a) Trayectoria polinómica, b) línea con origen (0,0.3) y c) círculo de radio 0.1m

## Capítulo 5

### 9. Conclusiones.

El tema abordado por esta tesis puede ser considerado como una opción a observar a futuro en el área rehabilitación, ya que esta área por lo general está en búsqueda de nuevos proyectos que pueda darle diferentes alternativas. El tema principal de este trabajo es la propuesta de una aplicación práctica a la generación de trayectorias en apoyo a la rehabilitación por medio de un robot de 2 GDL. Se presenta un estudio completo que comprende el análisis matemático del modelo dinámico y el diseño del controlador de un sistema que consiste en un robot de 2 GDL con efector final, que puede ser útil en la aplicación de terapias para rehabilitación. El trabajo, establece un sistema disponible de referencia para futuros proyectos que tenga en cuenta la interacción con el paciente.

Por medio de la simulación de Matlab® se pueden observar las alteraciones de las posiciones y pares adquiridos comparados con las trayectorias deseadas. Esto se presenta de manera natural dado que el paciente, por su problema, ejerce. Se espera que esto tienda a minimizarse conforme se lleven a cabo las sesiones de terapia ya en una aplicación real. Esto quiere decir que, en la medida que el paciente se somete a neuro rehabilitación, al ir recobrando la acción motriz tratará de oponer menos resistencia a la trayectoria ejecutada por el robot.

Para la plataforma experimental se determinó que el control clásico PID es una buena opción para la aplicación el controlador por la simplicidad de su cálculo e implementación en el sistema del microcontrolador, debido a que no requiere una mayor capacidad computacional.

Los resultados obtenidos de la medición de los pares mecánicos durante la experimentación de la plataforma son considerablemente diferentes a los resultantes por medio de la simulación, los cuales se deben en gran parte a dinámicas no modeladas que se omitieron debido al poco tiempo de experimentación que se tuvo con la plataforma.

La variación en los pares mecánicos del manipulador tomando en cuenta los pares externos ejercidos por el paciente es dependiente del tiempo de duración de la trayectoria seleccionada, lo cual se interpreta como una mayor o menor resistencia por parte del paciente.

Los valores obtenidos para la trayectorias tuvieron una diferencia significativa entre un 5% y 7%, dependiendo de la complejidad de la trayectoria esto en comparación a los resultados esperados de la simulación, mientras que los datos obtenidos por medio del sensor para el cálculos de los pares mecánicos presentaron un diferencia en comparación a los obtenidos por medio del modelo, la cual ronda entre un 14% y 16% para las trayectorias establecidas, sin embargo las gráficas para estos datos obtenidos mantienen una similitud a la forma de la gráfica simulada, pero siendo la mayor variabilidad según los datos recopilados para el primer eslabón, todos estos para los valores PID aplicados que mejor obtuvieron resultados los obtenidos por el sintonizador de Simulink.

A pesar de los resultados que se han presentado en base de las investigaciones, aun no existe un método óptimo en la terapia robótica, ya que aún se presentan ciertas cuestiones a resolver como su costo económico y la capacidad de ser transferible más fácilmente [60].

## **10. Trabajo Futuro**

En cuanto al uso del robot manipulador se recomienda que al momento de usar no se le ingresen dimensiones arriba de los 135 grados en ambas orientaciones partiendo de la posición home, debido a que los eslabones tiene la misma distancia el segundo eslabón puede golpear la base principal y forzar al motor; se puede cambiar el primer eslabón por uno de menor dimensión al primer teniendo la sección de la base como referencia para evitar este problema aunque deberán recalcularse los datos presentados nuevamente.

Se recomienda un posterior proyecto para la medición directa de los pares, podría mejorar el alcance del actual proyecto, aunque esto implicaría el desarrollo de una herramienta adaptable al manipulador mencionado en el proyecto. La inclusión de

una forma más adecuada para medir los pares actuales sería lo más recomendable. El uso de sensores del tipo de galgas extensiométricas colocadas en las articulaciones podría ser una opción. El monitoreo del par resultante del sistema en vacío y con un paciente, fue lo implementado en este trabajo con la intención de tener un monitoreo de la mejora en la rehabilitación de los pacientes. Los sensores de clasificación industrial son caros y obligaría al uso de otras tarjetas más sofisticadas.

## Referencias

---

- [1] Recuperado de [www.imss.gob.mx/salud-en-linea/enfermedad-vascular-cerebral](http://www.imss.gob.mx/salud-en-linea/enfermedad-vascular-cerebral). consultada el 25/04/2018.
- [2] Poli P, Morone G, Rosati G, Masiero S. (2013) "Robotic Technologies and Rehabilitation: New Tools for Stroke Patients' Therapy", *BioMed Research International* <http://dx.doi.org/10.1155/2013/153872>.
- [3] Recuperado [www.conacytprensa.mx/index.php/ciencia/salud/3661-rehabilitacion-asistida-por-tics](http://www.conacytprensa.mx/index.php/ciencia/salud/3661-rehabilitacion-asistida-por-tics), consultada el 25/04/2018.
- [4] Hoppenfeld S, Murthy V. (2001), *Fracturas tratamiento y rehabilitación*, Edición Marbán.
- [5] Rodríguez L, Cano R, Cuesta A, Alguacil I, Molina F. (2014) "Terapia robótica para la rehabilitación del miembro superior en patología neurológica", *EISEVIER* <https://doi.org/10.1016/j.rh.2014.01.001>.
- [6] Gopura R, Bandara D, Kiguchi K, Man G. (2015) "Developments in Hardware Systems of Active Upper-Limb Exoskeleton Robots", *EISEVIER* <http://dx.doi.org/10.1016/j.robot.2015.10.001>.
- [7] Perry J, Oblak J, Jung J, Cikajlo I, Veneman J, Goljar N, Bizoviar N, Matja Z, Keller T. (2011) "Variable structure pantograph mechanism with spring suspension system for comprehensive upper-limb haptic movement training", *The Journal of Rehabilitation Research and Development*, 48 (4) 317, doi: 10.1682/JRRD.2010.03.0043.
- [8] Kwakkel G, Kollen B, Krebs H. (2008) "Effects of robot-assisted therapy on upper limb recovery after stroke: a systematic review *Neurorehabilitation and Neural Repair*", Sage, doi:10.1177/1545968307305457.
- [9] Timmermans A, Seelen H, Willmann R, Kingma H. (2009) "Technology-assisted training of arm-hand skills in stroke: concepts on reacquisition of motor control and therapist guidelines for rehabilitation technology design," *Journal of NeuroEngineering and Rehabilitation*, vol. 6, no. 1, article 1
- [10] Poli P, Morone G, Rosati G, and Masiero S. (2013) "Robotic Technologies and Rehabilitation: New Tools for Stroke Patients' Therapy", *Biomed Res Int.*, doi: 10.1155/2013/153872.
- [11] Mani S, Mutha P, Przybyla A, Haaland K, Sainburg R. (2013) "Contralesional motor deficits after unilateral stroke reflect hemisphere-specific control mechanisms", *Brain*, doi: 10.1093/brain/aws283.

- 
- [12] Elliott D, Grierson L, Hayes S, Lyons, J, (2011) "Action representations in perception, motor control and learning: implications for medical education", *Med. Educ.* doi: 10.1111/j.1365-2923.2010.03851.
- [13] Schmidt, R, Lee T, Winstein C, Wulf G, Zelaznik H. (2011) *Motor Control and Learning: A Behavioral Emphasis*, Human Kinetics, Estados Unidos de America.
- [14] Kang N, Cauraugh J. (2015) "Force control in chronic stroke", *Neuroscience & Biobehavioral Reviews*, doi: 10.1016/j.neubiorev.2015.02.005
- [15] Hasegawa Y, Muramatsu M. (2013) "Wearable Lower-limb Assistive Device for Physical Load Reduction of Caregiver on Transferring Support." In *Advanced Intelligent Mechatronics (AIM)*, IEEE. DOI: 10.1109/AIM.2013.6584229
- [16] Wehner M, Rempel D, Kazerooni H. (2009) "Lower Extremity Exoskeleton Reduces Back Forces in Lifting." In *ASME Dynamic Systems and Control Conference*. <https://doi.org/10.1115/DSCC2009-2644>
- [17] Ulrey B, Fathallah F. (2013) "Subject-specific, Whole-body Models of the Stoop Posture with a Personal Weight Transfer Device." *Journal of Electromyography and Kinesiology*. DOI: 10.1016/j.jelekin.2012.08.016
- [18] Loose M, Bosch T, Krause F, Stadler K. (2016) "Exoskeletons for industrial application and their potential effects on physical work load", *Ergonomics*, doi: 10.1080/00140139.2015.1081988.
- [19] D'Adarrio M. (2017) *Mecatrónica Procesos, métodos y sistemas de la Ingeniería industrial*, Editorial Createspace Independent Publishing Platform.
- [20] Sciavicco L, Siciliano B. (2012) *Modelling and Control of Robot Manipulators*, Inglaterra, Edicion Springer-Verlag.
- [21] Ruiz E., Sánchez V. (2007) *Educatrónica: innovación en el aprendizaje de las ciencias y la tecnología*, México, Ediciones Díaz de Santos.
- [22] Maxinez D. (2014), *Programación de Sistemas Digitales con VHDL*, México, Grupo editorial Patria.
- [23] D'Addario M. (2017), *Mecatrónica: Procesos, métodos y sistemas*, España Edición Createspace.
- [24] Ollero A. (2001), *Robótica: manipuladores y robots móviles*, España, Boixareu Editores.
- [25] Niku S. (2010), *Introduction to robotics Analysis, Control, Applications*, Estados Unidos de América, Editorial Wiley.

- 
- [26] Altintas Y. (2000), *Manufacturing Automation*, Estados Unidos de América, Cambridge University Press.
- [27] Poslad S. (2011) *Ubiquitous Computing: Smart Devices, Environments and Interactions*, Editorial Wiley.
- [28] Lantos B, Marton L. (2011) *Nonlinear Control of vehicles and Robots*, Inglaterra, editorial Springer-Verlag.
- [29] Kanade T, Asada H, Reddy R (1981) "Design Concept of Direct-Drive Manipulators Using Rare-Earth DC Torque Motors", citeseerx, doi: 10.1.1.77.3311
- [30] Hogan N, Krebs H, Charnnarong J, Srikrishna P, Sharon A. (1992), "MIT-MANUS: a workstation for manual therapy and training", IEEE, doi: 10.1109/ROMAN.1992.253895.
- [31] Torres J. (2016) *Manual de prácticas de la tecnología de la fabricación*, España, Editorial universidad de Almeida.
- [32] Hartenberg R, Denavit J. (1968) *Kinematic Synthesis of Linkages*, Estados Unidos de América, McGraw-Hill Book Company.
- [33] Aubry C. (2012) *Flash Professional CS6 para PC/Mac: Crear animaciones atractivas para la Web*, España, Edición Eni.
- [34] Pérez M, Cuevas E, Zaldivar N. (2014) *Fundamentos de robótica y mecatrónica con MATLAB y Simulink*, España, Editorial RA-MA
- [35] Hernández M, Ortiz M, Calles C, Rodríguez J (2015) *Robótica: Análisis, modelado, control e implementación*, México; Editorial OmniaScience.
- [36] Merchan E. (2000) *Metodología para generación de trayectorias de manipuladores robóticos, su cinemática y dinámica (Tesis Maestría)* recuperado en <http://tesis.ipn.mx/bitstream/handle/123456789/15945>.
- [37] Lagrange J. (1788) *Tratado de Mecánica Analítica*, Francia.
- [38] Kumar S. (2010) *Introducción a la robótica*, India, Editorial McGraw-Hill Companies Inc.
- [39] Cañas J. (2003) *Jerarquía dinámica de esquemas para la generación de comportamiento autónomo*, Tesis Doctoral, Universidad Politécnica de Madrid, España.
- [40] Ogata K. (2003) *Ingeniería de control moderna*, España, cuarta edición, Editor Pearson Educación.



---

[41] Acedo J. (2003) *Control Avanzado de Procesos: (Teoría y Práctica)*, España, Edición Díaz de Santos.

[42] Morales A, Cortes C, Toletino M, Mendez F. (2009) “Control de un Robot planar de 2 GDL”, consultado: 05/05/2020  
[http://amca.mx/memorias/amca2009/articulos/amca2009\\_35.pdf](http://amca.mx/memorias/amca2009/articulos/amca2009_35.pdf)

[43] Reyes F. (2011) *Robótica: control de robots manipuladores*, México, Primera Edición, Alfaomega Grupo Editor

[44] Ogata K. (2011) *Sistemas de control en tiempo discreto*, México, Primera Edición, Prentice Hall Hispanoamérica, México

[45] Kelly R, Santibañez V, Loria A. (2005) *Control of robot manipulators in joint space*, Londres, Springer-Verlag.

[46] Spong M, Hutchinson S, Vidyasagar M. (2008) *Robot modeling and control*, Estados Unidos de América, Wiley.

[47] Spong M, Vidyasagar M. (2008) *Robot dynamics and control*, Estados Unidos de América, Wiley India.

[48] Merola A, Colacino D, Cosentino C, Amato F. (2017) “Model-based tracking control design, implementation of embedded digital controller and testing of a biomechatronic device for robotic rehabilitation”, Elsevier Ltd  
<https://doi.org/10.1016/j.mechatronics.2018.04.006>.

[49] Gopura R, Bandara D, Kiguchi K, Man G. (2015) “Developments in Hardware Systems of Active Upper-Limb Exoskeleton Robots: A Review”, Elsevier Ltd  
<http://dx.doi.org/10.1016/j.robot.2015.10.001>.

[50] Liu Y, Li C, Ji L, Bi S, Zhang X, Huo J, Ji, R. (2017) “Development and Implementation of an End-Effector Upper Limb Rehabilitation Robot for Hemiplegic Patients with Line and Circle Tracking Training”, *Journal of Healthcare Engineering*, doi:10.1155/2017/4931217.

[51] Saadatzi M, David C, Ozkan C. (2015) “Torque estimation in a wrist rehabilitation robot using a nonlinear disturbance observer”. *ASME*  
<https://doi.org/10.1115/DSCC2015-9717>.

[52] Peng L, Zeng-Guang H, Wang W. (2015) “Dynamic Modeling and Control of a Parallel Upper-Limb Rehabilitation Robot”, *IEEE*, doi: 10.1109/ICORR.2015.7281254.

[53] Nader N, Mohammad J. (2015) “Design and control of an active 1-DoF mechanism for knee rehabilitation”, *Inform healthcare*  
<https://doi.org/10.3109/17483107.2015.1027299>.

- 
- [54] Nielsen J, Stengaard A, Christensen T, Rajeeth T, Kulvicius T.(2017) "Individualized and adaptive upper limb rehabilitation with industrial robot using dynamic movement primitives" *Icra*.
- [55] Mancisidor A, Zubizarreta A, Cabanes I, Bengoa P, Jung J. (2017) "Kinematical and dynamical modeling of a multipurpose upper limbs rehabilitation robot", Elsevier Ltd. <http://dx.doi.org/10.1016/j.rcim.2017.08.013>.
- [56] Xiang L, Yun-Hui L, Haoyong Y. (2017) "Iterative learning impedance control for rehabilitation robots driven by series elastic actuator", Elsevier <https://doi.org/10.1016/j.automatica.2017.12.031>.
- [57] Mancisidor A, Zubizarreta A, Cabanes I, Bengoa P, Jung J. (2017). "Kinematical and dynamical modeling of a multipurpose upper limbs rehabilitation robot". Elsevier Ltd. <http://dx.doi.org/10.1016/j.rcim.2017.08.013>.
- [58] Craig J. (2006) *Robótica*, México, Editor Pearson Educación.
- [59] Boylestad R. (2004) *Introducción al análisis de circuitos*, México, Editor Pearson Educación.
- [60] Loeza P. (2015) "Introducción a la rehabilitación robótica para el tratamiento de la enfermedad vascular cerebral: revisión", *Revista Mexicana de Medicina Física y Rehabilitación*, <https://www.medigraphic.com/pdfs/fisica/mf-2015/mf152c.pdf> revisado: 30/01/2020

---

## Apéndices

### Apéndice A. Programación

#### 6.1.1 Programa principal

```
// Contiene el control de velocidad del motor de acuerdo con la posición
//
// Autor: Fernando Monrreal
//
// Revision:
// 06/05/20 – Archivo creado

#include <SD.h>
File myFile;

//inializacion de las variables
volatile long int encoder_pos = 0;
volatile long int encoder_pos2 = 0;
////////////////////posiciones en arduino de puentes H////////////////
int LPWM=4;
int RPWM=5;
int L_EN=6;
int R_EN=7;
int LPWM2=8;
int RPWM2=9;
int L_EN2=10;
int R_EN2=11;
////////////////////valores para motores////////////////
int motor_value2 = 255;
int motor_value = 255;
////////////////////Constantes de posicion home////////////////
int PULSADOR = 12; // Pin digital para el pulsador
int LED = 13; // Pin digital para el pulsador
int encender = 0;
int anterior = 0;
int estado = 0;
////////////////////Constantes de trayectoria////////////////
long int datos[n] ={}; //se coloca la trayectoria a seguir para q1
long int datos2[n] ={}; //se coloca la trayectoria a seguir para q2
////////////////////Constantes de tiempo //////////////////
int T1=n; //se coloca el tiempo en segundos para q1
int T2=n; //se coloca el tiempo en segundos para q2
//////////////////// Variables control de velocidad //////////////////
int i;
int j;
long int q1;
long int q2;
int a=0;
```

---

```

int sentido;
int b=0;
int sentido2;
int c=0;
int d=0;
long int regreso1;
long int regreso2;
int T=2;
int n=4;
double tiempo;
float Setpoint;
float Setpoint2;
//////////////////////Constantes para el PID //////////////////
double kp=1;
double ki=0;
double kd=0;
////////////////////// Variables para el PID general //////////////////
float errorx,error_prevx,errorx2,error_prevx2;
float tiempotrans, time, tiempoprev,tiempotrans2, time2, tiempoprev2;
float PIDM1=0;
float ITerm=0;
float dInput=0;
float PIDM2=0;
float ITerm2=0;
float dInput2=0;
//////////////////////
////////////////////// Variables para el Encoder //////////////////

void setup() {

  Serial.begin(57600);
  pinMode(2, INPUT);      //conectamos el canal B del encoder primer motor
  pinMode(3, INPUT);      //conectamos el canal A del encoder primer motor
  pinMode(19, INPUT);     //conectamos el canal B del encoder segundo motor
  pinMode(18, INPUT);     //conectamos el canal A del encoder segundo motor
  pinMode(4, OUTPUT);     // Salida PWM primer motor
  pinMode(5, OUTPUT);     //Salida PWM primer motor
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);     // Salida PWM segundo motor
  pinMode(9, OUTPUT);     // Salida PWM segundo motor
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);

  attachInterrupt(digitalPinToInterrupt(2), encoder, RISING); //cuenta los pulsos del encoder
  attachInterrupt(digitalPinToInterrupt(19), encoder2, RISING); //cuenta los pulsos del encoder
  pinMode(PULSADOR, INPUT); // Pin digital del pulsador como entrada
  pinMode(LED, OUTPUT); // Pin digital del LED como salida
  digitalWrite(LED, LOW); // LED apagado

```



---

```

    myFile.println(encoder_pos);
    //////////////////////////////////////
    myFile.close();//cierra memoria SD
    goto prueba3;
}
}else{
prueba4:
if((encoder_pos)<=(q1)){
i++;
a=q1;
motor_value = 0;
if(motor_value > 0){
MotorCounterClockwise(motor_value);
}else{
MotorClockwise(abs(motor_value));
}
delay(1);
goto prueba1;
}else{
delay(1);
myFile = SD.open("datalog.txt", FILE_WRITE); //abre memoria SD
//guarda valores en memoria SD//
myFile.print("Tiempo(ms)=");
myFile.println(millis());
myFile.print(", encoder_pos=");
myFile.println(encoder_pos);
////////////////////////////////////
myFile.close();//cierra memoria SD
goto prueba4;
}
}
}
////////////////////////////////////segundo motor////////////////////////////////////
j=0;
prueba21:
delay(1);
q2=datos2[j]*(-1); //se toman datos de la matriz de posiciones para el segundo motor
sentido2=q2-b; //registra la orientación para el botón home para el segundo motor
regreso2=q2+b; //registra la posición para el botón home para el segundo motor
while(j<=50){
Setpoint2=(q2); //toma el valor de los encoder para el segundo motor
tiempoprev2 = time2; //El tiempo anterior es guardado antes que la lectura del tiempo actual
time2=millis(); //Lectura del tiempo actual
tiempotrans2=(time2 - tiempoprev2)/1000;
controlmotor2();
error_prevx2=errorx2;
motor_value2 = PIDM2; //marca el valor de la velocidad del segundo motor
if(motor_value2 > 0){
MotorCounterClockwise2(motor_value2); //marca la orientación en sentido contrario a las
manecillas
}else{
MotorClockwise2(abs(motor_value2)); //marca la orientación en sentido de las manecillas

```

---

```

}
if(sentido2<=0){
  prueba23:
  if((encoder_pos2)<=(q2)){
    j++;
    b=q2;
    // en esta parte del programa se debera establecer el set point en pulsos del encoder, la relacion es
    // de 736.78:1 x 500 pulsos por revolucion siendo 368,390 pulsos por una vuela completa
    motor_value2 = 0;
  if(motor_value2> 0){
    MotorCounterClockwise2(motor_value2); //marca la orientación en sentido contrario a las
    manecillas
  }else{
    MotorClockwise2(abs(motor_value2)); //marca la orientación en sentido de las manecillas
  }
  }
  delay(1);
  myFile = SD.open("datalog2.txt", FILE_WRITE); //abre memoria SD
  //guarda valores en memoria SD//
  myFile.print("Tiempo(ms)=");
  myFile.println(millis());
  myFile.print(", encoder_pos2=");
  myFile.println(encoder_pos2);
  //////////////////////////////////////
  myFile.close();//cierra memoria SD
  goto prueba21;
  }else{
  delay(1);
  goto prueba23;
  }
  }else{
  prueba24:
  if((encoder_pos2)>=(q2)){
    j++;
    b=q2;
    motor_value2 = 0;
  if(motor_value2> 0){
    MotorCounterClockwise2(motor_value2);
  }else{
    MotorClockwise2(abs(motor_value2));
  }
  }
  delay(1);
  goto prueba21;
  }else{
  delay(1);
  myFile = SD.open("datalog2.txt", FILE_WRITE); //abre memoria SD
  //guarda valores en memoria SD//
  myFile.print("Tiempo(ms)=");
  myFile.println(millis());
  myFile.print(", encoder_pos2=");
  myFile.println(encoder_pos2);
  //////////////////////////////////////
  myFile.close();//cierra memoria SD

```

---

```

goto prueba24;
}
}
}
//////////regresión de posición a home//////////
regreso1=-1*regreso1;
regreso2=-1*regreso2;
prueba2:
volatile long int encoder_pos1 = 0;
volatile long int encoder_pos2 = 0;
motor_value = 0;
if(motor_value > 0){
  MotorCounterClockwise(motor_value);
}else{
  MotorClockwise(abs(motor_value));
}
delay(1); // en esta parte del programa se debera establecer el set point en pulsos del
encoder, la relacion es de 736.78:1 x 500 pulsos por revolucion siendo 368,390 pulsos por una
vuela completa
motor_value2 = 0;
if(motor_value2 > 0){
  MotorCounterClockwise2(motor_value2);
}else{
  MotorClockwise2(abs(motor_value2));
}
delay(1);
if(digitalRead(PULSADOR) == HIGH){
  Setpoint=(regreso1);
  tiempoprev = time; //El tiempo anterior es guardado antes que la lectura del tiempo actual
  time=millis(); //Lectura del tiempo actual
  tiempo2=(time - tiempo2)/1000;
  controlmotor1();
  error_prevx=errorx;
  motor_value = PIDM1;
  if(motor_value > 0){
    MotorCounterClockwise(motor_value);
  }else{
    MotorClockwise(abs(motor_value));
  }
  delay(1);

  Setpoint2=(regreso2);
  tiempo2=millis(); //El tiempo anterior es guardado antes que la lectura del tiempo actual
  tiempo2=millis(); //Lectura del tiempo actual
  tiempo2=(time2 - tiempo2)/1000;
  controlmotor2();
  error_prevx2=errorx2;
  motor_value = PIDM2;
  if(motor_value2 > 0){
    MotorCounterClockwise2(motor_value2);
  }else{
    MotorClockwise2(abs(motor_value2));
  }
}

```



---

```

}
  delay(1);
bailout:
  motor_value =0;
if(motor_value > 0){
  MotorCounterClockwise(motor_value);
}else{
  MotorClockwise(abs(motor_value));
}
  delay(1);

  motor_value2 = 0;
if(motor_value2 > 0){
  MotorCounterClockwise2(motor_value2);
}else{
  MotorClockwise2(abs(motor_value2));
}
  delay(1);
  goto bailout;
}
goto prueba2;
}
////////////////////////////////////
void encoder(){
  if(digitalRead(3) == LOW){
    encoder_pos++;
  }else{
    encoder_pos--;
  }
}

void encoder2(){
  if(digitalRead(18) == LOW){
    encoder_pos2++;
  }else{
    encoder_pos2--;
  }
}

void MotorClockwise(int power){
  if(power > 20){
    digitalWrite(R_EN,HIGH);
    digitalWrite(L_EN,HIGH);
    analogWrite(RPWM,power);
  }else{
    digitalWrite(LPWM, LOW);
    digitalWrite(RPWM, LOW);
  }
}

void MotorCounterClockwise(int power){
  if(power > 20){

```

---

```

    digitalWrite(R_EN,HIGH);
    digitalWrite(L_EN,HIGH);
    analogWrite(LPWM,power);
}else{
    digitalWrite(RPWM, LOW);
    digitalWrite(LPWM, LOW);
}
}

void MotorClockwise2(int power2){
    if(power2 > 20){
        digitalWrite(R_EN2,HIGH);
        digitalWrite(L_EN2,HIGH);
        analogWrite(RPWM2,power2);

    }else{
        digitalWrite(LPWM2, LOW);
        digitalWrite(RPWM2, LOW);
    }
}

void MotorCounterClockwise2(int power2){
    if(power2 > 20){
        digitalWrite(R_EN2,HIGH);
        digitalWrite(L_EN2,HIGH);
        analogWrite(LPWM2,power2);

    }else{
        digitalWrite(RPWM2, LOW);
        digitalWrite(LPWM2, LOW);
    }
}

void xyz() {
    if (estado && anterior == 0) // Comparamos el estado actual y el anterior del pulsador
    {
        encender = 1 - encender;
    }
    anterior = estado; // Actualizamos el estado del pulsador.

    if (encender) // Si el estado interno del pulsador pasa de "LOW" a "HIGH".
    {
        digitalWrite(LED, HIGH); // LED encendido
        delay(300); // Evita los rebotes del pulsador.
    }
    else // Si el estado interno del pulsador pasa de "HIGH" a "LOW".
    {
        digitalWrite(LED, LOW); // LED apagado
    }
}

void controlmotor1()

```

---

```

{
double errorx = Setpoint - encoder_pos;
ITerm+=(ki*errorx);
dInput=kd*((errorx - error_prevx)/tiempotrans);
//El control final es la suma de las 3 partes
PIDM1=kp * errorx + ITerm + dInput;
if(PIDM1 < -150)
{
PIDM1=-150;
}else{
if(PIDM1 > 150)
{
PIDM1=150;
}else{
PIDM1=PIDM1;
}
}
}
}

```

```

void controlmotor2()
{
double errorx2 = Setpoint2 - encoder_pos2;
ITerm2+=(ki*errorx2);
dInput2=kd*((errorx2 - error_prevx2)/tiempotrans2);
//El control final es la suma de las 3 partes
PIDM2=kp * errorx2 + ITerm2 + dInput2;
if(PIDM2 < -150)
{
PIDM2=-150;
}else{
if(PIDM2 > 150)
{
PIDM2=150;
}else{
PIDM2=PIDM2;
}
}
}
}

```

## 6.1.2 Programa para calibración del sensor ACS712

```

// Contiene calibración para el sensor ACS712
//
// Autor: Fernando Monrreal
//
// Revision:
// 06/05/20 – Archivo creado
//////////////////////////////////////Iniciación de variables//////////////////////////////////////
float voltaje_sensor

```

---

```

float volt=0;

void setup() {
  Serial.begin(9600);// velocidad de transferencia
}

void loop() {
  voltaje_sensor =get_volt(10000);//Se obtiene el voltaje del sensor(10000 muestras)
  Serial.print("Voltaje del sensor: ");
  Serial.println(voltaje_sensor ,3);
}

float get_volt(int muestras)
{
  for(int i=0;i<muestras;i++)
  {
    volt =volt+analogRead(A0) * (5.0 / 1023.0);
  }
  volt=volt/muestras;
  return(volt);
}

```

### 6.1.3 Programa Simulink® para función transferencia del motor

```

// Contiene programa para la obtención de función transferencia
//
// Autor: Fernando Monrreal
//
// Revision:
// 06/05/20 – Archivo creado
////////////////////////////////////valores de las constantes del motor////////////////////////////////

J = 0.00000710615;// valor de la inercia
b = 0.0003059; // valor del coeficiente de fricción
K = 0.0229;//Valor de constante de torque
R = 0.71; //Valor de la resistencia
L = 0.00066; //Valor de la inductancia
s = tf('s');// transformación de variables a Laplace
P_motor = K/((J*(s^2)+(b*s))*((L*s)+R)+K^2)

```

---

### 6.1.4 Programa Matlab® para generación de trayectoria polinómica

```
// Contiene programa para la generación de la trayectoria polinómica
//
// Autor: Fernando Monrreal
//
// Revision:
// 06/05/20 – Archivo creado
////////////////////constantes para la trayectoria////////////////
function [poli2]= tray_poliv3(pos_ini,pos_fin,vel_ini,vel_fin,tf)//se colocan valores como posición inicial,
posición final velocidad inicial, velocidad final y tiempo en la forma de vector

//////////////////// variables para la trayectoria //////////////////
syms t a0 a1 a2 a3;

theta=a0 + a1*t + a2*t^2 + a3*t^3;

theta1=diff(theta);
theta2=diff(theta,2);

a0=pos_ini;
a1=vel_ini;
a2=(3/tf^2)*(pos_fin-pos_ini)-(2/tf)*vel_ini-(1/tf)*vel_fin;
a3=(-2/tf^3)*(pos_fin-pos_ini)+(1/tf^2)*(vel_fin+vel_ini);

A=subs(theta);
B=subs(theta1);
C=subs(theta2);

simplify(A);
simplify(B);
simplify(C);

t=0:0.1:tf;

poli2=eval(theta)

time=t;

poli2 = timeseries(poli2, t, 'name', 'Vector1')//nombre deseado del archivo
save poli2 -v7.3 poli2//guardar la trayectoria en tiempo en forma de vector
```

### 6.1.5 Programa Matlab® para solución de la cinemática directa.

```
// Contiene programa para la solución de la cinemática directa
//
// Autor: Fernando Monrreal
//
// Revision:
```

---

```

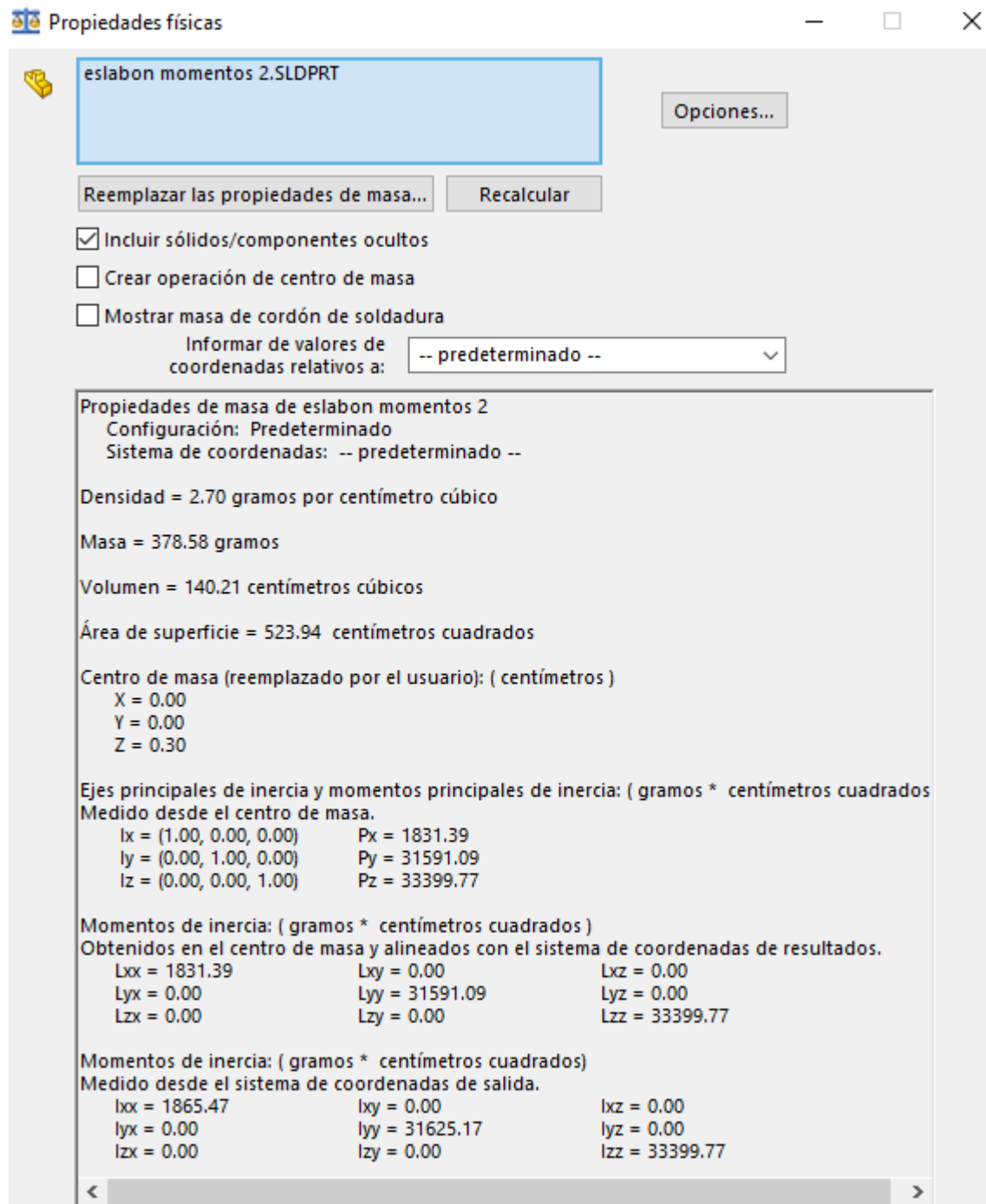
// 06/05/20 – Archivo creado
////////////////////introducción de variables DH////////
function [Px,Py,Pz]=cinematicadirecta(q1,q2)//se introduce los valores de q1 q2
////////////////////constantes DH////////

alfa1 = 0;
alfa2 = 0;
a1 = 0.264;
a2 = 0.224;
d1 = 0;
d2 = 0;
////////////////////variables DH////////
A01=[(cos((q1)*(pi/180)),-
(sin((q1)*(pi/180)))*(cos((alfa1)*(pi/180))),sin((q1)*(pi/180))*(sin((alfa1)*(pi/180))),a1*(cos((q1)*(pi
/180)));sin((q1)*(pi/180)),cos((q1)*(pi/180))*(cos((alfa1)*(pi/180))),-
(cos((q1)*(pi/180)))*(sin((alfa1)*(pi/180))),a1*(sin((q1)*(pi/180))),0,(sin((alfa1)*(pi/180))),cos((alfa
)*(pi/180))),d1;0,0,0,1];
A12=[(cos((q2)*(pi/180)),-
(sin((q2)*(pi/180)))*(cos((alfa2)*(pi/180))),sin((q2)*(pi/180))*(sin((alfa2)*(pi/180))),a2*(cos((q2)*(pi
/180)));sin((q2)*(pi/180)),cos((q2)*(pi/180))*(cos((alfa2)*(pi/180))),-
(cos((q2)*(pi/180)))*(sin((alfa2)*(pi/180))),a2*(sin((q2)*(pi/180))),0,(sin((alfa2)*(pi/180))),cos((alfa2
)*(pi/180))),d2;0,0,0,1];
A012 = A01*A12;
r11=A012(1,1);
r12=A012(1,2);
r13=A012(1,3);
r21=A012(2,1);
r22=A012(2,2);
r23=A012(2,3);
r31=A012(3,1);
r32=A012(3,2);
r33=A012(3,3);
Px=A012(1,4);
Py=A012(2,4);
Pz=A012(3,4);

```

## Apéndice B. Anexos de documentos

### 6.2.1 Valores de Eslabón en Solidworks®



eslabon momentos 2.SLDPRT

Opciones...

Reemplazar las propiedades de masa... Recalcular

Incluir sólidos/componentes ocultos

Crear operación de centro de masa

Mostrar masa de cordón de soldadura

Informar de valores de coordenadas relativos a: -- predeterminado --

Propiedades de masa de eslabon momentos 2  
Configuración: Predeterminado  
Sistema de coordenadas: -- predeterminado --

Densidad = 2.70 gramos por centímetro cúbico

Masa = 378.58 gramos

Volumen = 140.21 centímetros cúbicos

Área de superficie = 523.94 centímetros cuadrados

Centro de masa (reemplazado por el usuario): ( centímetros )  
X = 0.00  
Y = 0.00  
Z = 0.30

Ejes principales de inercia y momentos principales de inercia: ( gramos \* centímetros cuadrados  
Medido desde el centro de masa.  
Ix = (1.00, 0.00, 0.00) Px = 1831.39  
Iy = (0.00, 1.00, 0.00) Py = 31591.09  
Iz = (0.00, 0.00, 1.00) Pz = 33399.77

Momentos de inercia: ( gramos \* centímetros cuadrados )  
Obtenidos en el centro de masa y alineados con el sistema de coordenadas de resultados.  
Lxx = 1831.39 Lxy = 0.00 Lxz = 0.00  
Lyx = 0.00 Lyy = 31591.09 Lyz = 0.00  
Lzx = 0.00 Lzy = 0.00 Lzz = 33399.77

Momentos de inercia: ( gramos \* centímetros cuadrados )  
Medido desde el sistema de coordenadas de salida.  
Ixx = 1865.47 Ixy = 0.00 Ixz = 0.00  
Iyx = 0.00 Iyy = 31625.17 Iyz = 0.00  
Izx = 0.00 Izy = 0.00 Izz = 33399.77

## 6.2.2 ficha técnica GM9236S019



### GM9236S019

Lo-Cog® DC Servo Gearmotor



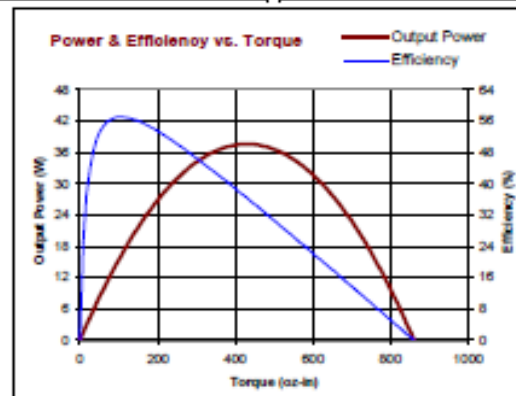
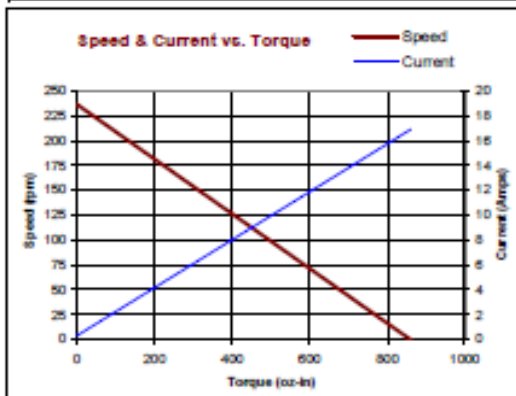
Assembly Data	Symbol	Units	Value
Reference Voltage	E	V	12
No-Load Speed	$\omega_{NL}$	rpm (rad/s)	236 (24.7)
Continuous Torque (Max.) <sup>1</sup>	$T_C$	oz-in (N-m)	153 (1.1E+00)
Peak Torque (Stall) <sup>2</sup>	$T_{PK}$	oz-in (N-m)	860 (6.1E+00)
Weight	$W_M$	oz (g)	23.3 (661)
Motor Data			
Torque Constant	$K_T$	oz-in/A (N-m/A)	3.25 (2.29E-02)
Back-EMF Constant	$K_E$	V/krpm (V/rad/s)	2.40 (2.29E-02)
Resistance	$R_T$	$\Omega$	0.71
Inductance	L	mH	0.66
No-Load Current	$i_{NL}$	A	0.33
Peak Current (Stall) <sup>2</sup>	$i_P$	A	16.9
Motor Constant	$K_M$	oz-in/ $\sqrt{W}$ (N-m/ $\sqrt{W}$ )	4.11 (2.90E-02)
Friction Torque	$T_F$	oz-in (N-m)	0.80 (5.6E-03)
Rotor Inertia	$J_M$	oz-in-s <sup>2</sup> (kg-m <sup>2</sup> )	1.0E-03 (7.1E-06)
Electrical Time Constant	$\tau_E$	ms	1.06
Mechanical Time Constant	$\tau_M$	ms	8.5
Viscous Damping	D	oz-in/krpm (N-m-s)	0.053 (3.5E-06)
Damping Constant	$K_D$	oz-in/krpm (N-m-s)	12.5 (8.5E-04)
Maximum Winding Temperature	$\theta_{MAX}$	$^{\circ}F$ ( $^{\circ}C$ )	311 (155)
Thermal Impedance	$R_{TH}$	$^{\circ}F/watt$ ( $^{\circ}C/watt$ )	56.3 (13.5)
Thermal Time Constant	$\tau_{TH}$	min	13.5
Gearbox Data			
Reduction Ratio			19.7
Efficiency <sup>3</sup>			0.84
Maximum Allowable Torque		oz-in (N-m)	500 (3.53)
Encoder Data			
Channels			3
Resolution		CPR	500

1 - Specified at max. winding temperature at 25°C ambient without heat sink. 2 - Theoretical values supplied for reference only.  
3 - Effective gearbox efficiency for this unit improved by use of ball bearings.

Included Features
2-Pole Stator
Ceramic Magnets
Heavy-Gauge Steel Housing
7-Slot Armature
Silicon Steel Laminations
Stainless Steel Shaft
Copper-Graphite Brushes
Diamond Turned Commutator
Motor Ball Bearings
Output Ball Bearing
Wide Face Gears

Customization Options
Alternate Winding
Sleeve or Ball Bearings
Modified Output Shaft
Custom Cable Assembly
Special Brushes
EMI/RFI Suppression
Alternate Gear Material
Special Lubricant
Optional Encoder
Fail-Safe Brake



All values are nominal. Specifications subject to change without notice. Graphs are shown for reference only.

© 2001 Pittman.



---

### 6.2.3 Distribución de terminales del microcontrolador

Distribución de pines (terminales) en microcontroladores

PIN ARDUINO	Función	Elemento
4	LPWM	IBT_2(1)
5	RPWM	IBT_2(1)
6	L_EN	IBT_2(1)
7	R_EN	IBT_2(1)
8	LPWM2	IBT_2(2)
9	RPWM2	IBT_2(2)
10	L_EN2	IBT_2(2)
11	R_EN2	IBT_2(2)
12	PULSADOR	PUSH BOTON
13	LED	PUSH BOTON
50	MISO	LECTOR SD
51	MOSI	LECTOR SD
52	SCK	LECTOR SD
53	CS	LECTOR SD

## 6.2.4 Constancia de participación en congreso



**EDUCACIÓN**  
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO  
NACIONAL DE MÉXICO

**EL TECNOLÓGICO NACIONAL DE MÉXICO  
A TRAVÉS DEL INSTITUTO TECNOLÓGICO DE CHIHUAHUA**

OTORGA LA PRESENTE

**CONSTANCIA**

A

**Luis Fernando Monrreal Arellano**

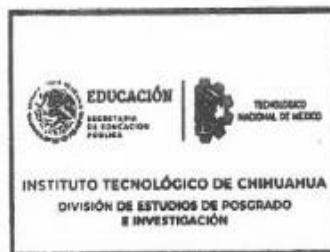
POR IMPARTIR LA CONFERENCIA

*Adaptación de trayectorias y simulación de un robot planar en un sistema  
de neurorehabilitación motriz de pacientes con enfermedad cerebro  
vascular*

EN EL CONGRESO INTERNACIONAL DE INGENIERÍA ELECTRÓNICA ELECTRO 2019,  
REALIZADO DEL 9 AL 11 DE OCTUBRE DEL PRESENTE AÑO.

CHIHUAHUA, CHIH., A OCTUBRE DE 2019.

**LUIS CARDONA CHACÓN**  
COORDINADOR GENERAL



INSTITUTO TECNOLÓGICO  
DE CHIHUAHUA



# ADAPTACIÓN DE TRAYECTORIAS Y SIMULACIÓN DE UN ROBOT PLANAR EN UN SISTEMA DE NEUROREHABILITACIÓN MOTRIZ DE PACIENTES CON ENFERMEDAD CEREBRO VASCULAR

L.F. Monrreal Arellano, J. A. Rojas Estrada\*, M. A. Ochoa Villegas, R. Sanjuan Galindo  
División de Estudios de Posgrado e Investigación  
Posgrado en Ingeniería

Tecnológico Nacional de México / Instituto Tecnológico de Nuevo León, Av. Eloy Cavazos 2001,  
Guadalupe, N.L. 67170  
Tel. 81 8157 0500

Fernando.monreal91@hotmail.com

\*Autor de correspondencia: juan.antonio.rojas@itnl.edu.mx

## RESUMEN.

Este artículo presenta un sistema para la rehabilitación de pacientes que han sufrido un accidente cerebro vascular y que consiste en la generación y aplicación de trayectorias en un robot planar de 2 GDL. Mediante la aplicación de trayectorias, y por medio del robot, se realizan sesiones de terapias para el brazo del paciente para rehabilitarlo. Comercialmente existen algunas plataformas y por ser de alto costo no están disponibles en los centros de salud. En este trabajo se toma en cuenta el par de reacción del paciente y se refleja en el modelo. Se estudió el modelo dinámico del robot tomando en cuenta la fuerza que ejerce el paciente expresándolo en el modelo global. Se realizaron simulaciones utilizando un controlador PD más retro de velocidad y en los resultados obtenidos se observó el seguimiento de la trayectoria mostrando un cierto grado de error por el efecto de la reacción.

Palabras Clave: Neurorehabilitación, GDL, robot planar

## ABSTRACT.

This article presents a system for the rehabilitation of patients who have had a cerebral brain accident and consists of the generation and application of trajectories in a 2 dof planar robot. Through the application of trajectories and by means of the robot, therapy sessions are carried out for the patient's arm to rehabilitate it. Commercially there are some platforms but because they are expensive, they aren't available at the health centers. In this work, the reaction torque of the patient that generate is taken into account and is reflected in the model. The dynamic model of the robot was studied taking into account the force exerted by the patient expressing in the global model. Simulations were carried out using a PD controller plus retro speed and in the results a tracking of the trajectory was achieved for a certain degree of error by the effect of the reaction.

Keywords: neurorehabilitation, DOF, planar robot

## 1. INTRODUCCIÓN

Actualmente en México ocurren problemas de enfermedad cerebro vascular a una razón de 118 por cada 100,000 habitantes al año. Se conoce como enfermedad vascular cerebral (EVC) a una alteración en las neuronas, que provoca disminución de flujo sanguíneo en el cerebro, acompañada de alteraciones cerebrales de manera momentánea o permanente.

Una de cada 3 personas presenta un grado de discapacidad después de sufrir una EVC[1].

Se ha demostrado, por medio de varios estudios, que los pacientes afectados por las EVC pueden mejorar sus habilidades realizando una apropiada secuencia de rehabilitación la cual se lleva a cabo por medio de ejercicios terapéuticos [2].

Una sesión de terapia consiste en repeticiones de algunos movimientos planeados sistemáticamente que permiten al usuario fortalecer el área afectada, mejorar el funcionamiento del miembro tratado y así como remediar o prevenir deterioros. Son considerados los principales objetivos de la rehabilitación como el recuperar la fuerza muscular que básicamente es la capacidad del músculo para contraerse y segundo recuperar la resistencia que es la capacidad de realizar el mismo movimiento repetidas veces.

Para llevar a cabo estas sesiones de terapias es necesario la asistencia del fisioterapeuta, quien es el especialista en la recuperación óptima de las secuelas que puede obtenerse de algún accidente; el especialista puede ir dejando que el paciente haga los movimientos completamente solo, ayudar a realizar el ejercicio o incluso asistirlo completamente. Existen diferentes tipos de ejercicios de rehabilitación como los ejercicios de fortalecimiento, estos incrementan la cantidad de fuerza que el músculo puede generar [3].

Estos ejercicios deben realizarse diariamente; sin embargo, esto no siempre es posible y el paciente en ocasiones recibe dos o tres sesiones por semana debido a la poca capacidad que pueden presentar algunos centros de salud que es donde generalmente se llevan a cabo estas secciones de terapia.

Desde hace varios años, diversos tipos de robots se han comenzado a utilizar en el área de medicina como apoyo en rehabilitación para pacientes de accidentes cerebro vasculares, ya que pueden replicar algunos de los movimientos que usan los fisioterapeutas, además de presentar diferentes ventajas como la facilidad para generar un número elevado repeticiones sin necesidad de detenerse así como también el almacenamiento de información objetiva sobre el estado del paciente [4].

Existen dos tipos de robots o dispositivos que se usan en el apoyo de rehabilitaciones, los primeros dispositivos están hechos para alinear el efector final con el usuario (el que se maneja en este proyecto, ver figura 2), y los otros son aquellos que alinean mecánicamente sus articulaciones con las articulaciones del usuario (exoesqueletos), estas son externas al cuerpo y proporcionan fuerza para el movimiento del usuario a una posición [5].

La organización del artículo se presenta de la siguiente forma, en la sección dos se desarrollan y verifican los modelos dinámicos que se emplearon, en la sección tres se muestra el desarrollo de la ley de control que se aplica al manipulador, en la sección cuatro se abarca principalmente la simulación del sistema, y por último en la sección cinco se concluye a partir de los resultados obtenidos.

## 2. DESARROLLO DEL MODELO DINÁMICO

### 2.1 Modelo dinámico del manipulador

En la figura 1 se muestra el concepto del robot planar (vista superior) del cual se desarrollará el modelo dinámico siendo  $l_1$  y  $l_2$  las longitudes de los eslabones respectivamente,  $m_1$  y  $m_2$  las masas de cada uno de estos, las distancias del centro de masas están dadas como  $l_{c1}$  para el primer eslabón y  $l_{c2}$  para el segundo,  $I_1$  e  $I_2$  representan los momentos de inercia respectivamente para cada eslabón, estas se dan debido a la forma y material del que está hecho cada eslabón. Y por último se tienen las posiciones articulares  $q_1$  y  $q_2$ . En figura 2, se observa una vista superior del sistema del paciente y el robot.

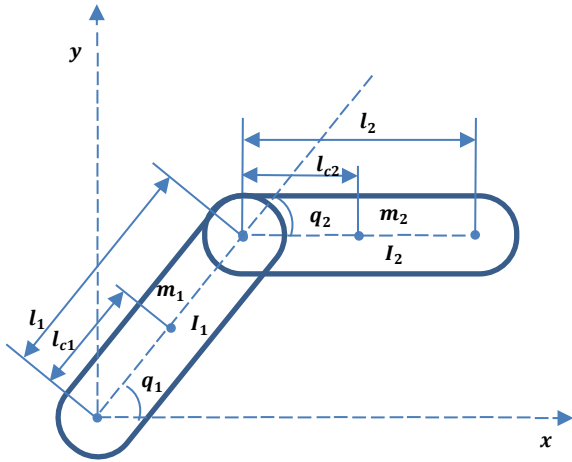


Fig. 1. Robot planar de 2 GDL (vista superior).

Para la obtención del modelo matemático, para el caso particular del robot planar de 2 GDL, se usó la formulación Euler-LaGrange la cual se basa en la obtención del Lagrangiano ( $L$ ) que representa la diferencia de las energías cinética ( $K$ ) y la energía potencial ( $U$ ) [6], una vez que se han aplicado las ecuaciones queda definido por la ecuación 1.

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}} \right) - \left( \frac{\partial L}{\partial \theta} \right) = \tau \quad (1)$$

Después de su aplicación se obtiene por último la ecuación dinámica que modela este tipo de robot manipulador como se expresa en la ecuación 2.

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \quad (2)$$

La cual en su forma compacta matricial quedaría expresada en la ecuación 3.

$$\begin{bmatrix} M_{11}(q) & M_{12}(q) \\ M_{21}(q) & M_{22}(q) \end{bmatrix} \ddot{q} + \begin{bmatrix} C_{11}(q, \dot{q}) & C_{12}(q, \dot{q}) \\ C_{21}(q, \dot{q}) & C_{22}(q, \dot{q}) \end{bmatrix} \dot{q} + \begin{bmatrix} g_1(q) \\ g_2(q) \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (3)$$

Donde las siguientes ecuaciones por separado completan la ecuación 3.

$$\begin{aligned} M_{11} &= l_{c1}^2 m + l_1^2 m_2 + m_2 l_{c2} + 2m_2 l_1 l_{c2} (\cos q_2) + I_1 + I_2 \\ M_{12} &= l_{c2}^2 m_2 + m_2 l_1 l_{c2} (\cos q_2) + I_2 \\ M_{21} &= l_{c2}^2 m_2 + m_2 l_1 l_{c2} (\cos q_2) + I_2 \\ M_{22} &= l_{c2}^2 m_2 + I_2 \\ C_{11} &= 2m_2 l_1 l_{c2} (\sin q_2) (\dot{q}_2) \\ C_{12} &= m_2 l_1 l_{c2} (\sin q_2) (\dot{q}_1 + \dot{q}_2) \\ C_{21} &= m_2 l_1 l_{c2} (\sin q_2) (\dot{q}_1) \\ C_{22} &= 0 \\ g_1 &= 0 \\ g_2 &= 0 \end{aligned}$$

Siendo el vector de los pares gravitacionales  $G(q)$  igual a cero debido a que el robot planar solo se maneja dentro de las dimensiones x/y, por lo que no existe momento fuera en x ó y.

### 2.2 Modelo dinámico del brazo humano

Para la simulación del modelo dinámico para el par generado por el brazo humano se tomó la velocidad del robot manipulador como referencia como la entrada al bloque *sign* y combinado con una señal de una función aleatoria se simulan los diferentes pares que puede generar el paciente ( $\tau_{pi}$ ). Esto únicamente para fines de simulación.

### 2.3 Modelo completo dinámico manipulador-paciente

Podemos definir el movimiento de nuestro robot planar para rehabilitación como el par necesario para generar movimiento en las articulaciones que está compuesto por la suma de dos pares ( $\tau_{pi}$  y  $\tau_{ri}$ ), donde  $\tau_{p1}$  y  $\tau_{p2}$  son los pares generados por la fuerza humana del brazo y el hombro respectivamente mientras que los pares mecánicos  $\tau_{r1}$  y  $\tau_{r2}$  son generados por

cada uno de los actuadores para cada eslabón; esta se expresan de la siguiente forma:

$$\tau_1 = \tau_{p1} + \tau_{r1} \quad (5)$$

$$\tau_2 = \tau_{p2} + \tau_{r2} \quad (6)$$

Sustituyéndolos en la ecuación (2) quedaría expresada de la siguiente forma en (7).

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau_{pi} + \tau_{ri} \quad (7)$$

Una vez que se ha validado nuestro modelo dinámico se procede a llevarlo a ecuaciones de estado, en las cuales las variables de estado que quedaron representadas son la posición ( $x_1$ ) y la velocidad ( $x_2$ ), por lo que las ecuaciones 9 y 10 quedan como:

$$\dot{x}_1 = x_2 \quad (9)$$

$$\dot{x}_2 = M^{-1}(\tau - Cx_2 - G) \quad (10)$$

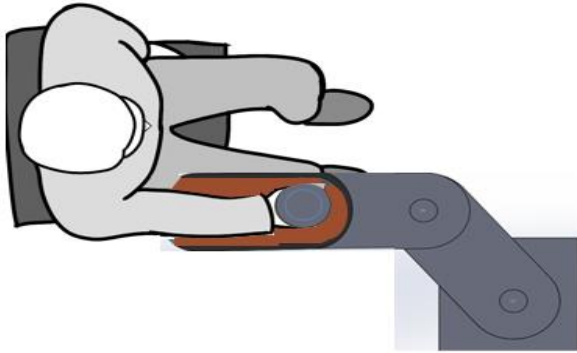


Fig. 2. Vista superior sistema paciente/robot de 2GDL

### 3. CONTROL

En cuanto al control de posición angular para el manipulador, se tuvo en cuenta que el robot debería de realizar movimientos relativamente lentos las cuales serían generadas de una velocidad y una aceleración pequeñas.

Para su aplicación se utilizó un control proporcional más retroalimentación de velocidad, ya que el control P puro no garantiza el cumplimiento del objetivo de control de posición en sistemas cuyos modelos dinámicos varían el valor de su entrada constantemente.

Considerando los pares mecánicos generados los términos de la diferencia entre error de la posición y la derivada de la posición, ambas multiplicadas por matrices simétricas positivas definidas conocidas como ganancia de posición  $K_p$  y velocidad  $K_v$  respectivamente quedando expresado en (11) de la siguiente manera:

$$\tau = K_p \tilde{q} + K_v \dot{q} \quad (11)$$

Mientras que el error de posición se entiende como la diferencia entre la posición deseada y la posición actual, por lo que queda escrita en (12).

$$\tilde{q}_i = q_{di} - q_i \quad (12)$$

Expresando el modelo por medio del software matlab, como se muestra en la figura 3, donde dentro de bloque de función de matlab fue colocado nuestro modelo dinámico escrito en su forma de variable de estado y siendo nuestra posición deseada dada por una secuencia de puntos en intervalos pequeños generando una trayectoria [7].

Para las trayectorias con puntos en ( $x$ - $y$ ) es necesario su conversión a las posiciones deseadas del eslabón por medio de la solución de la cinemática inversa [6], de la cual se obtienen las posiciones articulares en un archivo de datos, este se utiliza como referencia de entrada mediante un bloque de datos. Ver figura 3.

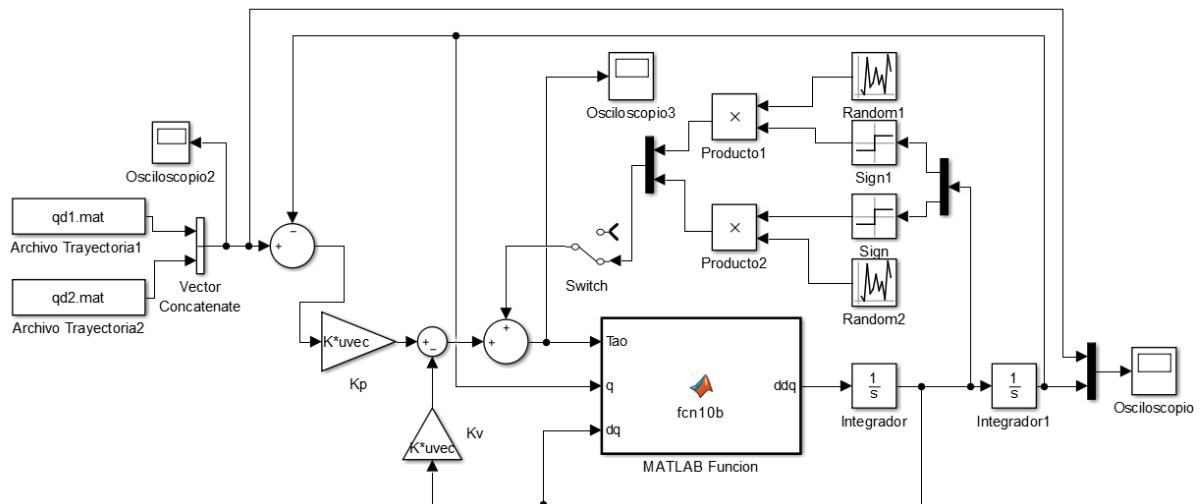


Fig. 3. Esquemas de control en Matlab para el robot de 2GDL

Para el estudio del equilibrio del sistema, se usa el método directo de Lyapunov. Considerando la función candidata de Lyapunov como se ve en (13) [8]. Tomando en cuenta solamente los pares de los actuadores.

$$V(\tilde{q}, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q} + \frac{1}{2} \tilde{q}^T K_p \tilde{q} \quad (13)$$

La cual al derivarse con respecto al tiempo queda expresado en (14) de la siguiente forma:

$$\dot{V}(\tilde{q}, \dot{q}) = \dot{q}^T M(q) \ddot{q} + \frac{1}{2} \dot{q}^T \dot{M}(q) \dot{q} + \tilde{q}^T K_p \dot{\tilde{q}} \quad (14)$$

Por lo que al ser sustituido el valor de  $M(q)\ddot{q}$  y al reagrupar los términos se puede ver que se cumple con la forma  $q^T \left[ \frac{1}{2} \dot{M}(q) - C(q, \dot{q}) \right] q = 0$  y por lo tanto la ecuación se puede resumir en (15).

$$\dot{V}(\tilde{q}, \dot{q}) = -\dot{q}^T K_v \dot{q} \quad (15)$$

Por lo que si  $K_v$  es una matriz positiva definida se entiende que el valor de  $\dot{V}(\tilde{q}, \dot{q}) \leq 0$ , por lo que se considera una función de Lyapunov y por lo tanto el origen es estable y las soluciones  $\tilde{q}$  y  $\dot{q}$  están acotadas [9].

#### 4. SIMULACION Y RESULTADOS

Para la validación de nuestro modelo de control se recurrió al software Matlab para realizar pruebas de la simulación de las trayectorias realizadas. Los datos simulados que se colocaron para cierto robot manipulador se encuentran en la tabla 1.

Tabla 1. Datos para simulación de un robot manipulador.

Descripción	Notación	Valor	Unidad
Longitud del eslabón 1	$l_1$	0.3	m
Longitud del eslabón 2	$l_2$	0.3	m
Distancia al centro de masas del eslabón 1	$l_{c1}$	0.15	m
Distancia al centro de masas del eslabón 2	$l_{c2}$	0.15	m
Masa del eslabón 1	$m_1$	7	kg
Masa del eslabón 2	$m_2$	3	kg
Inercia al centro de masa del eslabón 1	$I_1$	0.1213	Kg m <sup>2</sup>
Inercia al centro de masa del eslabón 2	$I_2$	0.0116	Kg m <sup>2</sup>

En la gráfica de la figura 4 se puede ver la simulación de una trayectoria polinómica que estará siguiendo el robot manipulador y el brazo del paciente.

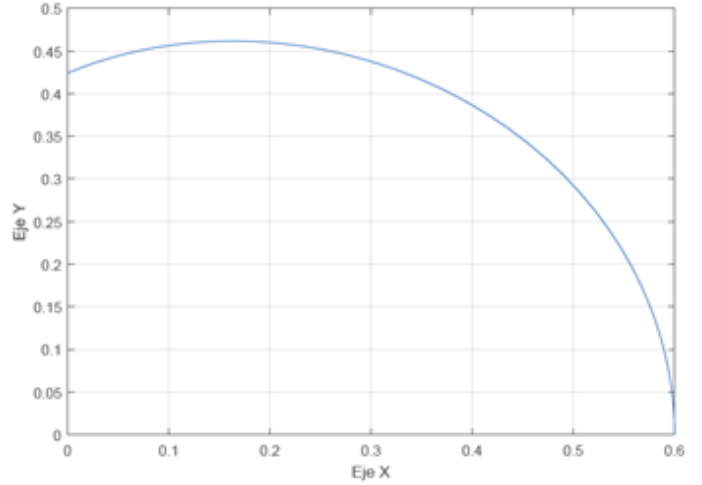
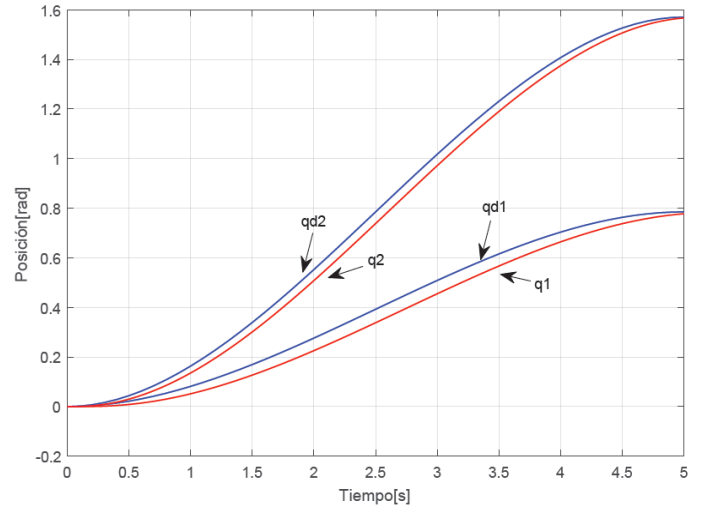


Fig. 4. Gráfica de la trayectoria polinómica en el plano XY

Las trayectorias deseadas tienen velocidades y aceleraciones angulares muy pequeñas que están determinadas dentro de un tiempo de ciclo, en este caso de 5 segundos.

En la figura 5, se aprecia la posición deseada y las posiciones generadas para las posiciones de las articulaciones.

Fig. 5. Gráfica de posición de las articulaciones.



En las pruebas experimentales, el valor de la posición angular obtenida ( $q_{ai}$ ) reflejará, mediante los sensores, el deseo del paciente de extender o flexionar las articulaciones con una determinada fuerza.

El error de posición se entiende como la diferencia entre la posición deseada y la actual, para esta sección de simulación, este caso en particular, puede verse representado en la gráfica de la figura 6.

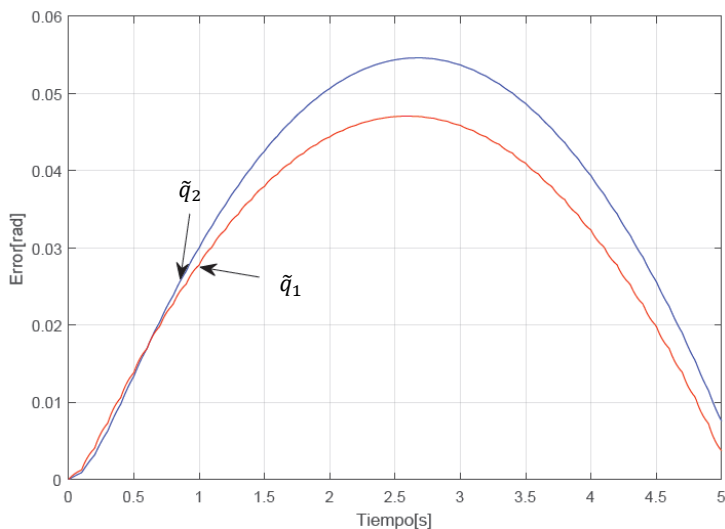


Fig. 6. Gráfica del error de posición

En la figura 7 se muestran la simulación de los pares aplicados por parte del manipulador ( $\tau_{ri}$ ) durante el ciclo.

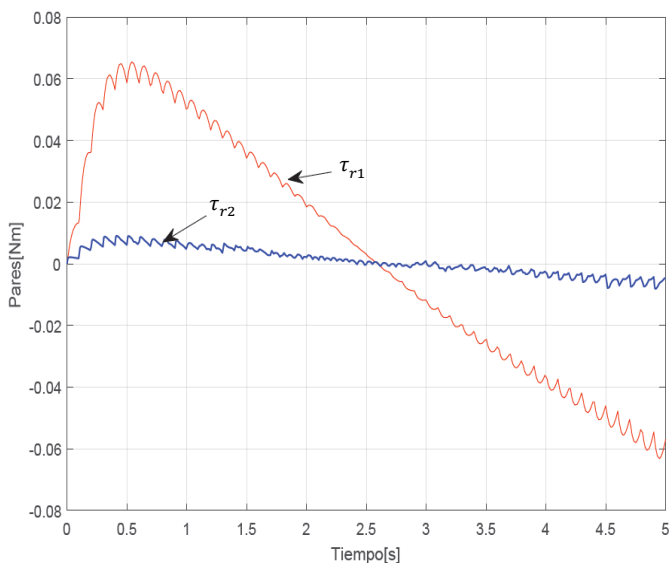


Fig. 7. Gráfica de pares aplicados por el manipulador  $\tau_{ri}$

Mientras que en la gráfica 8 se puede ver la simulación de los pares generados por parte de los actuadores  $\tau_{ri}$  y el par de reacción aplicado por el paciente ( $\tau_{pi}$ ), el cual es simulado por la función descrita en el punto 2.2 de la sección 2.

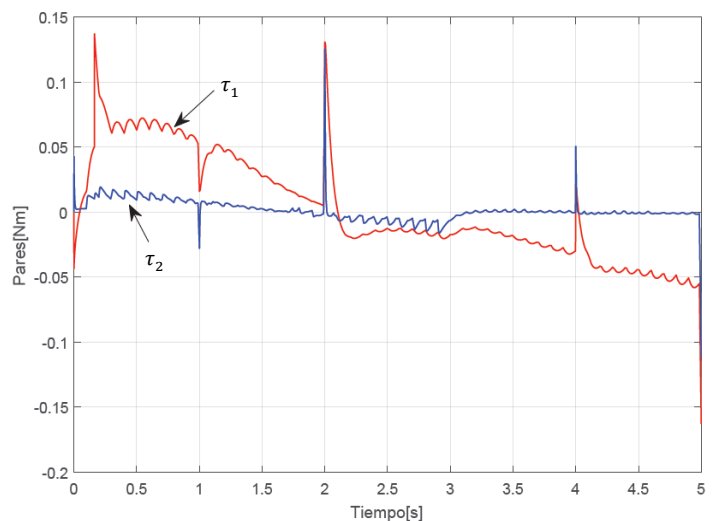


Fig. 8. Gráfica de los pares aplicados por el manipulador  $\tau_{ri}$  y los pares del paciente  $\tau_{pi}$

En la figura 8 se observan desviaciones de los pares con respecto a los pares presentados en la gráfica de la figura 7.

## 5. CONCLUSIONES

El tema abordado aporta de manera importante a los sistemas de rehabilitación, dando como resultado nuevas búsquedas por la innovación en apoyo a estas áreas. El tema principal de este artículo es el de la proposición de una aplicación práctica a la generación de trayectorias en apoyo a la rehabilitación por medio de un robot de 2 GDL. Un estudio completo que comprende el análisis matemático del modelo dinámico y el diseño del controlador de un sistema que consiste en un robot de 2 GDL con efector final, resulta útil para la aplicación de terapias, estableciendo un sistema disponible de referencia para futuros proyectos que tomen en cuenta la interacción con el paciente.

Por medio de la simulación de matlab se puede observar el grado del error que se genera al tratar de dar seguimiento a la trayectoria deseada teniendo en cuenta los diferentes pares que ejerce el usuario, por lo que se espera tienda a minimizarse conforme se lleven a cabo las sesiones ya en una aplicación real. Esto quiere decir que, en la medida que el paciente se somete a neurorehabilitación, al ir recobrando la acción motriz tratará de oponer menos resistencia a la trayectoria ejecutada por el robot.

La variación en los pares mecánicos del manipulador tomando en cuenta los pares externos ejercidos por el paciente es dependiente del tiempo de duración de la trayectoria seleccionada, lo cual se interpreta como una mayor o menor resistencia por parte del paciente.

## AGRADECIMIENTOS:

Fernando Monrreal Arellano, agradece el apoyo de beca No.889446 del CONACyT y parcialmente al proyecto de TecNM 6772.18-P convocatoria 2018-2.

## 6. REFERENCIAS.

- [1] Recuperado de [www.imss.gob.mx/salud-en-linea/enfermedad-vascular-cerebral](http://www.imss.gob.mx/salud-en-linea/enfermedad-vascular-cerebral). consultada el 25/05/2019.
- [2] Poli P., Morone G., Rosati G., and Masiero S. (2013) "Robotic Technologies and Rehabilitation: New Tools for Stroke Patients' Therapy", *BioMed Research International* <http://dx.doi.org/10.1155/2013/153872>.
- [3] Glynn, A., Fiddler, H., 2009. "The physiotherapist's pocket guide to exercise, assessment, prescription and training". ELSEVIER.
- [4] Rodríguez L, Cano R, Cuesta A, Alguacil I, Molina F. (2014), "Terapia robótica para la rehabilitación del miembro superior en patología neurológica", ELSEVIER <https://doi.org/10.1016/j.rh.2014.01.001>.
- [5] Gopura R, Bandara D, Kiguchi K, Man G. (2015), "Developments in Hardware Systems of Active Upper-Limb Exoskeleton Robots", ELSEVIER <http://dx.doi.org/10.1016/j.robot.2015.10.001>.
- [6] Kelly, R., Santibañez, V. y Loria, A. Control of robot manipulators in joint space, Springer-Verlag, 2005.
- [7] Ortega R., Loria A., Nicklasson P. J., Sira-Ramirez H., 1998, "Passivity-based control of Euler-Lagrange Systems Mechanical, Electrical and Electromechanical Applications", Springer-Verlag: London.
- [8] Craig J. J., *Robótica tercera edición*, Pearson, 2006
- [9] Spong, M., y Vidyasagar, M., *Robot dynamics and control*, Wiley India, 2008.