



# **ESCUELA POLITÉCNICA NACIONAL**



## **FACULTAD DE INGENIERÍA MECÁNICA**

### **DISEÑO E IMPLEMENTACIÓN DEL SISTEMA DE CONTROL, INSTRUMENTACIÓN E INTERFAZ DE MANDO HUMANO – MÁQUINA PARA UN EXOESQUELETO DE REHABILITACIÓN DE MANO**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE  
MAGÍSTER EN MECATRÓNICA Y ROBÓTICA**

**CHRISTIAN SEBASTIAN BONILLA RIBADENEIRA**

**DIRECTOR:  
ING. JORGE ANDRÉS ROSALES ACOSTA, Ph.D**

**CODIRECTOR:  
ING. CARLOS CEVALLOS BARRAGÁN, Ph.D**

**FECHA  
Quito, febrero, 2020**

## **CERTIFICACIÓN**

Certifico que el presente trabajo fue desarrollado por **CHRISTIAN SEBASTIAN BONILLA RIBADENEIRA**, bajo mi supervisión.

---

**Dr. JORGE ANDRÉS ROSALES ACOSTA**

## **CERTIFICACIÓN**

Certifico que el presente trabajo fue desarrollado por **CHRISTIAN SEBASTIAN BONILLA RIBADENEIRA**, bajo mi supervisión.

---

**Dr. CARLOS CEVALLOS BARRAGÁN**

## DECLARACIÓN

Yo, **Christian Sebastian Bonilla Ribadeneira**, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondiente a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.

---

Christian Bonilla R.

## **DEDICATORIA**

El presente proyecto está dedicado a:

Mi padre Marco Bonilla, quien me ha enseñado a seguir siempre adelante en la vida.

Mi madre Teresa Ribadeneira, que es la persona que me ha apoyado siempre.

Mi esposa Andrea Molina, compañera de mi vida y que me ha permitido mirar la vida de una forma distinta. Lo logramos.

## **AGRADECIMIENTO**

A Dios, por permitirme finalizar mis estudios de postgrado, gracias por darme la fuerza necesaria para seguir adelante en los buenos y malos momentos.

A la Escuela Politécnica Nacional, gracias a todos los docentes, amigos y personal administrativo de esta prestigiosa institución quienes han sido parte fundamental de mi carrera estudiantil y profesional. A las Escuela de Formación de Tecnólogos quienes me han apoyado siempre durante mi carrera y quienes han sido parte importante de todo lo logrado. A la Facultad de Ingeniería Mecánica quienes acertadamente ofertaron la Maestría en Mecatrónica y Robótica y a todos los profesores que fueron parte de la Maestría y de quienes he aprendido los conocimientos impartidos. A los profesores que dirigieron este proyecto de Titulación Dr. Andrés Rosales y Dr. Carlos Cevallos y en especial al Ing. Gabriel Bravo quien fue el profesor y amigo que me propuso realizar este proyecto.

A toda mi familia quienes han sido un apoyo fundamental para poder seguir estudiando, a mis padres por darme la sabiduría para escoger el camino que he seguido. A Andre, gracias por estar junto a mi siempre, sin tu ayuda no lo hubiera podido lograr, deseo que podamos compartir muchos logros más juntos. Gracias a la fidelidad de mis queridas mascotas Princesa y Martín, quienes me acompañaron en esas largas noches mientras estudiaba y realizaba este proyecto.

Finalmente, gracias a todos mis amigos con quienes tuve el honor de compartir muchos momentos importantes de mi vida y que me han apoyado para poder conseguir mis metas.

# ÍNDICE

INTRODUCCIÓN .....	1
<b>Objetivo general</b> .....	<b>2</b>
<b>Objetivos específicos</b> .....	<b>2</b>
1. MARCO TEÓRICO .....	3
1.1. Conceptos de robótica .....	3
1.2. Biomecánica de la mano .....	7
1.3. Tecnologías de exoesqueletos de la mano .....	8
1.4. Exoesqueletos de rehabilitación.....	10
1.4.1. SAFER.....	10
1.4.2. HEXOTRAC [15] .....	11
1.4.3. Exoesqueleto para rehabilitación de mano – Andreas Wege [2].....	12
1.4.4. Exoesqueleto de rehabilitación de mano – Jamshed Iqbal [16] .....	13
1.4.5. Comparación de los modelos analizados .....	14
1.5. Modelo cinemático de la mano.....	14
2. METODOLOGÍA .....	20
2.1. Estudio de la cinemática del mecanismo.....	20
2.1.1. Mecanismo para los dedos trifalángicos [5].....	20
2.1.2. Mecanismo para el dedo pulgar .....	21
2.1.3. Modelo cinemático del mecanismo.....	21
2.2. Diseño de la Instrumentación .....	27
2.2.1. Medición de posición con flexosensores .....	28
2.2.2. Medición de posición con sensores de posición angular .....	30
2.2.3. Medición de Fuerza.....	32
2.3. Diseño del Sistema de Control .....	34
2.3.1. Elementos del Sistema de Control .....	34
2.3.2. Programa de Control.....	40
2.3.3. Lazo Cerrado de Control.....	40
2.4. Diseño de la Interfaz de Mando.....	42

2.5.	Implementación del Sistema de Rehabilitación en Conjunto .....	45
3.	RESULTADOS Y DISCUSIÓN .....	49
<b>3.1.</b>	<b>Resultados</b> .....	49
3.1.1.	Resultados del Estudio Cinemático .....	49
3.1.2.	Resultados del Funcionamiento del Exoesqueleto .....	56
<b>3.2.</b>	<b>Discusión</b> .....	61
3.2.1	Discusión del Estudio Cinemático .....	61
3.2.2.	Discusión del Funcionamiento del Exoesqueleto.....	63
4.	CONCLUSIONES Y RECOMENDACIONES.....	65
4.1.	CONCLUSIONES .....	65
4.2.	RECOMENDACIONES .....	66
<b>Anexos 70</b>		
<b>Anexo I. Simulaciones del Modelo Cinemático desarrolladas en MATLAB.....</b>		<b>70</b>
<b>Anexo II. Matrices de Rotación obtenidas de los modelos cinemáticos .....</b>		<b>80</b>
<b>Anexo III. Programa de Control del Arduino MEGA .....</b>		<b>83</b>
<b>Anexo IV. Programa de Control del Arduino UNO.....</b>		<b>90</b>
<b>Anexo V. Programa de la interfaz de mando en labview.....</b>		<b>96</b>



## ÍNDICE DE FIGURAS

Figura 1.1. Articulaciones de un grado de libertad. (a) Articulación de revolución. (b) Articulación prismática.....	3
Figura 1.2. Ejemplo de parámetros Denavit – Hartenberg modificado aplicados a un brazo robótico cilíndrico.....	5
Figura 1.3. Parámetros Denavit – Hartenberg modificado aplicados a una articulación rotativa.....	6
Figura 1.4. Huesos de la mano derecha que conforman cada dedo y la palma de la mano. Articulaciones de los dedos.....	8
Figura 1.5. Tipos de mecanismo, dependiendo de los ejes de rotación. (a) Coincidencia directa de los ejes de las articulaciones. (b) Enlaces para un centro de rotación remoto. (c) Estructura de enlaces redundantes. (d) Mecanismo accionado por tendones. (e) Actuador de flexión unido a las juntas. (f) Eslabones seriales unidos al segmento distal. ....	9
Figura 1.6. Criterios de clasificación de exoesqueletos de mano según cuatro criterios importantes. Tipo de actuador, propósito, transmisión de potencia y método de detección de intención .....	10
Figura 1.7. Modelo CAD del mecanismo SAFER. Cadena de eslabones con articulaciones móviles cuya transmisión del movimiento es en base a cables y poleas. ....	11
Figura 1.8. HEXOTRAC, prototipo de exoesqueleto para 3 dedos. Diseño modular que consta de una cadena serial de eslabones unidos a los dedos en la punta, el movimiento se genera con los motores unidos a la palma de la mano y la muñeca y la parte electrónica es parte del exoesqueleto.....	12
Figura 1.9. Diseño del exoesqueleto con juntas alineadas presentado por Andreas Wege.....	13
Figura 1.10. Diseño Conceptual del Exoesqueleto – Jamshed Iqbal.....	13
Figura 1.11. Modelo cinemático de la mano de [18]. Tiene 26 grados de libertad y considera el movimiento de la palma de la mano con 6 grados de libertad adicionales. La articulación DIP se le considera dependiente de la articulación PIP. DOF = grados de libertad.....	15
Figura 1.12. Modelo cinemático de la mano de [19]. Presenta dos modelos distintos uno para el pulgar con 4 grados de libertad y otro para el resto de los dedos, con 5 grados de libertad. No se considera el movimiento de la muñeca, sino que es la referencia para el movimiento del resto de eslabones. ....	16

Figura 1.13. Diagrama de bloques de la cinemática directa .....	17
Figura 1.14. Configuración cinemática de los dedos de la mano propuesta en [11]. Tiene 24 grados de libertad y toma como punto de referencia la muñeca. En la izquierda se observan los ejes de coordenadas de cada articulación, aplicando el criterio de Denavit - Hartenberg. En la derecha se observa cada uno de los grados de libertad que se consideran en las articulaciones. ....	17
Figura 2.1. Mecanismo para los dedos trifalángicos. Se observa el mecanismo diseñado en [5] en conjunto con la mano. Los eslabones del mecanismo unidos a la punta del dedo, y por el otro lado se unen a los engranes que son movidos por el motor. ....	20
Figura 2.2. Mecanismo del exoesqueleto para el pulgar .....	21
Figura 2.3. Mecanismo de eslabones para los dedos trifalángicos. En azul el mecanismo del exoesqueleto, la mano con color piel. ....	22
Figura 2.4. Representación de las articulaciones para un dedo trifalángico. Las articulaciones de color amarillo son rotativas y las de color rojo son prismáticas. ....	23
Figura 2.5. Representación del mecanismo para el dedo pulgar. En color amarillo articulaciones rotativas para el movimiento de flexión, en color azul articulaciones rotativas para el movimiento de abducción, en rojo articulaciones prismáticas. ....	26
Figura 2.6. Flexosensor FS-L-0055-253-ST. Variación de la Resistencia dependiendo del ángulo de curvatura del sensor. ....	28
Figura 2.7. Rangos de las variables del acondicionamiento del flexosensor. La primera etapa el flexosensor recibe la variación del ángulo de rotación y convierte la señal en variación de resistencia. La siguiente etapa es el circuito de acondicionamiento que cambia los valores de resistencia a voltaje normalizado. ....	29
Figura 2.8. Circuito de Acondicionamiento del Flexosensor. Primera etapa convierte la señal de resistencia a voltaje. Segunda etapa amplifica la señal de voltaje a un valor normalizado. Tercera etapa limita el valor máximo de voltaje a 5.1 V. ....	30
Figura 2.9. Sensor de posición rotativo. Izquierda SV01 Murata, derecha CJMCU-103 .....	31
Figura 2.10. Circuito de Acondicionamiento del Potenciómetro. Desde la izquierda se tiene el sensor, a continuación, el amplificador U1:A para el seguidor y la resistencia R5 y Zener D1 corresponden al limitador .....	31
Figura 2.11. Sensor de fuerza resistivo. FSR 402. La parte circular es donde se aplica la fuerza. En la parte izquierda se tiene la conexión eléctrica. ....	32
Figura 2.12. Curva característica del sensor y circuito de acondicionamiento. Izquierda: Curva de variación de la resistencia dependiendo de la fuerza aplicada al sensor. Derecha: Circuito recomendado por el fabricante. ....	32

Figura 2.13. Circuito de Acondicionamiento Sensor de Fuerza. Divisor de voltaje para cambiar la señal de resistencia a voltaje, el amplificador como seguidor para acoplar impedancias y un limitador para evitar que el voltaje supere los 5.1V.....	33
Figura 2.14. Esquema de conexión de los circuitos de acondicionamiento. El exoesqueleto envía las señales de los sensores a la placa de acondicionamiento. La placa envía las señales acondicionadas al controlador. El exoesqueleto recibe del controlador las señales para el movimiento de los motores. ....	34
Figura 2.15. Controladores utilizados. Izquierda: Arduino Mega. Derecha: Arduino Uno .....	35
Figura 2.16. Motor (izquierda) y Reductor (derecha) utilizados. La entrada del motor es un voltaje de alimentación de 0 a 12 V. La salida del motor es una salida de torque y velocidad angular, se conecta al reductor para disminuir la velocidad y aumentar el torque. ....	37
Figura 2.17. Motor Shield de Arduino L298.....	38
Figura 2.18. Fuente de Alimentación de los Controladores. Entrada de 120V de corriente alterna. Salida de 12V de corriente continua, 5A. ....	39
Figura 2.19. Módulo Bluetooth HC-05. Funciona como emisor y receptor. Se comunica directamente con el Bluetooth de la PC .....	39
Figura 2.20. Elementos del sistema de control en conexión con el resto de los elementos del sistema. ....	40
Figura 2.21. Lazo Cerrado de Control Dedos Trifalángicos.....	41
Figura 2.22. Lazo Cerrado de Control Dedo Pulgar .....	41
Figura 2.23. Interfaz de Mando del Exoesqueleto desarrollada en MATLAB.....	42
Figura 2.24. Interfaz de mando creada en Labview .....	44
Figura 2.25. Sistema de Rehabilitación en Conjunto.....	45
Figura 2.26. Mecanismo del Exoesqueleto utilizado en la mano izquierda. Imagen de izquierda arriba vista inferior. Imagen de derecha arriba vista lateral. Imagen de abajo vista superior .....	46
Figura 2.27. Circuitos Electrónicos. Controladores, fuente de alimentación, placa de circuitos de acondicionamiento, drivers de los motores .....	47
Figura 3.1. Simulación del dedo índice con el módulo Robot de Matlab. Junto al punto cero la falange proximal, a continuación, la falange media y luego la falange distal. (a) Posición 1 de la tabla Tabla 3.1. (b) Posición 2. (c) Posición 3. (d) Posición 4.....	50
Figura 3.2. Simulación del dedo índice utilizando las matrices de rotación. En amarillo la falange proximal, en azul la falange media y en rojo la falange distal. (a) Posición 1 de la Tabla 3.1. (b) Posición 2. (c) Posición 3. (d) Posición 4 .....	51

Figura 3.3. Simulación del mecanismo para dedos trifalángicos. Izquierda: modelo simulado en el toolbox robot de MATLAB. Derecha: Modelo obtenido con las matrices de rotación.....	52
Figura 3.4. Simulación del mecanismo para el dedo pulgar. Izquierda: simulación en el módulo Robot de MATLAB. Derecha: Modelo obtenido con matrices de rotación.....	52
Figura 3.5. Espacio de trabajo para el dedo índice, en azul el espacio de trabajo de la mano y en verde el espacio de trabajo del mecanismo. Izquierda: Medidas mínimas. Derecha: Medidas máximas.....	53
Figura 3.6. Espacio de trabajo para el dedo pulgar, en azul el espacio de trabajo de la mano y en verde el espacio de trabajo del mecanismo. Izquierda: Medidas mínimas. Derecha: Medidas máximas.....	54
Figura 3.7. Posición del mecanismo y dedo para máxima flexión .....	55
Figura 3.8. Trayectoria de un Movimiento de Flexión.....	56
Figura 3.9. Datos leídos de los sensores desde el terminal Realterm. Se observa las mediciones de las articulaciones DIP1, DIP2, PIP2, MCP2, PIP1, MCP1 en el orden mostrado.....	58

## ÍNDICE DE TABLAS

Tabla 1.1. Comparación entre modelos de exoesqueletos analizados.....	14
Tabla 1.2. Parámetros Denavit - Hartenberg para el dedo índice.....	18
Tabla 1.3. Parámetros Denavit - Hartenberg para el dedo índice sin considerar la articulación CMC .....	19
Tabla 1.4. Parámetros Denavit – Hartenberg para el dedo pulgar .....	19
Tabla 2.1. Medidas de los segmentos del mecanismo de los dedos trifalángicos .....	23
Tabla 2.2. Parámetros Denavit – Hartenberg para el mecanismo de los dedos trifalángicos .....	24
Tabla 2.3. Parámetros de diseño en condiciones críticas de los dedos trifalángicos...	24
Tabla 2.4. Longitudes de los segmentos del mecanismo del dedo pulgar.....	26
Tabla 2.5. Parámetros Denavit – Hartenberg para el mecanismo del dedo pulgar .....	26
Tabla 2.6. Valores de longitudes de las falanges que componen el dedo pulgar [4] ...	27
Tabla 2.7. Restricciones estáticas del pulgar .....	27
Tabla 2.8. Ubicación de Flexosensores .....	29
Tabla 2.9. Características del Arduino Mega 2560.....	35
Tabla 2.10. Características del Arduino Uno.....	35
Tabla 2.11. Características del Arduino Motor Shield.....	38
Tabla 2.12 Códigos del sistema de comunicación .....	44
Tabla 2.13. Elementos del mecanismo del exoesqueleto.....	46
Tabla 2.14. Elementos de los Circuitos Electrónicos.....	47
Tabla 3.1. Valores considerados y obtenidos en la simulación del dedo índice.....	49
Tabla 3.2. Ángulos obtenidos en máxima flexión para dedos trifalángicos .....	54
Tabla 3.3. Ángulos simulados para una trayectoria de flexión.....	55
Tabla 3.4. Ejemplo de valores tomados de los flexosensores .....	58
Tabla 3.5. Valores tomados por los flexosensores en grados .....	59
Tabla 3.6. Pruebas realizadas al Sistema de Control e Interfaz de Mando .....	61

## RESUMEN

El presente proyecto presenta el diseño del sistema de control, instrumentación y la interfaz de mando para un exoesqueleto de rehabilitación de mano. La implementación de la parte mecánica del exoesqueleto se construye previamente, por lo que esta propuesta busca la integralidad del proyecto. Se realiza un estudio de la cinemática del mecanismo del exoesqueleto para obtener los rangos de funcionamiento. También, el estudio cinemático permite determinar la trayectoria que sigue el dedo en conjunto con el mecanismo en un movimiento de flexión. Además, se presenta el diseño de la instrumentación que incluye sensores de posición y de fuerza con sus respectivos circuitos de acondicionamientos. El sistema de control es el encargado de manejar los motores eléctricos que generan los movimientos del mecanismo y también, recibir las señales de los sensores de posición y fuerza. Con dicha información, el controlador establece las señales de los motores que permiten realizar rutinas de rehabilitación con el exoesqueleto. Se puede escoger los tipos de rutinas de rehabilitación a través de un interfaz humano – máquina que funciona en la computadora.

**Palabras clave:** exoesqueleto de mano, rehabilitación física, cinemática de la mano, sensores de fuerza.

## ABSTRACT

This article presents the design of the control system, instrumentation and human – machine interface for a hand rehabilitation exoskeleton. The design and implementation of the mechanical part of the exoskeleton is previously built, so this proposal seeks the integrality of the project. A study of the kinematics of the exoskeleton mechanism is performed to obtain the operating ranges. Also, the kinematic study allows to determine the trajectory that the finger follows in conjunction with the mechanism in a flexion movement. In addition, the design of the instrumentation that includes position and force sensors with their respective conditioning circuits is presented. The control system is responsible for operating the electric motors that generate the movements of the mechanism and also, receives the signals from the position and force sensors. With this information, the controller establishes the motor signals that allow rehabilitation routines with the exoskeleton. The types of rehabilitation routines can be chosen through a human-machine interface that run on the computer.

**Keywords:** hand exoskeleton, physical rehabilitation, hand kinematic, force sensors

# **DISEÑO E IMPLEMENTACIÓN DEL SISTEMA DE CONTROL, INSTRUMENTACIÓN E INTERFAZ DE MANDO HUMANO – MÁQUINA PARA UN EXOESQUELETO DE REHABILITACIÓN DE MANO**

## **INTRODUCCIÓN**

Las lesiones y accidentes en la mano son un problema que luego de un diagnóstico médico son tratados con rehabilitación. Una eficiente rehabilitación es importante para que el paciente pueda volver a desempeñar sus actividades de manera normal [1]. Existe un riesgo de quedar permanentemente incapacitado en caso de no realizar una rehabilitación adecuada [2].

La rehabilitación pasiva es un procedimiento manual hecho por el fisioterapeuta aplicando movimiento pasivo continuo a la mano del paciente [2]. Debido a que los movimientos hechos por el fisioterapeuta junto con el paciente son continuos y durante varios ciclos, se puede complementar la rehabilitación manual por dispositivos que generen el movimiento pasivo en la mano del paciente [1]. Existen dispositivos en el mercado que permiten realizar los ejercicios de rehabilitación, sin embargo, son costosos y existe la dificultad de obtener dichos dispositivos ya que no son comerciales en el país.

La falta de especialistas médicos restringe la capacidad de atención a pacientes en las diferentes ramas de salud, en el caso de fisioterapia existe la necesidad de reducir costos operativos, mediante el uso de dispositivos de bajo costo. [3]

El Proyecto de Vinculación con la Colectividad de la Escuela Politécnica Nacional [3], “Diseño y Construcción de un exoesqueleto para rehabilitación de la mano” propuesto por Víctor Hugo Guerrero Barragán, Ph.D., dentro de su metodología incluye:

- Diseñar algoritmos de control automático para la programación y ejecución de rutinas de rehabilitación de mano mediante el uso del exoesqueleto mecatrónico;
- Automatizar los mecanismos del exoesqueleto de la mano en base a la utilización de controladores.



El mecanismo desarrollado dentro del proyecto se construye previamente [4] [5] [6], sin embargo, es necesario complementar al mecanismo con un sistema que permita realizar el mando y control automático del mecanismo, por lo que, esta propuesta busca la integralidad del proyecto.

El proyecto realizado es para una rehabilitación física de tipo pasiva, en la cual el paciente recibe el movimiento del exoesqueleto, lo cual le permite ejercitar los dedos y de esta forma lograr un funcionamiento correcto de los mismos. En la rehabilitación física activa el paciente genera el movimiento y el exoesqueleto debe realizar un seguimiento del movimiento de los dedos del paciente [7]. Para lograr una rehabilitación activa se tiene el mecanismo como una cadena cinemática subactuada, en la cual hay más grados de libertad que articulaciones motorizadas, condición necesaria para realizar los movimientos de rehabilitación activa. Además, se tiene las medidas de los sensores de fuerza, los cuales permiten conocer la intención de movimiento del paciente y de esta forma generar el movimiento del exoesqueleto.

## **Objetivo general**

Diseñar e implementar el control automático, la instrumentación y la interfaz de mando humano – máquina de un exoesqueleto para rehabilitación de mano.

## **Objetivos específicos**

- Analizar el estado del arte referente a los exoesqueletos para rehabilitación de mano.
- Realizar un estudio de la cinemática de la mano y del mecanismo para obtener los rangos de movimiento del conjunto mano – mecanismo.
- Diseñar el sistema de control, la instrumentación y la interfaz de mando del mecanismo que permita escoger los distintos tipos de rehabilitación que se pueden generar con el exoesqueleto.
- Implementar un sistema de control y la interfaz de mando para comprobar el funcionamiento de la propuesta.

# 1. MARCO TEÓRICO

## 1.1. Conceptos de robótica

Los cuerpos individuales que forman un robot se llaman eslabones. Cuando varios cuerpos individuales se juntan sin movimientos relativos entre ellos se los considera como un solo eslabón. Los eslabones se conectan a través de articulaciones. Una articulación conecta dos eslabones y genera movimientos relativos entre ambos. [8]

Las principales articulaciones que existen son: articulación de revolución y articulación prismática, son articulaciones de un grado de libertad y pueden ser utilizados en mecanismos planares y espaciales. La articulación de revolución permite que dos eslabones conectados giren el uno respecto del otro, alrededor del eje de la articulación. En la Figura 1.1 (a), se observa una articulación de revolución. La articulación prismática se basa en eslabones prismáticos congruentes o cilíndricos que tienen un movimiento deslizante sobre uno de los ejes. En la Figura 1.1(b), se observa una articulación prismática.

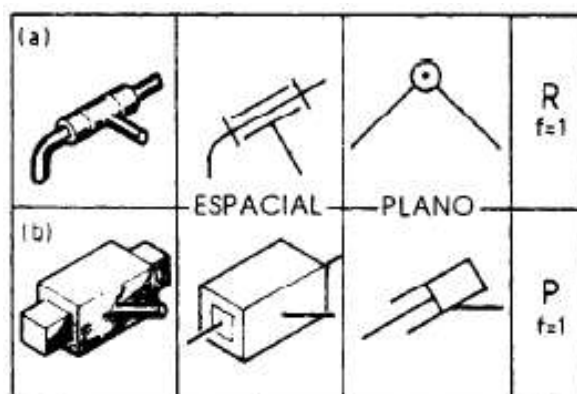


Figura 1.1. Articulaciones de un grado de libertad. (a) Articulación de revolución. (b)

Articulación prismática

(Fuente: [9])

Los grados de libertad de un mecanismo determinan la movilidad del mecanismo. La movilidad se define como el número de variables independientes que es necesario especificar para ubicar los miembros del mecanismo con respecto a otros miembros. Un cuerpo que se mueve libremente en el espacio cartesiano tiene 6 grados de libertad, 3 definen su posición y 3 la orientación. Para definir los grados de libertad de un mecanismo se utiliza el método conocido como Grubler – Kutzbach [8]. En dicho método se pretende encontrar el número de grados de libertad del mecanismo n.

$$\mathbf{n} = \mathbf{s}(\mathbf{r} - \mathbf{p} - \mathbf{1}) + \sum_i^p \mathbf{n}_i \quad \text{Ecuación 1.1.}$$

Donde  $n$  son los grados de libertad del mecanismo,  $s$  es la dimensión del espacio de trabajo: para mecanismo planares es 3 y para mecanismos espaciales es 6. También  $r$  define el número de cuerpos rígidos que tiene el mecanismo,  $p$  es el número de articulaciones del sistema,  $n_i$  son los grados de libertad relativos de cada articulación. [8]

Un manipulador robótico está formado por varios eslabones que se conectan a través de articulaciones, a estos se les conoce como cadenas cinemáticas. La cinemática directa en una cadena cinemática consiste en obtener la posición y orientación del efector final, considerando que las posiciones de las articulaciones están determinadas. En la cinemática inversa para un determinado estado del efector final, en posición y orientación, se debe obtener los valores de los ángulos de las articulaciones.

Para definir la posición del efector final con respecto a la posición de la base de una cadena cinemática, es necesario encontrar la relación entre los sistemas de coordenadas del efector final con respecto al sistema de coordenadas de la base. Se puede encontrar esta relación utilizando matrices de transformación de sistemas de coordenadas. El procedimiento sistemático para encontrar dicha relación se conoce como los parámetros de Denavit – Hartenberg. Con este procedimiento es posible encontrar matrices de transformación que permiten relacionar los sistemas de coordenadas deseados. En la Figura 1.2, se observa un ejemplo de los parámetros Denavit – Hartenberg aplicados a un manipulador robótico.

Existen dos tipos de variantes en dichos parámetros y son los Denavit – Hartenberg y Denavit – Hartenberg modificado. En los parámetros Denavit -Hartenberg se considera desde el punto de vista de los eslabones y dichos parámetros se encuentran para cada eslabón. Los parámetros para el  $i$  – ésimo eslabón son los siguientes:

- $d_i$ , Desplazamiento de la articulación, posición relativa de los eslabones  $i-1$  e  $i$ . Se mide como la distancia entre  $X_i$  y  $X_{i+1}$  a lo largo de  $Z_i$ .
- $\theta_i$ , Ángulo de la articulación, ángulo relativo entre los eslabones  $i-1$  e  $i$ . Se mide como el ángulo entre  $X_i$  y  $X_{i+1}$  alrededor de  $Z_i$ .
- $a_i$ , Longitud del eslabón, distancia entre las perpendiculares comunes a los ejes  $Z_i$  y  $Z_{i+1}$  a lo largo de  $X_{i+1}$ .

- $\alpha_i$ , Ángulo de Torsión, ángulo entre las proyecciones ortogonales de los ejes de articulación  $Z_i$  y  $Z_{i+1}$ , alrededor del eje  $X_{i+1}$ .

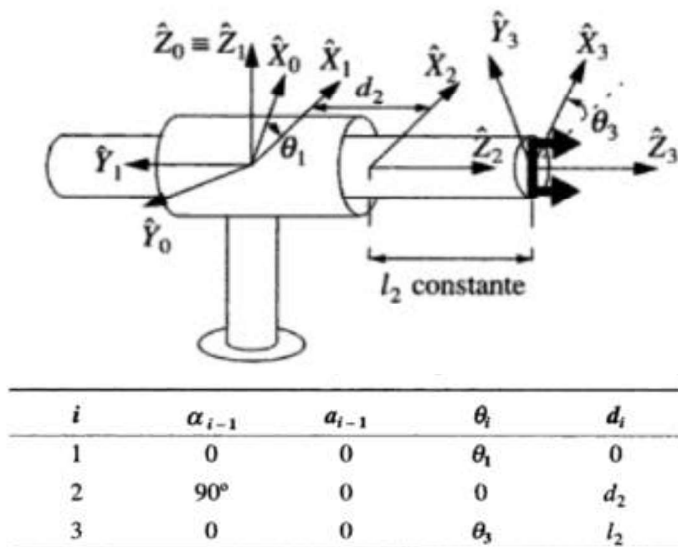


Figura 1.2. Ejemplo de parámetros Denavit – Hartenberg modificado aplicados a un brazo robótico cilíndrico

(Fuente: [27])

Para determinar las matrices de transformación de un sistema de coordenadas relativo a un eslabón con respecto a otro se puede utilizar la matriz que se muestra en la Ecuación 1.2.

$$T_i = \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i S\alpha_i & a_i C\theta_i \\ S\theta_i & C\theta_i C\alpha_i & -C\theta_i S\alpha_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ecuación 1.2}$$

Donde C es coseno y S es seno. Para obtener la matriz de transformación del sistema de coordenadas del efector final relativo al sistema de coordenadas de la base se multiplica las matrices de transformación de cada eslabón.

Los parámetros Denavit – Hartenberg modificado se obtienen relativos a cada articulación, por lo tanto, se definen los parámetros para cada articulación, Figura 1.3.

Los parámetros son los siguientes:

- $\alpha_{i-1}$ , Ángulo de Torsión, ángulo entre las proyecciones ortogonales de los ejes de articulación  $Z_{i-1}$  y  $Z_i$ , alrededor del eje  $X_i$ .
- $a_{i-1}$ , Longitud del eslabón, distancia entre las perpendiculares comunes a los ejes  $Z_{i-1}$  y  $Z_i$  a lo largo de  $X_i$ .
- $\theta_i$ , Ángulo de la articulación, ángulo relativo entre los eslabones  $i-1$  e  $i$ . Se mide como el ángulo entre  $X_{i-1}$  y  $X_i$  alrededor de  $Z_i$ .
- $d_i$ , Desplazamiento de la articulación, posición relativa de los eslabones  $i-1$  e  $i$ . Se mide como la distancia entre  $X_{i-1}$  y  $X_i$  a lo largo de  $Z_i$ .

La matriz que relaciona los sistemas de coordenadas de una articulación con otra es la que se muestra en la Ecuación 1.3, en los parámetros Denavit – Hartenberg modificado.

$$T_i = \begin{bmatrix} C\theta_i & -S\theta_i & 0 & a_{i-1} \\ S\theta_i C\alpha_{i-1} & C\theta_i C\alpha_{i-1} & -S\alpha_{i-1} & -S\alpha_{i-1}d_i \\ S\theta_i S\alpha_{i-1} & C\theta_i S\alpha_{i-1} & C\alpha_{i-1} & C\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ecuación 1.3}$$

Donde C es coseno y S es seno. De igual forma, para obtener la matriz de transformación del sistema de coordenadas del efector final relativo al sistema de coordenadas de la base, se multiplica las matrices de transformación de cada articulación.

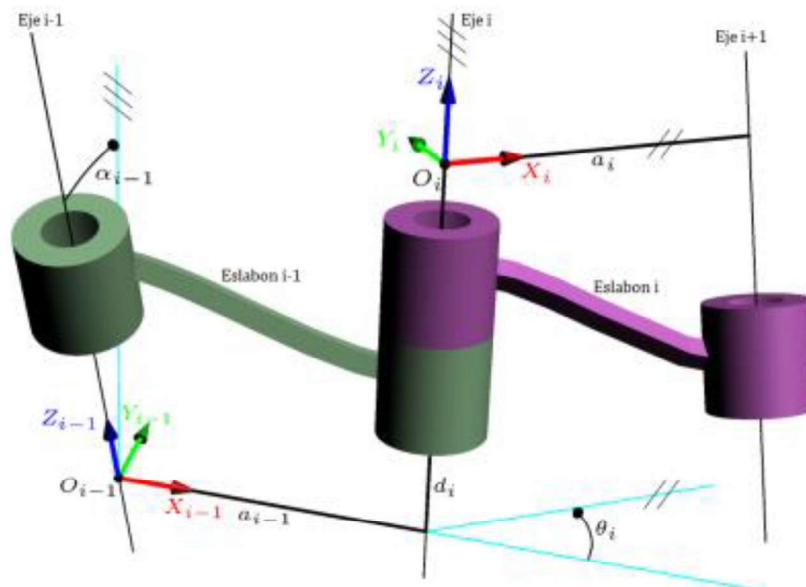


Figura 1.3. Parámetros Denavit – Hartenberg modificado aplicados a una articulación rotativa  
(Fuente: [10])

## 1.2. Biomecánica de la mano

El estudio de la biomecánica de la mano es muy complejo debido a su gran número de huesos y articulaciones los cuales permiten un amplio rango de posibles posiciones de los dedos. Es necesario describir los grados de libertad y los rangos de movimiento de los dedos, para desarrollar un exoesqueleto de la mano. Con este conocimiento se puede entender los movimientos posibles de la mano y generar las rutinas de rehabilitación. [11]

La mano debe tomar distintas formas que le permiten a la persona interactuar con el medio. La forma cóncava de la mano permite agarrar objetos, el movimiento de pinza, en el que se genera una fuerza contrapuesta entre el pulgar con el resto de los dedos y permite manipular objetos y generar movimientos con una destreza fina. [12] Los requerimientos del proyecto a implementarse incluyen el movimiento de flexión y extensión y el movimiento de pinza.

La mano está compuesta por 5 dedos: pulgar, índice, medio, anular y meñique. De los cuales 4 dedos son conocidos como trifalángicos el índice, medio, anular y meñique, ya que están compuestos por 3 huesos: falange distal, falange media y falange proximal. El pulgar está compuesto por dos falanges: falange distal y falange proximal. Todos los dedos están unidos en la palma de la mano y tienen un hueso por cada dedo conocido como el metacarpiano [1]. Por motivos de nomenclatura en el presente escrito se numera a los dedos de la siguiente forma: pulgar – 1, índice - 2, medio - 3, anular - 4 y meñique - 5.

Los dedos trifalángicos tienen 3 articulaciones, la interfalángica distal DIP (por sus siglas en inglés distal interphalangeal), la interfalángica proximal PIP (por sus siglas en inglés proximal interphalangeal), la metacarpofalángica MCP (por sus siglas en inglés metacarpophalangeal) [11]. Las articulaciones DIP y PIP son de tipo troclear (forma similar a una bisagra), por lo que permiten únicamente movimientos de flexión y extensión. Mientras que la MCP es de tipo condílea, y permite un rango de movilidad de abducción y aducción cuando la articulación se encuentra en extensión, pierde esta capacidad cuando los dedos se encuentran totalmente en flexión [12]. Las articulaciones y huesos de la mano se pueden observar en la Figura 1.4.

El movimiento del dedo pulgar se lo considera desde la muñeca, debido a que el movimiento del hueso metacarpiano es independiente de los dedos trifalángicos. La primera articulación es la trapezio-metacarpiana TMC (por sus siglas en inglés de trapeziometacarpal), la segunda articulación es la metacarpofalángica MCP y la tercera es la interfalángica IP (por sus siglas en inglés de interphalangeal). Se observa también en la Figura 1.4.

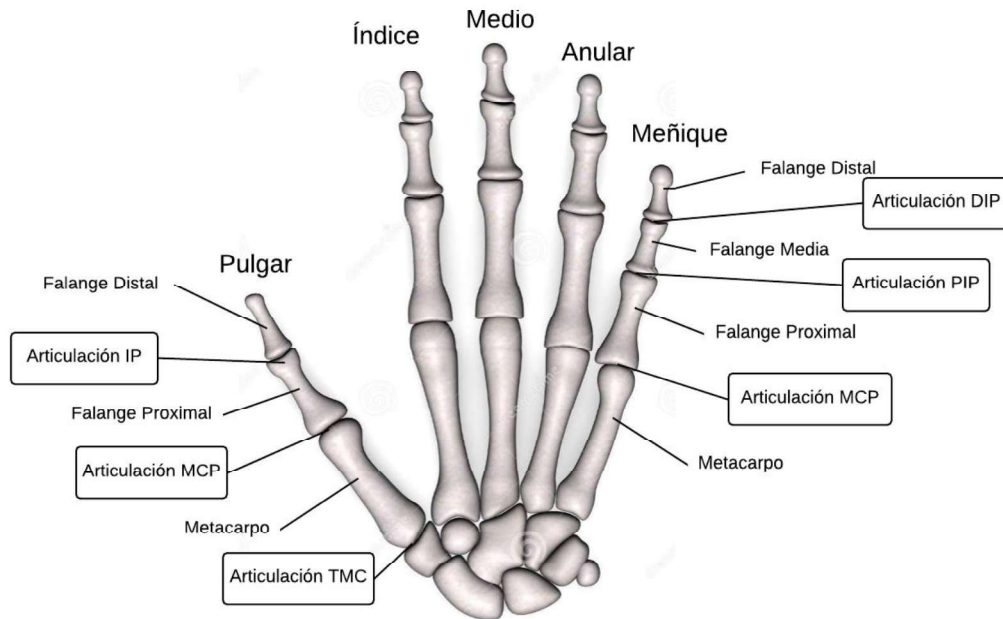


Figura 1.4. Huesos de la mano derecha que conforman cada dedo y la palma de la mano.

Articulaciones de los dedos

(Fuente: propia)

### 1.3. Tecnologías de exoesqueletos de la mano

Existen varios aspectos a tomar en cuenta al momento de diseñar un exoesqueleto para rehabilitación de la mano, entre ellos la ergonomía, seguridad, peso total, resistencia. Uno de los aspectos más importantes en las tecnologías de exoesqueletos de la mano es la seguridad. Debido a que cualquier malfuncionamiento del exoesqueleto puede desencadenar en una lesión de la mano [1]. Los límites del rango de funcionamiento del exoesqueleto deben ser con topes mecánicos. El diseño mecánico de exoesqueletos para la mano incluye varias posibilidades que se describen a continuación.

La primera permite coincidir el centro de rotación de los eslabones rígidos con el centro de rotación de la mano como se observa en la Figura 1.5 (a), sin embargo, esta opción incluye un espacio adicional entre los dedos que puede cambiar el movimiento natural de los dedos.

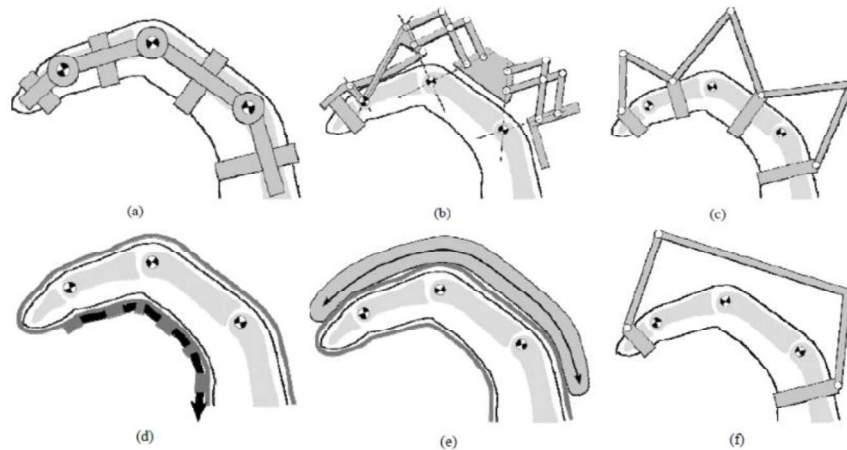


Figura 1.5. Tipos de mecanismo, dependiendo de los ejes de rotación. (a) Coincidencia directa de los ejes de las articulaciones. (b) Enlaces para un centro de rotación remoto. (c) Estructura de enlaces redundantes. (d) Mecanismo accionado por tendones. (e) Actuador de flexión unido a las juntas. (f) Eslabones seriales unidos al segmento distal.

(Fuente: [1])

Por otro lado, hay varias formas de tener un centro de rotación distante del centro de rotación de los dedos. A pesar de esto, la coincidencia del centro de rotación del mecanismo puede ser ignorada cuando se tiene un mecanismo sub - actuado con varios grados de libertad. Existe la opción de crear un mecanismo con varios eslabones en serie los cuales están unidos al dedo únicamente en la punta, de esta forma se libera totalmente las alineaciones de los ejes del mecanismo con los ejes de rotación de los dedos, Figura 1.5 (f).

Los exoesqueletos de mano pueden ser clasificados por varios criterios como: el tipo de actuador, el método de transmisión de la potencia, los grados de libertad, método medición de las intenciones y método de control. De acuerdo con estos criterios se puede clasificar a los exoesqueletos como se observa en la Figura 1.6.



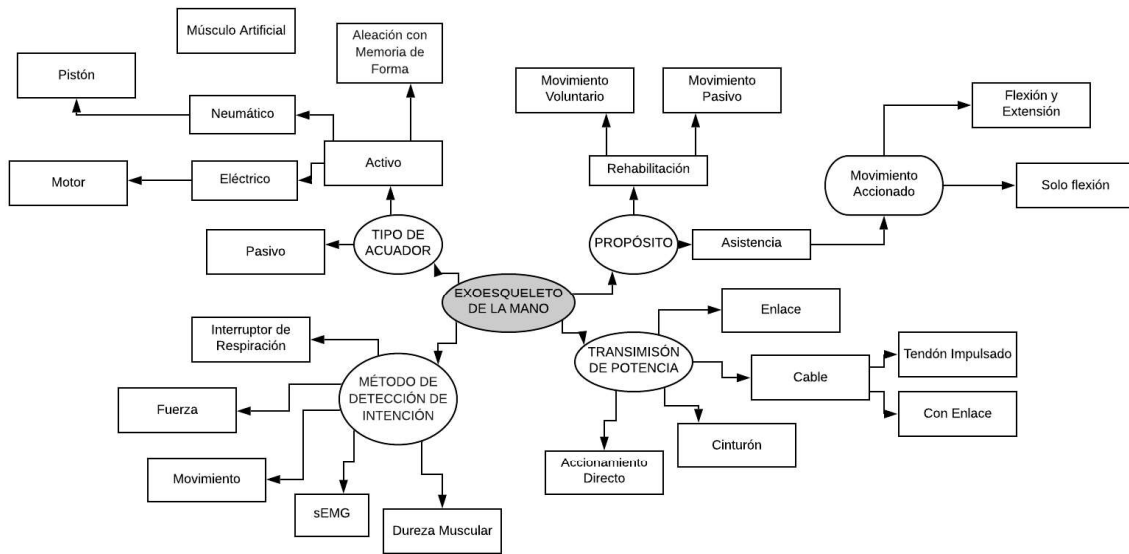


Figura 1.6. Criterios de clasificación de exoesqueletos de mano según cuatro criterios importantes. Tipo de actuador, propósito, transmisión de potencia y método de detección de intención  
(Fuente: [1])

## 1.4. Exoesqueletos de rehabilitación

### 1.4.1. SAFER

El exoesqueleto conocido como SAFER se presenta en [13]. Es un exoesqueleto que está conformado por varios eslabones seriales unidos por articulaciones móviles. El movimiento rotacional de las articulaciones se trasmite a través de cables que van unidos al motor principal ubicado encima de la palma de la mano. La unión del mecanismo con el dedo se lo hace en la punta del dedo. La continuación del diseño del guante SAFER se presenta en [14]. En el citado artículo se presenta el diseño y la optimización del mecanismo del guante háptico. Dentro de la optimización se presenta un criterio para calcular la longitud de los eslabones de modo que se obtenga un rango de movimiento del exoesqueleto que sea útil para distintas longitudes de los dedos, y se maximice la relación de transmisión de fuerza. En la Figura 1.7, se presenta el modelo CAD del mecanismo SAFER. Finalmente, en [15] se presenta la evolución del mecanismo del guante háptico, y los resultados cuando se utiliza en movimientos de rehabilitación física.

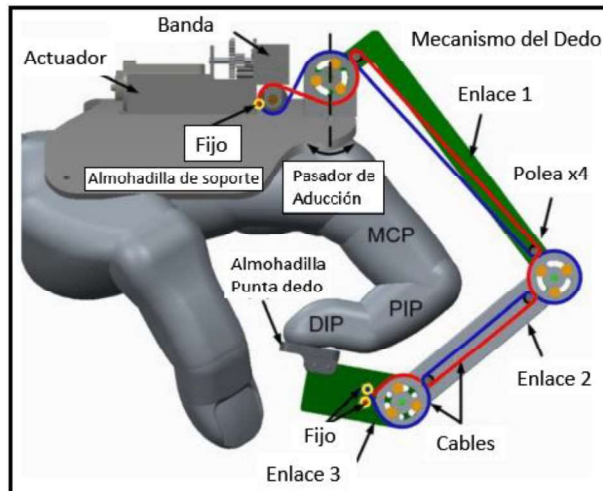


Figura 1.7. Modelo CAD del mecanismo SAFER. Cadena de eslabones con articulaciones móviles cuya transmisión del movimiento es en base a cables y poleas.

(Fuente: [16])

#### 1.4.2. HEXOTRAC [16]

HEXOTRAC es un diseño de exoesqueleto de la mano para tres dedos: pulgar, índice y medio, Figura 1.8. Tiene realimentación de fuerza y seguimiento de la trayectoria de los dedos. Se propone un nuevo enfoque para la realimentación de fuerza con un solo punto de unión del exoesqueleto con la punta del dedo, a través de una cadena cinemática de 6 grados de libertad. Esta cadena cinemática permite alcanzar a los dedos todo su espacio de trabajo sin restricciones, y permite un sistema para realimentar la posición de los dedos en su trayectoria. Al mismo tiempo el mecanismo sub - actuado permite la realimentación bidireccional de fuerza en la punta de los dedos. El mecanismo puede ser ajustado para varios tamaños de manos sin ajuste mecánico entre el mecanismo y los dedos de la mano. Los objetivos que se persiguen en HEXOTRAC son los siguientes: Portabilidad, debe ser usado en varios tamaños de manos y debe ser simple de utilizar. Comodidad, no debe causar fatiga ni lastimar a la mano en periodos largos de uso. Espacio de trabajo, debe permitir el uso en todo el espacio de trabajo de la mano. Sistema actuador, el sistema debe ser sub-actuado con un sistema de un actuador simple por cada dedo. Realimentación de fuerza, los niveles de fuerzas deben ser adecuados para generar la percepción de las interacciones. Seguimiento, debe incluir un sistema de alta resolución para el seguimiento de trayectorias. Peso, debe tener un bajo peso.

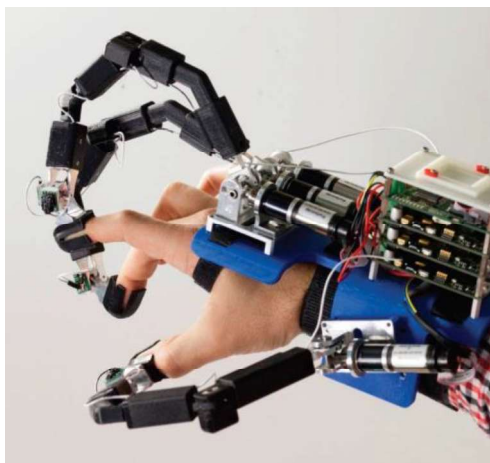


Figura 1.8. HEXOTRAC, prototipo de exoesqueleto para 3 dedos. Diseño modular que consta de una cadena serial de eslabones unidos a los dedos en la punta, el movimiento se genera con los motores unidos a la palma de la mano y la muñeca y la parte electrónica es parte del exoesqueleto.

(Fuente: [16])

#### **1.4.3. Exoesqueleto para rehabilitación de mano – Andreas Wege [2]**

Otra solución de un exoesqueleto para rehabilitación de mano se presenta en (A. Wege, 2005), dicho exoesqueleto ha sido diseñado específicamente para aplicaciones médicas. Se presenta la construcción e implementación del sistema de control para aplicaciones en terapia física en caso de lesiones de la mano. El exoesqueleto se ha diseñado para satisfacer los siguientes requerimientos: movimiento bidireccional, tener 4 grados de libertad para cada dedo, la palma debe estar libre de elementos mecánicos para interactuar con el ambiente. Dentro de la construcción mecánica se presenta el diseño del mecanismo el cual está formado por una serie de palancas y poleas que van unidas a cada hueso del dedo, y que permiten transmitir el movimiento de los motores al dedo. Para configurar el mecanismo para diferentes tamaños de dedos se puede ajustar a través de tornillos. Se puede lograr movimientos independientes de cada junta del dedo. La construcción mecánica está formada por una serie de eslabones alineados a los ejes de los dedos, cada eslabón se junta con una de las falanges del dedo, haciendo que el movimiento de la articulación del dedo sea dependiente de la articulación del exoesqueleto. Se incluye una junta libre para el movimiento de abducción. La construcción se observa en la Figura 1.9.

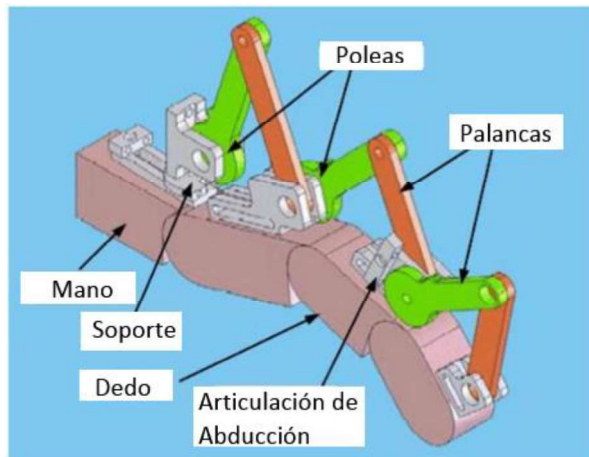


Figura 1.9. Diseño del exoesqueleto con juntas alineadas presentado por Andreas Wege.  
(Fuente: [2])

#### 1.4.4. Exoesqueleto de rehabilitación de mano – Jamshed Iqbal [17]

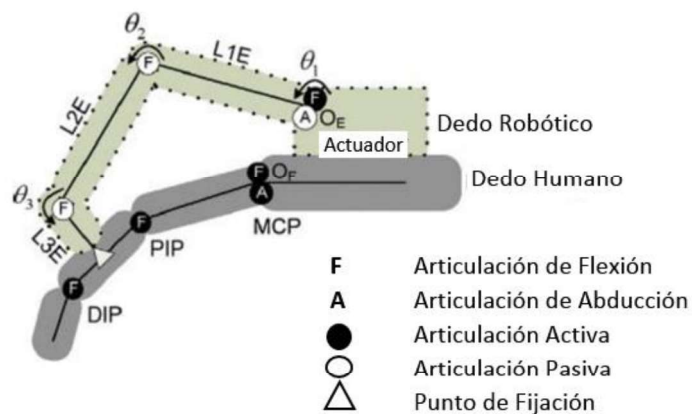


Figura 1.10. Diseño Conceptual del Exoesqueleto – Jamshed Iqbal  
(Fuente: [17])

Se presenta un dispositivo para rehabilitación de mano, diseñado para asistir en movimientos de flexión y extensión. El sistema consiste en un exoesqueleto compuesto por varios grados de libertad y sub – actuado con múltiples eslabones conectados en serie y que ejerce fuerza perpendicular en las puntas de los dedos. Es un exoesqueleto con un mecanismo sub - actuado tipo RRR (tres juntas de revolución), compuesto de tres eslabones planos conectados en serie y unido al usuario en un punto simple. El punto de unión puede ser localizado en el segmento proximal o medio del dedo. El mecanismo sub – actuado permite tener menos actuadores y permite la adaptación pasiva de cada dedo según la forma del objeto agarrado. Todo el mecanismo se encuentra en la parte dorsal de la mano, dejando la parte palmar y los dedos libres para

el agarre de objetos. El exoesqueleto es accionado por una unidad simple de actuación que se encuentra en la base de la primera juntura. El diseño conceptual del exoesqueleto se muestra en la Figura 1.10.

#### 1.4.5. Comparación de los modelos analizados

Para la comparación de los modelos analizados se utiliza los criterios que se presentan en la Figura 1.6. es decir, se clasifican los modelos de acuerdo con el tipo de actuador, propósito, método de detección de la intención y la transmisión de potencia. Los resultados de dicha comparación se observan en la Tabla 1.1.

Tabla 1.1. Comparación entre modelos de exoesqueletos analizados.

	<b>SAFER</b>	<b>HEXOTRAC</b>	<b>Andreas Wege</b>	<b>Jamshed Iqbal</b>
<b>Actuador</b>	Motor Eléctrico	Motor Eléctrico	Motor Eléctrico	Motor Eléctrico
<b>Propósito</b>	Rehabilitación – Movimiento Voluntario	Rehabilitación – Movimiento Voluntario	Rehabilitación – Movimiento Voluntario	Rehabilitación – Movimiento Voluntario
<b>Método de detección</b>	Movimiento y Fuerza	Movimiento y Torque de motor	Movimiento, Fuerza y sEMG	Movimiento y Fuerza
<b>Transmisión de Potencia</b>	Cable	Eslabones Seriales	Palancas y Poleas	Eslabones Seriales

#### 1.5. Modelo cinemático de la mano

Existen varios modelos para describir la cinemática de la mano, muchos de ellos son simplificados debido a la complejidad de los grados de libertad y a las restricciones. Se consideran tres modelos utilizados en distintos artículos, el primero es un modelo de 26 grados de libertad [18], el segundo es un modelo de 24 grados de libertad que utiliza los parámetros Denavit – Hartenberg standard [19], y el tercero es un modelo con 24 grados de libertad que utiliza los parámetros Denavit – Hartenberg modificado [11].

El primer modelo presentado en [18], es definido por eslabones correspondientes a cada uno de los huesos de los dedos y con articulaciones giratorias entre las uniones de los huesos. El ángulo medido entre cada eslabón se lo conoce como el ángulo de la articulación. La mano humana está restringida por dependencias que reducen el modelo a 30 grados de libertad. Se consideran 5 ángulos de articulación por cada dedo, 3 para el movimiento de flexión, 1 para el movimiento de abducción y 1 para el movimiento de la muñeca, este modelo cambia para el dedo pulgar. En el dedo pulgar se consideran 4

ángulos de articulación, 3 para flexión y 1 para abducción. La palma de la mano tiene 6 grados de libertad para la translación y rotación de la muñeca, en total se tienen 30 grados de libertad. Para simplificar el modelo se utiliza la dependencia existente entre la articulación DIP y PIP, lo cual hace que se elimine un grado de libertad por cada dedo, excepto en el pulgar. Con esta simplificación se obtienen 26 grados de libertad. Dichos ángulos de las articulaciones se observan en la Figura 1.11. No se utiliza este modelo debido a que considera el movimiento desde la muñeca de la mano, para el presente exoesqueleto el movimiento de la muñeca no influye en el movimiento del mecanismo.

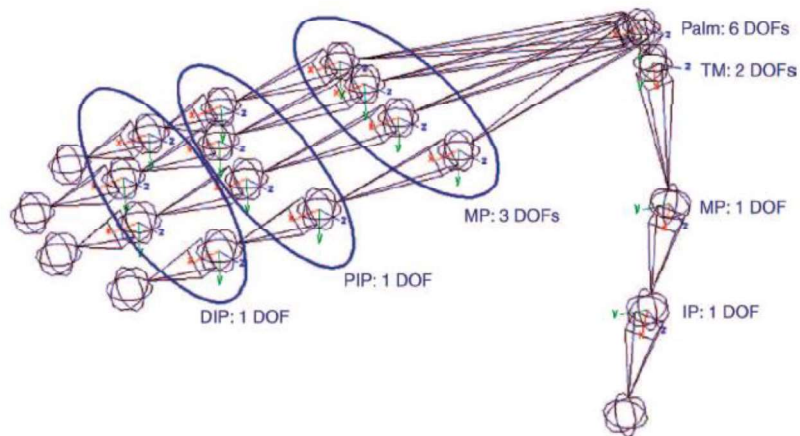


Figura 1.11. Modelo cinemático de la mano de [18]. Tiene 26 grados de libertad y considera el movimiento de la palma de la mano con 6 grados de libertad adicionales. La articulación DIP se le considera dependiente de la articulación PIP. DOF = grados de libertad.

(Fuente: [18])

El segundo modelo es presentado en [19] y tiene 24 grados de libertad. El modelo cinemático utilizado se basa en el esqueleto humano, en el cual se utilizan 19 eslabones correspondientes a cada hueso que conforma la palma de la mano y los dedos, los 24 grados de libertad corresponden a las articulaciones que permiten los movimientos angulares de cada eslabón. Se presentan dos modelos cinemáticos uno para el pulgar y otro para el resto de los dedos. Para los dedos índice, medio, anular y meñique se utiliza un modelo de 4 eslabones y 5 grados de libertad, 4 grados de libertad para el movimiento de flexión y 1 para el movimiento de abducción. Mientras que para el pulgar se utiliza un modelo con 3 eslabones y 4 grados de libertad, 3 grados de libertad para el movimiento de flexión y 1 para el movimiento de abducción.

Para obtener los modelos cinemáticos se utiliza los parámetros Denavit-Hartenberg standard. El primer modelo es para los dedos índice, medio, anular y meñique, en este cada hueso del dedo es tomado como un eslabón de una cadena cinemática serial. La

articulación MCP es de dos grados de libertad, mientras que las articulaciones CMC, PIP y DIP son de un grado de libertad. La referencia para este modelo se ubica en la muñeca. De igual forma se presentan los modelos cinemáticos para el dedo pulgar, en el cual se consideran los cuatro huesos que lo componen, trapecio, metacarpo, falange proximal y falange distal. Las articulaciones consideradas son la TMC con dos grados de libertad y las articulaciones MCP e IP con un grado de libertad. Dichos modelos se observan en la Figura 1.12.

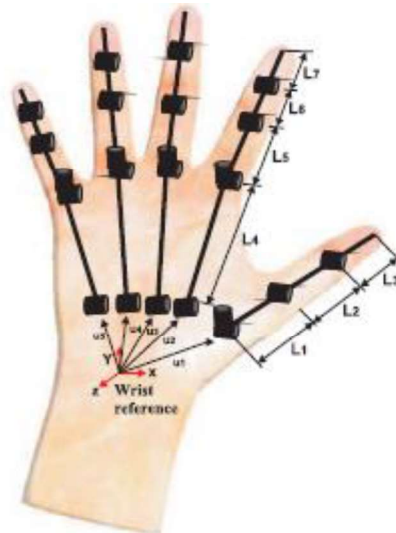


Figura 1.12. Modelo cinemático de la mano de [19]. Presenta dos modelos distintos uno para el pulgar con 4 grados de libertad y otro para el resto de los dedos, con 5 grados de libertad. No se considera el movimiento de la muñeca, sino que es la referencia para el movimiento del resto de eslabones.

(Fuente: [19])

El modelo cinemático presentado en [11], es muy similar al anterior, ya que considera 19 huesos y 24 grados de libertad modelados por juntas. También, se utilizan dos modelos cinemáticos uno para el dedo pulgar y otro para los dedos trifalángicos. La cinemática directa se utiliza para obtener las coordenadas de la punta del dedo en función de los ángulos de las juntas Figura 1.13. La diferencia con el modelo anterior es que se utiliza los parámetros de Denavit – Hartenberg modificado. En la Figura 1.14, se observa la configuración cinemática de los dedos de la mano.

Se utiliza como referencia el modelo propuesto en [11], debido a que el autor prefiere utilizar los parámetros Denavit – Hartenberg modificado. Es importante recalcar que el modelo propuesto en [11] es un modelo general, del cual se realizan ciertas modificaciones para adaptarlo al sistema que se necesita para el presente desarrollo.



Figura 1.13. Diagrama de bloques de la cinemática directa  
(Fuente: propia)

El modelo presentado en [11] se utilizan dos modelos distintos, uno para los dedos trifalángicos y otro para el dedo pulgar. En los dedos trifalángicos se tiene una junta rotacional en la articulación CMC (en inglés Carpometacarpal), dicha articulación es la unión de la palma de la mano con la muñeca y permite un movimiento conjunto de todos los dedos incluido la palma de la mano. Dos juntas rotacionales en la articulación MCP, la primera permite el movimiento de flexión y extensión de cada dedo y la segunda permite el movimiento de abducción y aducción de cada dedo. Una junta rotacional en la articulación PIP, que permite el movimiento de flexión y extensión de la articulación. Una junta rotacional en la articulación DIP, que permite el movimiento de flexión y extensión de la articulación. Dichas articulaciones se pueden observar en la Figura 1.14

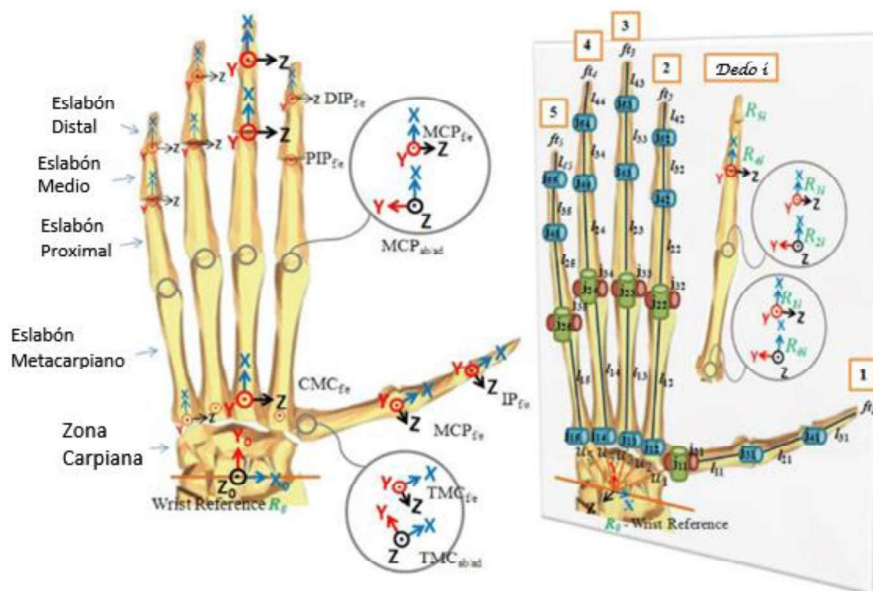


Figura 1.14. Configuración cinemática de los dedos de la mano propuesta en [11]. Tiene 24 grados de libertad y toma como punto de referencia la muñeca. En la izquierda se observan los ejes de coordenadas de cada articulación, aplicando el criterio de Denavit - Hartenberg. En la derecha se observa cada uno de los grados de libertad que se consideran en las articulaciones.

(Fuente: [11])



Para el dedo pulgar la configuración cinemática es distinta que en los dedos trifalángicos. En el caso del dedo pulgar se tiene dos juntas rotacionales en la articulación TMC, una para el movimiento de flexión y extensión y otra para el movimiento de abducción y aducción. En la articulación MCP se tiene una junta rotacional que permite el movimiento de flexión y extensión. En la articulación IP se tiene una junta rotacional que permite el movimiento de flexión y extensión. Se puede observar en la Figura 1.14.

Los parámetros de Denavit – Hartenberg para los dedos trifalángicos que se proponen en [11], se presentan en la Tabla 1.2 . Para el ejemplo se utiliza los parámetros del dedo índice. Hay que considerar que en el estudio se toma en cuenta las juntas desde la muñeca, incluido la articulación CMC.

Tabla 1.2. Parámetros Denavit - Hartenberg para el dedo índice

<b>Junta</b>	$\alpha_{i-1}$	$a_{i-1}$	$d_i$	$\theta_i$
$j_{12}$	$\pi/2$	0	0	$\theta_{CMC\ f/e}$
$j_{22}$	$-\pi/2$	$L_{12}$	0	$\theta_{MCP\ ab/ad}$
$j_{32}$	$\pi/2$	0	0	$\theta_{MCP\ f/e}$
$j_{42}$	0	$L_{22}$	0	$\theta_{PIP\ f/e}$
$j_{52}$	0	$L_{32}$	0	$\theta_{DIP\ f/e}$

(Fuente: [11])

Los valores  $L_{12}$ ,  $L_{22}$  y  $L_{32}$  corresponden a los tamaños de los huesos metacarpo, falange proximal y falange media, correspondientemente. Los valores de  $\theta$  corresponden a los ángulos de movimiento de cada articulación, dependiendo si el movimiento es de flexión y extensión o de abducción y aducción.

Para el exoesqueleto propuesto en el presente trabajo se omite la articulación CMC, debido a que el movimiento de la palma y de la muñeca no influyen en el movimiento del mecanismo del exoesqueleto diseñado, por tanto, el estudio cinemático de la mano se realiza variando los parámetros DH propuestos en [11]. Se elimina los parámetros correspondientes a la articulación CMC, quedando, como se observan en la Tabla 1.3. Además, se considera la falange distal que lleva desde la junta 42 hasta la punta del dedo, por esto se incluye la  $L_{42}$ . Además, se elimina la distancia  $L_{12}$ , que corresponde al metacarpo, el cual está en la palma de la mano, eslabón que no se considera en el presente proyecto. Por tanto, los nuevos parámetros corresponden al dedo sin tomar en

cuenta la palma de la mano, la base de la cadena está en la articulación MCP, con sus movimientos de abducción y aducción y de flexión y extensión.

Tabla 1.3. Parámetros Denavit - Hartenberg para el dedo índice sin considerar la articulación CMC

<b>Junta</b>	$\alpha_{i-1}$	$a_{i-1}$	$d_i$	$\theta_i$
$j_{22}$	0	0	0	$\theta_{MCP\ ab/ad}$
$j_{32}$	$\pi/2$	0	0	$\theta_{MCP\ f/e}$
$j_{42}$	0	$L_{22}$	0	$\theta_{PIP\ f/e}$
$j_{52}$	0	$L_{32}$	0	$\theta_{DIP\ f/e}$
$j_{62}$	0	$L_{42}$	0	0

En el caso del modelo cinemático utilizado para el dedo pulgar se considera la articulación TMC, la cual es la unión entre el dedo pulgar y la muñeca, debido a que los movimientos de flexión, extensión, abducción y aducción se presentan desde dicha articulación. Los parámetros Denavit Hartenberg para el dedo pulgar se obtienen de [11]. Y son los parámetros que se indican en la Tabla 1.4.

Tabla 1.4. Parámetros Denavit – Hartenberg para el dedo pulgar

<b>Junta</b>	$\alpha_{i-1}$	$a_{i-1}$	$d_i$	$\theta_i$
$j_{11}$	0	0	0	$\theta_{TMC\ ab/ad}$
$j_{21}$	$\pi/2$	0	0	$\theta_{TMC\ f/e}$
$j_{31}$	0	$L_{11}$	0	$\theta_{MCP\ f/e}$
$j_{41}$	0	$L_{21}$	0	$\theta_{IP\ f/e}$

(Fuente: [11])

Ambos modelos cinemáticos son comprobados utilizando el módulo robot de MATLAB, pero también se utilizan los parámetros para obtener las matrices de rotación que permiten calcular la posición de la punta del dedo en función de los ángulos de rotación de cada articulación.

## 2. METODOLOGÍA

### 2.1. Estudio de la cinemática del mecanismo

El diseño mecánico del exoesqueleto es realizado con proyectos de titulación previos de la carrera de Ingeniería Mecánica [5], [4]. El diseño del mecanismo se divide en dos partes principales, la primera constituye los mecanismos necesarios del exoesqueleto de los dedos trifalángicos, es decir el índice, medio, anular y meñique, dicho diseño es realizado en [5]. La segunda parte contiene todos los mecanismos necesarios para el movimiento del dedo pulgar, dicho diseño se presenta en [4].

#### 2.1.1. Mecanismo para los dedos trifalángicos [5]

El mecanismo diseñado para los dedos trifalángicos se presenta en la Figura 2.1. Cada dedo está constituido por una cadena de eslabones que por un lado están sujetos al mecanismo y por el otro se unen a la punta del dedo correspondiente. Para generar los movimientos se utilizan 2 motores. El primer motor está conectado a través de engranes a los mecanismos del dedo índice y medio, y el segundo motor está unido a los mecanismos del dedo anular y meñique. Los mecanismos de cada dedo son iguales, pero tienen regulaciones para ajustar dependiendo de la longitud del dedo que se desea manejar. Sobre la palma de la mano se juntan los mecanismos de cada dedo a una placa común, la cual se une a la muñeca con una cinta velcro.

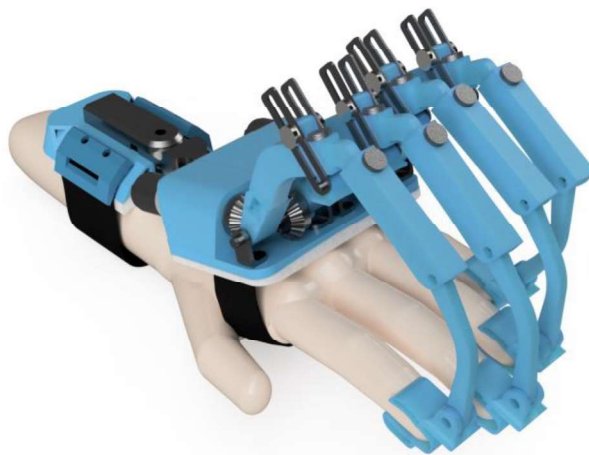


Figura 2.1. Mecanismo para los dedos trifalángicos. Se observa el mecanismo diseñado en [5] en conjunto con la mano. Los eslabones del mecanismo unidos a la punta del dedo, y por el otro lado se unen a los engranes que son movidos por el motor.

(Fuente: [5])

### 2.1.2. Mecanismo para el dedo pulgar

El mecanismo diseñado para el dedo pulgar se presenta en la Figura 2.2. Dentro del movimiento del pulgar se compone de 2 movimientos principales, definidos como flexión – extensión y abducción - aducción. El mecanismo se diseña para satisfacer ambos movimientos. Por tanto, se utilizan dos motores, uno para generar el movimiento de flexión – extensión y otro para el movimiento de abducción – aducción.

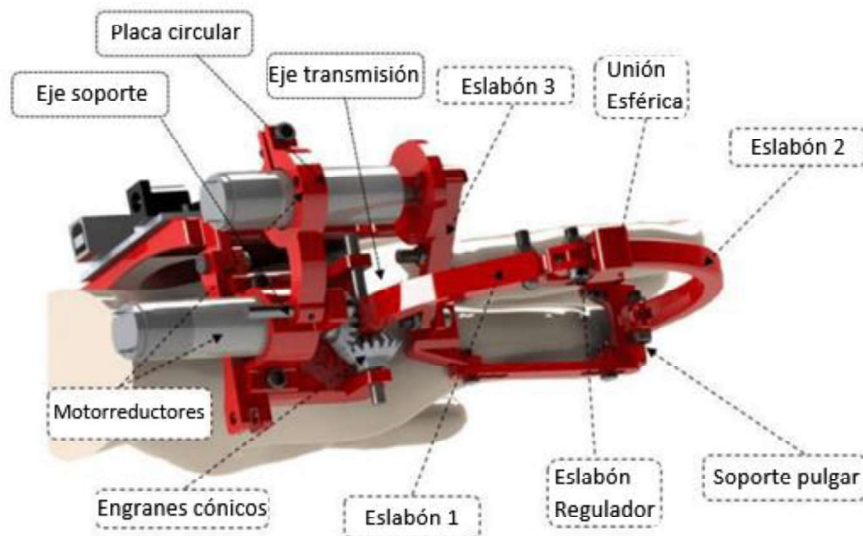


Figura 2.2. Mecanismo del exoesqueleto para el pulgar

(Fuente: [4])

### 2.1.3. Modelo cinemático del mecanismo

Para el estudio del modelo cinemático del mecanismo es importante considerar que tipo de exoesqueleto se tiene de acuerdo con la Figura 1.5. Para el criterio del tipo de actuador, el exoesqueleto tiene un actuador activo que es un motor de corriente continua. En el criterio de propósito, es para rehabilitación en primera instancia el desarrollo se hace para movimientos pasivos, sin embargo, el diseño mecánico permite generar movimientos voluntarios. La transmisión de potencia es en base a eslabones seriales que se unen al dedo en la punta, siendo un mecanismo subactuado. En el método de detección de la intención es en base a sensores de movimiento para el control de posición, pero en caso de utilizarlo para movimientos activos se cuenta con medición de fuerza. Es importante recalcar que para el estudio cinemático del mecanismo se considera por separado al mecanismo de los dedos trifalángicos y del pulgar.

El mecanismo de los dedos trifalángicos, se observa en la Figura 2.1. Está formado por dos eslabones y tres articulaciones. La articulación 1 es motorizada, mientras que la articulación 2 y 3 están libres, por lo tanto, se define al sistema como subactuado.

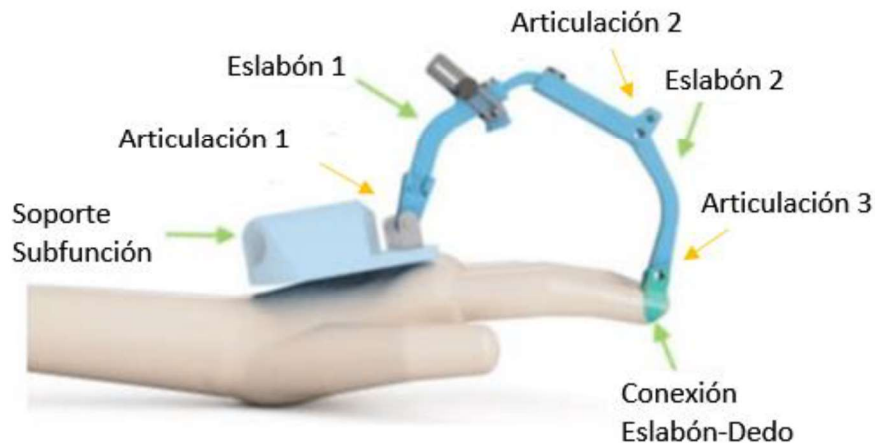


Figura 2.3. Mecanismo de eslabones para los dedos trifalángicos. En azul el mecanismo del exoesqueleto, la mano con color piel.

(Fuente: [13])

Para obtener el modelo cinemático del mecanismo es necesario definir la cadena cinemática. El eslabón 1 (Figura 2.3) está formado por tres segmentos rectos, unidos por ángulos fijos. El eslabón 2 está formado por 2 segmentos unidos por un ángulo fijo.

El diagrama esquemático de las articulaciones del mecanismo en unión con el dedo se muestra en la Figura 2.4, se observan 3 articulaciones de rotación en el dedo, cuyos ángulos están representados por  $\theta_1$ ,  $\theta_2$  y  $\theta_3$  para las articulaciones MCP, PIP y DIP, correspondientemente. El mecanismo del exoesqueleto tiene 3 articulaciones de rotación representadas por  $\alpha_1$ ,  $\alpha_2$  y  $\alpha_3$  en los puntos A, F y H.

Los puntos B, D y G son uniones de eslabones que no rotan, además, se observan 2 articulaciones prismáticas (color rojo, puntos C y E), las cuales permiten incrementar la longitud de los eslabones correspondientes, cabe recalcar que durante el movimiento del mecanismo estas articulaciones permanecen fijas y se ajustan dependiendo de la longitud del dedo que se va a colocar en el mecanismo. Para la simulación se toma estas distancias como fijas, por tanto, se omiten por facilidad en la simulación. Las longitudes de cada eslabón se presentan en la Tabla 2.1.

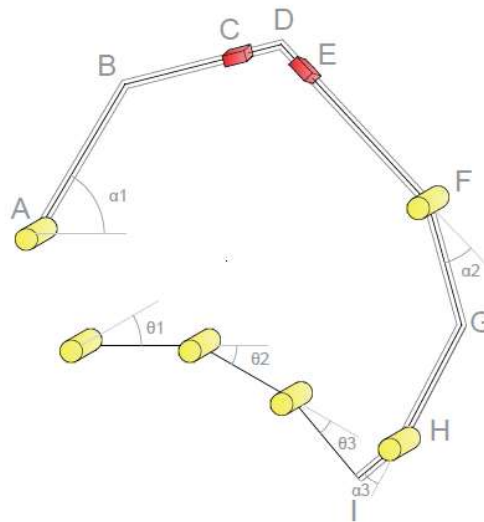


Figura 2.4. Representación de las articulaciones para un dedo trifalángico. Las articulaciones de color amarillo son rotativas y las de color rojo son prismáticas.  
(Fuente: propia)

Tabla 2.1. Medidas de los segmentos del mecanismo de los dedos trifalángicos

Segmento	Medida [cm]
AB	3.9
BC	2.6
CD (longitud mínima)	1
CD (longitud máxima)	1
DE (longitud mínima)	3.2
DE (longitud máxima)	3.2
EF	3.9
FG	3
GH	3
HI	1.4

Para definir los parámetros Denavit – Hartenberg del mecanismo se utiliza el algoritmo Denavit – Hartenberg, el cual incluye 16 pasos. Los resultados obtenidos del algoritmo se presentan en la Tabla 2.2. Hay que considerar que estos parámetros son para obtener una simulación más didáctica en la que se observen los eslabones completos del mecanismo, ya que, si solo se consideran las juntas móviles, el mecanismo se vería reducido a 3 eslabones rectos. Por lo tanto, a pesar de que los puntos B, D y G son ángulos fijos se los incluye como articulaciones para que se observe cada segmento del

mecanismo. De igual forma se utiliza el módulo robot de MATLAB para obtener la simulación.

Tabla 2.2. Parámetros Denavit – Hartenberg para el mecanismo de los dedos trifalángicos

<b>Junta</b>	$\alpha_{i-1}$	$a_{i-1}$	$d_i$	$\theta_i$
$j_{11}$	$\pi/2$	0	0	$\alpha_1$
$j_{21}$	0	3.9	0	$-\pi/4$
$j_{31}$	0	5	0	$-\pi/3$
$j_{41}$	0	7	0	$\alpha_1$
$j_{51}$	0	3	0	$-42.08*\pi/180$
$j_{61}$	0	3	0	$\alpha_1$
$j_{71}$	0	1.4	0	0

Uno de los problemas que existen al momento de realizar el diseño de un exoesqueleto mecatrónico, es la adaptabilidad. Cada mano tiene diferentes dimensiones, por lo tanto, el exoesqueleto debe ser adaptable a los tamaños de cada mano. Dentro del diseño mecánico se ha definido las articulaciones prismáticas para que el exoesqueleto sea adaptable a varios tamaños de manos.

En el proyecto de titulación que se realiza el diseño mecánico del exoesqueleto [5] de los dedos trifalángicos, se presenta las condiciones críticas de diseño que se observan en la Tabla 2.3. Se utilizan dichos parámetros dentro de los modelos cinemáticos para determinar los espacios de trabajo de la mano y del mecanismo. Los espacios de trabajo se presentan en los resultados

Tabla 2.3. Parámetros de diseño en condiciones críticas de los dedos trifalángicos

<b>Parámetros de Diseño</b>	
Flexión mínima MCP	$0^\circ$
Flexión mínima PIF	$0^\circ$
Flexión mínima DIF	$0^\circ$
Flexión máxima MCP	$-90^\circ$
Flexión máxima PIF	$-90^\circ$
Flexión máxima DIF	$-60^\circ$
Longitud máxima de falange proximal	0.064 [m]
Longitud máxima de falange media	0.039 [m]

Longitud máxima de falange distal	0.03 [m]
Longitud mínima de falange proximal	0.022 [m]
Longitud mínima de falange media	0.015 [m]
Longitud mínima de falange distal	0.01 [m]
Longitud eslabón 2	0.056 [m]
Distancia de acople entre dedo y eslabones (longitud mínima)	Y: 0.01 [m] X: 0.012
Distancia de acople entre dedo y eslabones (longitud máxima)	Y: 0.01 [m] X: 0.03

Fuente: [5]

En el caso del mecanismo para el dedo pulgar, en la Figura 2.5 se tiene la representación del mecanismo. La articulación TMC del dedo pulgar es una articulación de dos grados de libertad, en esta se observan los movimientos de abducción – aducción (articulación color azul) cuyo movimiento está representado por el ángulo  $\theta_1$  y de flexión – extensión (articulación color amarillo) representado por el ángulo  $\theta_2$ . La articulación MCP e IP tienen un grado de libertad cada uno cuyos ángulos de movimiento están representados por  $\theta_3$  y  $\theta_4$  respectivamente. El mecanismo del exoesqueleto está formado por el eslabón AB, el cual tiene una forma arqueada y que está unida en el punto A al cuerpo del exoesqueleto, la articulación A es una articulación motorizada cuyo ángulo de movimiento está representado por  $\alpha_1$ , y que genera el movimiento de flexión – extensión sobre el pulgar. La articulación B es de dos grados de libertad y el movimiento de dicha articulación se representa con los ángulos  $\alpha_2$  y  $\alpha_3$ . El eslabón BC también tiene forma de arco y une al mecanismo con la punta del dedo en el punto C.

Para generar el movimiento de abducción – aducción se utiliza el eslabón que se une desde el punto D hasta la articulación MCP del pulgar. Dicha articulación es prismática cuyo movimiento es motorizado. Dentro de las condiciones de diseño mecánico se incluye que el exoesqueleto se pueda utilizar para la mano derecha e izquierda. El mecanismo para el movimiento de abducción – aducción se cambia para utilizar en la otra mano. Las longitudes de los segmentos del mecanismo se presentan en la Tabla 2.4, dichas longitudes se han medido en línea recta en los eslabones curvos, por facilidad en las simulaciones.



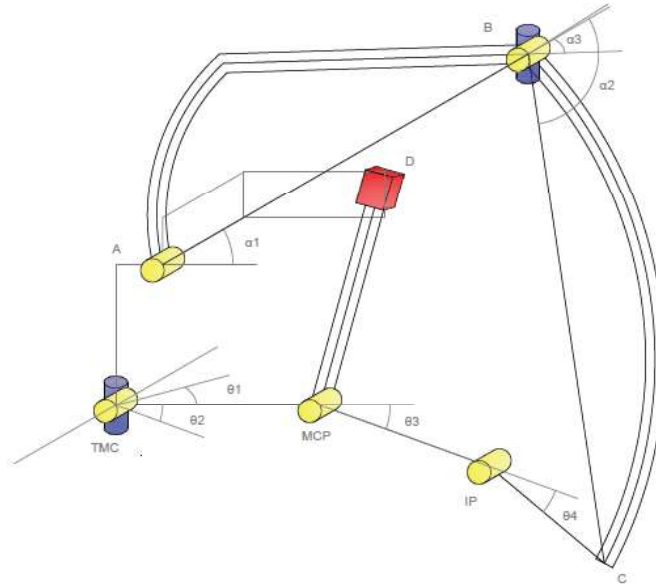


Figura 2.5. Representación del mecanismo para el dedo pulgar. En color amarillo articulaciones rotativas para el movimiento de flexión, en color azul articulaciones rotativas para el movimiento de abducción, en rojo articulaciones prismáticas.

(Fuente: propia)

Tabla 2.4. Longitudes de los segmentos del mecanismo del dedo pulgar

Eslabón	Longitud [cm]
AB	11
BC	9
D-MCP	5

Se obtienen los parámetros Denavit – Hartenberg para el mecanismo del dedo pulgar, de igual forma se utiliza la metodología para determinar dichos parámetros. También se consideran los eslabones curvos como líneas rectas por facilidad en la simulación. Los parámetros Denavit – Hartenberg se obtienen para el movimiento de flexión – extensión y se presentan en la Tabla 2.5.

Tabla 2.5. Parámetros Denavit – Hartenberg para el mecanismo del dedo pulgar

Junta	$\alpha_{i-1}$	$a_{i-1}$	$d_i$	$\theta_i$
		1		
$j_{11}$	$\pi/2$	0	0	$\alpha_1$
$j_{21}$	0	11	0	$\alpha_2$
$j_{31}$	$-\pi/2$	0	0	$\alpha_3$
$j_{41}$	0	9	0	0

En el proyecto de titulación [4], donde se realiza el diseño mecánico del exoesqueleto para el pulgar no se presentan de forma clara las condiciones críticas de diseño para los distintos tamaños de falanges del pulgar. Sin embargo, se presenta tablas en las que se muestran datos recolectados de distintos tamaños de las falanges del pulgar, se considera estos tamaños para el estudio cinemático del mecanismo del pulgar. La Tabla 2.6, presenta las medidas recolectadas de los tamaños de las falanges y los ángulos para los cuales se varía el espacio de trabajo del pulgar se presentan en la Tabla 2.7, valores que son tomados de [11].

Tabla 2.6. Valores de longitudes de las falanges que componen el dedo pulgar [4]

<b>Falanges</b>		[cm]
<b>Distal</b>	Longitud Superior	1.9
	Longitud Inferior	4
	Promedio	2.5
<b>Proximal</b>	Longitud Superior	2.5
	Longitud Inferior	5.2
	Promedio	3.3
<b>Metacarpal</b>	Longitud Superior	4.2
	Longitud Inferior	5.2
	Promedio	3.1

Tabla 2.7. Restricciones estáticas del pulgar

<b>Pulgar</b>	<b>Flexión</b>	<b>Extensión</b>	<b>Abducción/aducción</b>
<b>TMC</b>	-50°	15°	45° - 60°
<b>MCP</b>	-75°	0°	0°
<b>IP</b>	-75°	5°	0°

De igual forma se utilizan los datos obtenidos del modelo cinemático, para utilizarlo en conjunto con el modelo cinemático del dedo y de esta forma analizar el espacio de trabajo que se puede lograr con el mecanismo, los resultados se presentan en el Capítulo 3.

## 2.2. Diseño de la Instrumentación

El exoesqueleto para su funcionamiento necesita realimentar señales tomadas del mecanismo. Se ha determinado que lo necesario para poder controlar el mecanismo es

conocer su posición y además medir la fuerza que se aplica desde el mecanismo a la mano y viceversa. Por tanto, se utilizan 3 tipos de sensores en el exoesqueleto 2 para medir posiciones angulares y uno para medir la fuerza aplicada.

### 2.2.1. Medición de posición con flexosensores

Para medir la posición angular de las falanges de los dedos trifalángicos se utilizan flexosensores FS-L-0055-253-ST, que son sensores que permiten medir el desplazamiento angular. El principio de transducción se basa en la variación de la resistencia del material piezoeléctrico del cual está formado, dependiendo de la variación del ángulo en el cual está dispuesto el sensor. En la Figura 2.6, se observa al sensor y un esquema de su funcionamiento.

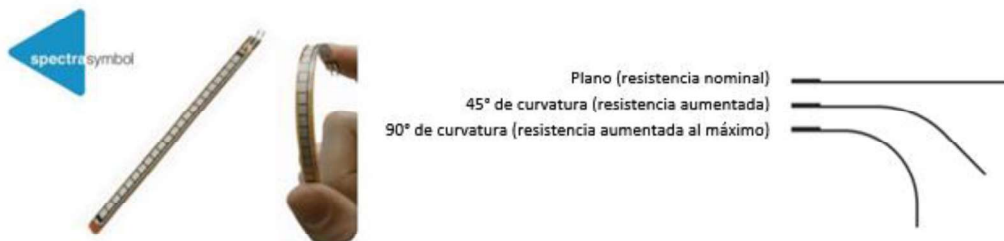


Figura 2.6. Flexosensor FS-L-0055-253-ST. Variación de la Resistencia dependiendo del ángulo de curvatura del sensor.

(Fuente: [20])

Para la medición de los ángulos de las articulaciones de los dedos trifalángicos se utilizan 6 flexosensores, que se ubican 4 de ellos dentro de un guante y 2 en el mecanismo. Considerando que el mecanismo de los dedos trifalángicos tiene dos motores que permiten el movimiento de los dedos en pares, es decir un motor mueve los dedos índice y medio y el otro motor mueve los dedos anular y meñique, el movimiento de ambos dedos es similar, por tanto, la medición de los ángulos de los dedos se hace en el par de dedos que se mueven juntos. Entonces 3 flexosensores se utilizan para medir las articulaciones MCP, DIP y PIP del primer par de dedos y los otros 3 flexosensores miden las articulaciones del segundo par de dedos. La ubicación de los sensores considerando que se utiliza el mecanismo en la mano izquierda se muestra en la Tabla 2.8.

Tabla 2.8. Ubicación de Flexosensores

	<b>Código</b>	<b>Dedo</b>	<b>Articulación</b>
1	MCP1	Meñique	MCP
2	PIP1	Anular	PIP
3	DIP1	Meñique	DIP
4	MCP2	Medio	MCP
5	PIP2	Indice	PIP
6	DIP2	Medio	DIP

La hoja de datos del sensor muestra la salida del sensor. El rango de variación de la resistencia dependiendo del ángulo en el que está flexionado el sensor es desde 45 a 125 k $\Omega$ . Se consideran estos datos para diseñar el circuito de acondicionamiento que necesita el sensor. Es importante considerar que este valor de resistencia hay que cambiar a niveles de voltaje, y este nivel de voltaje acondicionar al rango de entrada que recibe el controlador, que es de 0 a 5V. En la Figura 2.7, se muestra un diagrama de bloques donde se explica los rangos de las variables que se miden y se entregan del sensor.

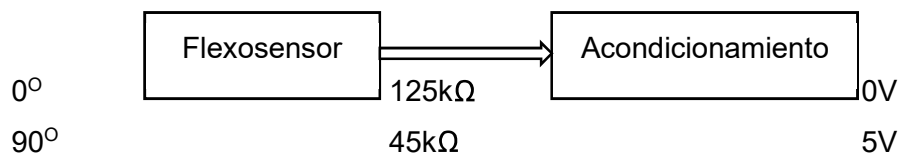


Figura 2.7. Rangos de las variables del acondicionamiento del flexosensor. La primera etapa el flexosensor recibe la variación del ángulo de rotación y convierte la señal en variación de resistencia. La siguiente etapa es el circuito de acondicionamiento que cambia los valores de resistencia a voltaje normalizado.

(Fuente:propia)

El circuito de acondicionamiento se muestra en la Figura 2.8, e incluye la primera etapa que permite cambiar los niveles de resistencia a voltaje, se lo hace a través de un puente de Wheatstone. La siguiente etapa permite amplificar el nivel de voltaje del puente Wheatstone para convertirlo en una señal de voltaje de 0 a 5V. Finalmente se pone un circuito limitador para evitar que el voltaje que llegue al microcontrolador sea mayor de 5V.

Debido a que se utilizan 6 flexosensores, este circuito se replica para cada flexosensor y se envía las señales a las entradas analógicas del controlador. En el controlador se

utiliza dichas señales para saber la posición angular de cada articulación de los dedos trifalángicos.

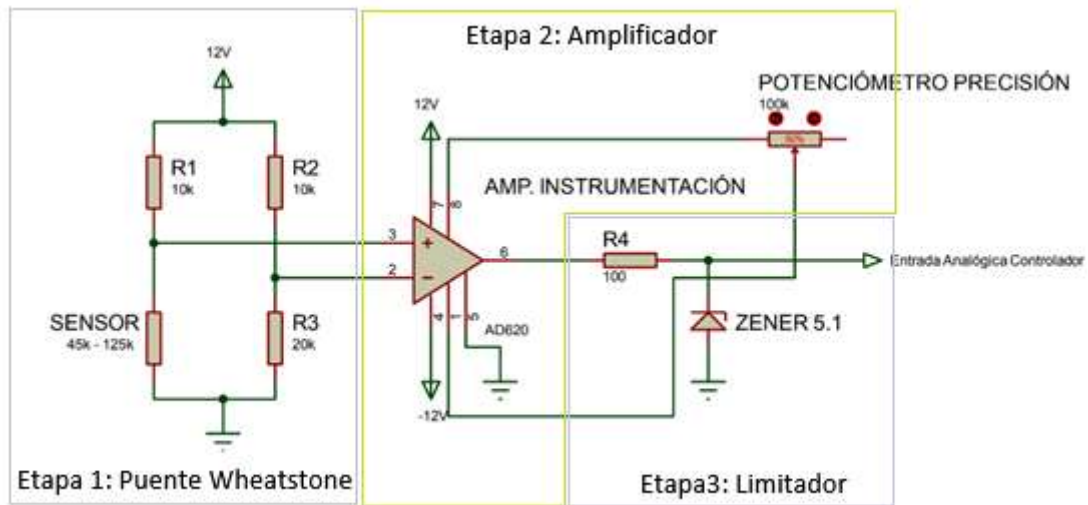


Figura 2.8. Circuito de Acondicionamiento del Flexosensor. Primera etapa convierte la señal de resistencia a voltaje. Segunda etapa amplifica la señal de voltaje a un valor normalizado.

Tercera etapa limita el valor máximo de voltaje a 5.1 V.

(Fuente: propia)

### 2.2.2. Medición de posición con sensores de posición angular

Para medir la posición angular del dedo pulgar, debido a la forma del dedo, no se puede determinar ciertos ángulos de las articulaciones con flexosensores, por lo tanto, se decide utilizar sensores de posición rotativos basados en potenciómetros para medir la posición angular de los motores utilizados. Como se utilizan dos motores para el movimiento del mecanismo del pulgar, se utilizan dos potenciómetros adaptados a los ejes de cada motor. El sensor utilizado es CJMCU-103 basado en el potenciómetro SV01 de murata y su imagen se observa en la Figura 2.9.

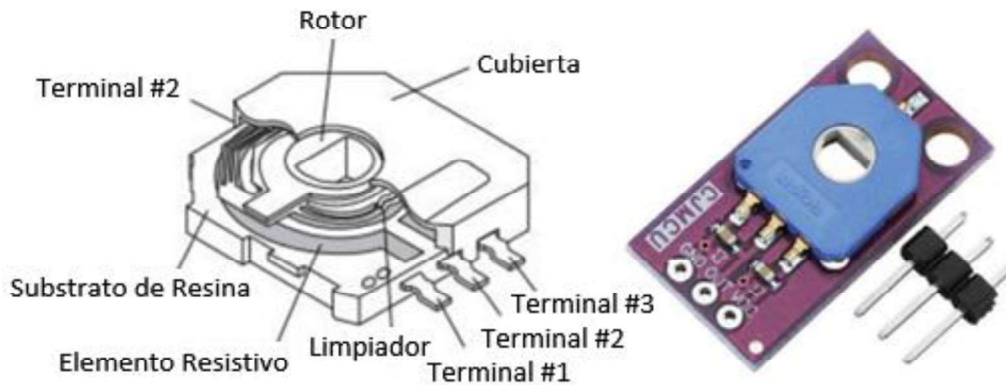


Figura 2.9. Sensor de posición rotativo. Izquierda SV01 Murata, derecha CJMCU-103  
(Fuente: [21])

Debido a que este sensor es un potenciómetro se conecta los terminales fijos a la fuente de voltaje de 5V y el punto variable entrega ya una señal de voltaje proporcional a la posición angular de los motores. Dicha señal se la pasa a través de un circuito seguidor para acoplar impedancias del sensor con la entrada al controlador. Luego se ubica un limitador para evitar que la señal se pase de los 5.1V. El circuito que se utiliza se observa en la Figura 2.10. Hay que considerar que se utilizan dos sensores de este tipo, por lo tanto, se replica el circuito observado para cada sensor utilizado.

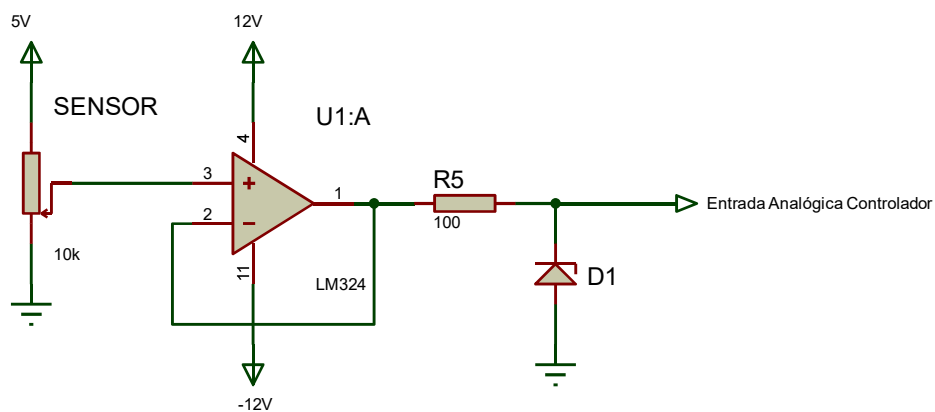


Figura 2.10. Circuito de Acondicionamiento del Potenciómetro. Desde la izquierda se tiene el sensor, a continuación, el amplificador U1:A para el seguidor y la resistencia R5 y Zener D1 corresponden al limitador  
(Fuente: propia)

### 2.2.3. Medición de Fuerza

Para medir la fuerza que se aplica desde el mecanismo a la mano y viceversa se utiliza sensores de fuerza resistivos FSR 402. Dicho sensor tiene una salida que varía su resistencia dependiendo de la fuerza que se aplique sobre el sensor. En el diseño inicial se ubican dos sensores de fuerza en cada dedo, para medir la fuerza que se genera en ambos sentidos, movimiento de flexión y movimiento de extensión. Por lo tanto, se utilizan 10 sensores resistivos.



Figura 2.11. Sensor de fuerza resistivo. FSR 402. La parte circular es donde se aplica la fuerza. En la parte izquierda se tiene la conexión eléctrica.

(Fuente: [22])

La hoja de datos del sensor explica que el rango de medición del sensor es de 0 a 10N, la salida de la resistencia en función de la fuerza aplicada se observa en la Figura 2.12. También se observa un circuito recomendado para su utilización. Sin embargo, por facilidad se cambia este circuito a uno más apropiado a la situación, el cual se describe más adelante.

Figura 1 – Curva de Fuerza

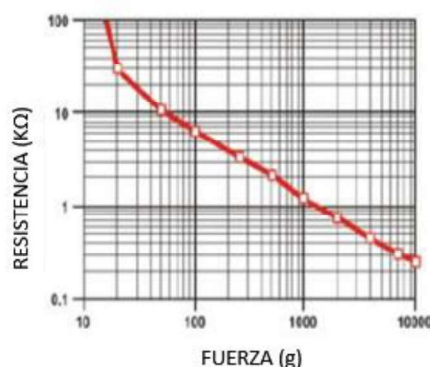


Figura 2 – Esquema

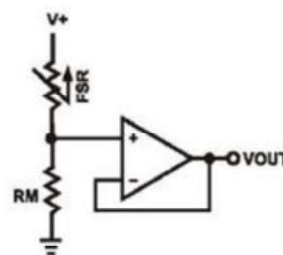


Figura 2.12. Curva característica del sensor y circuito de acondicionamiento. Izquierda: Curva de variación de la resistencia dependiendo de la fuerza aplicada al sensor. Derecha: Circuito recomendado por el fabricante.

(Fuente: [22])

El circuito utilizado para el acondicionamiento de este sensor se observa en la Figura 2.13. Debido a que los cables que se utilizan para llevar la señal desde el sensor que está ubicado en el exoesqueleto hasta el circuito de acondicionamiento son cables trenzados STP, se prefiere enviar la señal del sensor con el punto de tierra y el enchaquetado del cable conectado también a tierra para evitar interferencias. Por este motivo se cambia la posición del sensor en el divisor de voltaje. Como se utilizan 10 sensores de fuerza, se replica el circuito para cada uno de los sensores.

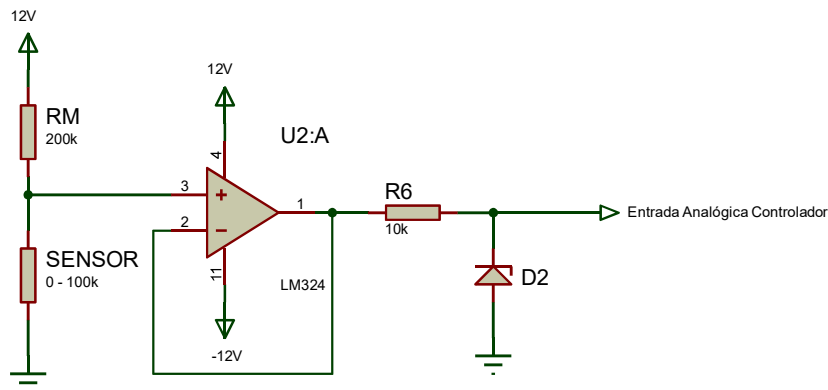


Figura 2.13. Circuito de Acondicionamiento Sensor de Fuerza. Divisor de voltaje para cambiar la señal de resistencia a voltaje, el amplificador como seguidor para acoplar impedancias y un limitador para evitar que el voltaje supere los 5.1V.

(Fuente: propia)

Todos los circuitos de acondicionamiento presentados se los coloca en una placa electrónica, la cual se comunica con el exoesqueleto a través de cables y también con el circuito de control. El diagrama esquemático que incluye todos los circuitos se observa en la Figura 2.14.

Como se observa en la Figura 2.14, el circuito de acondicionamiento incluye una fuente de +-12V. Dicha fuente se utiliza para la polarización de los circuitos de acondicionamiento. Existen partes del circuito que deben ser alimentadas con 5V, dicha alimentación se la toma desde las fuentes de los controladores. Además, se observa que se incluyen 8 pulsantes para control manual de los motores, dichos pulsantes se utilizan para generar movimientos directamente sobre los motores.



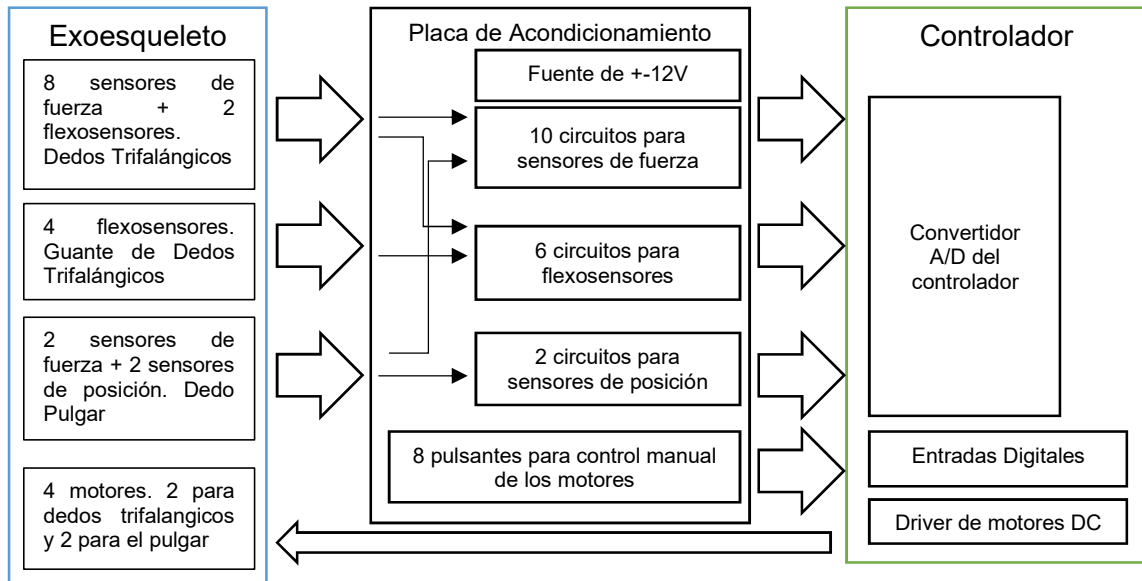


Figura 2.14. Esquema de conexión de los circuitos de acondicionamiento. El exoesqueleto envía las señales de los sensores a la placa de acondicionamiento. La placa envía las señales acondicionadas al controlador. El exoesqueleto recibe del controlador las señales para el movimiento de los motores.

(Fuente: propia)

## 2.3. Diseño del Sistema de Control

El sistema de control toma las señales que vienen de los sensores y utiliza esa información para determinar las acciones que deben hacer los motores para generar los movimientos del mecanismo. Para el sistema de control se determinó el uso de dos tarjetas de desarrollo, la primera un Arduino Mega 2560 y la segunda un Arduino Uno. Para el manejo de los motores se utiliza el driver Motor Shield del mismo fabricante Arduino.

### 2.3.1. Elementos del Sistema de Control

Los controladores utilizados se observan en la Figura 2.15. El primero es un Arduino Mega, cuyas características más importantes se observan en la Tabla 2.9. El segundo es un Arduino Uno y sus características se observan en la Tabla 2.10.



Figura 2.15. Controladores utilizados. Izquierda: Arduino Mega. Derecha: Arduino Uno  
(Fuente: [18])

Tabla 2.9. Características del Arduino Mega 2560

Microcontrolador	ATmega2560
Voltaje de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (límite)	6-20V
Pines Digitales I/O	54 (15 salidas PWM)
Pines Análogos de Entrada	16
Corriente DC por Pin I/O	20 mA
Corriente DC para Pin de 3.3V	50 mA
Memoria Flash	256 KB (8 KB bootloader)
SRAM	8 KB
EEPROM	4 KB
Velocidad de Reloj	16 MHz
LED_BUILTIN	13
Longitud	101.52 mm
Ancho	53.3 mm
Peso	37 g

Fuente: [23]

Tabla 2.10. Características del Arduino Uno

Microcontrolador	ATmega328P
Voltaje de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V

Voltaje de entrada (límite)	6-20V
Pines Digitales I/O	14 (6 salidas PWM)
Pines PWM Digitales I/O	6
Pines Analógicos de Entrada	6
Corriente DC por Pin I/O	20 mA
Corriente DC para Pin de 3.3V	50 mA
Memoria Flash	32 KB
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Velocidad de Reloj	16 MHz
LED_BUILTIN	13
Longitud	68.6 mm
Ancho	53.4 mm
Peso	25 g

(Fuente: [23])

Las razones por las que se utilizan dos controladores son: en primer lugar, se tiene un amplio número de sensores que deben ser leídos por canales analógicos del controlador, en total son 18 sensores y las entradas totales que tiene el Arduino Mega son 16. Por otro lado, el exoesqueleto se ha dividido en 3 sistemas de control en lazo cerrado, el primero controla el motor de los dedos meñique y anular, el segundo para el control del motor del dedo medio e índice y el tercero para los motores del dedo pulgar. El utilizar 2 controladores permite que 2 lazos de control de los dedos trifalángicos sean manejados por el Arduino Mega y el tercer lazo del dedo pulgar, por el Arduino Uno.

Los motores que se utilizan para el movimiento del mecanismo son Faulhaber 1717T012s RL y se usan en conjunto con una caja de engranes planetarios Faulhaber 17/1 con reducción 494:1, la caja de engranes se encarga de reducir la velocidad del motor y de amplificar el torque de este. La velocidad sin carga del motor es de 14000 rev/min. Con la caja de engranes la velocidad disminuye a 28.34 rev/min, implica que una revolución la completa en 2.11 segundos, sin carga conectada.



Figura 2.16. Motor (izquierda) y Reductor (derecha) utilizados. La entrada del motor es un voltaje de alimentación de 0 a 12 V. La salida del motor es una salida de torque y velocidad angular, se conecta al reductor para disminuir la velocidad y aumentar el torque.

(Fuente: [21])

Los cálculos mecánicos de torque de los motores se realizan en los proyectos de titulación previos [5] [4]. En este caso es necesario considerar las características eléctricas de los motores. Primero el voltaje de alimentación del motor es de 12V, la corriente máxima del motor se calcula con el voltaje nominal y la resistencia en los terminales, de esta forma se calcula la corriente a rotor bloqueado, dichos valores se toman de la hoja de datos del motor.

$$I_{max} = \frac{V_{nom}}{R} = \frac{12V}{17.1\Omega} = 0.701 A$$

Los 4 motores del mecanismo con rotor bloqueado podrían consumir una corriente de hasta 2.8 A en el caso que todos se bloqueen al mismo tiempo. Por lo tanto, el resto de los elementos que se utilizan para el manejo de los motores se dimensionan con dicha corriente máxima.

El driver que se utiliza para el manejo de los motores es el Motor Shield de Arduino, Figura 2.17. Está basado en un puente H L298, permite el manejo de 2 motores DC. Por lo tanto, se utilizan dos Motor Shield, el primero será manejado desde el Arduino Mega y el segundo desde el Arduino Uno. Las características técnicas del Motor Shield se describen en la Tabla 2.11.

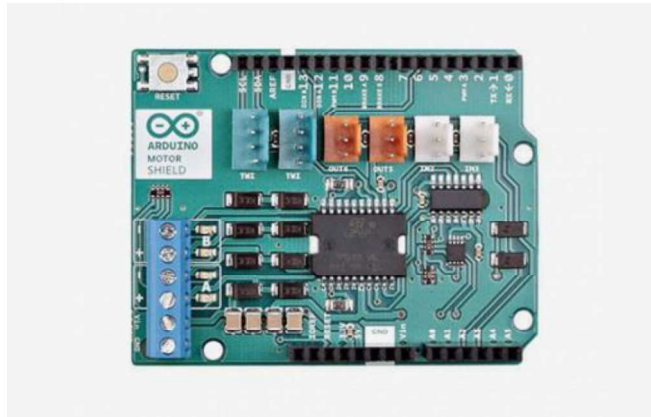


Figura 2.17. Motor Shield de Arduino L298  
(Fuente: [23])

Tabla 2.11. Características del Arduino Motor Shield

MOTOR SHIELD	
Voltaje de funcionamiento	5V to 12V
Controlador del motor	L298P, Acciona 2 motores de DC o 1 motor paso a paso
Corriente Máxima	2 <sup>a</sup> por canal o 4 <sup>a</sup> máx. (Con fuente de alimentación externa)
Corriente Sensada	1.65V/A
Función de parada y frenado de funcionamiento libre.	

(Fuente: [23])

Se observa que la corriente máxima por canal del Motor Shield es de 2 amperios, y los motores que se van a manejar tienen una corriente máxima de 0.701 A. Por lo tanto, no se va a exceder la corriente máxima del driver.

Como se observa en la Tabla 2.9 y en la Tabla 2.10, el voltaje de alimentación necesario en ambos casos es de 7 a 12V, por lo tanto se utiliza una fuente de 12V para su alimentación, en este caso se utiliza una fuente de alimentación comercial Mean Well de 12 V, 5A y 60 W. La fuente de alimentación se observa en la figura Figura 2.18. Hay que considerar que esta fuente se utiliza también para la alimentación de los drivers y los motores del exoesqueleto, los motores que son los que mayor potencia necesitan,

consumen una corriente máxima de 2.8 A, por lo que no van a exceder la corriente máxima de la fuente.



Figura 2.18. Fuente de Alimentación de los Controladores. Entrada de 120V de corriente alterna. Salida de 12V de corriente continua, 5A.

(Fuente: [24])

Para la comunicación entre los controladores Arduino Mega y Arduino Uno se utiliza comunicación serial directa entre ambos controladores. Para la comunicación con la interfaz de mando se utiliza comunicación serial a través de un módulo bluetooth HC – 05. El módulo bluetooth está conectado al Arduino Mega, por lo tanto, se utiliza como una configuración Master – Slave, en la que el Arduino Mega es el Master, la Pc y el Arduino Uno son los esclavos. De esta forma se comparte información de mando entre ambos controladores y permite la sincronización de los movimientos de rehabilitación.



Figura 2.19. Módulo Bluetooth HC-05. Funciona como emisor y receptor. Se comunica directamente con el Bluetooth de la PC

(Fuente: [23])

Los elementos antes mencionados se conectan entre sí como se observa en el diagrama esquemático de la Figura 2.20.

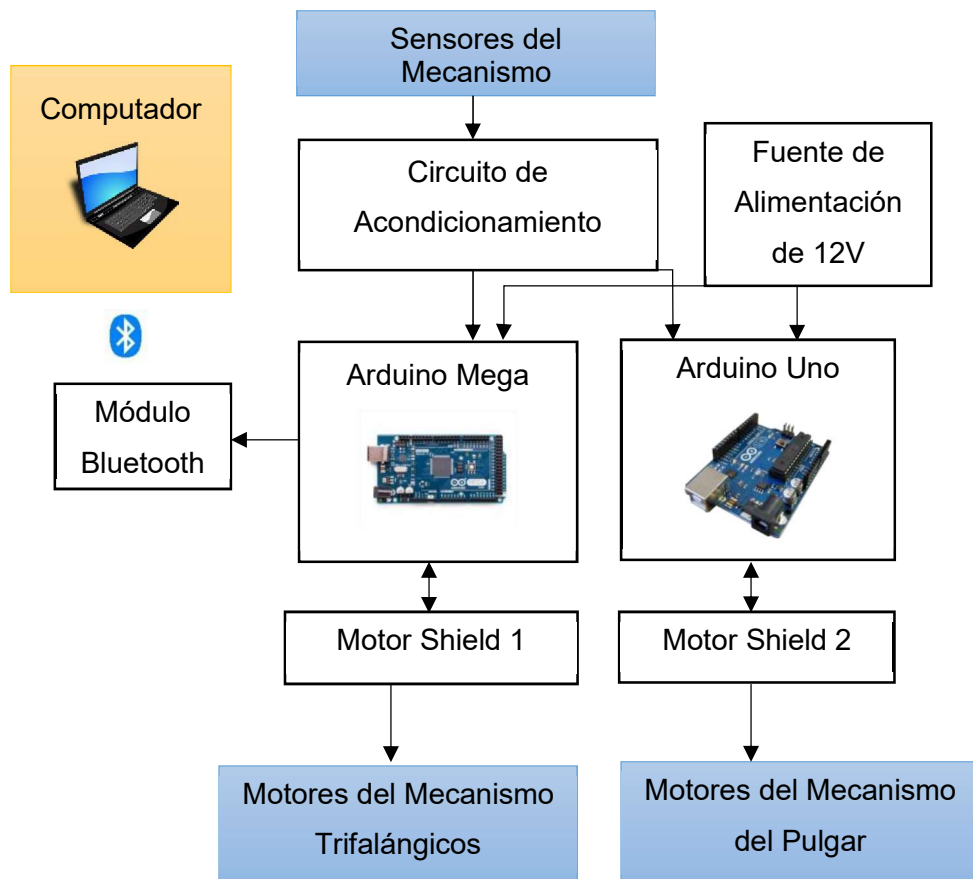


Figura 2.20. Elementos del sistema de control en conexión con el resto de los elementos del sistema.

(Fuente: propia)

### 2.3.2. Programa de Control

Para el programa de control se define lo siguiente: las dos rutinas de rehabilitación que se implementan son los movimientos de flexión – extensión y el movimiento de pinza. El movimiento de flexión – extensión hace que los dedos se muevan desde la posición de máxima extensión a la posición de máxima flexión. Mientras que el movimiento de pinza hace que la punta del dedo pulgar se encuentre con la punta de los dedos índice y medio, y en otro movimiento se encuentre con la punta de los dedos anular y meñique.

El programa de control permite escoger el tipo de movimiento de rehabilitación y el número de repeticiones del movimiento. Los mandos se los realiza desde la interfaz de mando. Además, se ha incluido pulsantes en la placa de acondicionamiento para controlar manualmente el movimiento de los motores.

### 2.3.3. Lazo Cerrado de Control

Como se ha descrito anteriormente, el primer lazo cerrado de control se utiliza para el movimiento de los dedos meñique y anular, es el mismo lazo de control para los dedos medio e índice. El diagrama de bloques del lazo cerrado de control se observa en la Figura 2.21.

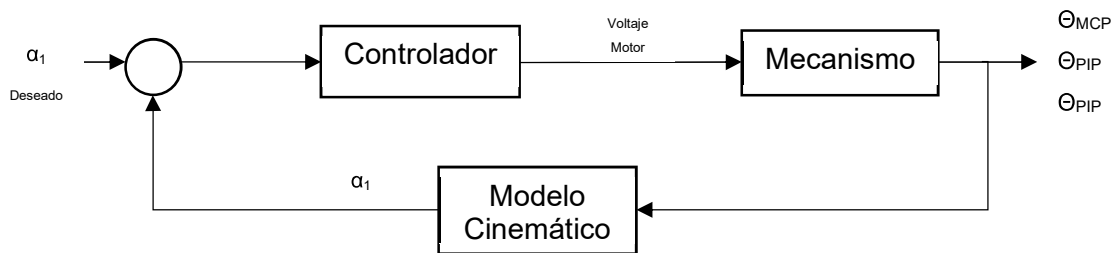


Figura 2.21. Lazo Cerrado de Control Dedos Trifalángicos  
(Fuente: propia)

En el lazo de control se observa que la medición de los sensores se hace en los ángulos de los dedos  $\theta_{MCP}$ ,  $\theta_{PIP}$  y  $\theta_{DIP}$ , luego se utiliza los modelos cinemáticos para determinar la posición del motor  $\alpha_1$ . El valor deseado es el valor de la posición angular del motor  $\alpha_1$  deseado, y el controlador determina el movimiento del motor dependiendo del error que se está midiendo en cada momento. Se define como un control de posición. Como se deben establecer varias repeticiones de las rutinas de rehabilitación, ejemplo 10 repeticiones del movimiento de flexión - extensión, se escoge la cantidad de movimientos que se desean realizar y se cambia el valor deseado cada vez que el mecanismo ha alcanzado la posición deseada.

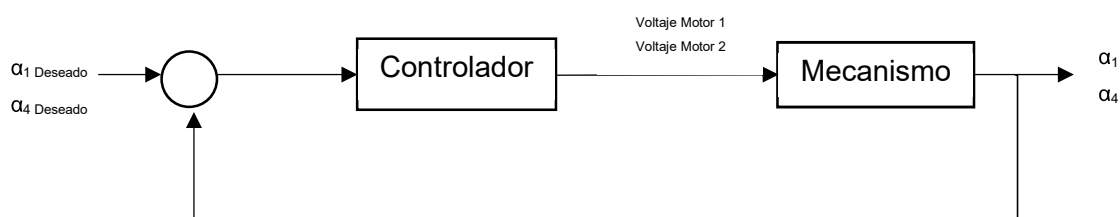


Figura 2.22. Lazo Cerrado de Control Dedo Pulgar  
(Fuente: propia)

El segundo lazo de control Figura 2.22, se utiliza para el dedo pulgar, en este caso los sensores no miden la posición de las articulaciones del dedo, sino que miden las posiciones angulares de los motores  $\alpha_1$  y  $\alpha_4$  y de esta forma con el modelo cinemático se obtiene la posición del dedo. Sin embargo, el movimiento de los eslabones está



definido para las posiciones de los motores por lo tanto no se utiliza el modelo cinemático dentro del lazo de control.

## 2.4. Diseño de la Interfaz de Mando

La interfaz de mando es la encargada de enviar las instrucciones de mando al mecanismo. Como se describió en el Diseño del Sistema de Control, existen dos tipos de movimientos que se realizan con el exoesqueleto, el de flexión – extensión y el de pinza. Para cada uno de estos movimientos se puede escoger el número de repeticiones que debe realizarse cada rutina de rehabilitación. Por lo tanto, la interfaz de mando debe permitir escoger el tipo de movimiento y el número de repeticiones que se realizarán.

La interfaz de mando se realiza en el software MATLAB, el cual permite generar la comunicación serial a través del módulo bluetooth. Es requisito que la computadora o el dispositivo de mando tenga acceso a este tipo de comunicación. En la Figura 2.23, se observa la imagen de la interfaz gráfica que permite el mando sobre el exoesqueleto.



Figura 2.23. Interfaz de Mando del Exoesqueleto desarrollada en MATLAB

(Fuente: propia)

El primer botón Conectar, permite la conexión del puerto serial de la PC, y el botón Desconectar realiza la desconexión de la comunicación, es necesario siempre que se termina la conexión desconectar, ya que el canal puede quedarse encendido, sin permitir que se vuelva a conectar.

El tipo de movimiento de rehabilitación se escoge a través de un menú, que contiene los dos tipos, flexión – extensión y pinza. El número de repeticiones se ingresa a través de un texto editable.

Luego de comprobar el funcionamiento de la interfaz de mando, empieza a generar problemas de conexión a través de bluetooth. Es importante considerar que la conexión a través del módulo bluetooth es una comunicación serial, que utiliza el módulo HC-05 para enviar los mensajes seriales al receptor bluetooth de la computadora. Por lo tanto, el controlador y la interfaz deben generar una comunicación serial. El problema específico que se genera en la interfaz creada en MATLAB es que al abrir el puerto de comunicación hay veces que se queda abierto, y a pesar de que se cierra posteriormente, no se puede realizar la conexión nuevamente, ya que aparecen mensajes de error en la comunicación.

Debido a que la interfaz no es muy estable en MATLAB, se procede a buscar otras opciones para la creación de la interfaz de mando. Se utiliza el software Labview para generar una interfaz de mando que funcione de manera más estable. Prácticamente la función de dicha interfaz es la misma que la creada en MATLAB ya que establece una comunicación serial y envía los mensajes necesarios para que se de las órdenes necesarias al controlador para que permita su funcionamiento. La interfaz creada en Labview se presenta en la Figura 2.24. De igual forma permite escoger el número de repeticiones que se desea del ejercicio de rehabilitación, el tipo de movimiento que se desea generar. Tiene un botón para iniciar la rutina de rehabilitación, es importante recalcar, que solo puede enviar la orden mientras el exoesqueleto no se encuentra en funcionamiento.

La interfaz además incluye una lista para escoger el puerto por la cual se va a realizar la comunicación, un indicador de que la conexión por el puerto se ha logrado, un interruptor que permite la lectura de datos que provienen del controlador y un botón de STOP que cierra la comunicación

Para el envío de mensajes por el puerto serial fue necesario generar una lista de códigos que se envían por comunicación serial para el mando del exoesqueleto, pero además se incluye una comunicación serial entre los dos controladores. Esto es posible debido a que el controlador Arduino Mega tiene 3 canales de comunicación serial. Por lo tanto, el canal 1 se utiliza para la comunicación con la interfaz de mando, mientras que el canal 2 se utiliza para la comunicación con el otro controlador.



Figura 2.24. Interfaz de mando creada en Labview  
(Fuente: propia)

La codificación de los mensajes que se envían se la realiza con cada byte enviado. En general se envían 3 bytes, el primero sirve para determinar si el mensaje que se está enviando es una orden o un mensaje genérico. Si el valor es de uno, implica que el mensaje es una orden. El segundo byte determina el tipo de movimiento que se realiza, si es de flexión – extensión o si es de pinza. El tercer byte envía el número de repeticiones del movimiento de rehabilitación.

Para la comunicación entre controladores se utiliza una trama similar, sin embargo, los mensajes que se envían entre controladores sirven para dar la orden de inicio de los movimientos de rehabilitación y para sincronizar los movimientos. La sincronización de los movimientos es necesaria debido a que los controladores únicamente conocen la posición del lazo de control que manejan. Con esta sincronización se logra que los movimientos de flexión o extensión empiecen al mismo tiempo. El resumen de los códigos utilizados para el mando desde la interfaz y para la comunicación entre controladores se presenta en la Tabla 2.12.

Tabla 2.12 Códigos del sistema de comunicación

Byte 1	Byte 2	Byte 3	Descripción	Origen	Destino
--------	--------	--------	-------------	--------	---------

1	1	#	Orden de inicio de secuencia de flexión extensión	PC o Arduino Mega	Arduino Mega o Arduino Uno
1	2	#	Orden de inicio de secuencia de pinza	PC o Arduino Mega	Arduino Mega o Arduino Uno
1	3	1	Movimiento de flexión completo	Arduino Mega o Arduino Uno	Arduino Mega o Arduino Uno
1	3	2	Cambio de movimiento de flexión a extensión o viceversa	Arduino Mega o Arduino Uno	Arduino Mega o Arduino Uno

## 2.5. Implementación del Sistema de Rehabilitación en Conjunto

Para la mejor comprensión del Sistema de Rehabilitación, se ha dividido al sistema en 3 partes importantes. La primera es el mecanismo del exoesqueleto, la segunda son los circuitos electrónicos necesarios para el funcionamiento y finalmente la interfaz de mando del sistema. Como se observa en la Figura 2.25.

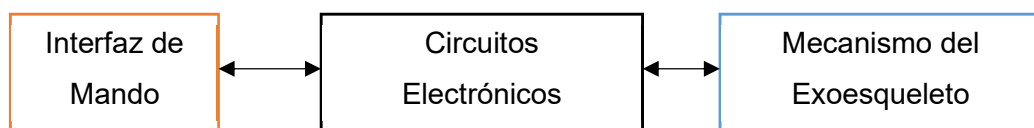


Figura 2.25. Sistema de Rehabilitación en Conjunto  
(Fuente: propia)

En la Figura 2.26, se observa el mecanismo del exoesqueleto. En el mecanismo están todos los elementos mecánicos que conforman el exoesqueleto, los motores que generan los movimientos del mecanismo y los sensores que permiten conocer el estado del mecanismo. Se observa que son similares los mecanismos para los dedos trifalángicos, mientras que para el dedo pulgar tiene diferencias importantes.

Todos los elementos que se encuentran en el mecanismo del exoesqueleto se detallan en la Tabla 2.13, considerando que el exoesqueleto se utiliza en la mano izquierda. La comunicación desde el exoesqueleto a los circuitos electrónicos se hace a través de cables STP para las señales de los sensores y cables multihilos para los motores.

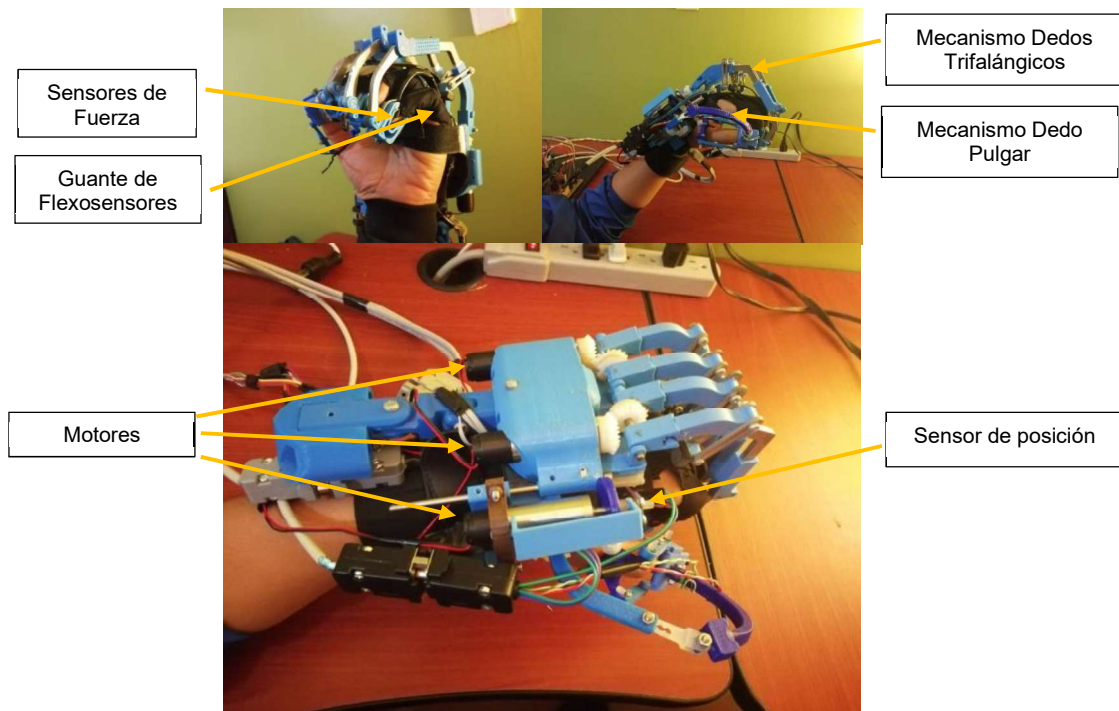


Figura 2.26. Mecanismo del Exoesqueleto utilizado en la mano izquierda. Imagen de izquierda arriba vista inferior. Imagen de derecha arriba vista lateral. Imagen de abajo vista superior  
(Fuente: propia)

Tabla 2.13. Elementos del mecanismo del exoesqueleto

Elemento	Ubicación	Descripción
Mecanismo Dedo Trifalángico 1	Dedo meñique	Conjunto de eslabones que transfieren el movimiento para el dedo meñique
Mecanismo Dedo Trifalángico 2	Dedo anular	Conjunto de eslabones que transfieren el movimiento para el dedo anular
Mecanismo Dedo Trifalángico 3	Dedo medio	Conjunto de eslabones que transfieren el movimiento para el dedo medio
Mecanismo Dedo Trifalángico 4	Dedo índice	Conjunto de eslabones que transfieren el movimiento para el dedo índice
Mecanismo Dedo Pulgar	Dedo pulgar	Conjunto de eslabones que transfieren el movimiento para el dedo pulgar
Motor 1	Dedo meñique y anular	Genera el movimiento en conjunto para los mecanismos del dedo meñique y anular
Motor 2	Dedo medio e índice	Genera el movimiento en conjunto para los mecanismos del dedo medio e índice
Motor 3	Dedo pulgar	Genera el movimiento de flexión y extensión del dedo pulgar
Motor 4	Dedo pulgar	Genera el movimiento abducción a aducción del dedo pulgar
Sensor de Fuerza 1	Dedo meñique	Medición de fuerza de flexión del dedo meñique
Sensor de Fuerza 2	Dedo meñique	Medición de fuerza de extensión del dedo meñique
Sensor de Fuerza 3	Dedo anular	Medición de fuerza de flexión del dedo anular
Sensor de Fuerza 4	Dedo anular	Medición de fuerza de extensión del dedo anular
Sensor de Fuerza 5	Dedo medio	Medición de fuerza de flexión del dedo medio
Sensor de Fuerza 6	Dedo medio	Medición de fuerza de extensión del dedo medio
Sensor de Fuerza 7	Dedo índice	Medición de fuerza de flexión del dedo índice
Sensor de Fuerza 8	Dedo índice	Medición de fuerza de extensión del dedo índice
Sensor de Fuerza 9	Dedo pulgar	Medición de fuerza de flexión del dedo pulgar

Sensor de Fuerza 10	Dedo pulgar	Medición de fuerza de extensión del dedo pulgar
Flexosensor 1	Dedo meñique	Medición del ángulo articulación MCP
Flexosensor 2	Dedo anular	Medición del ángulo articulación PIP
Flexosensor 3	Dedo medio	Medición del ángulo articulación MCP
Flexosensor 4	Dedo índice	Medición del ángulo articulación PIP
Flexosensor 5	Dedo meñique	Medición del ángulo articulación DIP
Flexosensor 6	Dedo medio	Medición del ángulo articulación DIP
Sensor de posición 1	Dedo pulgar	Medición del ángulo del motor 1 dedo pulgar
Sensor de posición 2	Dedo pulgar	Medición del ángulo del motor 2 dedo pulgar

Los elementos que se encuentran dentro de los circuitos electrónicos están instalados sobre una tabla para su mejor disposición. Dichos circuitos se observan en la Figura 2.27.

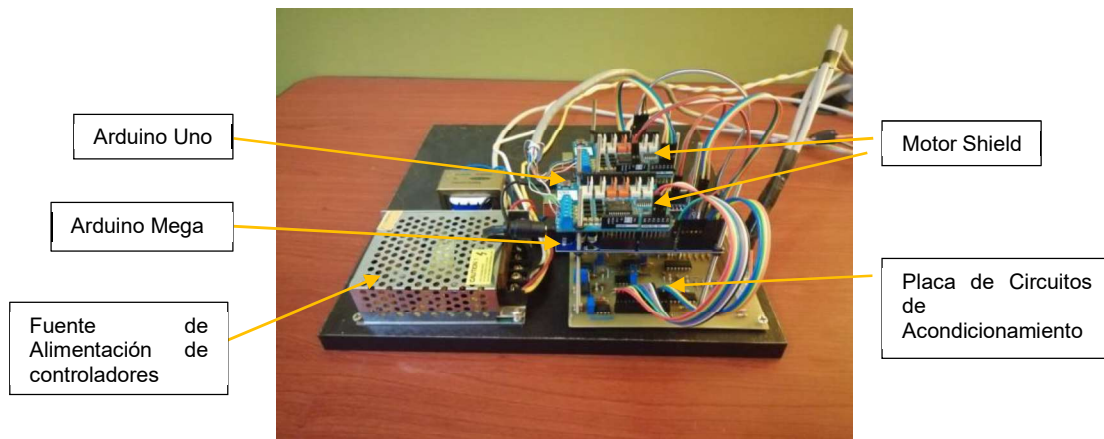


Figura 2.27. Circuitos Electrónicos. Controladores, fuente de alimentación, placa de circuitos de acondicionamiento, drivers de los motores  
(Fuente: propia)

Los elementos que se encuentran dentro de los circuitos electrónicos se observan en la Tabla 2.14. Finalmente tienen que conectarse con el último elemento que es la interfaz de mando que se corre en la computadora. Dicha conexión es de forma inalámbrica a través del módulo Bluetooth. El último elemento es la computadora en la que corre la interfaz de mando.

Tabla 2.14. Elementos de los Circuitos Electrónicos

Elemento	Descripción
Arduino Mega	Control del Mecanismo de dedos trifalángicos, maestro en la comunicación entre controladores y PC
Arduino Uno	Control del Mecanismo del dedo pulgar, esclavo en la comunicación entre controladores y PC
Motor Shield 1	Control de motores de dedos trifalángicos

Motor Shield 2	Control de motores dedo pulgar
Fuente de Alimentación	Alimentación de Controladores
Módulo Bluetooth	Comunicación entre Arduino Mega y la PC
Placa de Circuitos Electrónicos	Acondicionamiento de las señales de los sensores, pulsantes de mando manual, fuentes de alimentación de circuitos de acondicionamiento, módulo bluetooth

## 3. RESULTADOS Y DISCUSIÓN

### 3.1. Resultados

#### 3.1.1. Resultados del Estudio Cinemático

El estudio cinemático se divide en dos partes, el primero abarca el análisis del movimiento de los dedos de la mano, el segundo abarca el análisis del movimiento del mecanismo. Al revisar los dos modelos cinemáticos en conjunto se determina el rango de trabajo del mecanismo en conjunto con la mano. El resultado de todos estos análisis son las simulaciones realizadas en MATLAB cuyos programas se presentan como anexos.

El primer modelo obtenido es de los dedos trifalángicos, del cual se obtiene el modelo cinemático directo. Se utilizan los ángulos de las articulaciones de los dedos para determinar la posición de la punta del dedo. Esta simulación se realiza en el módulo Robot de MATLAB, sin embargo, la simulación es una comprobación del modelo presentado en [11], con la diferencia que se elimina la articulación CMC, ya que, en el diseño del exoesqueleto presentado, no influye el movimiento de dicha articulación. También, se obtienen las matrices de rotación con las que se obtiene el modelo cinemático directo de los dedos, estos programas se presentan como anexos junto con las matrices resultantes del análisis.

Los parámetros Denavit Hartenberg correspondientes a los dedos trifalángicos se presentan en la Tabla 1.2, se les ingresa en módulo robot de MATLAB para la simulación. Los valores utilizados se muestran en la Tabla 3.1. donde  $L_{22}$ ,  $L_{32}$ , y  $L_{42}$  son las longitudes de las falanges proximal, media y distal correspondientemente. Los valores  $\theta_{MCP\ ab/ad}$ ,  $\theta_{MCP\ f/e}$ ,  $\theta_{PIP\ f/e}$  y  $\theta_{DIP\ f/e}$  son las entradas del modelo y se simula para 4 posiciones distintas de dichos ángulos. Además, se presentan los resultados de la posición de la punta del dedo, en cada coordenada x y z en cada una de las 4 posiciones. En la Figura 3.1, se muestra la simulación para distintos valores de los ángulos de las articulaciones. Las longitudes se toman de [11], en dicho artículo se muestra valores promedios de cada hueso, en este caso es de manos izquierdas de varones. Los valores de los ángulos se consideran los datos de máxima flexión y extensión y dos valores intermedios.

Tabla 3.1. Valores considerados y obtenidos en la simulación del dedo índice



<b>Parámetro</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
L <sub>22</sub> (cm)	2.65	2.65	2.65	2.65
L <sub>32</sub> (cm)	2.37	2.37	2.37	2.37
L <sub>42</sub> (cm)	2.32	2.32	2.32	2.32
$\theta_{MCP\ ab/ad}$ (grados)	0	0	0	0
$\theta_{MCP\ f/e}$ (grados)	30	-10	-50	-90
$\theta_{PIP\ f/e}$ (grados)	0	-35	-75	-110
$\theta_{DIP\ f/e}$ (grados)	5	-25	-50	-80
Posición de la punta en x (cm)	6.248	5.079	-1.967	-1.824
Posición de la punta en y (cm)	0	0	0	0
Posición de la punta en z (cm)	3.841	-4.316	-4.174	0.445

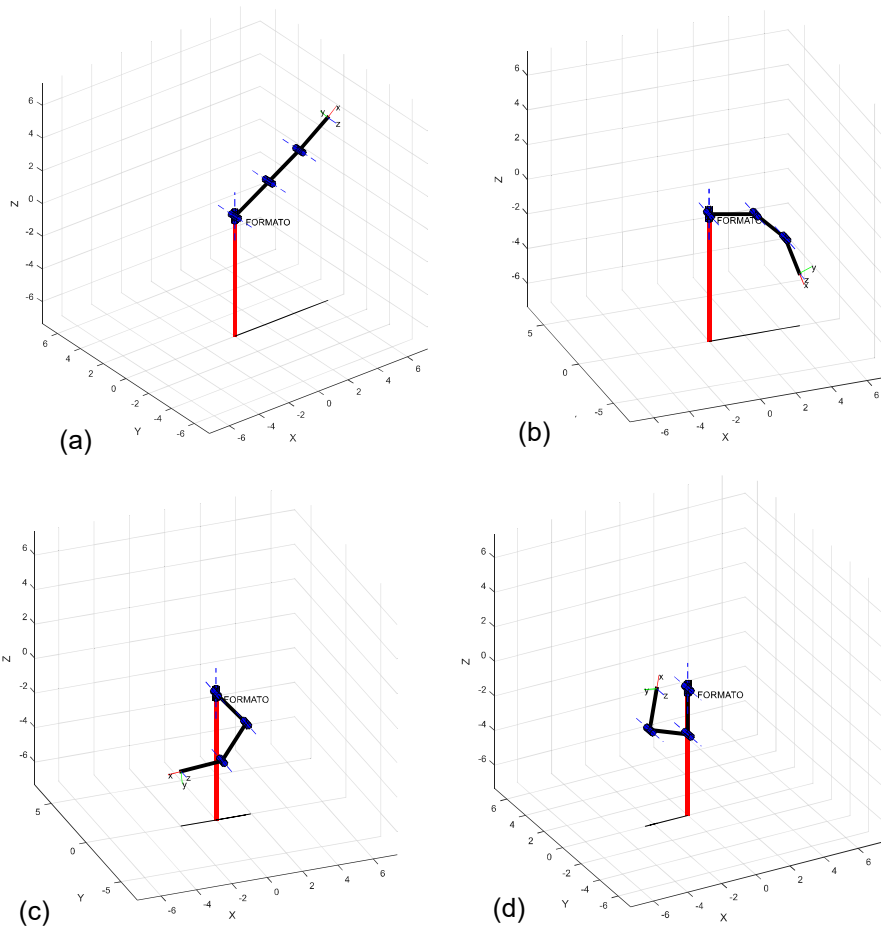


Figura 3.1. Simulación del dedo índice con el módulo Robot de Matlab. Junto al punto cero la falange proximal, a continuación, la falange media y luego la falange distal. (a) Posición 1 de la tabla Tabla 3.1. (b) Posición 2. (c) Posición 3. (d) Posición 4.

(Fuente: propia)

Además, se determinan las matrices de rotación con los parámetros DH que se muestran en la Tabla 1.3. Se utiliza el software MATLAB para encontrar las matrices.

Se utiliza las matrices de rotación para determinar la posición del dedo, en los mismos ángulos utilizados con el módulo robot. Los resultados se muestran en la Figura 3.2. Se comprueba que las posiciones obtenidas con el módulo robot de MATLAB y las obtenidas con las matrices de rotación son las mismas.

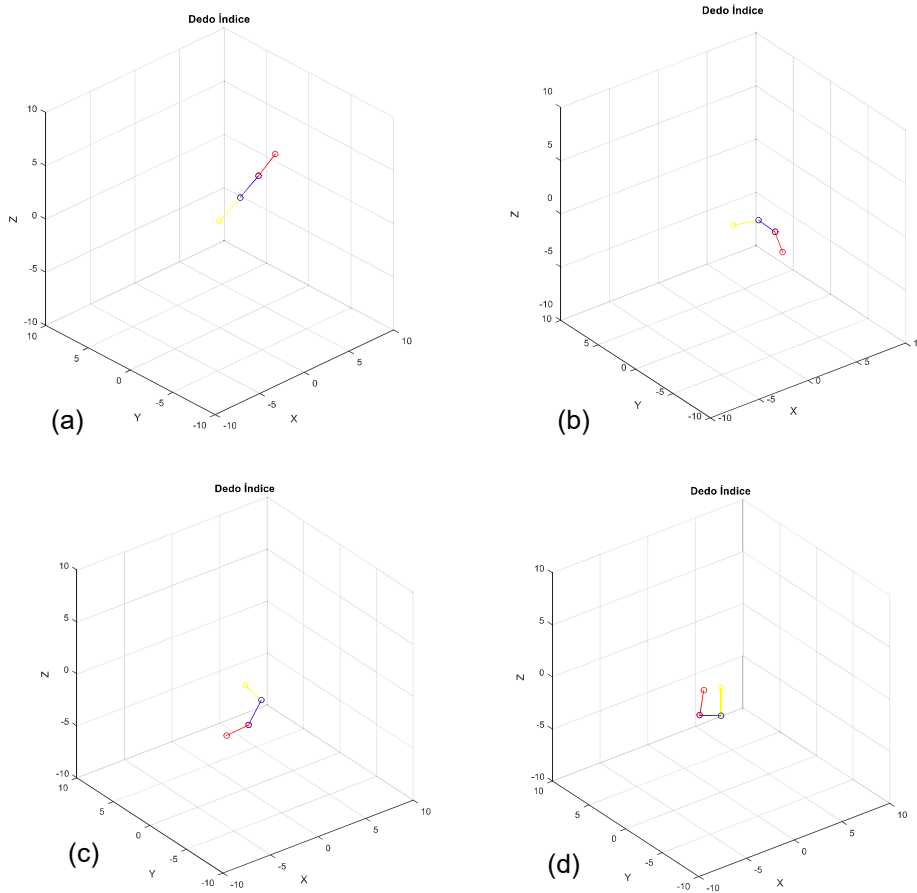


Figura 3.2. Simulación del dedo índice utilizando las matrices de rotación. En amarillo la falange proximal, en azul la falange media y en rojo la falange distal. (a) Posición 1 de la Tabla 3.1. (b) Posición 2. (c) Posición 3. (d) Posición 4

(Fuente: propia)

También se obtienen los modelos cinemáticos directos de los mecanismos por separado, uno para los dedos trifalángicos y otro para el dedo pulgar. De igual forma es un modelo cinemático directo ya que permite conocer la posición de la punta del mecanismo, en el punto donde se une con el dedo, en función de los ángulos de las articulaciones que lo conforman. Los parámetros Denavit – Hartenberg se presentan en el capítulo 2 del escrito, Tabla 2.2, y además se presentan los programas utilizados como anexos junto con las matrices resultantes del análisis.

La simulación en el módulo robot de MATLAB de los dedos trifalángicos se presentan en la Figura 3.3. Además, se calculan las matrices de rotación con las que se utiliza para comprobar la simulación realizada, el gráfico obtenido también se presenta en la Figura 3.3.

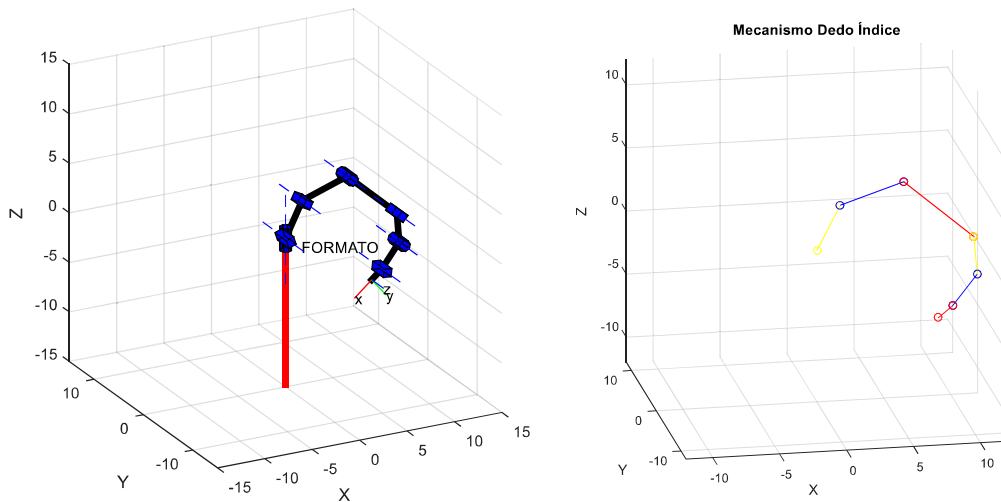


Figura 3.3. Simulación del mecanismo para dedos trifalángicos. Izquierda: modelo simulado en el toolbox robot de MATLAB. Derecha: Modelo obtenido con las matrices de rotación.  
(Fuente: propia)

De igual forma con los parámetros Denavit – Hartember obtenidos para el mecanismo del dedo pulgar se hace la simulación utilizando el módulo robot de MATLAB. También se encuentran las matrices de rotación y se presenta los resultados de ambas simulaciones en la Figura 3.4.

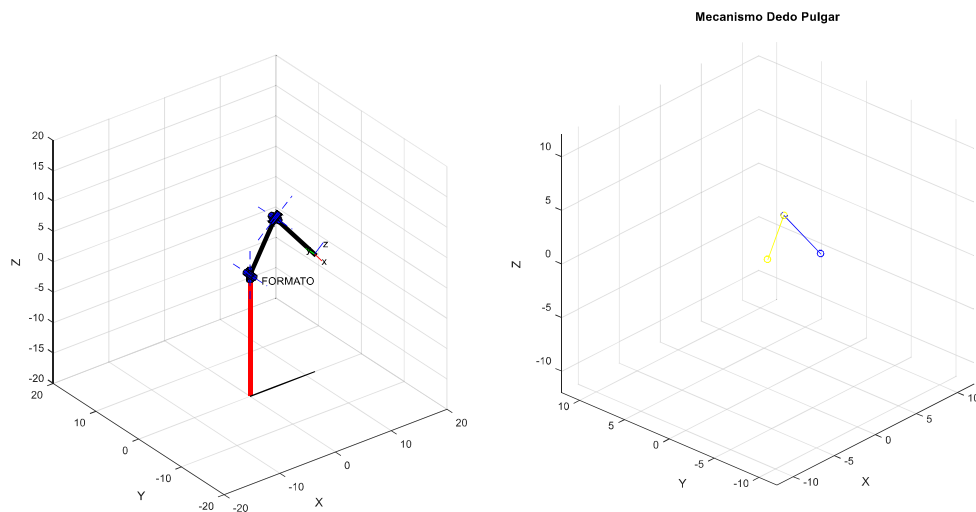


Figura 3.4. Simulación del mecanismo para el dedo pulgar. Izquierda: simulación en el módulo Robot de MATLAB. Derecha: Modelo obtenido con matrices de rotación.

Los datos observados se utilizan en el modelo cinemático de los dedos y del mecanismo para encontrar el espacio de trabajo de la mano y ponerle en conjunto con el espacio de trabajo del mecanismo. De esta forma se puede determinar si el mecanismo funciona para los distintos tamaños de mano para los que fue diseñado.

Los espacios de trabajo para los dedos trifalángicos se obtienen considerando tamaños de mano máximos y mínimos. El primer tamaño de mano utilizado es con las longitudes mínimas de las falanges (Tabla 2.3), en este caso se utiliza el mecanismo con los ajustes de los segmentos CD y DE al mínimo (Tabla 2.1). El siguiente espacio de trabajo se utiliza con las longitudes máximas de las falanges de la mano y en el mecanismo se utilizan los ajustes de los segmentos CD y DE al máximo. Los espacios de trabajo se observan en la Figura 3.5, la primera imagen con los espacios de trabajo de las medidas mínimas, en la segunda imagen con las medidas máximas.

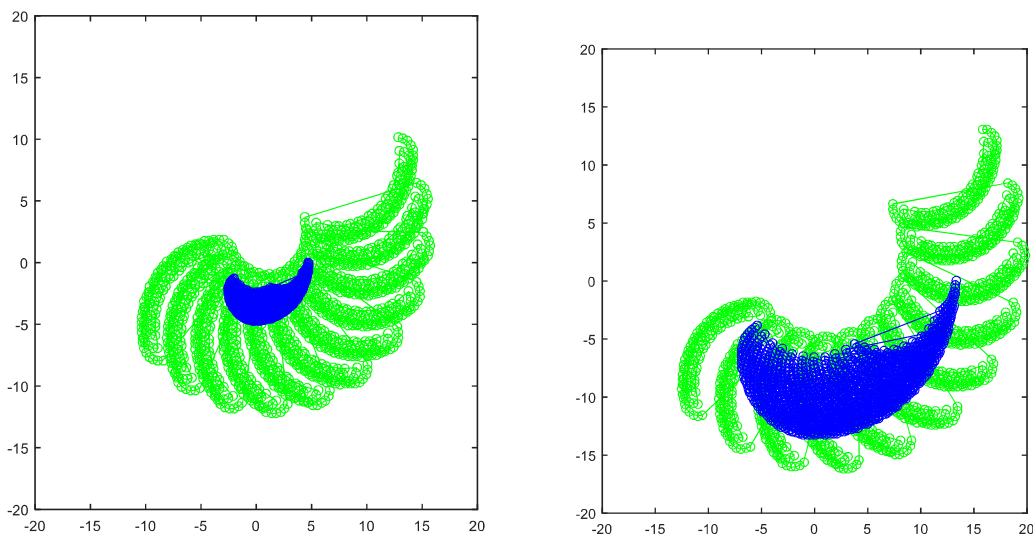


Figura 3.5. Espacio de trabajo para el dedo índice, en azul el espacio de trabajo de la mano y en verde el espacio de trabajo del mecanismo. Izquierda: Medidas mínimas. Derecha: Medidas máximas.

(Fuente: propia)

Algo similar se hace con el dedo pulgar. Se utiliza los modelos cinemáticos de la mano y del mecanismo para superponerlos y comprobar si se puede utilizar para los valores diseñados. En la Figura 3.6, la primera imagen muestra los espacios de trabajo considerando las medidas mínimas de las falanges del pulgar, mientras que la segunda

imagen muestra el espacio de trabajo considerando las medidas máximas de las falanges del pulgar.

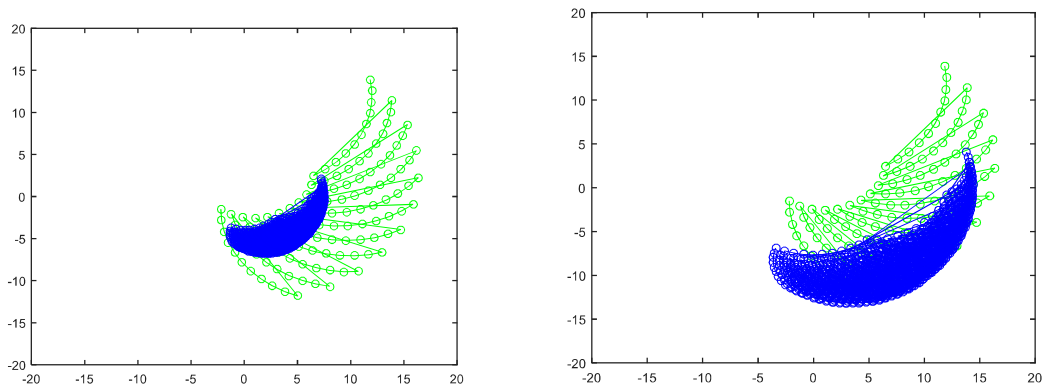


Figura 3.6. Espacio de trabajo para el dedo pulgar, en azul el espacio de trabajo de la mano y en verde el espacio de trabajo del mecanismo. Izquierda: Medidas mínimas. Derecha: Medidas máximas

(Fuente: propia)

Los modelos cinemáticos obtenidos se utilizan para determinar la trayectoria que debe seguir el mecanismo en conjunto con el dedo en los movimientos de rehabilitación. Para el movimiento de flexión – extensión se utiliza al mecanismo de forma manual para ubicarlo en posición de máxima flexión y se aproxima los ángulos de las articulaciones de los dedos que ha alcanzado. Dichos ángulos se presentan en la Tabla 3.2. Y la simulación de la posición del dedo en conjunto con el mecanismo se presenta en la Figura 3.7.

Tabla 3.2. Ángulos obtenidos en máxima flexión para dedos trifalángicos

<b>Dedos Trifalángicos</b>	
MCP	-72°
PIP	-90°
DIP	-50°

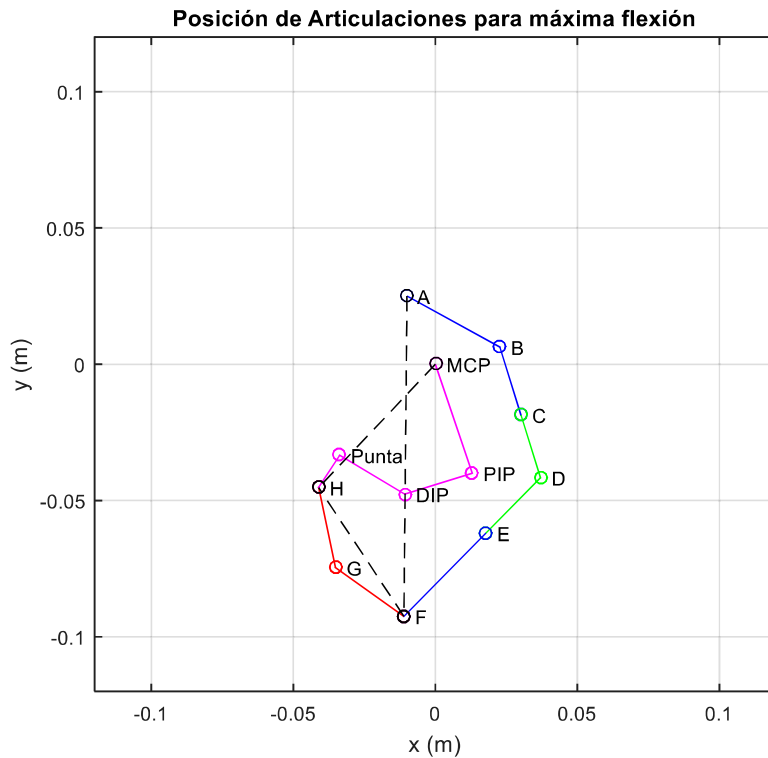


Figura 3.7. Posición del mecanismo y dedo para máxima flexión  
(Fuente: propia)

Estos ángulos se utilizan para determinar la trayectoria de cada falange de los dedos en un movimiento de flexión. Para esto se simula varias posiciones de ángulos de los dedos empezando por el punto de máxima flexión, considerando las condiciones de diseño en las que el punto de  $0^\circ$  en cada una de las articulaciones es la máxima extensión. Además, se encuentra la cinemática inversa del mecanismo, ya que ahora es necesario con la posición de la punta del dedo determinar los ángulos en los cuales se debe ubicar el mecanismo. Con esta simulación se determina una trayectoria esperada para el movimiento de flexión. Dicha trayectoria se presenta en la Figura 3.8. Para la trayectoria mencionada, se determina los ángulos de los dedos y los ángulos de las articulaciones del mecanismo. Dichos valores se presentan en la Tabla 3.3.

Tabla 3.3. Ángulos simulados para una trayectoria de flexión

	MCP	PIP	DIP	alfa1	alfa2	alfa3
Extensión 1	0	0	0	83,5	-52,2	25,7
2	-7,2	-9	-5	71,51	-45,4	9,7
3	-14,4	-18	-10	59,5	-40,4	-4,4

4	-21,6	-27	-15	47,97	-37,64	-16,85
5	-28,8	-36	-20	36,7	-37,1	-27,3
6	-36	-45	-25	26,06	-39,06	35,92
7	-43,2	-54	-30	15,8	-43,3	-42,4
8	-50,4	-63	-35	5,51	-49,78	-47,05
9	-57,6	-72	-40	-4,8	-58,2	-49,3
10	-64,8	-81	-45	-16,06	-68,87	-48,78
Flexión	-72	-90	-50	-29,5	-82,1	-43,4

Se utiliza estas trayectorias para el control de posición del mecanismo, sin embargo, debido a que la capacidad de cálculo de los controladores no es muy amplia, se decide utilizar tablas, y de esta forma obtener el valor de posición del motor dependiendo de la posición de las articulaciones de los dedos.

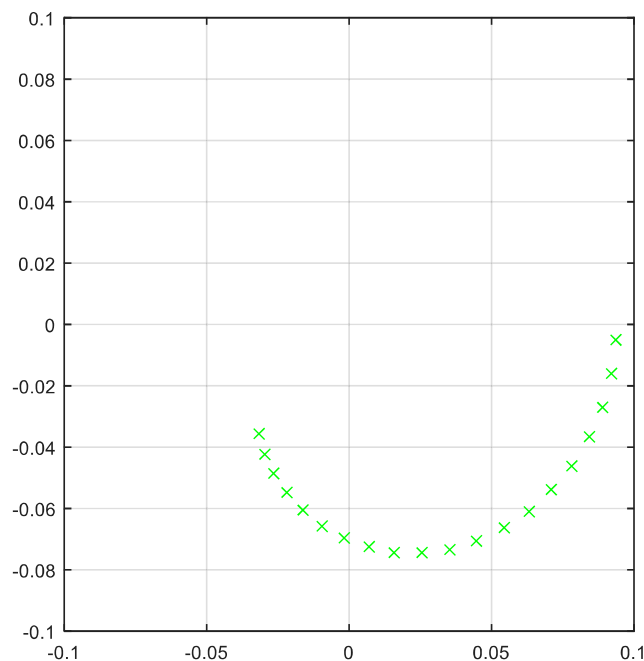


Figura 3.8. Trayectoria de un Movimiento de Flexión  
(Fuente: propia)

### 3.1.2. Resultados del Funcionamiento del Exoesqueleto

Para comprobar el funcionamiento del exoesqueleto se determina un procedimiento de prueba. El cual empieza comprobando la parte de instrumentación, luego se continúa

comprobando el sistema de control y finalmente se comprueba el funcionamiento de la interfaz gráfica.

En la parte de instrumentación es necesario considerar que se utilizan flexosensores para determinar la posición de los dedos trifalángicos. Para poder leer la señal de los flexosensores en el controlador se utiliza un circuito de acondicionamiento. Y finalmente en el controlador se utiliza estas señales para generar las acciones de control.

Para comprobar las señales de los sensores se hace lo siguiente, con todo el sistema en funcionamiento se utiliza un programa de control que únicamente permite el movimiento de los motores a través de los pulsantes, es decir un control manual. Con este control manual se ubica al mecanismo en posición de máxima extensión. En este punto el controlador lee las señales de los flexosensores y les envía directamente a la PC. En la PC se utiliza el software Realterm, terminal de lectura de datos seriales en la computadora, para observar estos datos de los sensores. En la Figura 3.9, se observa un ejemplo de los datos leídos por los sensores. Por columnas se observan los datos de los sensores en el siguiente orden: DIP1, DIP2, PIP1, MCP1, PIP2, MCP2. Es necesario calibrar las señales que ingresan de los sensores, para que el rango de funcionamiento de aproximadamente de 0 a 5V, aun así, hay señales que no se logran calibrar en todo el rango de funcionamiento.



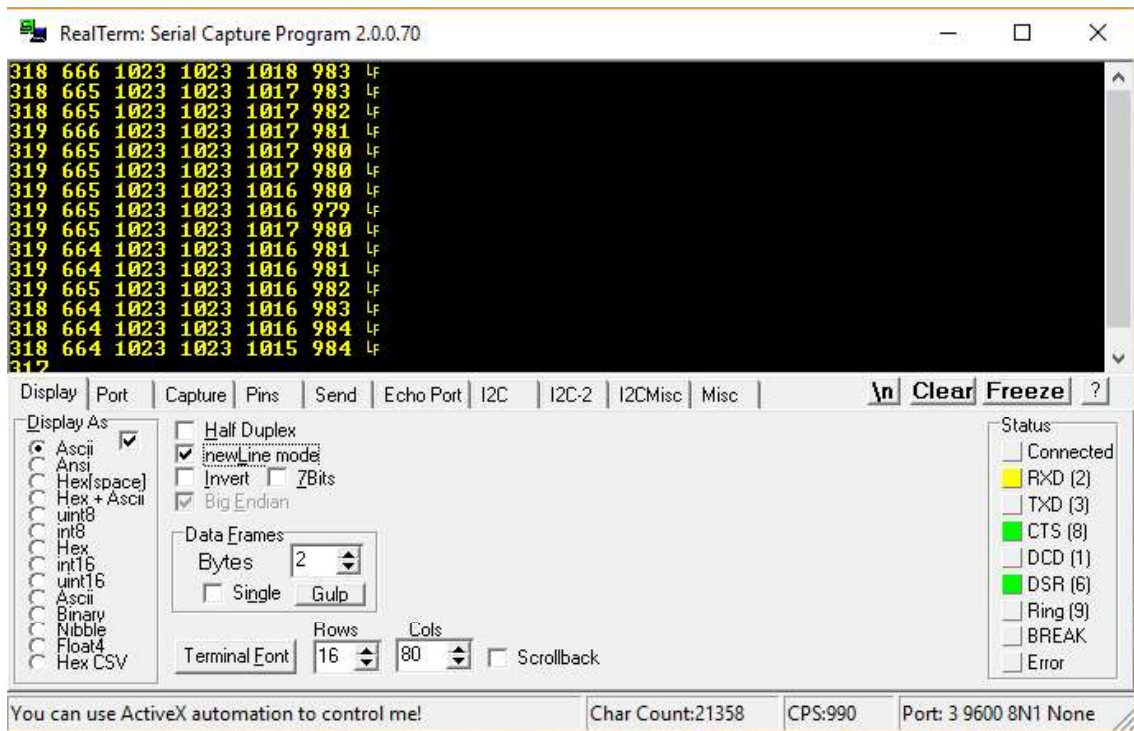


Figura 3.9. Datos leídos de los sensores desde el terminal Realterm. Se observa las mediciones de las articulaciones DIP1, DIP2, PIP2, MCP2, PIP1, MCP1 en el orden mostrado.

(Fuente: propia)

Ya con las señales calibradas se genera los movimientos de los motores para ir llevando el mecanismo paso a paso al punto de máxima flexión, hay que tomar en cuenta que las señales que no se calibran en todo el rango de funcionamiento tienen una limitación en los rangos de utilización, es decir no se usa desde los 0 a los 5V sino desde el valor más bajo en máxima flexión hasta el valor más alto en máxima extensión, en cada sensor se tienen valores distintos. Se va tomando los valores leídos de los sensores en cada paso. En este punto hay que considerar que el convertidor analógico digital del controlador es de 10 bits, por lo tanto, la señal que ingresa de 0 a 5V, se leerá en el controlador en un rango de 0 a 1023. Se tabulan estos datos para cada sensor y con esos valores se busca una ecuación que permita establecer el ángulo que está midiendo el sensor en cada valor medido. Un ejemplo de los datos tabulados se presenta en la Tabla 3.4. Se toman varias medidas y se utilizan estos datos para convertir el valor de la señal medida en un dato en grados.

Tabla 3.4. Ejemplo de valores tomados de los flexosensores

	<b>DIP1</b>	<b>DIP2</b>	<b>PIP2</b>	<b>MCP2</b>	<b>PIP1</b>	<b>MCP1</b>
1 (Máxima Flexión)	440	839	226	581	780	562

2	537	828	370	690	783	742
3	585	839	370	750	812	570
4	615	857	340	820	571	765
5	640	888	440	907	515	960
6	662	909	490	990	473	1000
7	682	916	720	995	679	990
8	720	955	990	999	1014	1000
9	727	978	1023	1000	1023	1023
10 (Máxima Extensión)	735	973	1023	1020	1023	1023

En este punto se evidencia que las señales que ingresan del circuito de acondicionamiento vienen con un ruido muy grande, por lo tanto, se decide utilizar un filtro por software en el controlador. Este filtro consiste en tomar los 10 últimos valores de la señal y promediarlos. Dentro de la memoria del controlador se guarda en un arreglo de 10 espacios, los valores leídos de cada sensor. Cada vez que ingresa un nuevo valor del sensor se desplazan las medidas desde la posición 1 del arreglo hasta la posición 9, se desecha la última posición y se guarda el valor leído en la primera posición del arreglo. Los valores dentro del arreglo se promedian y ese valor se utiliza como la posición de la articulación. El uso de este filtro mejora la calidad de las señales, sin embargo, aún presentan ruido.

Luego de convertir los valores de los sensores a ángulos, se vuelve a retomar el procedimiento, pero ahora se envía los valores de los ángulos y se obtienen valores como los que se observan en la Tabla 3.5. Se va calibra estos datos convertidos a ángulos para obtener mejores valores. Las señales ya calibradas se utilizan en el controlador para definir las acciones de control en lazo cerrado que se deben realizar para el control de posición.

Tabla 3.5. Valores tomados por los flexosensores en grados

	<b>DIP1</b>	<b>DIP2</b>	<b>PIP2</b>	<b>MCP2</b>	<b>PIP1</b>	<b>MCP1</b>
1 (Máxima Extensión)	16	51	1	21	-7	13
2	16	51	7	21	0	14
3	47	51	20	22	33	19
4	47	51	30	27	50	18

5	16	51	33	32	76	26
6	78	51	43	39	101	28
7	78	100	47	49	135	31
8	78	100	83	54	83	46
9	109	100	70	65	20	45
10 (Máxima Flexión)	78	100	74	62	11	64

Para el dedo pulgar, se obtienen las medidas de los dos sensores de posición. Estos sensores no presentan un ruido tan grande, sin embargo, se utiliza el mismo filtro para mejorar los valores obtenidos. Además, los valores de los sensores que se obtienen corresponden a la posición angular de los motores que controlan el mecanismo. De igual forma se obtiene las medidas y se las calibra para su lectura en el controlador. Es suficiente la cinemática directa del mecanismo para determinar la posición de la punta del dedo, debido a que los valores que se miden son las posiciones angulares de los motores.

A pesar de los problemas con los sensores, se utiliza estas señales para el sistema de control. Se realizan varias pruebas en el sistema de control en conjunto con la interfaz de mando y no presenta mayores problemas, excepto varios cambios en los puntos de máxima extensión y máxima flexión. Sin embargo, no se consideran como un error muy grande. Se observa en la Tabla 3.4 y Tabla 3.5, que los valores de los sensores DIP1 y DIP2 no tienen resultados lineales, lo que hace que no se pueda utilizar correctamente la cinemática inversa. Se utiliza una aproximación de dichos valores que se presenta en [11], en esta dice que los movimientos de la articulación DIP son dependientes de la articulación PIP.

Para comprobar el funcionamiento del sistema de control en conjunto con la interfaz de mando se realiza un nuevo procedimiento de pruebas, en el cual se escogen los tipos de movimientos y en distintos números de repeticiones al azar. Las pruebas realizadas se detallan en la Tabla 3.6.

Tabla 3.6. Pruebas realizadas al Sistema de Control e Interfaz de Mando

Prueba	Tipo de Movimiento	Número de Repeticiones
1	Flexión – Extensión	5
2	Flexión – Extensión	10
3	Flexión – Extensión	12
4	Pinza	5
5	Pinza	10

La alimentación de las tarjetas Arduino utilizadas permiten dos opciones para obtener la energía, una con un conector directo a una fuente de 6 a 12V, la otra con el puerto de conexión USB que tiene la tarjeta y que permite la programación de esta. En el proyecto se utiliza la fuente externa de 12V, pero se observaron varios problemas al conectar elementos externos a la tarjeta como pulsantes y el módulo bluetooth, haciendo que se sobrecargue y se quemara el regulador que tiene la tarjeta. Esto hizo que ya no se pueda utilizar la tarjeta con la fuente de 6 a 12V, sin embargo, continúa el funcionamiento con la alimentación por medio de la entrada USB. Este problema se observó en el Arduino Mega, la solución que se tomó fue alimentar los elementos externos desde la tarjeta Arduino Uno, lo que mejoró el funcionamiento.

## 3.2. Discusión

### 3.2.1 Discusión del Estudio Cinemático

En el capítulo 2 se ha considerado los rangos de funcionamiento del mecanismo y se los compara con los rangos de movimiento de los dedos. Dentro de los rangos de movimiento de los dedos se toman en cuenta todas las combinaciones posibles de los ángulos de las 3 articulaciones de los dedos trifalángicos, sin considerar que hay posiciones muy poco probables que se obtengan y mucho menos en un movimiento de flexión y extensión. Por lo tanto, al obtener la trayectoria posible de los dedos de la mano en el movimiento de flexión – extensión se restringe el rango de funcionamiento ya que se omiten los valores menos probables. Son valores menos probables debido a que existen restricciones internas de cada dedo, que hacen que el dedo no pueda ubicarse en ciertas posiciones. Dichos estudios de las restricciones se presentan en [11] o [19], donde se define una relación entre los movimientos de las articulaciones, por ejemplo se presenta la relación entre la articulación DIP y la articulación PIP en los dedos trifalángicos, que dice que  $\theta_{DIP}$  es  $2/3 \theta_{PIP}$ . Esto restringe el campo de trabajo en un

movimiento de flexión – extensión. Para determinar dichas restricciones se hace de manera experimental, es decir se realizan mediciones de los ángulos utilizando sensores y con dichas medidas se definen las restricciones como se hace en [18]. Por eso es importante generar la trayectoria, y de esta forma saber la posición de los ángulos que tendrá el mecanismo en conjunto con el dedo en el movimiento de flexión y extensión.

En el caso del estudio cinemático del dedo pulgar se observa en la Figura 3.6 que el rango de trabajo del dedo pulgar, cuando el dedo tiene longitudes máximas, ocupa un espacio más grande que el posible con el mecanismo. Es decir, hay puntos a los cuales el mecanismo no va a llegar cuando los dedos tienen longitudes máximas. Además, luego de observar el funcionamiento del mecanismo del dedo pulgar, se observa que la articulación que genera el movimiento de abducción y aducción también incluye ciertas restricciones de movimiento, tomando en cuenta que es una articulación prismática. Esto hace que el movimiento del dedo pulgar se restrinja y por tanto el espacio de trabajo se vea reducido. La diferencia que existe entre el modelo cinemático obtenido con el de la mano real es que, en el modelo cinemático que se obtiene del mecanismo no se considera la articulación prismática que genera el movimiento de abducción y aducción. En un modelo real de la mano el movimiento de abducción y aducción se genera desde la articulación TMC, mientras que el mecanismo tiene la articulación prismática unida a la articulación MCP. El movimiento de abducción y aducción del dedo pulgar es un movimiento rotacional que se pretende realizar con una articulación con movimiento lineal, lo que genera las restricciones de movimiento, haciendo que el rango de movimiento del dedo pulgar disminuya. Una solución para este problema está en la mecánica del exoesqueleto, en los dedos trifalángicos para no restringir el movimiento de abducción y aducción se tiene en el mecanismo una articulación de libre movimiento que se posiciona dependiendo de la abducción y aducción del dedo, dando libertad de movimiento al dedo. Se puede eliminar la articulación prismática que genera el movimiento de abducción y aducción, una solución parecida se presenta en [16], donde se tiene un mecanismo para el dedo pulgar que es una cadena cinemática serial subactuada con una sola articulación motorizada que realiza el movimiento de flexión y extensión, y se deja libre el movimiento de abducción y aducción.

La modelación cinemática en el presente proyecto se realiza por separado en el dedo y el mecanismo, existen autores que hacen la modelación del conjunto dedo mecanismo y le interpretan como una cadena cinemática cerrada. Un ejemplo de esto se tiene en [14], donde los ángulos de los dedos se consideran como las entradas del modelo

cinemático directo y los ángulos del mecanismo son las salidas del modelo cinemático directo.

### **3.2.2. Discusión del Funcionamiento del Exoesqueleto**

El procedimiento de pruebas del circuito de acondicionamiento presenta datos que permiten observar que el sensor no entrega una señal muy estable. En primer lugar, se tiene ruido, para lo que se utiliza un filtro por software que da buenos resultados. También, existe un problema en la estabilidad de la señal ya que a pesar de que el exoesqueleto está fijo, la señal presenta pequeños cambios, lo que puede generar problemas en el programa de control. También presenta variaciones entre una prueba y otra, implica que no tiene buena repetibilidad. El uso del filtro mejora mucho el problema de repetibilidad, sin embargo, en las pruebas finales se observan pequeñas variaciones entre los puntos máximos y mínimos de la trayectoria. Para mejorar dichos resultados se debería cambiar la instrumentación con otro tipo de sensores que no tengan los problemas que presentan los flexosensores.

Uno de los problemas principales que se tiene al momento de manejar el exoesqueleto es la gran cantidad de señales analógicas que se utilizan. El controlador utiliza un algoritmo que representa un tiempo alto de procesamiento para convertir la señal analógica en digital, esto hace que el programa de control sea más lento. Por esto en varias pruebas se han omitido las señales de los sensores de fuerza para obtener mejores resultados. Los sensores de fuerza en el presente proyecto se utilizan como limitadores para evitar fuerzas excesivas que se puedan aplicar a los dedos. Dentro del lazo de control de posición no son de vital importancia, por tanto, su omisión no afecta el funcionamiento del programa de control. Sin embargo, en el caso de una rehabilitación activa en la que se deba realimentar las señales de los sensores de fuerza se los puede utilizar ya que están las señales listas para su funcionamiento. Además, en un diseño inicial se consideró el uso de 10 sensores de fuerza, 5 para determinar el movimiento de flexión y 5 para el movimiento de extensión, es decir dos sensores en cada dedo. Con los cambios finales del diseño mecánico se incluye una pieza que genera presión en el dedo y es donde se ubican los sensores de fuerza. Debido al uso de esta pieza se puede omitir el uso de 10 sensores y reducir a 5, ya que el movimiento de flexión y extensión genera fuerzas de acción y reacción que permiten determinar la fuerza en los dos sentidos, movimiento de flexión y de extensión.

La comunicación entre la interfaz y el sistema de control se planifica con el uso de MATLAB, sin embargo, este software presenta varios inconvenientes al momento de

dejar los canales seriales abiertos, debido a esto se presenta inestabilidad en la interfaz mencionada. La interfaz es la encargada de enviar las sentencias de mando hacia el exoesqueleto, no se realizan cálculos en la interfaz, por lo que no es de importancia el uso de MATLAB. Por lo tanto, se puede generar la interfaz de mando en otro software que permita enviar las mismas sentencias de mando y que le dé mejor estabilidad en la comunicación hacia el exoesqueleto, a través del canal serial por medio del módulo bluetooth.

## 4. CONCLUSIONES Y RECOMENDACIONES

### 4.1. CONCLUSIONES

- El estudio cinemático ha permitido conocer los movimientos que se generan en el exoesqueleto, de esta forma se puede simular las distintas posiciones en las que se debe ubicar al mecanismo en conjunto con el dedo. Con el estudio cinemático y, luego con la utilización del exoesqueleto se puede predecir la trayectoria que seguirán los dedos cuando se tienen los movimientos de flexión y extensión.
- Para tener un mejor control del exoesqueleto es importante desarrollar los modelos cinemáticos ya que esto nos permite conocer la forma en que se mueve el mecanismo en conjunto con el dedo.
- En este proyecto se trabaja por separado los modelos cinemáticos. Únicamente en la obtención de las trayectorias se pone el conjunto dedo – mecanismo. En este punto se verifica que los movimientos sean posibles con la forma del mecanismo.
- Se define al sistema de control como un control de posición. En futuros proyectos se puede mejorar este sistema para tener un control de trayectoria, sin embargo, se pueden lograr muchas mejoras utilizando el modelo dinámico del sistema y hacer un control en tiempo real que permita generar trayectorias de funcionamiento.
- Los valores medidos de la articulación DIP1 y DIP2 de los dedos trifalángicos no se pueden utilizar debido a que no reflejan el ángulo de la articulación que se desea medir, esto debido a que no están dentro del guante, sino atados al exoesqueleto haciendo que no midan el valor de la articulación deseada.
- Para solucionar el problema con la comunicación entre la PC y el controlador, se utiliza el software Labview que presenta mejores resultados al momento de comunicarse por el canal serial, sin embargo, es necesario siempre cerrar el canal serial, para lo cual se utiliza un control a través de un botón de STOP.



## 4.2. RECOMENDACIONES

- El dispositivo no tiene topes mecánicos entonces depende de los sensores cuando se ha llegado a los valores máximos y mínimos, en este contexto, se recomienda no utilizar el dispositivo si los sensores no funcionan correctamente, ya que se podría causar daños en el mecanismo.
- Se recomienda, para tener mejores resultados en el mecanismo del dedo pulgar, mejorar la articulación que permite el movimiento de abducción y aducción, debido a que genera limitaciones en el movimiento del dedo, haciendo que el espacio de trabajo se reduzca.
- Al utilizar los canales seriales es recomendable siempre estar pendiente de abrir el canal y cerrar al finalizar, ya que el dejar encendido el canal puede causar daños en la comunicación evitando que funcione correctamente.
- Para mejorar el sistema de control se puede trabajar en el modelo dinámico del sistema con el cual se puede calcular los torques que deben desarrollar los motores para alcanzar las velocidades deseadas.
- Se recomienda en los controladores Arduino utilizar la energía que se toma desde la fuente de alimentación del puerto USB, ya que esta alimentación no pasa a través del regulador de voltaje, debido a que se suscitan problemas cuando se utilizan dispositivos externos a la tarjeta.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] P. Heo, G. Min Gu, S.-j. Lee, K. Rhee y J. Kim, «Current Hand Exoskeleton Technologies for Rehabilitation and Assistive Engineering,» *International Journal of Precision Engineering and Manufacturing*, pp. 807 - 824, 2012.
- [2] G. H. A. Wege, «Development and control of a hand exoskeleton for rehabilitation of hand injuries,» *IEEE Xplore*, p. 1, 2005.
- [3] V. Guerrero, *Proyecto de Vinculación EPN*, Quito, Pichincha, 2017.
- [4] M. Lozada, «<https://bibdigital.epn.edu.ec>,» 2018. [En línea]. Available: <https://bibdigital.epn.edu.ec/handle/15000/19746>. [Último acceso: 20 05 2019].
- [5] K. Zurita, «<https://biblioteca.epn.edu.ec>,» 2018. [En línea]. Available: [https://biblioteca.epn.edu.ec/cgi-bin/koha/opac-detail.pl?biblionumber=71839&query\\_desc=kw%2Cwrdl%3A%20Kevin%20Zurita](https://biblioteca.epn.edu.ec/cgi-bin/koha/opac-detail.pl?biblionumber=71839&query_desc=kw%2Cwrdl%3A%20Kevin%20Zurita). [Último acceso: 20 05 2019].
- [6] K. Zurita, P. Ruiz, F. Lozada, G. Bravo y C. Cevallos, «Mechatronic Interface Design for Hand Rehabilitation and Safe Recuperation,» de *Memorias del I Congreso Internacional de Bioingeniería y Sistemas Inteligentes de Rehabilitación*, Quito, 2017.
- [7] D. Reinkesmeyer, J. Emken y S. Cramer, «Robotics, Motor Learning, and Neurologic Recovery,» *Annu. Med. Biomed*, pp. 497-525, 2004.
- [8] S. Kumar, *Introducción a la Robótica*, México, D.F. : McGraw-Hill, 2008.
- [9] D. McCloy y D. Harris, *Robótica, una introducción*, México: Limusa, 1993.
- [10] M. Granja, «<https://bibdigital.epn.edu.ec/handle/15000/8544>,» 11 08 2014. [En línea]. Available: <https://bibdigital.epn.edu.ec/bitstream/15000/8544/3/CD-5756.pdf>. [Último acceso: 30 9 2019].
- [11] F. Chen Chen, S. Appendino, A. Battezzato y A. Favetto, «Constraint Study for a Hand Exoskeleton: Human Hand Kinematics and Dynamics,» *Hindawi Publishing Corporation*, 2013.
- [12] L. Arias, «Biomecánica y patrones funcionales de la mano,» *Morfologia*, vol. 4, nº 1, pp. 14-24, 2012.
- [13] Z. MA, P. Ben-Tzvi y J. Danoff, «Modeling Human Hand and Sensing Hand Motions with Five Fingered Haptic Glove Mechanism,» de *ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Buffalo, New York, 2014.

- [14] Z. MA y P. Ben-Tzvi, «Design and Optimization of a Five-Fingered Haptic Glove Mechanism,» *Journal Mechanism Robotics*, 2015.
- [15] P. Ben-Tzvi y J. Danoff, «The Design Evolution of a Sensing and Force-Feedback Exoskeleton Robotic (SAFER) Glove for Hand Rehabilitation Application,» *Journal of Mechanisms and Robotics*, pp. 3, 4, 5, 2015.
- [16] I. Sarakoglou, A. Brygo, D. Mazzanti, N. Garcia, D. Caldwell y N. Tsagarakis, «HEXOTRAC: A highly Under-Actuated Hand Exoskeleton for,» de *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, Korea, 2016.
- [17] J. Iqbal, H. Khan y N. Tsagarakis, «A novel exoskeleton robotic system for hand rehabilitation – Conceptualization to prototyping,» *Biocybernetics and Biomedical Engineering*, vol. 34, pp. 78 - 89, 2014.
- [18] M. Bray, E. Koller-Meier, P. Müller, N. Schraudolph y L. V. Gool, «Stochastic optimisation for high-dimensional tracking in dense range maps,» *IEE Proceedings - Vision, Image and Signal Processing*, vol. 152, nº 4, pp. 501-512, 2005.
- [19] S. Cobos, M. Ferre, M. S. Uran, J. Ortego y C. Pena, «Efficient human hand kinematics for manipulation tasks,» de *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, Francia, 2008.
- [20] Spectra symbol, «Flex Sensor Datasheet,» [En línea]. Available: <https://www.spectrasymbol.com/product/flex-sensors/>. [Último acceso: 19 10 2019].
- [21] Murata, «www.murata.com,» [En línea]. Available: <https://www.murata.com/~media/webrenewal/support/library/catalog/products/sensor/rotaryposition/r51e.ashx>. [Último acceso: 19 10 2019].
- [22] Interlink Electronics, «<https://www.interlinkelectronics.com>,» [En línea]. Available: <https://www.interlinkelectronics.com/request-data-sheets>. [Último acceso: 21 10 2019].
- [23] Arduino, «[Arduino.cc](https://www.arduino.cc),» [En línea]. Available: <https://www.arduino.cc>. [Último acceso: 31 10 2019].
- [24] Mean Well, «[farnell.com](http://www.farnell.com),» [En línea]. Available: [http://www.farnell.com/datasheets/1738947.pdf?\\_ga=2.126592589.1541603990.1573665396-1496042572.1573665396](http://www.farnell.com/datasheets/1738947.pdf?_ga=2.126592589.1541603990.1573665396-1496042572.1573665396). [Último acceso: 31 10 2019].
- [25] P. Heo, G. Min Gu, S.-j. Lee, K. Rhee y J. Kim, «Current Hand Exoskeleton Technologies for Rehabilitation and Assistive Engineering,» *INTERNATIONAL*

*JOURNAL OF PRECISION ENGINEERING AND MANUFACTURING*, vol. 13, nº 5, pp. 807-824, 2012.

[26] FAULHABER, «<https://www.faulhaber.com/>,» [En línea]. Available: [https://www.faulhaber.com/fileadmin/Import/Media/EN\\_1717\\_SR\\_DFF.pdf](https://www.faulhaber.com/fileadmin/Import/Media/EN_1717_SR_DFF.pdf).

[Último acceso: 31 10 2019].

[27] A. Ollero, «,» de *Robótica. Manipuladores y robots móviles*, Barcelona, Marcombo, 2001, p. 72.

## Anexos

### Anexo I. Simulaciones del Modelo Cinemático desarrolladas en MATLAB

#### 1. Modelo Cinemático Directo del dedo índice, realizado en Robot de MATLAB

```
%DH_R_indice_15

clear L
%           alpha-1 a-1      theta   d      sigma
L{1} = link([0      0      0      0      0], 'mod');
L{2} = link([pi/2  0      0      0      0], 'mod');
L{3} = link([0      2.65  0      0      0], 'mod');
L{4} = link([0      2.37  0      0      0], 'mod');
L{5} = link([0      2.32  0      0      0], 'mod');

dedoindice1 = robot(L, 'FORMATO', 'MAESTRIA', 'EPN');
clear L
```

#### 2. Modelo Cinemático Directo del dedo índice, se obtiene las matrices de Rotación en MATLAB

```
clc
clear
R_CMCfe=[0.087266 0]; %Rango de movimiento para la articulación
Metacarpofalangeal (MCP) en radianes
R_MCPaa=[-pi/3 pi/3];
R_MCPfe=[-pi/2 pi/6]; %Rango de movimiento para la articulación Proximal
Interfalangeal (PIP) en radianes
R_PIP=[-1.9198 0]; %Rango de movimiento para la articulación Distal
Interfalangeal (DIP) en radianes
R_DIP=[-pi/2 0.087266]; %Rango de movimiento para la articulación Distal
Interfalangeal (DIP) en radianes

theta1=0;
theta2=0;
theta3=-90*pi/180;
theta4=-110*pi/180;
theta5=-80*pi/180;
l1=11.9;
l2=2.65;
l3=2.37;
l4=2.32;

syms t2 t3 t4 t5 p2 p3 p4
q=[t2 t3 t4 t5 0];

L1= [0 0 0 0; pi/2 0 0 0; 0 p2 0 0; 0 p3 0 0; 0 p4 0 0];

for i=1:5

T(:, :, i)=[cos(q(i)) -sin(q(i)) 0 L1(i,2);...
sin(q(i))*cos(L1(i,1)) cos(q(i))*cos(L1(i,1)) -sin(L1(i,1)) -
L1(i,4)*sin(L1(i,1));...
sin(q(i))*sin(L1(i,1)) cos(q(i))*sin(L1(i,1)) cos(L1(i,1))
L1(i,4)*cos(L1(i,1));...
0 0 0 1];

end
```

```

T1=sym(T(:, :, 1))
T2=sym(T(:, :, 2))
T3=sym(T(:, :, 3))
T4=sym(T(:, :, 4))
T5=sym(T(:, :, 5))

Qs=T(:, :, 1)*T(:, :, 2)*T(:, :, 3)*T(:, :, 4)*T(:, :, 5)
Qf=sym(Qs);

x0=0;
y0=0;
z0=0;
Q1s=T(:, :, 1)*T(:, :, 2)*T(:, :, 3);

Q1=eval(subs(Q1s, {t2, t3, t4, t5, p2, p3, p4}, {theta2, theta3, theta4, theta5, l2, l3, l4}
));
x1=Q1(1, 4);
y1=Q1(2, 4);
z1=Q1(3, 4);
Q2s=T(:, :, 1)*T(:, :, 2)*T(:, :, 3)*T(:, :, 4);

Q2=eval(subs(Q2s, {t2, t3, t4, t5, p2, p3, p4}, {theta2, theta3, theta4, theta5, l2, l3, l4}
));
x2=Q2(1, 4);
y2=Q2(2, 4);
z2=Q2(3, 4);
Q3s=T(:, :, 1)*T(:, :, 2)*T(:, :, 3)*T(:, :, 4)*T(:, :, 5);

Q3=eval(subs(Q3s, {t2, t3, t4, t5, p2, p3, p4}, {theta2, theta3, theta4, theta5, l2, l3, l4}
));
x3=Q3(1, 4);
y3=Q3(2, 4);
z3=Q3(3, 4);

plot3([x0 x1], [y0 y1], [z0 z1], 'y-o', [x1 x2], [y1 y2], [z1 z2], 'b-o', [x2 x3], [y2
y3], [z2 z3], 'r-o')
axis([-10 10 -10 10 -10 10]);
title('Dedo Índice')
xlabel('X');
ylabel('Y');
zlabel('Z')
grid;

```

### 3. Modelo Cinemático Directo del dedo pulgar, realizado en Robot de MATLAB

```

clear L
%           alpha-1 a-1      theta   d           sigma
L{1} = link([0      0      0      0      0], 'mod');
L{2} = link([pi/2  0      0      0      0], 'mod');
L{3} = link([0      4.35  0      0      0], 'mod');
L{4} = link([0      3.85  0      0      0], 'mod');
L{5} = link([0      3.2   0      0      0], 'mod');

dedopulgar = robot(L, 'FORMATO', 'MAESTRIA', 'EPN');

```

### 4. Modelo Cinemático Directo del dedo pulgar, se obtiene las matrices de Rotación en MATLAB

```

clc
clear
%% Matriz de DH

```

```

theta1=-0*pi/180;
theta2=-50*pi/180;
theta3=-75*pi/180;
theta4=-75*pi/180;
theta5=0;
l1=4.35;
l2=3.85;
l3=3.2;

syms t1 t2 t3 t4 p1 p2 p3
q=[t1 t2 t3 t4 0];

L1=[0 0 0 0;pi/2 0 0 0;0 p1 0 0;0 p2 0 0;0 p3 0 0];

for i=1:5

T(:,:,i)=[cos(q(i)) -sin(q(i)) 0 L1(i,2);...
sin(q(i))*cos(L1(i,1)) cos(q(i))*cos(L1(i,1)) -sin(L1(i,1)) -
L1(i,4)*sin(L1(i,1));...
sin(q(i))*sin(L1(i,1)) cos(q(i))*sin(L1(i,1)) cos(L1(i,1))
L1(i,4)*cos(L1(i,1));...
0 0 0 1];

end

T1=sym(T(:,:,1))
T2=sym(T(:,:,2))
T3=sym(T(:,:,3))
T4=sym(T(:,:,4))
T5=sym(T(:,:,5))

Qs=T(:,:,1)*T(:,:,2)*T(:,:,3)*T(:,:,4)*T(:,:,5)
Qf=sym(Qs);

x0=0;
y0=0;
z0=0;
Q1s=T(:,:,1)*T(:,:,2)*T(:,:,3);

Q1=eval(subs(Q1s,{t1,t2,t3,t4,p1,p2,p3},{theta1,theta2,theta3,theta4,l1,l2,l3}
));
x1=Q1(1,4);
y1=Q1(2,4);
z1=Q1(3,4);
Q2s=T(:,:,1)*T(:,:,2)*T(:,:,3)*T(:,:,4);

Q2=eval(subs(Q2s,{t1,t2,t3,t4,p1,p2,p3},{theta1,theta2,theta3,theta4,l1,l2,l3}
));
x2=Q2(1,4);
y2=Q2(2,4);
z2=Q2(3,4);
Q3s=T(:,:,1)*T(:,:,2)*T(:,:,3)*T(:,:,4)*T(:,:,5);

Q3=eval(subs(Q3s,{t1,t2,t3,t4,p1,p2,p3},{theta1,theta2,theta3,theta4,l1,l2,l3}
));
x3=Q3(1,4);
y3=Q3(2,4);
z3=Q3(3,4);

plot3([x0 x1],[y0 y1],[z0 z1],'y-o',[x1 x2],[y1 y2],[z1 z2],'b-o',[x2 x3],[y2
y3],[z2 z3],'r-o')
axis([-15 15 -15 15 -15 15]);
title('Dedo pulgar')
xlabel('X');
ylabel('Y');
zlabel('Z')
grid;

```

## 5. Modelo Cinemático Directo del mecanismo del dedo índice, realizado en Robot de MATLAB

```
clear L
%           alpha-1 a-1      theta  d      sigma
L{1} = link([pi/2  0      0      0      0], 'mod'); %alfa1
L{2} = link([0      3.9      -pi/4  0      1], 'mod'); %Translación del
segmento AB, phi10 =-30
L{3} = link([0      5      -pi/3  0      1], 'mod'); %segmento BC para
rotar dos veces 2.1
L{4} = link([0      7      0      0      0], 'mod'); %segmento CD eslabón 1
junta prismática 3
L{5} = link([0      3      -42.08*pi/180 0      1], 'mod'); %segmento DE eslabón
1 junta prismática 4
L{6} = link([0      3      0      0      0], 'mod'); %segmento EF eslabón
1 para rotar dos veces 5
L{7} = link([0      1.4  0      0      1], 'mod'); %segmento EF eslabón
1 para rotar dos veces 5.1

dedoindice2 = robot(L, 'FORMATO', 'MAESTRIA', 'EPN');
clear L
```

## 6. Modelo Cinemático Directo del mecanismo del dedo índice, se obtiene las matrices de Rotación en MATLAB

```
clc
clear

theta1=63.3463*pi/180; %alfa1
theta2=-45*pi/180;
theta3=-60*pi/180;
theta4=-43.5674*pi/180; %alfa2
theta5=-42.08*pi/180;
theta6=-12.7*pi/180; %alfa3
theta7=0;

l2=3.9;
l3=5;
l4=7;
l5=3;
l6=3;
l7=1.4;

syms t1 t2 t3 t4 t5 t6 t7 p2 p3 p4 p5 p6 p7;
q=[t1 t2 t3 t4 t5 t6 0];

L1=[pi/2 0 0 0;0 p2 0 0;0 p3 0 0;0 p4 0 0;0 p5 0 0;0 p6 0 0;0 p7 0 0];

for i=1:7
T(:, :, i)=[cos(q(i)) -sin(q(i)) 0 L1(i,2);... %sin(L1(i,1))*sin(q(i))
L1(i,2)*cos(q(i));...
sin(q(i))*cos(L1(i,1)) cos(q(i))*cos(L1(i,1)) -sin(L1(i,1)) -
L1(i,4)*sin(L1(i,1));...
sin(q(i))*sin(L1(i,1)) cos(q(i))*sin(L1(i,1)) cos(L1(i,1))
L1(i,4)*cos(L1(i,1));...
0 0 0 1];
end

T1=sym(T(:, :, 1))
T2=sym(T(:, :, 2))
```



```

T3=sym(T(:, :, 3))
T4=sym(T(:, :, 4))
T5=sym(T(:, :, 5))
T6=sym(T(:, :, 6))
T7=sym(T(:, :, 7))

Qs=T(:, :, 1)*T(:, :, 2)*T(:, :, 3)*T(:, :, 4)*T(:, :, 5)*T(:, :, 6)*T(:, :, 7)
Qf=sym(Qs);

x0=0;
y0=0;
z0=0;
Q1s=T(:, :, 1)*T(:, :, 2);

Q1=eval(subs(Q1s, {t1,t2,t3,t4,t5,t6,t7,p2,p3,p4,p5,p6,p7}, {theta1,theta2,theta
3,theta4,theta5,theta6,theta7,12,13,14,15,16,17}));
x1=Q1(1,4);
y1=Q1(2,4);
z1=Q1(3,4);
Q2s=T(:, :, 1)*T(:, :, 2)*T(:, :, 3);

Q2=eval(subs(Q2s, {t1,t2,t3,t4,t5,t6,t7,p2,p3,p4,p5,p6,p7}, {theta1,theta2,theta
3,theta4,theta5,theta6,theta7,12,13,14,15,16,17}));
x2=Q2(1,4);
y2=Q2(2,4);
z2=Q2(3,4);
Q3s=T(:, :, 1)*T(:, :, 2)*T(:, :, 3)*T(:, :, 4);

Q3=eval(subs(Q3s, {t1,t2,t3,t4,t5,t6,t7,p2,p3,p4,p5,p6,p7}, {theta1,theta2,theta
3,theta4,theta5,theta6,theta7,12,13,14,15,16,17}));
x3=Q3(1,4);
y3=Q3(2,4);
z3=Q3(3,4);
Q4s=T(:, :, 1)*T(:, :, 2)*T(:, :, 3)*T(:, :, 4)*T(:, :, 5);

Q4=eval(subs(Q4s, {t1,t2,t3,t4,t5,t6,t7,p2,p3,p4,p5,p6,p7}, {theta1,theta2,theta
3,theta4,theta5,theta6,theta7,12,13,14,15,16,17}));
x4=Q4(1,4);
y4=Q4(2,4);
z4=Q4(3,4);
Q5s=T(:, :, 1)*T(:, :, 2)*T(:, :, 3)*T(:, :, 4)*T(:, :, 5)*T(:, :, 6);

Q5=eval(subs(Q5s, {t1,t2,t3,t4,t5,t6,t7,p2,p3,p4,p5,p6,p7}, {theta1,theta2,theta
3,theta4,theta5,theta6,theta7,12,13,14,15,16,17}));
x5=Q5(1,4);
y5=Q5(2,4);
z5=Q5(3,4);
Q6s=T(:, :, 1)*T(:, :, 2)*T(:, :, 3)*T(:, :, 4)*T(:, :, 5)*T(:, :, 6)*T(:, :, 7);

Q6=eval(subs(Q6s, {t1,t2,t3,t4,t5,t6,t7,p2,p3,p4,p5,p6,p7}, {theta1,theta2,theta
3,theta4,theta5,theta6,theta7,12,13,14,15,16,17}));
x6=Q6(1,4);
y6=Q6(2,4);
z6=Q6(3,4);

plot3([x0 x1],[y0 y1],[z0 z1], 'y-o', [x1 x2],[y1 y2],[z1 z2], 'b-o', [x2 x3],[y2
y3],[z2 z3], 'r-o', [x3 x4],[y3 y4],[z3 z4], 'y-o', [x4 x5],[y4 y5],[z4 z5], 'b-
o', [x5 x6],[y5 y6],[z5 z6], 'r-o')
axis([-12 12 -12 12 -12 12]);
title('Mecanismo Dedo Índice')
xlabel('X');
ylabel('Y');
zlabel('Z')
grid;

```

## 7. Modelo Cinemático Directo del mecanismo del dedo pulgar, realizado en Robot de MATLAB

```

clear L
%           alpha-1 a-1      theta   d           sigma
L{1} = link([pi/2      0      0      0      0], 'mod');
L{2} = link([0          9      0      0      0], 'mod');
L{3} = link([-pi/2     0      0      0      0], 'mod');
L{4} = link([0          11     0      0      0], 'mod');

```

```

dedopulgar2 = robot(L, 'FORMATO', 'MAESTRIA', 'EPN');
clear L

```

## 8. Modelo Cinemático Directo del mecanismo del dedo pulgar, se obtiene las matrices de Rotación en MATLAB

```

clc
clear

theta1=60*pi/180;           %alfa1
theta2=-110*pi/180;        %alfa2
theta3=0*pi/180;           %alfa3

l2=9;
l4=11;

syms t1 t2 t3 t4 p2 p4;
q=[t1 t2 t3 0];

L1=[pi/2 0 0 0;0 p2 0 0;-pi/2 0 0 0;0 p4 0 0];

for i=1:4

T(:,:,i)=[cos(q(i)) -sin(q(i)) 0 L1(i,2);...
          sin(q(i))*cos(L1(i,1))      cos(q(i))*cos(L1(i,1))      -sin(L1(i,1))      -
L1(i,4)*sin(L1(i,1));...
          sin(q(i))*sin(L1(i,1))      cos(q(i))*sin(L1(i,1))      cos(L1(i,1))
L1(i,4)*cos(L1(i,1));...
          0 0 0 1];

end

T1=sym(T(:,:,1))
T2=sym(T(:,:,2))
T3=sym(T(:,:,3))
T4=sym(T(:,:,4))

Qs=T(:,:,1)*T(:,:,2)*T(:,:,3)*T(:,:,4)
Qf=sym(Qs);

x0=0;
y0=0;
z0=0;
Q1s=T(:,:,1)*T(:,:,2);
Q1=eval(subs(Q1s, {t1,t2,t3,p2,p4}, {theta1,theta2,theta3,l2,l4}));
x1=Q1(1,4);
y1=Q1(2,4);
z1=Q1(3,4);
Q2s=T(:,:,1)*T(:,:,2)*T(:,:,3)*T(:,:,4);
Q2=eval(subs(Q2s, {t1,t2,t3,p2,p4}, {theta1,theta2,theta3,l2,l4}));
x2=Q2(1,4);
y2=Q2(2,4);
z2=Q2(3,4);

plot3([x0 x1],[y0 y1],[z0 z1], 'y-o', [x1 x2],[y1 y2],[z1 z2], 'b-o')
axis([-12 12 -12 12 -12 12]);
title('Mecanismo Dedo Pulgar')
xlabel('X');
ylabel('Y');
zlabel('Z')

```

## 9. Modelo Cinemático del dedo índice en conjunto con el mecanismo, generación de trayectoria de flexión

```

%% MODELO DEDO INDICE
clear all
clc
close all

ww = 5; hh = 5;
defpos = get(0,'defaultFigurePosition');
set(0,'defaultFigurePosition', [defpos(1) defpos(2) ww*100, hh*100]);
%% Coordenadas pares cinematicos

AB = 0.039; BC = 0.026; CD = 0.024; DE= 0.028 ;EF = 0.042; FG=0.03; GH=0.03;
FH=0.056; % [m]
JK = 0.042; KL=0.025; LI = 0.027; IH=0.014; beta4 = 90*(pi/180); JH= 0.045;
AF=0.1175;

%phi1 = 0.4*pi; % 90 grados [rad]
alpha2 = 135*(pi/180); % [rad]
alpha3 = 120*(pi/180); % [rad]
phi2 = pi-alpha2; % [rad]
phi3 = pi-alpha3; % [rad]
phi5 = 41*(pi/180); % [rad]
phi6 = (pi-(phi5))/2; % [rad]

beta1 = -72*(pi/180); % Angulo MCP [rad]
beta2= -90*(pi/180);
%beta2 = beta2*(pi/180); % Angulo PIP [rad]
beta3 = -50*(pi/180);
%beta3 = (beta3)*(pi/180); % Ángulo DIP [rad]
beta2=0;
beta3=0;
XH=[];
YH=[];
contadorh=0;
% DEDOS
for beta1 = 0:-3.6:-68.4
%MCP
beta1 = beta1*(pi/180);
beta2 = beta2 - 4.5*(pi/180);
beta3 = beta3 - 2.5*(pi/180);
xJ = 0; yJ = 0;
rJ = [xJ yJ 0];

%PIP
xK = JK*cos(beta1); yK = JK*sin(beta1);
rK = [xK yK 0];

%DIP
xL = xK + KL*cos(beta1+beta2); yL = yK + KL*sin(beta1+beta2);
rL = [xL yL 0];

%Punta
xI = xL + LI*cos(beta1+beta2+beta3); yI = yL + LI*sin(beta1+beta2+beta3);
rI = [xI yI 0];

%PAR H
xH = xI + IH*cos(beta1+beta2+beta3+beta4); yH = yI +
IH*sin(beta1+beta2+beta3+beta4);
rH = [xH yH 0];

% ESLABONES

% PAR A

```

```

xA = -0.01; yA = 0.025;
rA = [xA yA 0];

% PAR F
syms xF yF
%JH= ((xJ+xK+xL+0.002)^2+(yJ+yK+yL+0.012)^2)^0.5
eqnF1 = (xF-xH)^2+(yF-yH)^2-FH^2==0;
eqnF2 = (xF-xA)^2+(yF-yA)^2-AF^2==0;

solF = solve([eqnF1 eqnF2],[xF yF]);
xFpos = solF.xF;
yFpos = solF.yF;
xF1 = double(eval(xFpos(1))); xF2 = double(eval(xFpos(2)));
yF1 = double(eval(yFpos(1))); yF2 = double(eval(yFpos(2)));

if xF1 > xH
xF = xF1;
yF= yF1;
else
xF = xF2;
yF= yF2;
end

rF = [xF yF 0];

% PAR E
if (xF<xA && yF<yA)
alpha1=-atan((yF-yA)/(xF-xA));
else
alpha1=atan((yF-yA)/(xF-xA));
end
BF=((BC+CD)^2+(DE+EF)^2-(2*(BC+CD)*(DE+EF)*cos(alpha3)))^0.5;
phi7=acos((- (BC+CD)^2+BF^2+(DE+EF)^2)/(2*(DE+EF)*BF));
phi9= pi-alpha3-phi7;
phi10=alpha2-phi9;
AF=((xF-xA)^2+(yF-yA)^2)^0.5;
phi11=acos((- (BF^2)+AF^2+AB^2)/(2*AB*AF));
phi8=pi-phi11-phi10;
phi12=pi-phi7-phi8;

% PAR E
xE = xF + EF*cos(alpha1+phi12); yE = yF + EF*sin(alpha1+phi12);
rE = [xE yE 0];

% PAR D
xD = xE + DE*cos(alpha1+phi12); yD = yE + DE*sin(alpha1+phi12);
rD = [xD yD 0];

% PAR C
xC = xD + CD*cos(pi-alpha3+phi12+alpha1); yC = yD + CD*sin(pi-
alpha3+phi12+alpha1);
rC = [xC yC 0];

% PAR B
xB = xC+BC*cos(pi-alpha3+phi12+alpha1); yB = yC+BC*sin(pi-
alpha3+phi12+alpha1);
rB = [xB yB 0];

%PAR G
syms xG yG
eqnG1 = (xG-xF)^2+(yG-yF)^2-FG^2==0;
eqnG2 = (xH-xG)^2+(yH-yG)^2-GH^2==0;

```

```

solG = solve([eqnG1 eqnG2],[xG yG]);
xGpos = solG.xG;
yGpos = solG.yG;
xG1 = double(eval(xGpos(1))); xG2 = double(eval(xGpos(2)));
yG1 = double(eval(yGpos(1))); yG2 = double(eval(yGpos(2)));
xG = xG1;
yG = yG1;
rG = [xG yG 0];

phi4 = atan((yB-yC)/(xB-xC));

salphal=(alpha1+phi11)*180/pi;
phi13=acos((-((yG-yE)^2+(xG-xE)^2)+EF^2+FG^2)/(2*EF*FG));
salphal2=(pi-phi13)*180/pi;
phi14=acos((-((yG-yI)^2+(xG-xI)^2)+IH^2+GH^2)/(2*IH*GH));
salphal3=(pi-phi14)*180/pi;

SAB=((yA-yB)^2+(xA-xB)^2)^0.5;
SFH=((yF-yH)^2+(xF-xH)^2)^0.5;
SAF=((yF-yA)^2+(xF-xA)^2)^0.5;
SEF=((yF-yE)^2+(xF-xE)^2)^0.5;
SDE=((yD-yE)^2+(xD-xE)^2)^0.5;
SCD=((yC-yD)^2+(xC-xD)^2)^0.5;
SBC=((yB-yC)^2+(xB-xC)^2)^0.5;
% Resultados:
fprintf('COORDENADAS \n')
fprintf('rA = [ %g, %g, %g ] (m) \n', rA)
fprintf('rB = [ %g, %g, %g ] (m) \n', rB)
fprintf('rC = [ %g, %g, %g ] (m) \n', rC)
fprintf('rD = [ %g, %g, %g ] (m) \n', rD)
fprintf('rE = [ %g, %g, %g ] (m) \n', rE)
fprintf('rF = [ %g, %g, %g ] (m) \n', rF)
fprintf('rG = [ %g, %g, %g ] (m) \n', rG)
fprintf('rH = [ %g, %g, %g ] (m) \n', rH)
fprintf('rI = [ %g, %g, %g ] (m) \n', rI)
fprintf('rJ = [ %g, %g, %g ] (m) \n', rJ)
fprintf('rK = [ %g, %g, %g ] (m) \n', rK)
fprintf('rL = [ %g, %g, %g ] (m) \n\n', rL)
% fprintf('phi2 = %g (grados) \n', phi2*180/pi)

% Gráfico
%hold on
plot([xA,xB],[yA,yB], 'b-o', [xB,xC],[yB,yC], 'b-o', [xC,xD],[yC,yD], 'g-
o', [xD,xE],[yD,yE], 'g-o', [xE,xF],[yE,yF], 'b-o', [xF,xG],[yF,yG], 'r-
o', [xG,xH],[yG,yH], 'r-o', [xJ,xK],[yJ,yK], 'm-o', [xK,xL],[yK,yL], 'm-
o', [xL,xI],[yL,yI], 'm-o', [xI,xH],[yI,yH], 'm-o', [xA,xF],[yA,yF], 'k--
o', [xF,xH],[yF,yH], 'k--o', [xJ,xH],[yJ,yH], 'k--o')
xlabel('x (m)')
ylabel('y (m)')
title('Posición de Articulaciones para máxima flexión')
text(xA,yA, ' A'),
text(xB,yB, ' B'),
text(xC,yC, ' C'),
text(xD,yD, ' D'),
text(xE,yE, ' E'),
text(xF,yF, ' F'),
text(xG,yG, ' G'),
text(xH,yH, ' H'),
text(xI,yI, ' Punta'),
text(xJ,yJ, ' MCP'),
text(xK,yK, ' PIP'),
text(xL,yL, ' DIP'),
axis([-0.12 0.12 -0.12 0.12]),
grid

if
beta1==0
%Almacenamiento de la trayectoria de la punta de la Yema del dedo en
referencia a la base del mecanismo SAFER

```

```
XH=xI;
YH=yI;
else
XH=[XH xI];
YH=[YH yI];
end

hold off
pause(0.0001)

end

plot(XH,YH,'gx') %Grafica
de las coordenadas X y Y de la punta de la yema del dedo
grid;
axis([-0.10 0.10 -0.10 0.10]);
```

## Anexo II. Matrices de Rotación obtenidas de los modelos cinemáticos

### 1. Matrices de Rotación del dedo índice

T1 =

$$[\cos(t_2), -\sin(t_2), 0, 0]$$

$$[\sin(t_2), \cos(t_2), 0, 0]$$

$$[0, 0, 1, 0]$$

$$[0, 0, 0, 1]$$

T2 =

$$[\cos(t_3), -\sin(t_3), 0, 0]$$

$$[0, 0, -1, 0]$$

$$[\sin(t_3), \cos(t_3), 0, 0]$$

$$[0, 0, 0, 1]$$

T3 =

$$[\cos(t_4), -\sin(t_4), 0, p_2]$$

$$[\sin(t_4), \cos(t_4), 0, 0]$$

$$[0, 0, 1, 0]$$

$$[0, 0, 0, 1]$$

T4 =

$$[\cos(t_5), -\sin(t_5), 0, p_3]$$

$$[\sin(t_5), \cos(t_5), 0, 0]$$

$$[0, 0, 1, 0]$$

$$[0, 0, 0, 1]$$

T5 =

$$[1, 0, 0, p_4]$$

$$[0, 1, 0, 0]$$

$$[0, 0, 1, 0]$$

$$[0, 0, 0, 1]$$

### 2. Matrices de Rotación del dedo pulgar

T1 =

$$[\cos(t_1), -\sin(t_1), 0, 0]$$

$$[\sin(t_1), \cos(t_1), 0, 0]$$

$$[0, 0, 1, 0]$$

$$[0, 0, 0, 1]$$

T2 =

$$[\cos(t_2), -\sin(t_2), 0, 0]$$

$$[0, 0, -1, 0]$$

$$[\sin(t_2), \cos(t_2), 0, 0]$$

$$[0, 0, 0, 1]$$

$T3 =$   
 $[\cos(t3), -\sin(t3), 0, p1]$   
 $[\sin(t3), \cos(t3), 0, 0]$   
 $[0, 0, 1, 0]$   
 $[0, 0, 0, 1]$   
 $T4 =$   
 $[\cos(t4), -\sin(t4), 0, p2]$   
 $[\sin(t4), \cos(t4), 0, 0]$   
 $[0, 0, 1, 0]$   
 $[0, 0, 0, 1]$   
 $T5 =$   
 $[1, 0, 0, p3]$   
 $[0, 1, 0, 0]$   
 $[0, 0, 1, 0]$   
 $[0, 0, 0, 1]$

### 3. Matrices de Rotación del mecanismo del dedo índice

$T1 =$   
 $[\cos(t1), -\sin(t1), 0, 0]$   
 $[0, 0, -1, 0]$   
 $[\sin(t1), \cos(t1), 0, 0]$   
 $[0, 0, 0, 1]$   
 $T2 =$   
 $[\cos(t2), -\sin(t2), 0, p2]$   
 $[\sin(t2), \cos(t2), 0, 0]$   
 $[0, 0, 1, 0]$   
 $[0, 0, 0, 1]$   
 $T3 =$   
 $[\cos(t3), -\sin(t3), 0, p3]$   
 $[\sin(t3), \cos(t3), 0, 0]$   
 $[0, 0, 1, 0]$   
 $[0, 0, 0, 1]$   
 $T4 =$   
 $[\cos(t4), -\sin(t4), 0, p4]$   
 $[\sin(t4), \cos(t4), 0, 0]$   
 $[0, 0, 1, 0]$   
 $[0, 0, 0, 1]$   
 $T5 =$   
 $[\cos(t5), -\sin(t5), 0, p5]$   
 $[\sin(t5), \cos(t5), 0, 0]$   
 $[0, 0, 1, 0]$   
 $[0, 0, 0, 1]$   
 $T6 =$   
 $[\cos(t6), -\sin(t6), 0, p6]$



$$\begin{bmatrix} \sin(t_6) & \cos(t_6) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

T7 =

$$\begin{bmatrix} 1 & 0 & 0 & p_7 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

#### 4. Matrices de Rotación del mecanismo del dedo índice

T1 =

$$\begin{bmatrix} \cos(t_1) & -\sin(t_1) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin(t_1) & \cos(t_1) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

T2 =

$$\begin{bmatrix} \cos(t_2) & -\sin(t_2) & 0 & p_2 \\ \sin(t_2) & \cos(t_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

T3 =

$$\begin{bmatrix} \cos(t_3) & -\sin(t_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\sin(t_3) & -\cos(t_3) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

T4 =

$$\begin{bmatrix} 1 & 0 & 0 & p_4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

## Anexo III. Programa de Control del Arduino MEGA

```
//Motores
const int pwm_a = 3, dir_a = 12, brake_a = 9; //motor 1 Indice-Medio
const int pwm_b = 11, dir_b = 13, brake_b = 8; //motor 2 Anular-menique
const int m1_horario = 22, m1_antihorario = 23, m2_horario = 24, m2_antihorario
= 25; // pulsantes manejo manual motores

//Sensores de presión 8 sensores 4 dedos
int sp1 = A2, sp2 = A1, sp3 = A5, sp4 = A6, sp5 = A7, sp6 = A8, sp7 = A10, sp8
= A11;

//Flexosensores 6 sensores 3 en cada par de dedos
int fs[6] = {A2, A3, A4, A7, A8, A9};

//Variables para guardar los valores de los sensores
int fsv[6][10];
int fsvp[6];

//Contador de ciclos antes de lectura de canales analogicos
int cont=0, i=1, rl=0;

//Variables para obtener la lectura de los pulsantes
int buttonStatel = 0, buttonState2 = 0, buttonState3 = 0, buttonState4 = 0;

int func=0, tipo=1, rep=0, maxrep=3, flex=1;
int velm1=130, velm2=130;

void Analogicos(), LeerPulsantes();

String inputString = ""; // a String to hold incoming data
bool stringComplete2 = false, stringComplete1 = false; // whether the string
is complete
long LecturaSerial2[5], LecturaSerial1[5];
int k1, k2=0;
int Llego=0, Cambia=0;
int enviado=0;
int j;

int minvalue11=300, minvalue12=650, minvalue21=725, minvalue22=875,
maxvalue11=950, maxvalue12=950;
int minvalue1, minvalue2, maxvalue1, maxvalue2;

void setup()
{
  pinMode(m1_horario, INPUT);
  pinMode(m1_antihorario, INPUT);
  pinMode(m2_horario, INPUT);
  pinMode(m2_antihorario, INPUT);
  pinMode(brake_a, OUTPUT); // Brake pin on channel A
  pinMode(dir_a, OUTPUT); // Direction pin on channel A
  pinMode(brake_b, OUTPUT); // Brake pin on channel A
  pinMode(dir_b, OUTPUT); // Direction pin on channel A

  pinMode(26,OUTPUT);
  // Open Serial communication
  Serial.begin(9600); //Canal para enviar datos a la PC, valores de
angulos medidos
  Serial.print("Motor shield DC motor Test:");
  Serial.print("12345");

  Serial1.begin(9600); //Canal de comunicación entre los microcontroladores
  //Serial1.print("Prueba de Comunicación entre microcontroladores");
  Serial2.begin(9600); //Canal de comunicación con el HMI
}

void loop()
{
```

```

Analogicos();
SerialEntrada();

if (func==1)
{
    enviarserial(1);

    if (tipo==1)
    {
        minvalue1=minvalue11;
        minvalue2=minvalue12;
        maxvalue1=maxvalue11;
        maxvalue2=maxvalue12;
    }
    else
    {
        minvalue1=minvalue21;
        minvalue2=minvalue12;
        maxvalue1=maxvalue11;
        maxvalue2=maxvalue12;
    }
    while (rep<maxrep)
    {
        Analogicos();
        SerialEntrada();
        if (flex==1)
        {
            if (fsvp[2]>minvalue1)
            {
                digitalWrite(brake_a, LOW);
                digitalWrite(dir_a, LOW);
                analogWrite(pwm_a, velm1);
            }
            else if (fsvp[2]<minvalue1)
            {
                digitalWrite(brake_a, HIGH);
                analogWrite(pwm_a, 0);
            }
            if (fsvp[4]>minvalue2)
            {
                digitalWrite(brake_b, LOW);
                digitalWrite(dir_b, HIGH);
                analogWrite(pwm_b, velm2);
            }
            else if (fsvp[4]<minvalue2)
            {
                digitalWrite(brake_b, HIGH);
                analogWrite(pwm_b, 0);
            }
            if (fsvp[2]<minvalue1 && fsvp[4]<minvalue2)
            {
                if(Llego==1)
                {
                    Cambia=1;
                    enviarserial(3);
                }
                else
                {
                    if (enviado==0)
                    {
                        digitalWrite(26, HIGH);
                        enviarserial(2);
                        enviado=1;
                    }
                }
            }
            if(Cambia==1)
            {

```

```

        delay(500);
        enviado=0;
        flex=0;
        Cambia=0;
        Llego=0;
    }
}
if (flex==0)
{
    if (fsvp[2]<maxvalue1)
    {
        digitalWrite(brake_a, LOW);
        digitalWrite(dir_a, HIGH);
        analogWrite(pwm_a, velm1);
    }
    else if (fsvp[2]>maxvalue1)
    {
        digitalWrite(brake_a, HIGH);
        analogWrite(pwm_a, 0);
    }
    if (fsvp[4]<maxvalue2)
    {
        digitalWrite(brake_b, LOW);
        digitalWrite(dir_b, LOW);
        analogWrite(pwm_b, velm2);
    }
    else if (fsvp[4]>maxvalue2)
    {
        digitalWrite(brake_b, HIGH);
        analogWrite(pwm_b, 0);
    }
    if (fsvp[2]>maxvalue1 && fsvp[4] > maxvalue2)
    {
        if(Llego==1)
        {
            Cambia=1;
            enviarserial(3);
        }
        else
        {
            if (enviado==0)
            {
                enviarserial(2);
                enviado=1;
            }
        }
    }
    if(Cambia==1)
    {
        delay(500);
        flex=1;
        rep++;
        Cambia=0;
        Llego=0;
        enviado=0;
        enviarserial(5);
        if (tipo == 2)
        {
            int par = rep%2;
            if (par==0)
            {
                minvalue1=minvalue21;
                minvalue2=minvalue12;
            }
            else
            {
                minvalue1=minvalue11;
                minvalue2=minvalue22;
            }
        }
    }
}

```

```

        }
    }
}

}
enviarserial(6);
func=0;
tipo=0;
rep=0;

}
else
{
LeerPulsantes();
}
}

void LeerPulsantes()
{
    buttonState1 = digitalRead(m1_horario);
    buttonState2 = digitalRead(m1_antihorario);

    if (buttonState1 == LOW or buttonState2 == LOW)
    {

        digitalWrite(brake_a, LOW);
        if (buttonState1 == LOW)
        {
            digitalWrite(dir_a, HIGH);
            func=0;
        }
        if (buttonState2 == LOW)
        {
            digitalWrite(dir_a, LOW); // setting direction to HIGH the motor will spin
forward
            //func=1;
            rep=0;
        }
        analogWrite(pwm_a, 255);
    }
    else
    {
        analogWrite(pwm_a, 0);
        digitalWrite(brake_a, HIGH); // setting brake LOW disable motor brake
    }

    buttonState3 = digitalRead(m2_horario);
    buttonState4 = digitalRead(m2_antihorario);

    if (buttonState3 == LOW or buttonState4 == LOW)
    {
        digitalWrite(brake_b, LOW); // setting brake LOW disable motor brake
        if (buttonState3 == LOW)
        {
            digitalWrite(dir_b, HIGH); // setting direction to HIGH the motor will
spin forward
        }
        if (buttonState4 == LOW)
        {
            digitalWrite(dir_b, LOW); // setting direction to HIGH the motor will spin
forward
        }
        analogWrite(pwm_b, 255);
    }
}

```

```

else
{
analogWrite(pwm_b, 0);
digitalWrite(brake_b, HIGH); // setting brake LOW disable motor brake
}
}

void Analogicos()
{
if (cont == 100) //lectura de canales analogicos
{
if(i<6)
{
filtro();
//Serial.print(fsvp[i]);
//Serial.print(' ');
Serial.print(rep);
Serial.print(' ');
Serial.print(maxrep);
Serial.print(' ');
i++;
}
else
{
i=0;
Serial.print('\n');
}
cont=0;
}
else
{
cont++;
}
}

void filtro()
{
int suma=0,promedio=0;
int j;

for(j=10;j>1;j--)
{
fsv[i][j]=fsv[i][j-1];
suma=fsv[i][j]+suma;
}
fsv[i][1]=analogRead(fs[i]);
suma=fsv[i][1]+suma;
promedio=suma/10;
fsvp[i]=promedio;
}

void serialEvent2()
{
while (Serial2.available())
{
char inChar = (char)Serial2.read();

if (inChar == '\n') {
//Serial.print(inputString);
stringComplete2 = true;
k2=0;
}
else
{
LecturaSerial2[k2]= inChar;
}
}
}

```

```

    LecturaSerial2[k2]=LecturaSerial2[k2]-48;
    k2++;

}
}

void serialEvent1()
{

}

void SerialEntrada()
{
    while (Serial1.available())
    {

        char inChar = (char)Serial1.read();

        if (inChar == '\n') {
            //Serial.print(inputString);
            stringComplete1 = true;
            k1=0;
        }
        else
        {
            LecturaSerial1[k1]= inChar;
            k1++;
        }
    }
    if (stringComplete2)
    {

        func=LecturaSerial2[0];
        tipo=LecturaSerial2[1];
        maxrep=LecturaSerial2[2];
        enviarserial(4);
        Serial1.write(func);
        Serial1.write(tipo);
        Serial1.write(maxrep);
        stringComplete2=0;
    }

    if (stringComplete1)
    {

        if (LecturaSerial1[0]==1)
        {
            if(LecturaSerial1[1]==1 or LecturaSerial1[1]==2)
            {
                r1=1;
            }
            if(LecturaSerial1[1]==3)
            {

                switch(LecturaSerial1[2])
                {
                    case 1: Llego=1;
                    case 2: Cambia=1;
                    case 3: Cambia=1;
                }
            }
        }
        stringComplete1=0;
    }
}
}

```

```

void enviarserial(int mensaje)
{
    int mensaje1 = mensaje;
    if (mensaje1==1)
    {
        for (j=0;j<3;j++)
        {
            Serial1.write(func);
            Serial1.write(tipo);
            Serial1.write(maxrep);
            Serial1.print('\n');
        }
    }

    if (mensaje1==2)
    {
        for (j=0;j<3;j++)
        {
            Serial1.write(1);
            Serial1.write(3);
            Serial1.write(1);
            Serial1.print('\n');
        }
    }

    if (mensaje1==3)
    {
        for (j=0;j<3;j++)
        {
            Serial1.write(1);
            Serial1.write(3);
            Serial1.write(2);
            Serial1.print('\n');
        }
    }

    if (mensaje1==4)
    {
        Serial2.write("Orden Recibida, Funcionamiento: ");
        Serial2.print(func);
        Serial2.write(", Tipo de Movimiento: ");
        Serial2.print(tipo);
        Serial2.write(", Numero de repeticiones: ");
        Serial2.print(maxrep);
        Serial.print('\n');
    }

    if (mensaje1==5)
    {
        Serial2.write("Repetición Terminada, numero de repeticiones: ");
        Serial2.print(rep);
        Serial2.write(", repeticiones totales: ");
        Serial2.print(maxrep);
        Serial.print('\n');
    }

    if (mensaje1==6)
    {
        Serial2.write("Movimiento finalizado, numero de repeticiones: ");
        Serial2.print(rep);
        Serial2.write(", repeticiones totales: ");
        Serial2.print(maxrep);
        Serial.print('\n');
    }
}

```



## Anexo IV. Programa de Control del Arduino UNO

```
//Motores
const int pwm_a = 3, dir_a = 12, brake_a = 9; //motor 1 Indice-Medio
const int pwm_b = 11, dir_b = 13, brake_b = 8; //motor 2 Anular-menique
const int m1_horario = 4, m1_antihorario = 5, m2_horario = 6, m2_antihorario =
7; // pulsantes manejo manual motores

//Sensores de presión 8 sensores 4 dedos
int sp1 = A4, sp2 = A5;

//Sensores de posición
int pot1=A2, pot2=A3;

//variables para guardar los valores de los sensores
int potv1[10], potv2[10];
int potv1p, potv2p;

//contador de ciclos antes de la lectura de canales analogicos
int cont=0;

//Variables para obtener la lectura de los pulsantes
int buttonState1 = 0, buttonState2 = 0, buttonState3 = 0, buttonState4 = 0;

int velm1=200, velm2=100;

int func=0, tipo=1, rep=0, maxrep=3, flex=1;

String inputString = ""; // a String to hold incoming data
bool stringComplete = false; // whether the string is complete
int LecturaSerial[5];
int k=0;
int Llego=0, Cambia=0, enviado=0;
int j;

void setup() {
  // put your setup code here, to run once:
  pinMode(m1_horario, INPUT);
  pinMode(m1_antihorario, INPUT);
  pinMode(m2_horario, INPUT);
  pinMode(m2_antihorario, INPUT);
  pinMode(brake_a, OUTPUT); // Brake pin on channel A
  pinMode(dir_a, OUTPUT); // Direction pin on channel A
  pinMode(brake_b, OUTPUT); // Brake pin on channel A
  pinMode(dir_b, OUTPUT); // Direction pin on channel A
  pinMode(2, OUTPUT);
  // Open Serial communication
  Serial.begin(9600);
  Serial.print("Motor shield DC motor Test:");
  Serial.print("12345");
  inputString.reserve(200);
}

void loop() {

  Analogicos();
  SerialEntrada();

  if (func==1)
  {
    if (tipo==1)
    {
    }
    while (rep<maxrep)
    {
```

```

Analogicos();
SerialEntrada();
if (flex==1)
{
  if (potv1p>160)
  {
    digitalWrite(brake_b, LOW);
    digitalWrite(dir_b, LOW);
    analogWrite(pwm_b, velm1);
  }
  else if (potv1p<160)
  {
    digitalWrite(brake_b, HIGH);
    analogWrite(pwm_b, 0);
  }
  if (potv2p<680)
  {
    digitalWrite(brake_a, LOW);
    digitalWrite(dir_a, HIGH);
    analogWrite(pwm_a, velm2);
  }
  else if (potv2p>680)
  {
    digitalWrite(brake_a, HIGH);
    analogWrite(pwm_a, 0);
  }
  if (potv1p<160 && potv2p>680)
  {
    if(Llego==1)
    {
      Cambia=1;
      enviarserial(3);
    }
    else
    {
      if(enviado==0)
      {
        enviarserial(2);
        enviado=1;
      }
    }
    if(Cambia==1)
    {
      delay(500);
      flex=0;
      Cambia=0;
      Llego=0;
      enviado=0;
    }
  }
}

if (flex==0)
{
  if (potv1p<420)
  {
    digitalWrite(brake_b, LOW);
    digitalWrite(dir_b, HIGH);
    analogWrite(pwm_b, velm1);
  }
  else if (potv1p>420)
  {
    digitalWrite(brake_b, HIGH);
    analogWrite(pwm_b, 0);
  }
}

```

```

    if (potv2p>600)
    {
        digitalWrite(brake_a, LOW);
        digitalWrite(dir_a, LOW);
        analogWrite(pwm_a, velm2);
    }
    else if (potv2p<600)
    {
        digitalWrite(brake_a, HIGH);
        analogWrite(pwm_a, 0);
    }
    if (potv1p>420 && potv2p<600)
    {
        if(Llego==1)
        {
            Cambia=1;
            enviarserial(3);
        }
        else
        {
            if (enviado==0)
            {
                enviarserial(2);
                enviado=1;
            }
        }
        if(Cambia==1)
        {
            delay(500);
            flex=1;
            rep++;
            Cambia=0;
            Llego=0;
            enviado=0;
        }
    }
}
}
func=0;
tipo=0;
rep=0;

}
else
{
LeerPulsantes();
}
}

void LeerPulsantes()
{
    buttonStat1 = digitalRead(ml_horario);
    buttonState2 = digitalRead(ml_antihorario);

    if (buttonStat1 == LOW or buttonState2 == LOW)
    {
        digitalWrite(brake_a, LOW);
        if (buttonStat1 == LOW)
        {
            digitalWrite(dir_a, HIGH);
            func=0;
        }
        if (buttonState2 == LOW)
        {

```

```

    digitalWrite(dir_a, LOW); // setting direction to HIGH the motor will spin
forward
    //func=1;
    rep=0;
    }
    analogWrite(pwm_a, 255);
}
else
{
analogWrite(pwm_a, 0);
digitalWrite(brake_a, HIGH); // setting brake LOW disable motor brake
}

buttonState3 = digitalRead(m2_horario);
buttonState4 = digitalRead(m2_antihorario);

if (buttonState3 == LOW or buttonState4 == LOW)
{
    digitalWrite(brake_b, LOW); // setting brake LOW disable motor brake
    if (buttonState3 == LOW)
    {
        digitalWrite(dir_b, HIGH); // setting direction to HIGH the motor will
spin forward
    }
    if (buttonState4 == LOW)
    {
        digitalWrite(dir_b, LOW); // setting direction to HIGH the motor will spin
forward
    }
    analogWrite(pwm_b, 255);
}
else
{
    analogWrite(pwm_b, 0);
    digitalWrite(brake_b, HIGH); // setting brake LOW disable motor brake
}
}

void Analogicos()
{
    if (cont == 100) //lectura de canales analogicos
    {
        filtro();

        //Serial.print(potv1p);
        //Serial.print('\t');

        //Serial.print(potv2p);
        //Serial.print(' ');
        //Serial.print(flex);
        //Serial.print(' ');
        //Serial.print(func);
        //Serial.print(' ');
        //Serial.print('\n');
        cont =0;
    }
    else
    {
        cont=cont+1;
    }
}

void filtro()
{
int suma1=0, suma2=0, promedio1=0, promedio2;
int j;

for(j=10;j>1;j--)

```

```

{
potv1[j]=potv1[j-1];
suma1=potv1[j]+suma1;
potv2[j]=potv2[j-1];
suma2=potv2[j]+suma2;
}
potv1[1]=analogRead(pot1);
suma1=potv1[1]+suma1;
promedio1=suma1/10;
potv1p=promedio1;

potv2[1]=analogRead(pot2);
suma2=potv2[1]+suma2;
promedio2=suma2/10;
potv2p=promedio2;
}

void SerialEntrada()
{
while (Serial.available())
{

char inChar = (char)Serial.read();

if (inChar == '\n') {
//Serial.print(inputString);
stringComplete = true;
k=0;
}
else
{
LecturaSerial[k]= inChar;
k++;
}
}
if (stringComplete)
{
if (LecturaSerial[0]==1)
{
if(LecturaSerial[1]==1 or LecturaSerial[1]==2)
{
func=LecturaSerial[0];
tipo=LecturaSerial[1];
maxrep=LecturaSerial[2];
Serial.write(func);
Serial.write(tipo);
Serial.write(maxrep);
Serial.print('\n');
}
if(LecturaSerial[1]==3)
{
digitalWrite(2, HIGH);
switch(LecturaSerial[2])
{
case 1: Llego=1;
case 2: Cambia=1;
}
}
}
stringComplete=0;
}
}

void serialEvent()
{
while (Serial.available())
{

```

```

char inChar = (char)Serial.read();

if (inChar == '\n') {
    //Serial.print(inputString);
    stringComplete = true;
    k=0;
}
else
{
    LecturaSerial[k]= inChar;
    k++;
}
}
}

void enviarserial(int mensaje)
{
    int mensaje1 = mensaje;
    if (mensaje1==1)
    {
        for (j=0;j<3;j++)
        {
            Serial.write(func);
            Serial.write(tipo);
            Serial.write(maxrep);
            Serial.print('\n');
        }
    }

    if (mensaje1==2)
    {
        for (j=0;j<3;j++)
        {
            Serial.write(1);
            Serial.write(3);
            Serial.write(1);
            Serial.print('\n');
        }
    }

    if (mensaje1==3)
    {
        for (j=0;j<3;j++)
        {
            Serial.write(1);
            Serial.write(3);
            Serial.write(2);
            Serial.print('\n');
        }
    }
}
}

```

## Anexo V. Programa de la interfaz de mando en labview

