

2020-01

DISEÑO Y DESARROLLO DE DISPOSITIVO DE CAPTACIÓN DE MOVIMIENTOS DE UNA MANO HUMANA PARA CONTROL DE UNA MANO ROBÓTICA

JARA LIRA, ALEXIS ANDRES

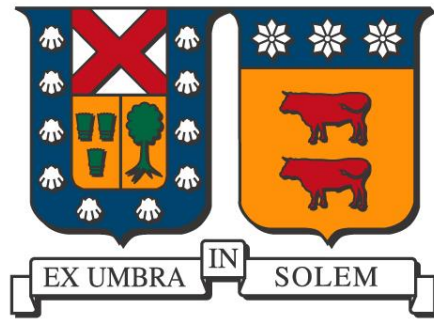
<https://hdl.handle.net/11673/49552>

Repositorio Digital USM, UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARIA

DEPARTAMENTO DE INGENIERÍA MECÁNICA

SANTIAGO – CHILE



“Diseño y desarrollo de dispositivo de captación
de movimientos de una mano humana para
control de una mano robótica”

ALEXIS ANDRES JARA LIRA

MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE INGENIERO CIVIL
MECÁNICO

PROFESOR GUÍA:

DR. ING. DANILO ESTAY B.

PROFESOR CO-REFERENTE:

DR. ING. SHEILA LASCANO F.

SANTIAGO, ENERO 2020

Resumen

En la búsqueda de la emulación del movimiento de una mano humana para realizar agarres básicos, en el presente trabajo de tesis se desarrolla un sistema de control para una mano robótica que emula los movimientos de flexión y extensión de los dedos y la muñeca de una mano humana. El sistema consiste en un dispositivo de captación de los movimientos de una mano humana a través de sensores MEMS con los que se obtiene la orientación de las falanges mediales de cada dedo lo que sirve como referencia para un control del tipo PID sobre los actuadores de la mano robótica, de esta forma se genera un control a lazo cerrado en donde la mano robótica replica los movimientos de un dedo y la muñeca de la mano humana.

Palabras Clave: Mano Robótica, Telecomando, Gripper, Control PID

Abstract

In the search for the emulation of the movement of a human hand to perform basic grips, in this thesis work a control system is developed for a robotic hand that emulates the flexion and extension movements of the fingers and wrist of one human hand. The system consists of a device for capturing the movements of a human hand through MEMS sensors with which the orientation of the medial phalanges of each finger is obtained, which serves as a reference for a PID control over the actuators of the robotic hand, in this way a closed control is generated where the robotic hand replicates the movements of a finger and the wrist of the human hand.

Keywords: Robotic Hand, Telecomand,Gripper, PID Control

Glosario

Baud Rate:	Velocidad de transmisión de datos en bits por segundo(baudios) para la transmisión de datos en serie.
Bit:	Unidad de almacenamiento básica que puede tener dos valores, 1 o 0.
Comunicación Serial:	Es un tipo de comunicación entre dos dispositivos que se realiza a través de envío de bits de forma secuencial.
Denavit-Hartenberg (DH):	Algoritmo que establece los sistemas de referencia para cada uno de los eslabones con los que cuenta un robot.
DoF:	Son el número mínimo de velocidades generales independientes necesarias para definir el estado cinemático de un mecanismo o sistema mecánico.
Dorso:	Parte exterior de la mano, opuesta a la palma.
Drift:	Error de arrastre presente en el cálculo del ángulo de inclinación a través del giroscopio al realizar la aproximación de la integral, el que va creciendo a medida que se procesan datos en el tiempo.
Eslabón o Link:	Es un miembro rígido del mecanismo capaz de causar o controlar movimiento y de transmitir fuerza de un miembro a otro.

Gymbal Lock:

Bloqueo de Cardán en español, en campo de la matemática aplicada, es un Fenómeno de la pérdida de un grado de libertad en la representación de la aplicación continua de los ángulos de Euler para las rotaciones.

I2C:

Sigla en inglés para “Inter-Integrated Circuit” es un bus de datos, diseñado para comunicación serial entre dispositivos de la forma “maestro-esclavo” , las conexiones para transmitir datos en el hardware electrónico generalmente son llamadas SDA y SCL.

IMU:

Sigla en inglés para “Inertial Measurement Unit” que hace referencia a un sistema de Medición inercial, se usa indistintamente para referir el sistema de medición del movimiento lineal y angular, como al encapsulado de sensores que traen incorporado un giroscopio y un acelerómetro.

Matlab:

Software y lenguaje de programación de alto nivel.

MEMS:

Sigla del inglés para “Microelectromechanical Systems” que denota la tecnología de Micro electromecánica.

Offset:

Diferencia entre el valor esperado y el valor obtenido.

OverShoot:

En procesamiento de señales se conoce como la aparición de una señal que excede su objetivo.

PID:

Sigla en inglés para “Proportional Integral Derivate Controller”, Sistema de control a lazo cerrado dependiente del error entre la entrada y la salida, compuesto por tres componentes: proporcional, integral, derivativo.

Spatial Tracking:

Es la metodología que mediante sensores determina el posicionamiento y orientación de un objeto, cosa o persona en un determinado espacio.

SPI:

Sigla en inglés para “*Serial Peripheral Interface*” es un *Bus estándar de comunicación, utilizado para transferencia de información entre circuitos integrados en equipos electrónicos.*

Tele operación:

Es el conjunto de tecnologías que comprenden la operación o gobierno a distancia de un dispositivo por un ser humano.

Valor RAW:

Dato bruto que entrega el sensor antes de ser procesado para obtener una medición de una variable física.

Índice

Resumen.....	i
Abstract	ii
Glosario	iii
Índice.....	vi
Índice de Figuras	xi
1 Marco Teórico.....	1
1.1 Introducción	1
1.2 Objetivos	2
1.2.1 Objetivo General	2
1.2.2 Objetivos Específicos	2
1.3 La mano humana y sus partes	3
1.4 Movimientos en la mano humana	5
1.4.1 Muñeca.....	5
1.4.2 Dedos.....	5

1.4.3	Rango de movimientos mano humana	7
1.5	Mano Robótica	8
1.5.1	Descripción General Mano Robótica	8
1.5.2	Opciones de Mano Robótica para control	9
1.6	Tipos de control.....	16
1.6.1	Control a lazo abierto	16
1.6.2	Control a lazo cerrado	16
1.6.3	Control PID	17
2	Implementación.....	23
2.1	Trabajo Previo con prototipo de Mano Mecánica	23
2.2	Selección de Mano Robótica.....	25
2.3	Arquitectura del Sistema	29
2.4	Sistema de Captación Movimientos	31
2.4.1	Sensores.....	31
2.4.2	Diseño Porta Sensores y prototipado.....	34

2.5	Procesamiento de datos y Control	40
2.5.1	Estrategia de control.....	40
2.5.2	Esquema de conexión.....	41
2.5.3	Estimación de la orientación de sensor IMU.....	43
2.5.4	Algoritmo de control actuadores	48
2.6	Validación de dispositivo de adquisición de datos.....	50
2.7	Experimentos Realizados	53
3	Resultados	56
3.1	Sintonización constantes PID dado un valor de referencia	56
3.1.1	Sintonización control proporcional (P) dedo índice.....	56
3.1.2	Sintonización control PI (Proporcional - Integral) dedo índice.....	57
3.1.3	Sintonización control PID (Proporcional-Integral-Derivativo) dedo índice	58
3.1.4	Sintonización constantes PID para Muñeca	60
3.2	Respuesta PID frente a referencia variable	62
3.3	Control proporcional entre posición de Servomotor y ángulo de dedo.....	65

4	Conclusiones	68
5	Trabajo a futuro.....	70
6	Referencias.....	74
	Anexos.....	80
A.	Descripción Prototipo de Mano Mecánica	80
B.	Diagrama de conexión para mano robótica completa	84
C.	Definición sensores micro electro-mecánicos	85
D.	Definición Acelerómetro.....	86
E.	Giroscopios	89
F.	Cinemática de un robot	93
G.	Hardware Utilizado	98
H.	Códigos de Programación	102
I.	Cálculo deformación cable.....	120
J.	Rango de captación de sensores	122

Índice de Figuras

Figura 1.1: Representación de las partes de la mano humana con los huesos que la conforman [1] .. 3

Figura 1.2: Articulaciones de la mano, incluyendo las de la muñeca. En la leyenda de colores se encuentra la abreviatura en inglés de cada articulación, las que corresponden a DIP: articulación interfalángica distal, PIP: articulación interfalángica proximal, MCP: articulación metacarpofalángica, CMC: articulación carpometacarpiana, TMC: articulación trapecio-metacarpiana, Radiocarpal Joint: Articulación radiocarpiana, Midcarpal Joint: articulación mediocarpiana. [2]..... 4

Figura 1.3: Movimientos que puede realizar la muñeca desde la posición anatómica [3] 5

Figura 1.4: Movimientos de los dedos Índice, Medio, Anular y Meñique en flexión-extensión, y en aducción-abducción. [4] 6

Figura 1.5: Movimientos que realiza el pulgar, en donde se añade el movimiento de oposición que puede realizar al igual que el dedo meñique. [5]..... 6

Figura 1.6: Prototipo Mano Robótica desarrollada por Jorge Salas como memoria del departamento de ingeniería mecánica de la Universidad Técnica Federico Santa María [6] 9

Figura 1.7: Dimensiones generales en milímetros de Mano de prototipo de Mano Mecánica desarrollada por [6] 10

Figura 1.8: Vista frontal y posterior de la mano robótica sobre la que se hace control 11

Figura 1.9: En el lado izquierdo vista del mecanismo de los dedos la mano robótica, al lado derecho vista lateral dedo, marcado en verde se muestran los canales por donde pasan los cables de nylon. El

cable que pasa por el lado de flexión va desde el servomotor hasta la falange medial, mientras que el del lado de extensión une la falange distal con la proximal	12
Figura 1.10: Ensamble mano robótica con servomotor para movimiento de flexión-extensión de la muñeca	13
Figura 1.11: Dimensiones generales mano robótica Hackerhand en milímetros (mm) [9].....	14
Figura 1.12: Representación en forma de diagrama de bloques de un sistema de lazo cerrado versus uno de lazo abierto. [10].....	17
Figura 1.13: Esquema control PID [11]	18
Figura 1.14: Gráfico de la posición angular de un servomotor al aplicar un control proporcional. En el gráfico se muestra la respuesta que tiene el servo con distintos valores de la constante kp [12].	19
Figura 1.15: Gráfico en donde se muestra el comportamiento oscilatorio de la señal de salida y los errores que se obtienen en un sistema al aplicar control PID [13]	20
Figura 1.16: Gráfico de control integral aplicado a la posición angular de un servomotor con distintos valores de constantes integrales [14].....	21
Figura 2.1: Simulación de la longitud de cable en el lado de flexión y en el de extensión para el dedo índice cuando falange proximal y medial están en 90° desde la posición anatómica	23
Figura 2.2: Componentes impresos del prototipo de mano mecánica desarrollada en el trabajo de tesis anterior por [6]	27
Figura 2.3: Arquitectura general del sistema en donde se muestran las 3 partes del sistema	29

Figura 2.4: Orientación de los ejes y sentido de rotación del sensor MPU9250 [15]	33
Figura 2.5: Concepto de dispositivo en el cual los sensores van adheridos en la tela de un guante que cubre la totalidad de la mano.....	35
Figura 2.6: Concepto de montaje de sensores en anillos y guante sin dedos	35
Figura 2.7: Prototipo físico de anillos y porta-sensores para falanges	36
Figura 2.8: Prototipo de anillo en dedo anular.	37
Figura 2.9: Vista frontal y superior de porta sensor en 90° y horizontal.....	38
Figura 2.10: Prototipo de guante con sensor montado en el dorso de la palma.	39
Figura 2.11: Diagrama control mano robótica	40
Figura 2.12: Diagrama de conexión para control de 1 dedo de la mano robótica	42
Figura 2.13: Representación de los ángulos de Euler en donde el sistema cartesiano en azul corresponde al sistema fijo y el rojo al móvil. [16].....	43
Figura 2.14: Representación de los ángulos de orientación de un cuerpo según la notación de Tait-Bryan [17]	44
Figura 2.15: Diagrama de descomposición de la fuerza de gravedad cuando el sensor está inclinado [18]	44

Figura 2.16: Gráfico del ángulo en función del tiempo obtenido por un giroscopio (azul), un acelerómetro (verde) y el valor con la combinación de ambos a través del filtro complementario (rojo). Nótese la alta variabilidad que presentan los datos del acelerómetro, mientras que los datos del giroscopio presentan un error de repetibilidad al final del gráfico [19]	47
Figura 2.17: Diagrama de flujo del algoritmo de control.....	48
Figura 2.18: Configuración Cinemática de la mano humana con la definición de la orientación del sistema de coordenadas para cada una de las uniones.....	51
Figura 2.19: Distintas posiciones del sistema de captación de movimientos instalado en las falanges del dedo índice para hacer validación de este por medio de un método gráfico utilizando cinemática directa.....	52
Figura 2.20: Posición inicial (izquierda) y final (derecha) a la que debe llegar dedo índice cuando se realizan los experimentos para encontrar valores de constantes PID frente a una referencia constante ($\theta = 50^\circ$).....	54
Figura 2.21: Posición inicial (izquierda) y final (derecha) a la que debe llegar la mano cuando se realizan los experimentos para encontrar valores de las constantes PID del movimiento de muñeca frente a una referencia constante ($\theta = 20^\circ$)	54
Figura 3.1: Tiempo de respuesta del sistema con control Proporcional para distintos valores de constante k_p , implementado en dedo índice con una señal de referencia $\theta = 50^\circ$	57
Figura 3.2: Tiempo de respuesta del sistema con distintos valores de constantes k_P y k_I en un control PI en un dedo para una señal de referencia $\theta = 50^\circ$	58

Figura 3.3: Tiempo de respuesta del sistema para distintas combinaciones de valores de constantes proporcional, integral, derivativa al seguir una referencia de $\theta = 50^\circ$ en el dedo medio de la mano robótica.....	59
Figura 3.4: Tiempo de respuesta de la muñeca con control proporcional al seguir una señal de referencia $\theta = 20^\circ$	60
Figura 3.5: Tiempo de respuesta de la muñeca con control PI , al seguir una referencia de $\theta = 20 [^\circ]$	61
Figura 3.6: Tiempo de respuesta de la muñeca con control PID al seguir una referencia de $\theta = 20^\circ$	61
Figura 3.7: Respuesta del dedo de la mano robótica frente a una referencia variable proveniente del sistema de captación de movimientos de la mano humana, con valor de constantes PID: $k_p = 0.013$, $k_I = 0,00015$, $k_D = 0,1$	62
Figura 3.8: Respuesta del dedo de la mano robótica frente a una referencia variable proveniente del sistema de captación de movimientos de la mano humana, con valor de constantes PID: $k_p = 0.013$, $k_I = 0,00015$, $k_D = 0,2$	63
Figura 3.9: Respuesta de la muñeca de la mano robótica frente a una referencia variable proveniente de la muñeca del sistema de captación de movimiento de la mano humana, con valor de constantes de PID: $k_p = 0.065$, $k_I = 0.000015$, $k_D = 0.0015$	64
Figura 3.10: Gráfico de Posición del servomotor en función del ángulo del dedo. En la esquina superior derecha se presenta la ecuación de la recta junto con el coeficiente de correlación entre ambas variables	65

Figura 3.11: Ángulo de la mano robótica al hacer seguimiento de la referencia variable dada por la mano humana, utilizando un control que relaciona la posición del servomotor y el ángulo del dedo de forma proporcional.....	67
Figura 5.1: Diferentes posiciones del dedo índice del prototipo de mano mecánica desarrollada en tesis anterior [6] para visualizar el rango de movimientos que tiene sus elementos.	71
Figura A.1: Prototipo Mano Robótica desarrollada por Jorge Salas como memoria del departamento de ingeniería mecánica de la Universidad Técnica Federico Santa María [6]	80
Figura A.2: Ensamble de las partes de la mano robótica sin los elementos de sujeción.....	81
Figura A.3: Vista explosionada del ensamble de los dedos de la mano robótica.....	82
Figura A.4: Vista isométrica del ensamble de la estructura que sostiene la muñeca con la mano....	83
Figura C.1: Resonador de superficie micro mecanizada desarrollado por la empresa MNX. Este dispositivo puede ser utilizado como un microsensor y como un microactuador [25]	85
Figura D.1: Esquema Acelerómetro mecánico [26]	86
Figura D.2: Esquema general de componentes de un acelerómetro basado en materiales piezoeléctricos.....	87
Figura D.3: Esquema de funcionamiento de un acelerómetro capacitivo. Este funcionamiento es el que se utiliza en sensores MEMS [27].....	88
Figura D.4: Esquema de funcionamiento de un acelerómetro capacitivo. Este funcionamiento es el que se utiliza en sensores MEMS [28].....	88

Figura E.1: Representación el mecanismo de un giroscopio rotativo [29].....	89
Figura E.2: Esquema de la estructura vibratoria que va sobre la plataforma que gira [30]	90
Figura E.3: Representación de la fuerza experimentada por la estructura vibratoria cuando gira la plataforma [30].....	90
Figura E.4: Diagrama explicativo efecto Sagnac en un circulo [30].....	91
Figura E.5: Diagrama funcionamiento giroscopio óptico [25].....	92
Figura F.1: Esquema del problema cinemático directo e inverso. Si se tiene el valor del movimiento que realizan los elementos del robot, se puede obtener la posición y orientación del extremo del robot (cinemática directa), al revés, si se cuenta con la información de la posición, se pueden obtener los movimientos que tiene que realizar cada uno de sus elementos para llegar a esa posición [31].....	93
Figura F.2: Representación de robot plano de 3 grados de libertad con los valores de movimiento en sus articulaciones [32].....	94
Figura G.1: Vista superior de la placa Arduino UNO con sus componentes [34]	98
Figura G.2: Sensor MPU9250 [35]	100
Figura G.3: Multiplexor I2C TCA9548A, distribuido por Adafruit [36].....	100
Figura G.4: Servomotor CYS S8218 utilizado para agregar movimiento flexión-extensión en la muñeca [37].....	101

Figura J.1: Representación del rango de movimiento angular que pueden captar los sensores en la muñeca y en una falange proximal cuando son instalados perpendicular a la superficie. La muñeca y la falange están representadas por un eslabón o link cada uno, los rectángulos morados corresponden a los sensores..... 122

Figura J.2: Representación del rango de medición angular si el sensor de la muñeca es instalado de forma perpendicular a la superficie, y el de la falange de forma horizontal. 123

Índice de tablas

Tabla 1.1: Magnitud de los movimientos de las diferentes partes de la mano, indicados en grados, a partir de la posición anatómica [6]	7
Tabla 2.2: Ángulos de movimiento mínimo y máximo de los dedos de la mano robótica HackerHand	15
Tabla 2.1: Tabla comparativa de manos robóticas disponibles	26
Tabla 3.2: Cuadro comparativo con las características de sensores disponibles en el mercado	32
Tabla 2.3: Valores Teóricos de lectura del acelerómetro y giroscopio del sensor MPU9250 en posición horizontal con eje Z+ apuntando hacia arriba.	34
Tabla 2.4: Parámetros de Denavit Hartenberg utilizados para realizar el esquema cinemático directo del dedo índice	51
Tabla G.1: Características placa Arduino UNO	99

1 Marco Teórico

1.1 Introducción

Las manos robóticas con un alto nivel de antropomorfismo son un objetivo deseable en el diseño de los efectores finales de un brazo robótico ya que permiten su uso como prótesis, operan en entornos orientados al hombre y también pueden ser teleoperadas o telecomandadas por humanos a través de interfaces que pueden reproducir el comportamiento de la mano del operador.

El presente trabajo de título se realizó con la finalidad de desarrollar un sistema de telecontrol para reproducir los movimientos de una mano humana, en una robótica. El dispositivo de captación de movimientos y posiciones se basará en sensores MEMS para la captura de la cinemática, información que permitirá controlar los actuadores de la mano robótica.

La idea surge del departamento de Ingeniería mecánica de la Universidad Técnica Federico Santa María para realizar una evaluación de los grados de libertad en las partes de la mano que son estrictamente necesarios para alcanzar posiciones básicas en la vida cotidiana, por medio de la emulación de movimientos en un prototipo de mano mecánica, este trabajo primero aborda un desarrollo de un trabajo de tesis anterior y luego un desarrollo en un producto validado comercialmente. El control presentado en este trabajo se desarrolla sobre una mano robótica que solo puede hacer movimientos de extensión-flexión en sus dedos y en la muñeca, pero se entregan los lineamientos para extender el control a los demás movimientos de la mano.

1.2 Objetivos

1.2.1 Objetivo General

Desarrollar un sistema para reproducir los movimientos de una mano humana, en una robótica. El dispositivo se basará en sensores MEMS para la captura de la cinemática, información que permitirá controlar los actuadores de la mano robótica.

1.2.2 Objetivos Específicos

- Identificar los movimientos y su rango en una mano humana, y analizar la cadena cinemática de la mano robótica de acuerdo con su diseño y construcción.
- Implementar sistema de adquisición de datos de movimiento en un dedo y la muñeca de una mano humana.
- Realizar una rutina de calibración entre el dispositivo de captación y la mano robótica.
- Definir e implementar un algoritmo para el control de actuadores de la mano robótica en base a la información recopilada por el dispositivo que adquiere la posición en un dedo y la muñeca.
- Establecer los lineamientos para extender el control a todos los grados de libertad de la mano robótica.

1.3 La mano humana y sus partes

La mano es una de las principales extremidades del cuerpo humano debido a que permite el agarre y manipulación de objetos. Las manos están localizadas en los extremos de los antebrazos y comprenden desde la muñeca hasta la punta de los dedos. Se pueden reconocer tres grandes secciones en la mano humana (**Figura 1.1**)

- **Muñeca:** Formada por 8 huesos denominados Carpo, es la articulación que une la mano con el antebrazo.
- **Palma-Dorso:** Compuesta por 5 huesos que forman el Metacarpo, es la parte central de la mano. Se denomina palma a la zona inferior y dorso a la superior.
- **Dedos:** 5 dedos constituidos por 14 huesos denominados falanges

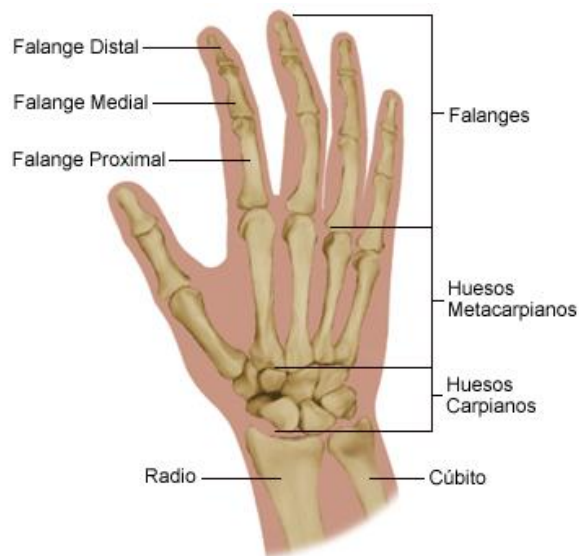


Figura 1.1: Representación de las partes de la mano humana con los huesos que la conforman [1]

Entre los distintos huesos se encuentran las articulaciones las que permiten la unión entre los huesos o entre un hueso y un cartílago. Además de facilitar la unión de los elementos, también otorgan

elasticidad y plasticidad entre ellos para que al menos uno de los elementos tenga movilidad. En la **Figura 1.2** se muestran las distintas articulaciones que existen en la mano

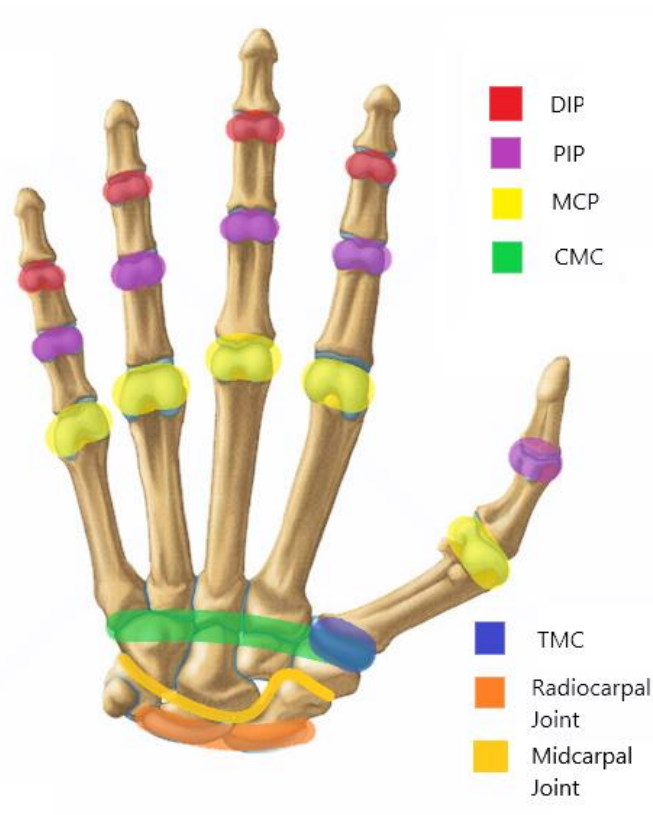


Figura 1.2: Articulaciones de la mano, incluyendo las de la muñeca. En la leyenda de colores se encuentra la abreviatura en inglés de cada articulación, las que corresponden a DIP: articulación interfalángica distal, PIP: articulación interfalángica proximal, MCP: articulación metacarpofalángica, CMC: articulación carpometacarpiana, TMC: articulación trapecio-metacarpiana, Radiocarpal Joint: Articulación radiocarpiana, Midcarpal Joint: articulación mediocarpiana. [2]

1.4 Movimientos en la mano humana

1.4.1 Muñeca

La muñeca puede realizar movimientos de abducción, aducción, flexión y extensión (**Figura 1.3**). Estos movimientos se realizan en los huesos del carpo de la muñeca, en las articulaciones radiocarpianas, mediocarpianas y carpometacarpianas. [3]

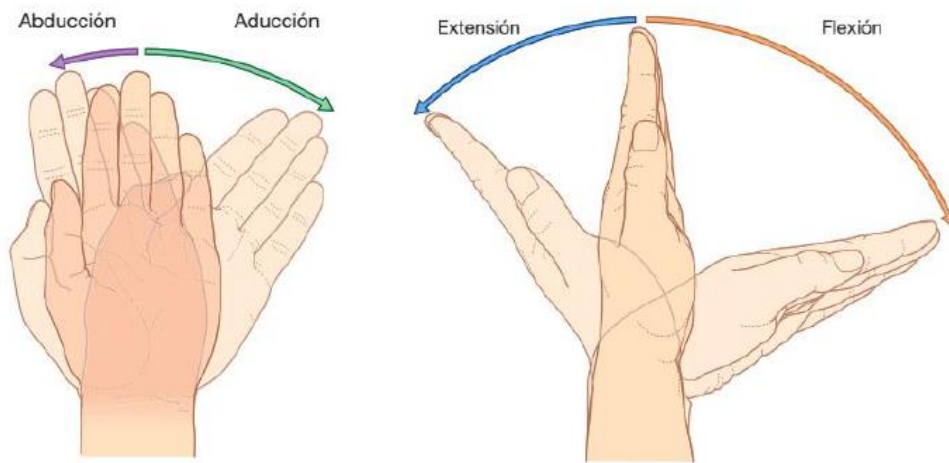


Figura 1.3: Movimientos que puede realizar la muñeca desde la posición anatómica [3]

El movimiento de rotación de la mano no se considera dentro de los movimientos de la muñeca debido a que es generado en el antebrazo por la rotación de esta parte.

1.4.2 Dedos

Respecto a los dedos, estos también pueden realizar los movimientos de flexión-extensión (**Figura 1.4**) en todas sus falanges gracias a las articulaciones interfalángicas (DIP, PIP, MCP) pero es en la articulación metacarpofalángica (MCP) en donde se produce el movimiento de aducción-abducción

de los dedos además de una rotación limitada, lo anterior es posible gracias a la forma de la articulación, que es del tipo esferoide.

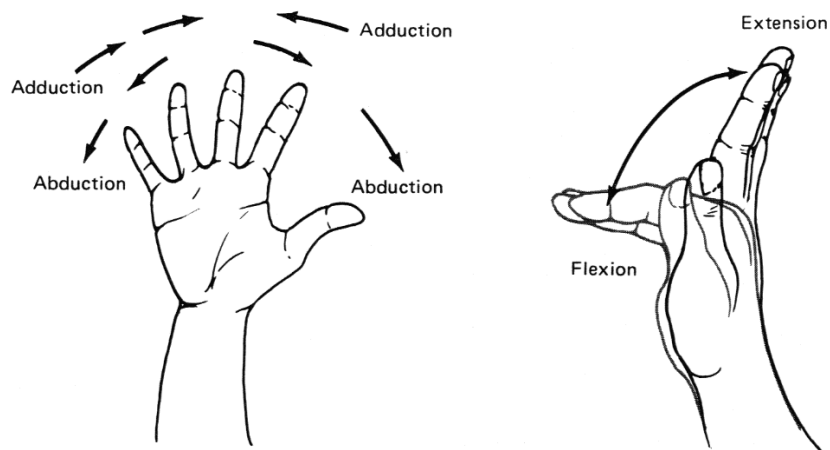


Figura 1.4: Movimientos de los dedos Índice, Medio, Anular y Meñique en flexión-extensión, y en aducción-abducción. [4]

En el caso del pulgar, si bien existen los movimientos de flexión-extensión, y de aducción-abducción, estos se definen en planos distintos a los de los demás dedos. En la **Figura 1.5** se muestra cómo se definen estos movimientos para el pulgar, nótese que se agrega un tercer par de movimientos que es el de oposición-reposición que también es efectuado por el dedo meñique.

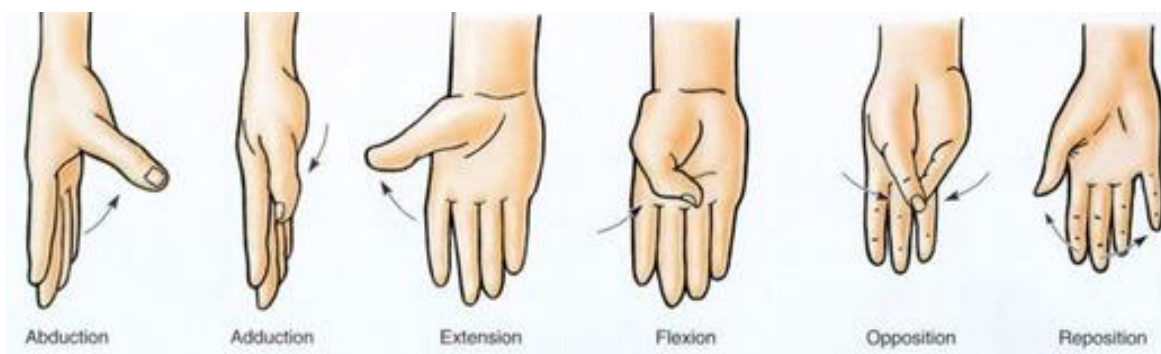


Figura 1.5: Movimientos que realiza el pulgar, en donde se añade el movimiento de oposición que puede realizar al igual que el dedo meñique. [5]

Como se aprecia en la **Figura 1.2**, el pulgar tiene 2 falanges y también tiene una articulación que los demás dedos no tienen (TMC), esta articulación es la que entrega el grado de libertad extra para hacer el movimiento de oposición.

1.4.3 Rango de movimientos mano humana

El rango de movimientos de la mano humana fue presentado en una tesis anterior [6], en donde se plantearon los rangos de movimientos como especificaciones de desempeño para el desarrollo del prototipo, en la **Tabla 1.1** se muestra el rango con magnitud en grados (°).

Tabla 1.1: Magnitud de los movimientos de las diferentes partes de la mano, indicados en grados, a partir de la posición anatómica [6]

Elemento a mover	Tipo de movimiento	Magnitud [°]
Muñeca	Abducción Radial	15
	Abducción Cubital	45
	Flexión	85
	Extensión	85
MCP (Articulación metacarpofalángica dedos)	Flexión	90
PIP (Articulación interfalángica proximal de los dedos)	Flexión	130
DIP (Articulación interfalángica distal de los dedos)	Flexión	90
MCP Pulgar (Articulación metacarpofalángica del pulgar)	Flexión	90
TMC (Articulación trapecio- metacarpiana Pulgar)	Proyección Medial	40
	Proyección Lateral	60
	Proyección Anterior	70

1.5 Mano Robótica

1.5.1 Descripción General Mano Robótica

Dentro del universo de la robótica debemos contextualizar a la “mano robótica” como un “Efector Final” de un brazo, ya sea robótico o no, o simplemente se prescindir de él. Este efector final está enmarcado en dos corrientes de desarrollo, uno enfocado en hacer tareas específicas, y otro que busca replicar la funcionalidad y/o semejanza con una mano humana.

Según [7] los dos puntos de vista en los cuales se enfoca el desarrollo son:

1. Hacer una mano antropomorfa (enfocado en la semejanza al humano)
2. Hacer un manipulador eficiente (enfoque minimalista centrado en la función)

El presente trabajo va enfocado en el desarrollo antropomorfo, con el propósito de imitar los movimientos de la mano humana, esto se denomina como “mano hábil o mano diestra”, de ahora en adelante “dextrous hands” como es denominada en inglés ya que la traducción no es directa y como concepto es mucho más complejo.

Dentro de las líneas de desarrollo, este trabajo se enfoca en la “Operación y Control” de la mano robótica, en particular en una guiada por una mano humana, según [8] las alternativas son a través de “enseñanza” (replicar movimientos grabados) o tele manipulación (replicar movimientos en tiempo real), en este caso se enfoca en la tele manipulación.

1.5.2 Opciones de Mano Robótica para control

1.5.2.1 Descripción Prototipo de Mano Mecánica

El prototipo de mano mecánica fue desarrollado en el contexto de una tesis anterior, este prototipo consta de una estructura de 8 bases de madera situadas de forma vertical, una sobre la otra como se muestra en la **Figura 1.6**. En la parte superior de esta estructura se ubica la mano robótica que emula la mano humana. En cada base se disponen de forma circular servomotores que tensionan hilos de pescar, simulando tendones. En total son 47 servomotores que a través de la tensión sobre los tendones permiten los movimientos en cada una de las partes de la mano a la que están conectadas.

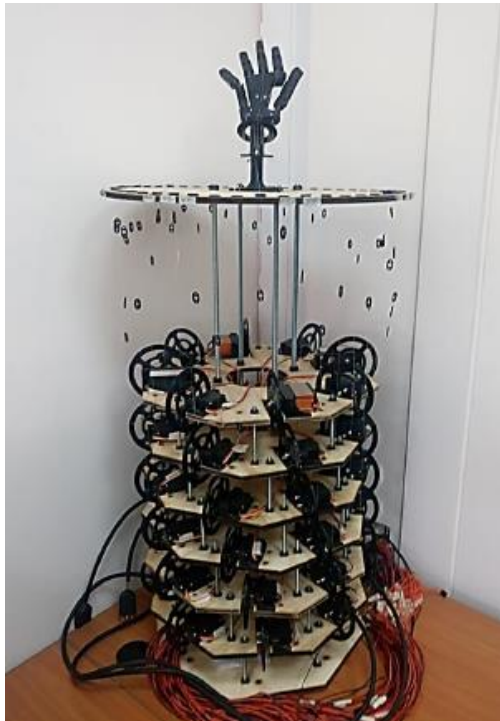


Figura 1.6: Prototipo Mano Robótica desarrollada por Jorge Salas como memoria del departamento de ingeniería mecánica de la Universidad Técnica Federico Santa María [6]

Las dimensiones generales del prototipo son 120 [cm] de alto por 40 [cm] de ancho, en la **Figura 1.7** se muestran las dimensiones de la mano ubicada en la parte superior del prototipo

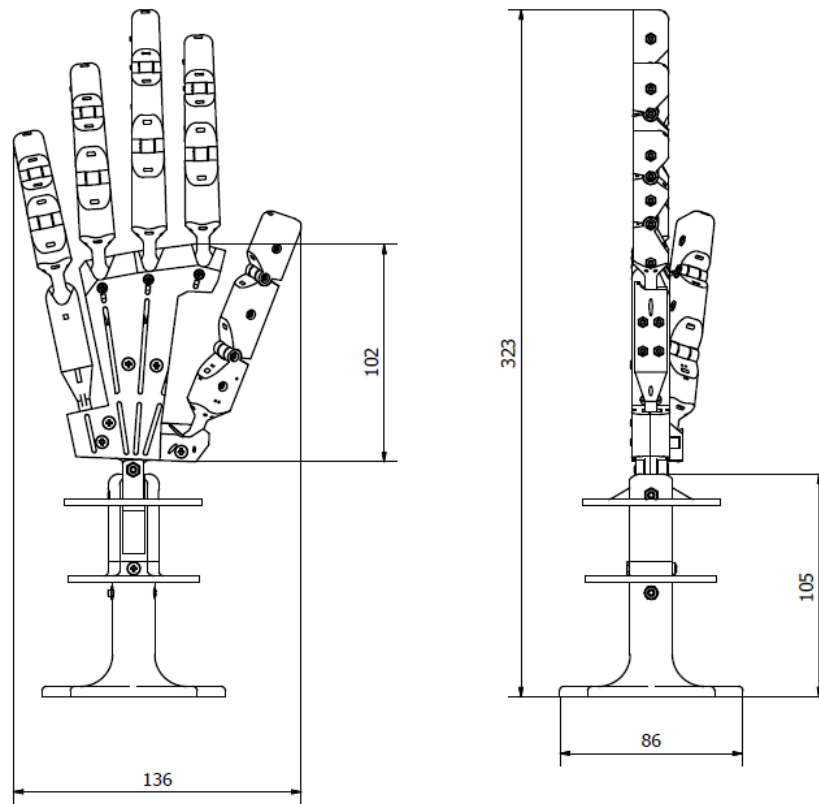


Figura 1.7: Dimensiones generales en milímetros de Mano de prototipo de Mano Mecánica desarrollada por [6] .

1.5.2.2 Descripción Mano Robótica HackerHand

El sistema de control se desarrolla sobre una mano robótica que está a la venta en el mercado, esta mano consta de 5 dedos que emulan una mano humana, cada dedo está compuesto de tres falanges, a excepción del pulgar que tiene dos. Estos dedos se encuentran unidos a una estructura rígida que emula la palma, donde además se encuentran los actuadores que permiten a los dedos realizar movimientos de flexión-extensión (**Figura 1.8**).

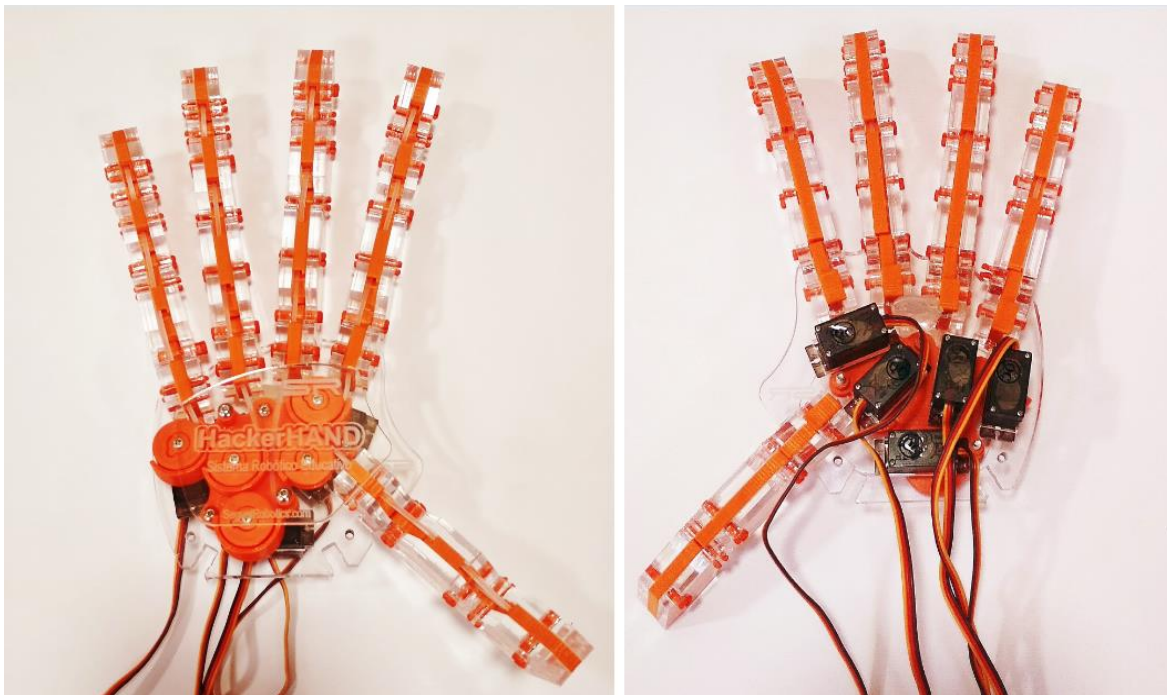


Figura 1.8: Vista frontal y posterior de la mano robótica sobre la que se hace control

Los movimientos de flexión son realizados a través de servomotores que tienen una polea conectada a su eje, al girar el eje en sentido horario, un cable de nylon semirrígido que se une a la falange medial se enrolla en la polea tirando de este punto generando la flexión. Por otro lado, la extensión se origina moviendo el servomotor en el sentido opuesto haciendo que el cable empuje la falange. Por el lado de extensión de la falange medial sale un cable de nylon que se conecta con la falange distal, este cable tiene como objetivo actuar como un resorte de tracción haciendo que las falanges vuelvan a su posición inicial (**Figura 1.9**).

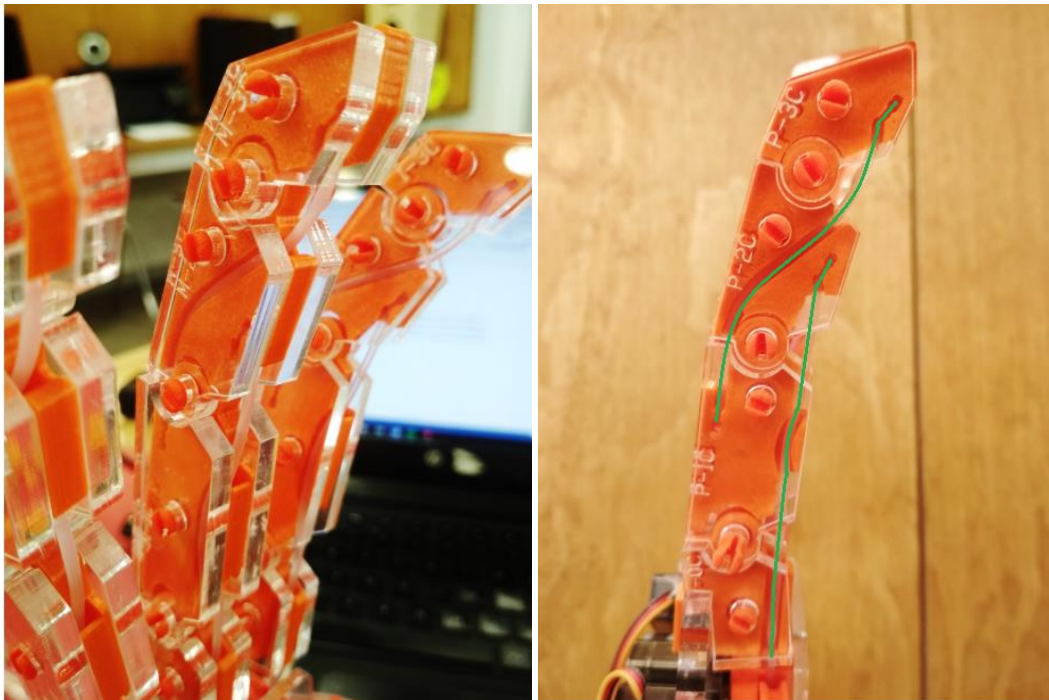


Figura 1.9: En el lado izquierdo vista del mecanismo de los dedos la mano robótica, al lado derecho vista lateral dedo, marcado en verde se muestran los canales por donde pasan los cables de nylon. El cable que pasa por el lado de flexión va desde el servomotor hasta la falange medial, mientras que el del lado de extensión une la falange distal con la proximal

Para agregar el movimiento de flexión-extensión de la muñeca se diseña una estructura que acopla la mano robótica al eje de un servomotor. Esta estructura es diseñada en el computador y fabricada a través de impresión 3D. (**Figura 1.10**)

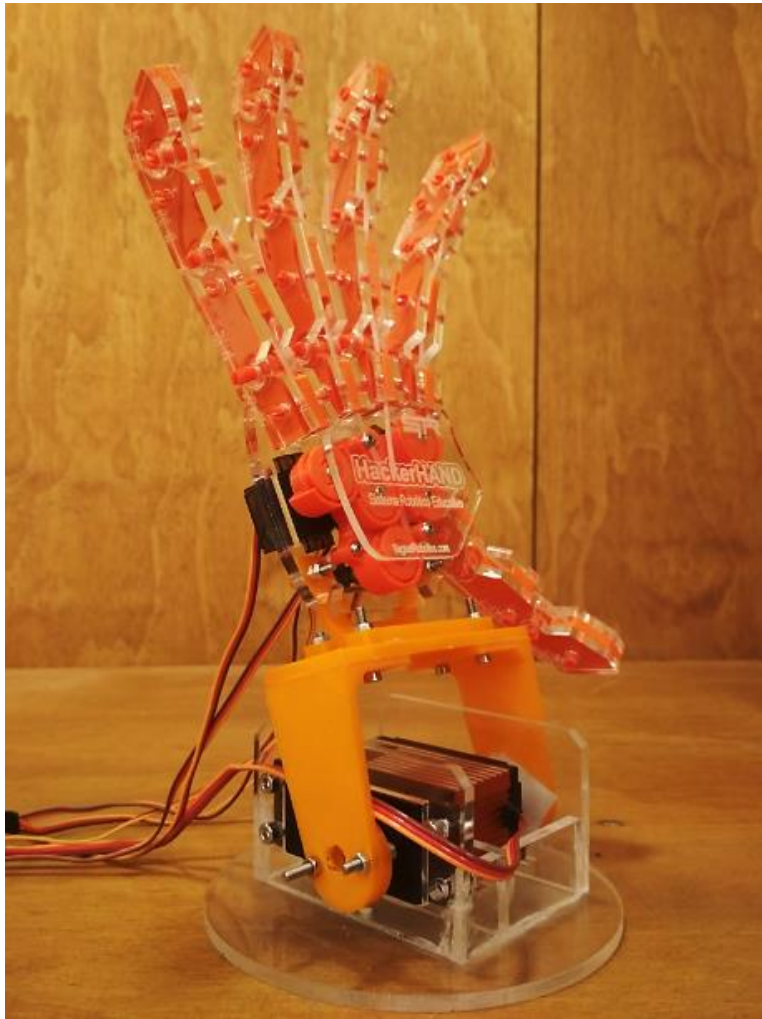


Figura 1.10: Ensamble mano robótica con servomotor para movimiento de flexión-extensión de la muñeca

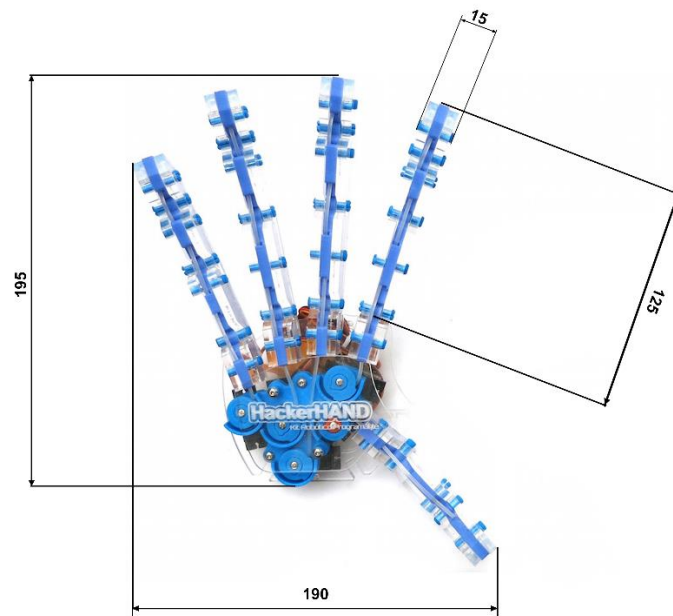


Figura 1.11: Dimensiones generales mano robótica Hackerhand en milímetros (mm) [9]

Rango de movimientos dedos mano robótica HackerHand

Para determinar el rango de movimientos de la mano robótica se utilizan sensores IMU MEMS, con ellos se obtiene el ángulo respecto de una línea vertical imaginaria respecto de la superficie para cada dedo, en la **Tabla 1.2** se muestran los rangos de movimiento:

Tabla 1.2: Ángulos de movimiento mínimo y máximo de los dedos de la mano robótica HackerHand

Dedo	Falange	Ángulo Mínimo [°]	Ángulo Máximo [°]
Índice	Proximal	18	64
	Medial	34	125
Medio	Proximal	2	62
	Medial	11	117
Anular	Proximal	4	63
	Medial	15	120
Meñique	Proximal	17	60
	Medial	19	125
Pulgar	Proximal	37	65

Cabe destacar que no se muestran los ángulos de movimiento de las falanges distales de los dedos debido a que esa falange no realiza el movimiento de flexión accionada directamente por el actuador, por lo que la cantidad de grados que se mueve no depende de este, sino más bien, de lo que se mueva la falange medial.

1.6 Tipos de control

Se conoce como sistema a la unión de piezas electrónicas y/o mecánicas que trabajan en conjunto para obtener un resultado. Un sistema tiene componentes de entrada y salida, en donde se trabaja con los elementos de entrada para obtener una salida. Se puede tomar como ejemplo básico el de una licuadora, la entrada del sistema podrían ser frutas, estas son procesadas al interior de un sistema, y la salida sería la fruta molida en forma de jugo. En el sistema podemos regular la forma en que la fruta es molida modificando el parámetro de la velocidad con la que giran las cuchillas, de manera de tener siempre un mismo resultado. Los sistemas de control son aquellos que controlan la salida del sistema frente a perturbaciones de forma de obtener un resultado a la salida que sea constante, independiente de la perturbación recibida.

1.6.1 Control a lazo abierto

Un sistema de control a lazo abierto es aquel que corrige la salida tomando en cuenta solo las perturbaciones que percibe en la entrada y realizando una corrección que es fija, no se relaciona la señal de salida para hacer una retroalimentación (o feedback) con la señal de entrada para retroalimentar el sistema. (Figura 1.12)

1.6.2 Control a lazo cerrado

Los sistemas de lazo cerrado corrigen el error a través de la retroalimentación que reciben de la comparación entre la señal de entrada y la de salida, por lo que el factor de corrección se ajusta de forma instantánea dependiendo de la diferencia entre la señal de entrada y la de salida para minimizar este error. (Figura 1.12)

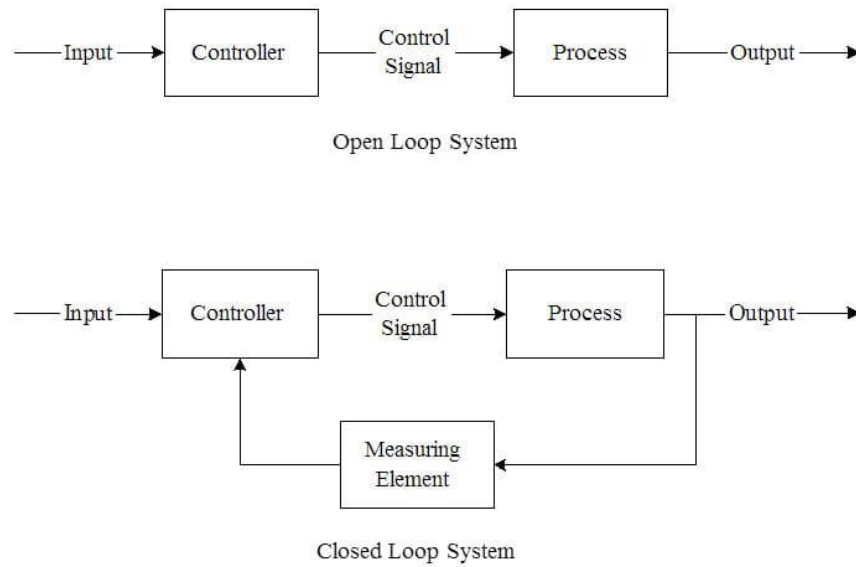


Figura 1.12: Representación en forma de diagrama de bloques de un sistema de lazo cerrado versus uno de lazo abierto. [10]

1.6.3 Control PID

Un control PID es un algoritmo de control a lazo cerrado en donde a través de la retroalimentación se configuran 3 tipos de variables que corrigen la señal del sistema hasta obtener la salida deseada. Estas 3 variables de corrección varían de acuerdo al error existente entre la entrada y la salida. En la **Figura 1.13** se puede observar un esquema de control PID:

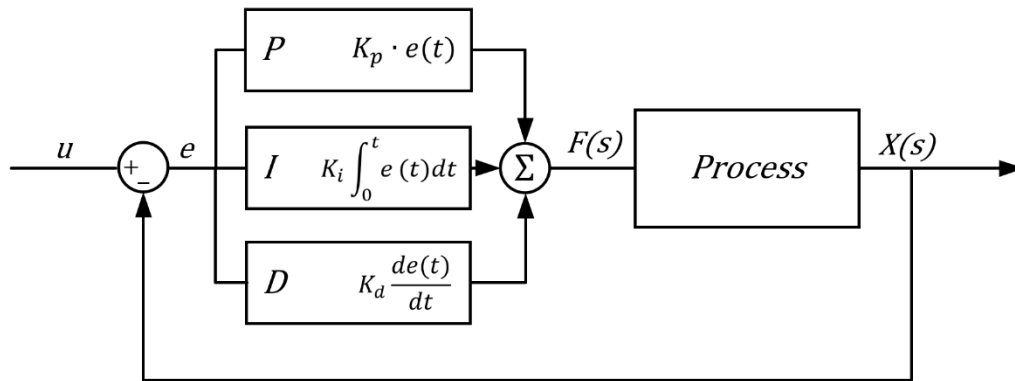


Figura 1.13: Esquema control PID [11]

En el esquema se muestra la señal de referencia u (también llamado set point) que es comparada con la señal de salida, esta diferencia entrega el error e que es utilizado por el control proporcional, integral y derivativo. La suma de estas tres componentes da como resultado la señal de control $F(s)$ que es aplicada sobre el sistema o planta, obteniendo la señal de salida $X(s)$.

Este tipo de control es ampliamente utilizado en la industria, varias aplicaciones utilizan solo la componente proporcional, la parte proporcional con la integral (conocido como control PI) o una combinación de ellas, por ejemplo, el control proporcional derivativo (PD).

1.6.3.1 Control Proporcional

Es un tipo de control lineal en donde la corrección es proporcional al error entre el valor definido o set point y la salida. La corrección actúa tomando en cuenta el error actual por lo que se relaciona con el presente. La definición matemática de este tipo de control se escribe como:

$$u(t) = e(t) * k_p \quad (1)$$

En donde $u(t)$ es la señal de corrección proporcional, $e(t)$ es el error o la diferencia entre la señal de referencia y la señal de salida del sistema, y k_p es la constante de corrección proporcional. Existen varios métodos para poder encontrar el valor de esta constante pero generalmente se encuentra iterando. El valor que se le asigne a k_p es importante ya que influye en que tan rápido se llegue al valor de referencia, en la **Figura 1.14** se observa cómo cambia la curva. Con un valor alto de k_p la señal de salida al comienzo genera un gran error que se atenúa en el tiempo, en cambio con un pequeño valor la señal demora en llegar a la referencia.

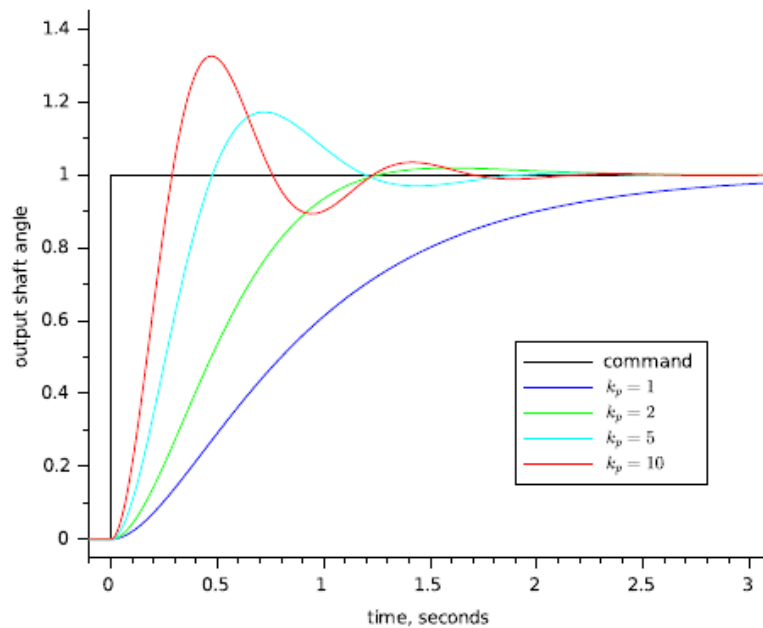


Figura 1.14: Gráfico de la posición angular de un servomotor al aplicar un control proporcional. En el gráfico se muestra la respuesta que tiene el servo con distintos valores de la constante k_p [12]

1.6.3.2 Control Integral

La acción integral permite eliminar los errores de offset que se producen cuando se regula el sistema, estos errores conocidos como overshoot pueden llevar a una oscilación en las señales de salida que se atenúan a medida que pasa el tiempo, hasta que el sistema llegue a la salida deseada como se muestra en la **Figura 1.15**.

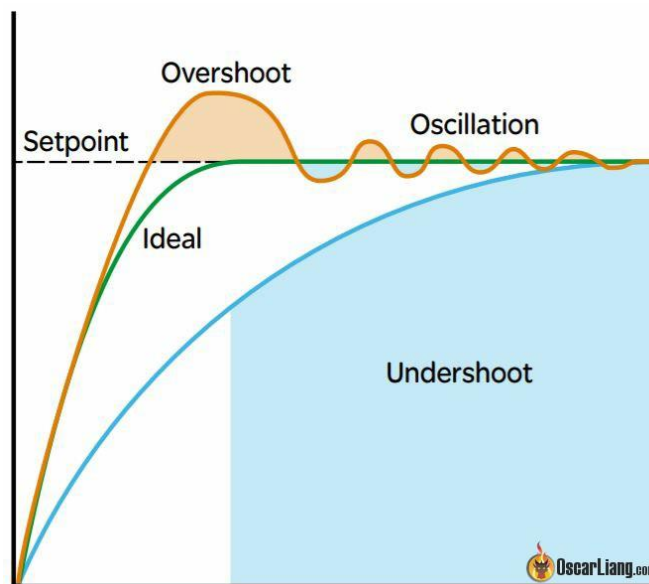


Figura 1.15: Gráfico en donde se muestra el comportamiento oscilatorio de la señal de salida y los errores que se obtienen en un sistema al aplicar control PID [13]

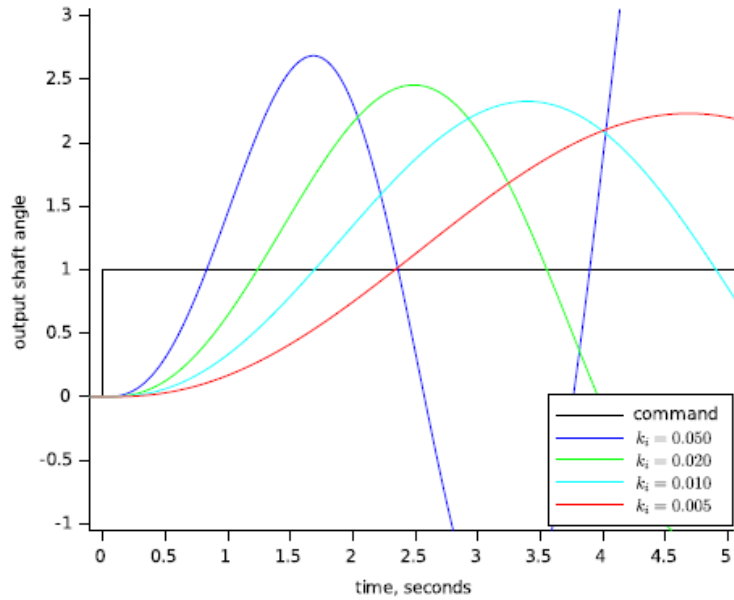


Figura 1.16: Gráfico de control integral aplicado a la posición angular de un servomotor con distintos valores de constantes integrales [14]

El control integral a diferencia del control proporcional que solo depende del error en el instante actual considera el historial de errores en el pasado, es decir, depende del error acumulado en el tiempo, disminuyendo el offset generado en primera instancia por el control proporcional y disminuyendo el tiempo que se demora en llegar al valor esperado

$$u(t) = k_i \cdot \left(e(t) \cdot t + \sum e(t) \right) \quad (2)$$

Si la acción proporcional (P) le dice a la salida cuando desplazarse cuando un error aparece, la acción integral (I) le dice a la salida que tan rápido moverse cuando un error aparece.

1.6.3.3 Control Derivativo

El control derivativo es proporcional a la tasa de cambio del error, permite aplicar una corrección dependiendo de la velocidad a la que se genera ese error, por lo que se anticipa a él. Este control regula los overshoots ocasionados por el control proporcional e integral. Se calcula de la forma:

$$u(t) = k_d \cdot \frac{(e_t - e_{t-1})}{\Delta t} \quad (3)$$

Donde e_t es el error en el instante actual, e_{t-1} es el error anterior, Δt es la diferencia de tiempo, y k_d es la constante derivativa.

2 Implementación

2.1 Trabajo Previo con prototipo de Mano Mecánica

De manera de continuar el trabajo desarrollado por [6] es que se intenta realizar el control sobre el prototipo de Mano Mecánica. Fueron propuestos diversos métodos para el control entre los que se encuentran:

- **Control por medio de longitud de Tendones:** Esta propuesta consistía en hacer un cálculo de la longitud de cada uno de los tendones del dedo índice para una posición determinada a través de un código en Matlab (**Figura 2.1**), luego la diferencia de longitud de cada cable de tendón con su posición original era llevada a una cantidad de grados que debía moverse el servomotor respectivo. Esta idea fue descartada debido a la deformación descontrolada del cable del tendón que produce pérdida de precisión excesiva en la posición final. (**Cálculo deformación cable**)

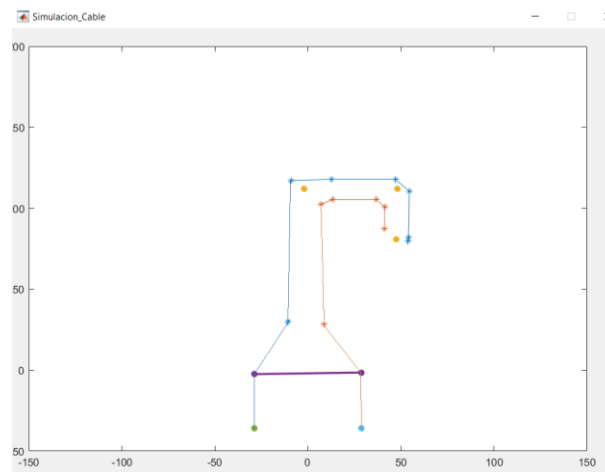



Figura 2.1: Simulación de la longitud de cable en el lado de flexión y en el de extensión para el dedo índice cuando falange proximal y medial están en 90° desde la posición anatómica

- **Control a lazo cerrado con corrección a valor constante:** Para implementar el lazo cerrado se colocaron sensores en el prototipo de mano mecánica para medir la posición angular de la muñeca y de la falange proximal, de esta forma se compara la posición con su homólogo en la mano humana por medio del sistema de captación de movimientos, en caso de existir diferencias estas serían corregidas mediante una instrucción para que el actuador se moviera $1 [^\circ]$, este ciclo se repite hasta que la diferencia entre las posiciones angulares fuese menor a $2 [^\circ]$. Al aplicar esta estrategia de control la variabilidad de los movimientos es mayor además de demorar bastante tiempo en llegar a la posición deseada.
- **Control PID:** Como en la estrategia de control anterior se obtuvo una alta variabilidad de la posición de las partes del prototipo antes de llegar a la posición final deseada, es que se opta por usar un control con base en la cantidad de error. Para llevarlo a cabo fue utilizada una configuración similar a la mencionada anteriormente en donde se agregan sensores al prototipo de mano mecánica para conocer su posición. Después de la búsqueda de las constantes proporcional e integral frente a una señal de referencia constante, fue observado el comportamiento del prototipo cuando sigue los movimientos de la mano humana, los resultados al igual que en el intento anterior fueron movimientos oscilatorios y con alta variabilidad.

2.2 Selección de Mano Robótica

Considerando que el prototipo de mano mecánica [6] no cumple con los objetivos funcionales, es realizado un análisis de otra alternativa de mano robótica que se encuentra a la venta en el mercado para realizar el control, a continuación, se muestra una tabla (Tabla 2.1) en donde se comparan las características de esta mano robótica con el prototipo de mano mecánica:

Tabla 2.1:Tabla comparativa de manos robóticas disponibles

		
Característica	Hacker Hand	Prototipo Mano Mecánica
Cuatro Dedos Superiores	✓	✓
Quinto Dedo	✓	✓
Pulgar Opuesto	✗	✓
Movimiento Muñeca	✗	✓
N° de Links (Eslabones)	14	16
N° de Uniones (Joints)	15	17
N° de DoF Controlados	5	24
Suavidad y Continuidad de los movimientos	Buena	Deficiente
Disposición de Actuador	Remota	Remota
Tipo de Actuador	Servo Motor	Servo Motor
Tipo de Transmisión	Tendón	Tendón

Ambas manos robóticas tienen 5 dedos, la cantidad de eslabones al igual que la cantidad de uniones es similar entre ellos, la diferencia entre ambas son los DoF presentes en estas uniones en donde la mano robótica HackerHand tiene solo 5 DoF mientras que el prototipo de mano mecánica tiene 24

DoF. La disposición que tienen los actuadores de las dos manos es de forma remota, esto es, los actuadores no se encuentran directamente dentro del dedo en las uniones sino fuera de este ya que utilizan una transmisión que emula los tendones a través de poleas y cables. Uno de los puntos importantes es que tan suaves y continuos son los movimientos, la Hackerhand tiene menos grados de libertad, pero puede realizar movimientos limpios a diferencia del prototipo de mano mecánica.

El prototipo de mano mecánica presenta los siguientes aspectos a mejorar:

- Proceso de manufactura en impresión 3D: Piezas presentan fallas en su calidad superficial (**Figura 2.2**) y estructural, capas se desprenden ante un esfuerzo mecánico, producto de ello las piezas presentan alto desgaste.



Figura 2.2: Componentes impresos del prototipo de mano mecánica desarrollada en el trabajo de tesis anterior por [6]

- Dificultad en el ensamble y montaje de componentes: La gran cantidad de componentes que tiene el prototipo hace que su ensamble sea complicado sumado a que la disposición de los elementos de la base no permite modificación si presentan problemas y se desea

cambiar uno de ellos, esto se visibiliza en los cables de los servomotores ya que cuesta acceder a ellos.

- Rigidez de la estructura: Los ejes roscados utilizados como columnas de soporte de cada base, requieren de un conjunto de Tuercas, golillas planas y golillas de presión para que la estructura no se desarme en el tiempo. Estos elementos no están presentes en el prototipo. Otro punto importante es que la base superior necesita mayor rigidez, en sus bordes la tensión de los cables produce flexión en la base de madera superior.
- Sistema de actuadores: no funciona de forma óptima ya que varía la cantidad de fuerza a aplicar dependiendo de la posición de las piezas de la mano. Dentro de este punto también se encuentra la alta deformación de los cables utilizados como tendones, estos deben ser ajustados frecuentemente, también, falta definir una fase de calibración y operación inicial.
- Sistema de Tensión de cables de tendones no permite un ajuste preciso.

Por lo mencionado anteriormente no es posible asegurar estabilidad en el control del prototipo de mano mecánica, ya que es capaz de hacer movimientos, pero no de forma óptima pues funciona bajo parámetros acotados por lo que no ejecuta movimientos limpios, esto demuestra que en su diseño no fue pensado para tele manipulación. Por otro lado, la mano robótica HackerHand tiene una menor cantidad de movimientos que es capaz de realizar, pero ofrece mayor estabilidad en su funcionamiento, tiene mayor fiabilidad, vida útil, no necesita de una calibración y, otro punto importante, es que los movimientos que realiza tienen mayor repetibilidad. Debido a lo anterior es que se descarta la utilización del prototipo de mano mecánica para realizar el control y se selecciona a la mano robótica Hackerhand.

2.3 Arquitectura del Sistema

Considerando un Actuador Final Antropomorfo, y que este es un sistema de telecontrol para una dextros hand, se define una arquitectura que nos sea útil para desarrollar y testear el “Sistema Mano Robótica Telecomandado”. En la **Figura 2.3** se visualiza un sistema general que engloba al conjunto de sistemas que son el sistema de captación de movimiento (en este caso particular spatial tracking), el sistema de procesamiento y el sistema que contempla al dispositivo con los actuadores montados (en este caso mano robótica Hacker Hand).

Para la implementación del control sobre la mano robótica se reconocen los sistemas que van a interactuar entre ellos para obtener el movimiento deseado. En el diagrama de la **Figura 2.3** se visualiza un gran sistema que sirve como interfaz para la emulación de los movimientos de la mano humana en una mano robótica, está compuesto por 3 partes que son el sistema de captación de movimientos de la mano humana, la unidad de procesamiento y por último el sistema de la mano robótica.

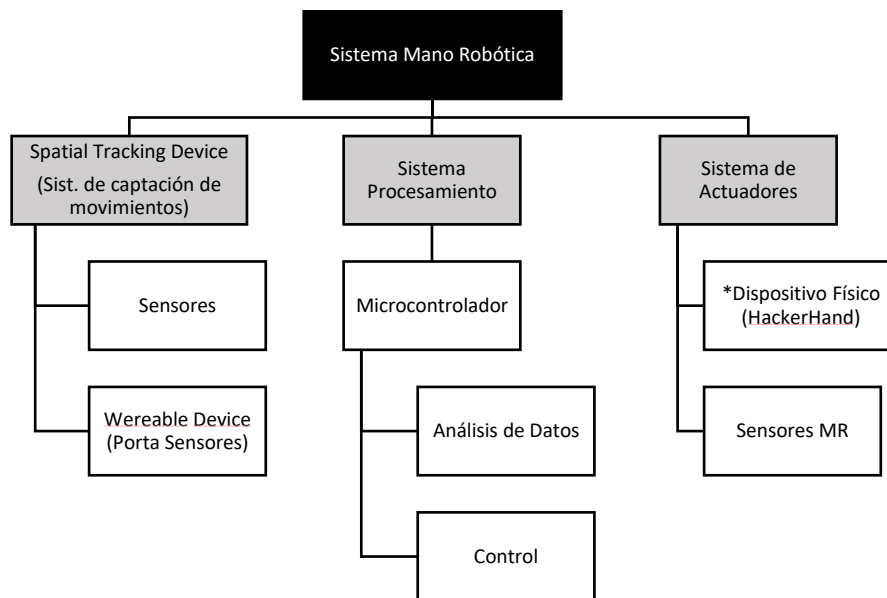


Figura 2.3: Arquitectura general del sistema en donde se muestran las 3 partes del sistema

Como se observa, el Spatial tracking device se organiza a través de dos elementos: el primero son los sensores encargados de captar las magnitudes físicas de los movimientos de la mano, y el segundo corresponde a un dispositivo que permite conectar los sensores con las distintas partes de la mano.

El procesamiento será realizado por un microcontrolador, en él se realiza el análisis de los datos de los sensores y además se hace el control de los actuadores de la mano robótica. La tercera parte del sistema corresponde a la mano robótica, en donde se encuentra el dispositivo físico de mano robótica y también se agregan sensores externos que sirven para conocer la posición de cada una de sus partes.

En el marco teórico fue detallado el tercer punto referente a la mano robótica, en los siguientes capítulos se explicará en mayor profundidad el sistema de captación de movimientos y la parte de procesamiento.

2.4 Sistema de Captación Movimientos

2.4.1 Sensores

Dentro del objetivo general es detallado la utilización de sensores MEMS (Anexo C) debido a que sus dimensiones son pequeñas. Dentro de la familia de sensores MEMS se encuentran acelerómetros y giroscopios, de 3 grados de libertad cada uno. Es común encontrar en el mercado sensores con giroscopios y acelerómetros que vienen en el mismo encapsulado, a estos se les conoce con el nombre de Inertial Measurement Unit (IMU). A continuación, se presenta un cuadro comparativo entre sensores disponibles en el mercado:

Tabla 2.2: Cuadro comparativo con las características de sensores disponibles en el mercado

Sensor	DoF	Protocolo Comunicación	Precio USD\$	Documentación en Internet	Dimensiones [mm]
Acelerómetro ADXL345	3	I2C	17.5	Alta	25x19x3,14
IMU MPU6050	6	I2C	16.95	Alta	21,2 x 16,4 x 3,3
IMU MPU9250	9	I2C/SPI	14.95	Alta	25,5 x 15,4 x 3
IMU BNO055 Bosch	9	I2C	43.18	Baja	20 x 27 x 4

Todos los sensores presentados en la tabla se comunican por protocolo I2C a excepción del MPU9250 en donde se puede seleccionar si la comunicación será por protocolo I2C o SPI. Esto entrega una ventaja frente a los otros sensores ya que la comunicación por protocolo SPI es más rápida que por I2C. Respecto a los grados de libertad, el acelerómetro ADXL345 es el que menos posee, los otros dispositivos pueden aumentar esta cantidad debido a que incorporan más sensores dentro del mismo encapsulado, como es el caso del IMU BNO055 de Bosch que incluye un giroscopio, acelerómetro y magnetómetro. Las dimensiones de los sensores son similares entre sí, por lo que no representan un factor decisivo a la hora de seleccionar uno. La documentación que hay en internet si es relevante, este ítem hace relación a la cantidad de información disponible en internet sobre todo en la resolución de problemas de comunicación con el sensor. En este sentido el sensor con menor documentación es el BNO055 de Bosch, los demás sensores tienen mayor respaldo de errores ya que son utilizados comúnmente en robótica.

Tomando en cuenta el análisis anterior es que se opta por utilizar el sensor MPU9250, su capacidad de utilizar un protocolo de comunicación I2C o SPI permite mejorar a futuro la rapidez del control sobre la mano robótica con el sistema de captación de movimientos, la incorporación de los 3 sensores (acelerómetro, giroscopio, magnetómetro) mejora el “Spatial Tracking” y, por último, su precio es menor a los otros sensores.

2.4.1.1 Rutina de calibración de sensores

Para obtener mediciones exactas por parte del sistema de captación de movimientos es necesario llevar a cabo la calibración de los sensores MEMS. Es importante que las lecturas de los sensores sean lo más parecidas entre ellas para evitar errores acumulativos, estas diferencias en sus lecturas se producen por diversos factores, entre ellos el ruido mismo de las señales del sensor, la posición en la que fue soldado, etc. En la **Figura 2.4** se muestra la orientación de los ejes del sensor; si este es situado sobre una superficie horizontal con el eje $z+$ apuntando hacia arriba, debiese obtener lecturas iguales a cero para los valores del acelerómetro y del giroscopio en todos sus ejes, a **excepción** de la lectura del acelerómetro en el eje z en donde debiese marcar un valor negativo ya que se encuentra afecto a la aceleración de gravedad ($g = 9.81 [m/s^2]$) y cuya magnitud depende de la configuración del rango de medición.

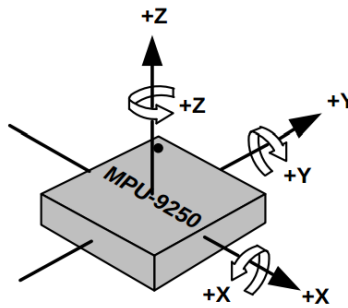


Figura 2.4: Orientación de los ejes y sentido de rotación del sensor MPU9250 [15]

Los sensores pueden ser configurados para distintos rangos de operación, a mayor rango de operación menor será la sensibilidad, para este caso ambos serán configurados para su mayor rango (2000 [dps] para el giroscopio y 16 [g] para el acelerómetro), de esta forma se obtiene una menor sensibilidad de parte del sensor para que disminuya la variabilidad de los datos. En base a esto, la lectura del sensor debiese ser cercana a estos valores: (ver Tabla 2.3)

Tabla 2.3: Valores Teóricos de lectura del acelerómetro y giroscopio del sensor MPU9250 en posición horizontal con eje Z+ apuntando hacia arriba.

Dato	Valor RAW [-]
a_x	0
a_y	0
a_z	-2048
g_x	0
g_y	0
g_z	0

Para la calibración se coloca el sensor en posición horizontal quieto, sin perturbaciones, apuntando hacia arriba el eje Z+ , luego se procede a tomar una muestra de 1000 datos del acelerómetro y del giroscopio con ellos se calcula el promedio para cada dato y se compara con los datos teóricos (**Tabla 2.3**) dando como resultado los offsets de cada sensor.

2.4.2 Diseño Porta Sensores y prototipado

El sistema de captación de movimientos se desarrolla en base a los sensores MPU9250, cada uno de ellos debe permitir la captación de datos en las distintas partes de la mano, en particular, movimientos de las falanges y de la muñeca

El diseño del sistema considera que cada sensor esté ubicado en una parte de la mano y se selecciona que esté por la parte dorsal debido a que los componentes tienen más espacio libre y de esta manera no limitan los movimientos. La primera propuesta de diseño considera el uso de un guante que contiene los sensores **Figura 2.5**.

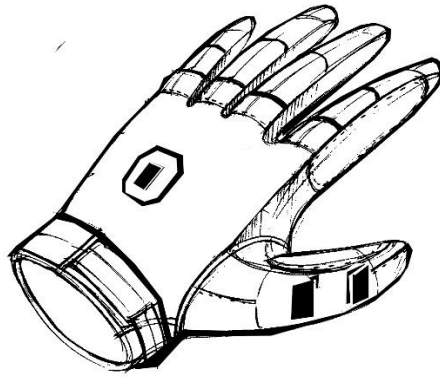


Figura 2.5: Concepto de dispositivo en el cual los sensores van adheridos en la tela de un guante que cubre la totalidad de la mano.

El problema de esta propuesta es que no permite que se adapte a las distintas fisonomías de mano humana, el guante es de un tamaño único y la posición de los sensores no se puede cambiar. La segunda propuesta es que cada uno de los sensores sea instalado de forma independiente a través de una pieza que sea colocada en cada una de las falanges de los dedos.

Este diseño se adapta bien a las falanges de los dedos, pero en la zona de la palma se opta por utilizar un guante sin dedos en donde se puede instalar el sensor, entonces la propuesta final queda como se muestra en la **Figura 2.6**.

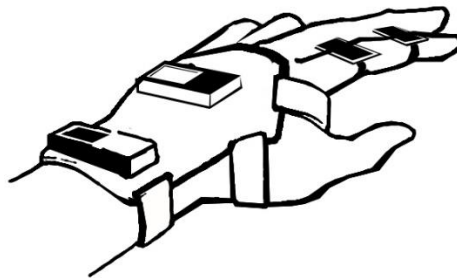


Figura 2.6: Concepto de montaje de sensores en anillos y guante sin dedos

Prototipo Final Sistema Captación de Movimientos

Los conceptos anteriores son llevados a prototipos físicos por medio de una impresora 3D marca Ultimaker 3, con filamento de material PLA. Los sensores son instalados en las falanges de los dedos por medio de dos piezas: la primera es un anillo abierto que es situado en la falange de los dedos, la otra es un porta-sensor que es ensamblado en el anillo (**Figura 2.7**). Esta decisión de separar los componentes se adopta para facilitar el proceso de manufactura por impresión 3D ya que ambas piezas se imprimen de forma independiente, pudiendo imprimir las piezas al mismo tiempo en impresoras distintas, y los tiempos de impresión son menores en comparación a si se imprime el conjunto completo.

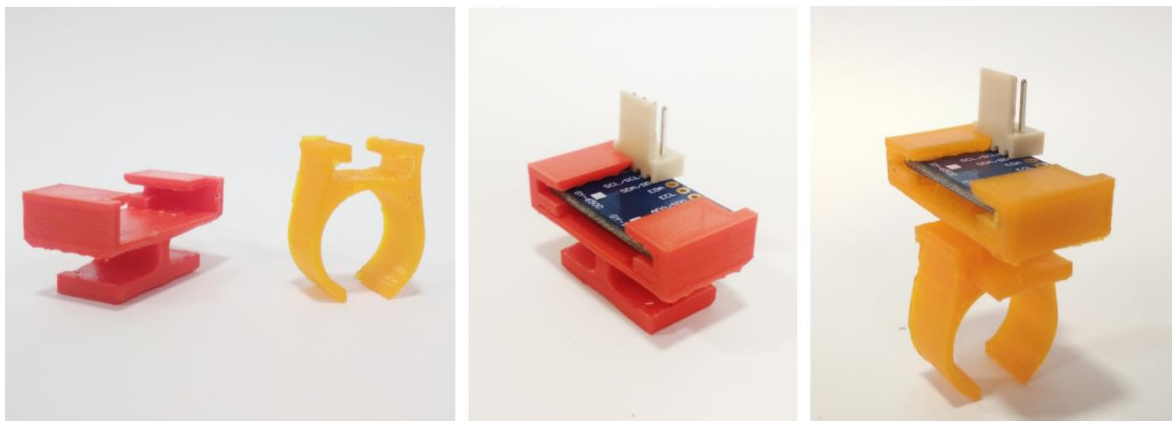


Figura 2.7: Prototipo físico de anillos y porta-sensores para falanges

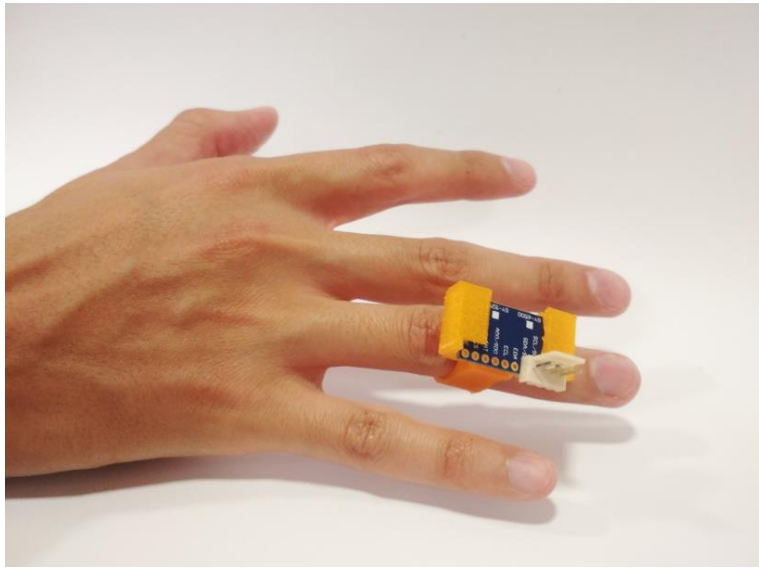


Figura 2.8: Prototipo de anillo en dedo anular.

Siguiendo con la idea de los porta-sensores fueron diseñados dos tipos dependiendo de la ubicación del sensor, si se encuentra en la palma el porta-sensor deja el sensor perpendicular a la mano, si el sensor se encuentra en alguna de las falanges el porta-sensor cambia su diseño para dejar el sensor de forma paralela a la falange (**Figura 2.9**). El motivo de estos dos diseños son problemas detectados en la entrega de datos de los sensores y rangos que estos alcanzan, esto es más detallado en el anexo **J Rango de captación de sensores**.

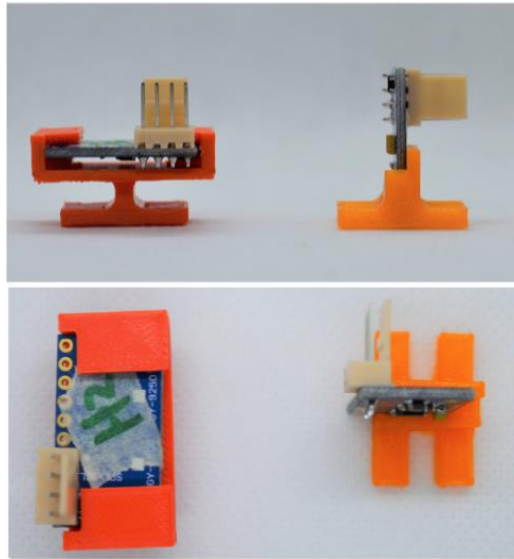


Figura 2.9: Vista frontal y superior de porta sensor en 90° y horizontal

Finalmente, para la parte de la palma es utilizado un guante corto sin dedos, que tiene una superficie desmontable por el lado del dorso. Esta superficie se puede desmontar gracias a un velcro que tiene en la parte inferior, en la parte superior de la superficie se adhiere el porta-sensor en 90°, en la **Figura 2.10** se muestra el prototipo sobre la palma.



Figura 2.10: Prototipo de guante con sensor montado en el dorso de la palma.

2.5 Procesamiento de datos y Control

2.5.1 Estrategia de control

La estrategia de control consiste en colocar sensores en cada una de las partes de la mano humana y de la mano robótica para obtener el ángulo con respecto a la horizontal. En ambas manos se utilizarán sensores por el lado de extensión, los que son ubicados en el dorso de la mano y en la falange medial del dedo índice. La idea es hacer un circuito de control a lazo cerrado en donde la variable de control es el ángulo en el que se encuentran las partes de la mano robótica.

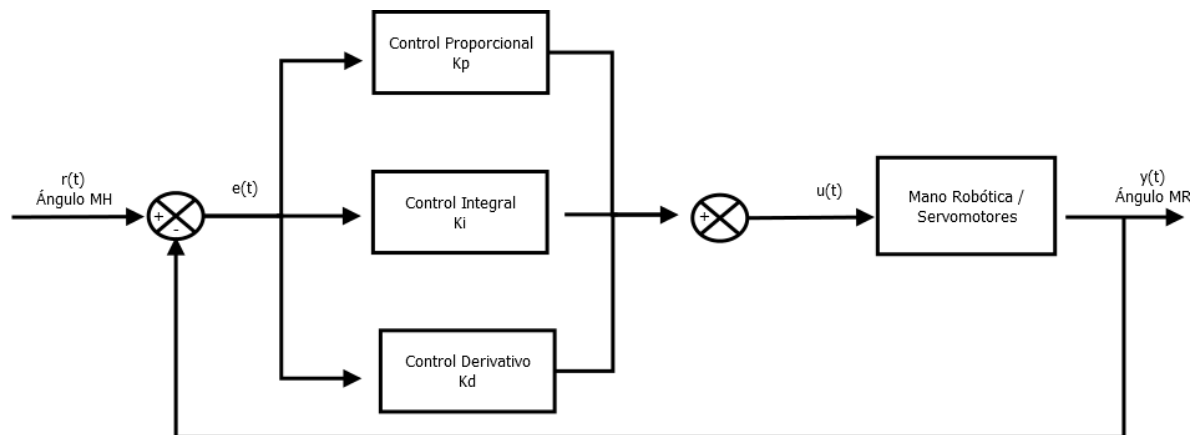


Figura 2.11: Diagrama control mano robótica

El valor de referencia es el ángulo que registra un sensor de la mano humana, este valor se compara con el ángulo del sensor de la mano robótica correlativo y se obtiene el error. Para controlar el error se utiliza un control del tipo PID que genera la señal de control $u(t)$ sobre los actuadores de cada parte de la mano que entrega una diferencia a sumar o restar en la posición del servomotor

2.5.2 Esquema de conexión

En la **Figura 2.12** se muestra el diagrama de conexión para el control de una de las partes de la mano robótica Hackerhand, ya sea un dedo o la muñeca. En el diagrama aparecen cuatro elementos: dos sensores MPU9250 (identificados como 9DoF IMU) utilizados para conocer la posición angular de la mano robótica y de la mano humana, un Servo que corresponde al actuador de la mano robótica, y la placa Arduino Uno que corresponde al sistema de procesamiento. Esta placa además de procesar las señales provenientes de los sensores y ejecutar el algoritmo de control también permite alimentar los sensores y al servomotor en el caso de que el control se realice en un dedo, en caso de que el control se lleve a cabo en la muñeca de la mano la alimentación del servomotor cambia a una fuente externa debido a que el servomotor utilizado en esa parte requiere mayor corriente de la que puede suministrar esta placa, teniendo en cuenta que deben compartir la línea de GND.

Para más información del esquema de conexión de los demás dedos de la mano robótica, esto aparece en el **anexo B*

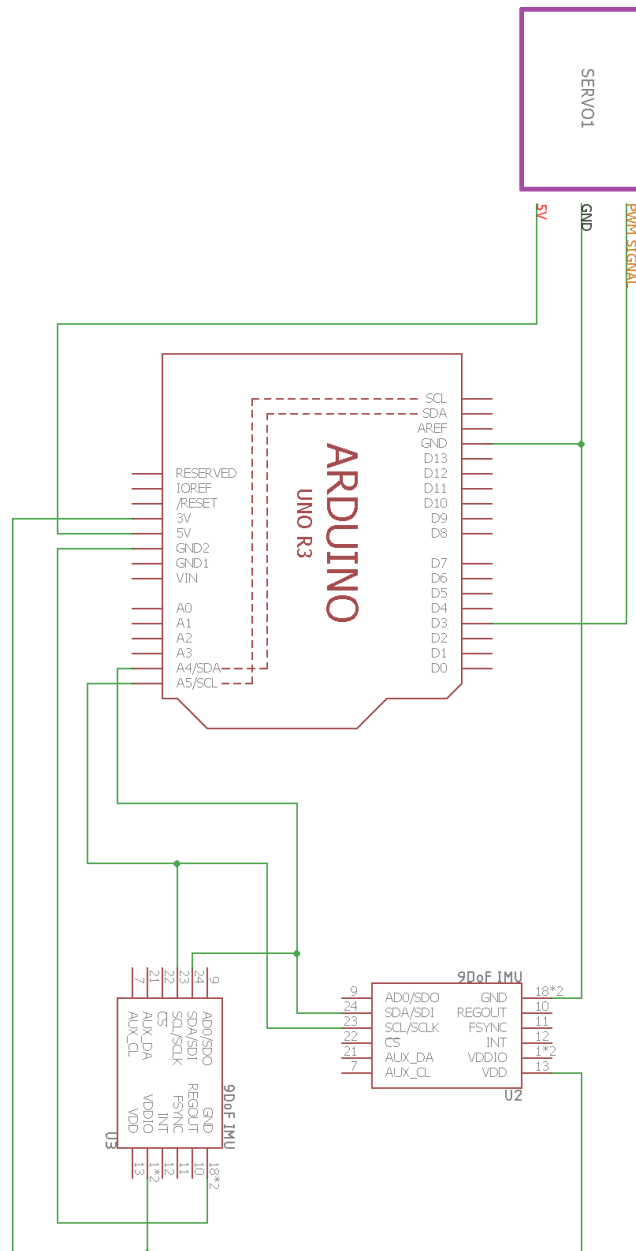


Figura 2.12: Diagrama de conexión para control de 1 dedo de la mano robótica

2.5.3 Estimación de la orientación de sensor IMU

A partir de las mediciones del giroscopio y de acelerómetro se puede obtener el ángulo de rotación en cada eje del sensor con respecto a un sistema fijo. Existen varias formas de representar estos ángulos ya que dependen del orden en el que se lleven las rotaciones del objeto, las formas más utilizadas son los ángulos de Euler y los ángulos de Tait-Bryan.

Los ángulos de Euler utilizan rotaciones en el orden $R_x R_y R_z$ para representar los ángulos, en cambio la notación de Tait-Bryan, utiliza las rotaciones en orden $R_z R_y R_x$. La notación de Tait-Bryan es usada por los sistemas de navegación aeronáuticos y a estos ángulos se les conoce como *Yaw* (ψ), *Pitch* (θ), *Roll* (ϕ).

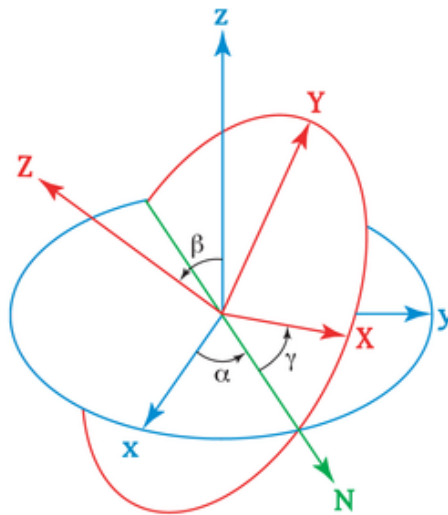


Figura 2.13: Representación de los ángulos de Euler en donde el sistema cartesiano en azul corresponde al sistema fijo y el rojo al móvil. [16]

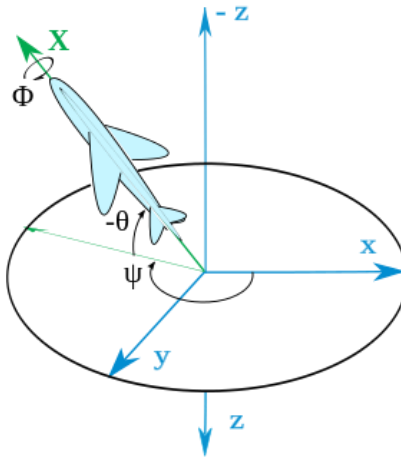


Figura 2.14: Representación de los ángulos de orientación de un cuerpo según la notación de Tait-Bryan [17]

2.5.3.1 Obtención a partir del acelerómetro

Como todo objeto en la tierra, el acelerómetro está siempre sujeto a la aceleración de gravedad y es a partir de esta aceleración que se calcula el ángulo de inclinación del sensor. Para ello, considérese el diagrama de cuerpo libre en dos dimensiones de la **Figura 2.15** Figura 2.15 .

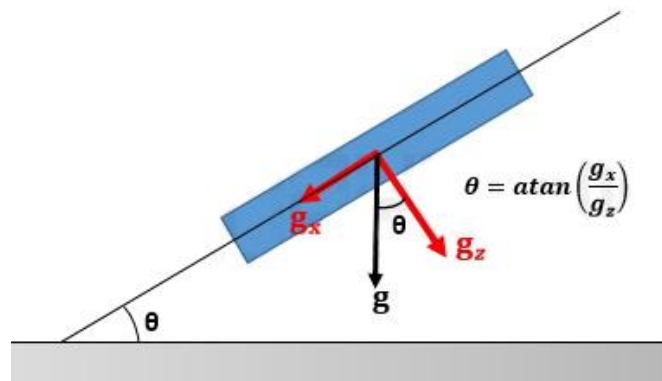


Figura 2.15: Diagrama de descomposición de la fuerza de gravedad cuando el sensor está inclinado [18]

En base a este mismo principio de descomposición de la fuerza de gravedad en los ejes XYZ es que se obtiene el ángulo de rotación en X e Y, el valor de la rotación en Z no es posible calcularlo debido a que la aceleración de gravedad está actuando en ese mismo eje, por lo que no hay un punto de referencia para comparar la inclinación. Si se desea tener la medición en este eje es necesario agregar otro sensor, por ejemplo, un magnetómetro que tiene como punto de referencia el norte magnético. Para efectos de este trabajo se omite el cálculo de la rotación en el eje Z ya que el movimiento de los dedos de la mano robótica se realiza sobre dos ejes.

$$\theta_x = \tan^{-1} \left(\frac{a_y}{\sqrt{a_x^2 + a_z^2}} \right) \quad (4)$$

$$\theta_y = \tan^{-1} \left(\frac{-a_x}{\sqrt{a_y^2 + a_z^2}} \right) \quad (5)$$

Los acelerómetros no están exentos de problemas al realizar la medición, ya que las componentes de la gravedad se pueden ver alteradas debido a perturbaciones en la aceleración, por lo que los resultados tienen alta variabilidad.

2.5.3.2 Obtención de ángulos a través de giroscopio

El giroscopio entrega información de la velocidad angular en cada eje, se puede obtener el valor del ángulo en un instante dado por medio de la aproximación:

$$\omega_i = \frac{\Delta\theta}{\Delta t} \approx \frac{\theta_i - \theta_{i-1}}{\Delta t} \quad (6)$$

$$\theta_i \approx \omega_i \cdot \Delta t + \theta_{i-1} \quad (7)$$

θ_i y ω_i es el ángulo y la velocidad angular en el instante actual respectivamente, Δt es la diferencia de tiempo, y θ_{i-1} es el ángulo en el tiempo anterior. Cabe destacar que la aproximación anterior genera un error llamado *drift*, esto es, a medida que pasa el tiempo el valor del ángulo aumenta, aunque no exista rotación en su eje, por lo que los ángulos obtenidos a través de un giroscopio son confiables solo en el corto plazo hasta que se genera *drift*.

2.5.3.3 Filtro Complementario

Como se mencionó anteriormente, tanto el acelerómetro como el giroscopio tienen errores en su medición de ángulo, es por ello que es necesario utilizar un filtro que combine ambas mediciones para obtener un resultado confiable. Las mediciones del acelerómetro son inestables ya que presentan alta variabilidad, es por ello que se les aplica un filtro pasa bajo para atenuar esta variabilidad. A diferencia del acelerómetro, el giroscopio no presenta variabilidad en el corto plazo, pero tiene el problema que a medida que pasa el tiempo, aumenta el error (*drift*), por lo que es necesario aplicar un filtro pasa alto. Al combinar ambos filtros, se obtiene el filtro complementario (**Figura 2.16**).

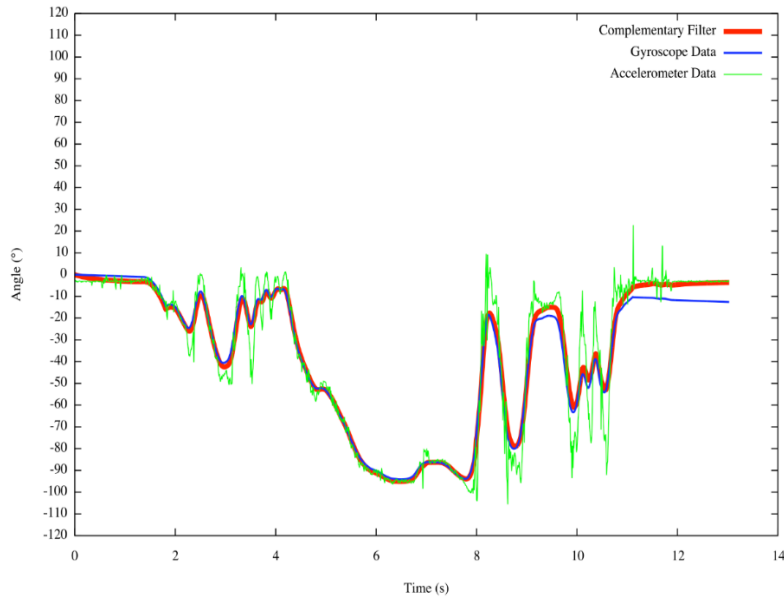


Figura 2.16: Gráfico del ángulo en función del tiempo obtenido por un giroscopio (azul), un acelerómetro (verde) y el valor con la combinación de ambos a través del filtro complementario (rojo). Nótese la alta variabilidad que presentan los datos del acelerómetro, mientras que los datos del giroscopio presentan un error de repetibilidad al final del gráfico [19]

El filtro complementario se calcula como:

$$\theta = \alpha \cdot \theta_{gyro} + (1 - \alpha) \cdot \theta_{acel} \quad (8)$$

En donde θ es el ángulo que combina las mediciones del acelerómetro y del giroscopio, θ_{gyro} es el ángulo obtenido por el giroscopio, θ_{acel} es el ángulo que da el acelerómetro, y α es una constante que tiene valores entre 0 y 1, y sirve para ver el peso que tiene cada una de las mediciones de los sensores, por lo general se considera un valor de $\alpha = 0.98$ debido a que las mediciones del giroscopio son más estables.

$$\theta = 0.98 \cdot \theta_{gyro} + (0.02) \cdot \theta_{acel} \quad (9)$$

2.5.4 Algoritmo de control actuadores

El sistema de control de los actuadores lo lleva a cabo el microcontrolador Arduino Uno que procesa los datos y efectúa el control PID. El algoritmo que realiza para hacer el seguimiento de un ángulo de referencia entregado por el sistema de captación de movimientos se muestra en la Figura 2.17:

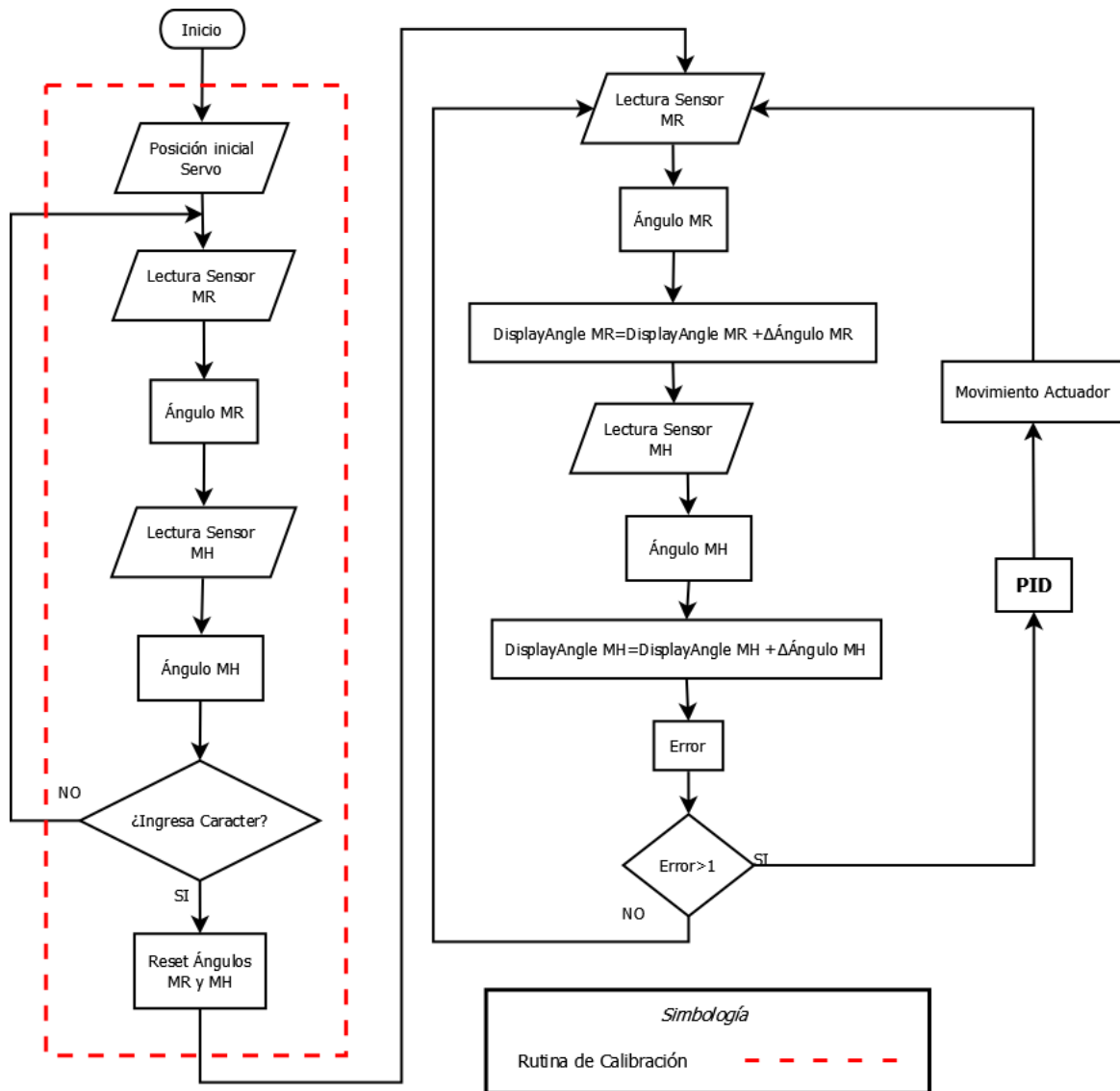


Figura 2.17: Diagrama de flujo del algoritmo de control

El punto de partida del algoritmo es el envío de la posición inicial del servomotor para que el dedo comience con las falanges completamente extendidas. Una vez que el dedo llega a esta posición inicial comienza la lectura de los datos del sensor de la mano robótica y de la mano humana, cabe destacar que inmediatamente después de realizar la lectura del sensor se calcula el ángulo, esto se debe a que el ángulo obtenido por el giroscopio depende de un delta de tiempo. Posteriormente, la lectura de los sensores es efectuada de forma cíclica hasta que el usuario ingrese un carácter, esta parte del algoritmo corresponde a la rutina de calibración entre los sensores de la mano robótica con los sensores de la mano humana, el objetivo es que ambas manos sepan dónde están y tengan un punto de partida común para los movimientos que van a realizar. Para llevar a cabo la calibración ambas manos deben estar con los dedos extendidos y de forma vertical (posición anatómica), cuando el usuario ingresa un carácter el programa resetea la posición angular de ambas manos. Después de volver a leer el ángulo de los sensores el programa ve la diferencia que hay entre las dos lecturas, si existe un error mayor a $1 [^\circ]$, es aplicado el control PID que entrega como resultado una señal de corrección $\Delta\theta$ a sumar o restar a la posición del servomotor para compensar el error.

2.6 Validación de dispositivo de adquisición de datos

Uno de los objetivos de este trabajo de memoria es desarrollar un sistema de captación de los movimientos de una mano humana. Este dispositivo es fundamental para el correcto control de la mano robótica por lo que se hace necesario poder validarlo antes de realizar el control. Tomando en cuenta los datos que entrega el sensor y acoplándolos a un modelo cinemático directo se puede obtener la posición de la mano, en particular de un dedo y la muñeca, entonces a través de una comparación visual se puede ver si los datos que están entregando los sensores son correctos

En el trabajo desarrollado por [20] se presenta un modelo cinemático de la mano humana compuesto por 19 links correspondientes a los huesos y 24 DoFs (Grados de libertad) formados por uniones. En la **Figura 2.18** se puede ver la configuración cinemática para cada dedo, cabe destacar que los nombres de las uniones son los mismos que los presentados en el **capítulo 1.3**. De la imagen se observa que cada dedo es enumerado del 1 al 5 y cada una de las uniones está indicada con la letra **j**. En la parte inferior del hueso metacarpiano se encuentra la primera unión que es modelada a través de dos ejes coordenados distintos para representar los movimientos de flexión-extensión y los de aducción-abducción. En la unión de la falange proximal con el hueso metacarpiano también existen dos definiciones de ejes coordenados para representar los mismos movimientos, diferente al caso de las otras dos uniones superiores en las falanges medial y distal donde las definiciones de los ejes coordenados permiten solo los movimientos de flexión-extensión.

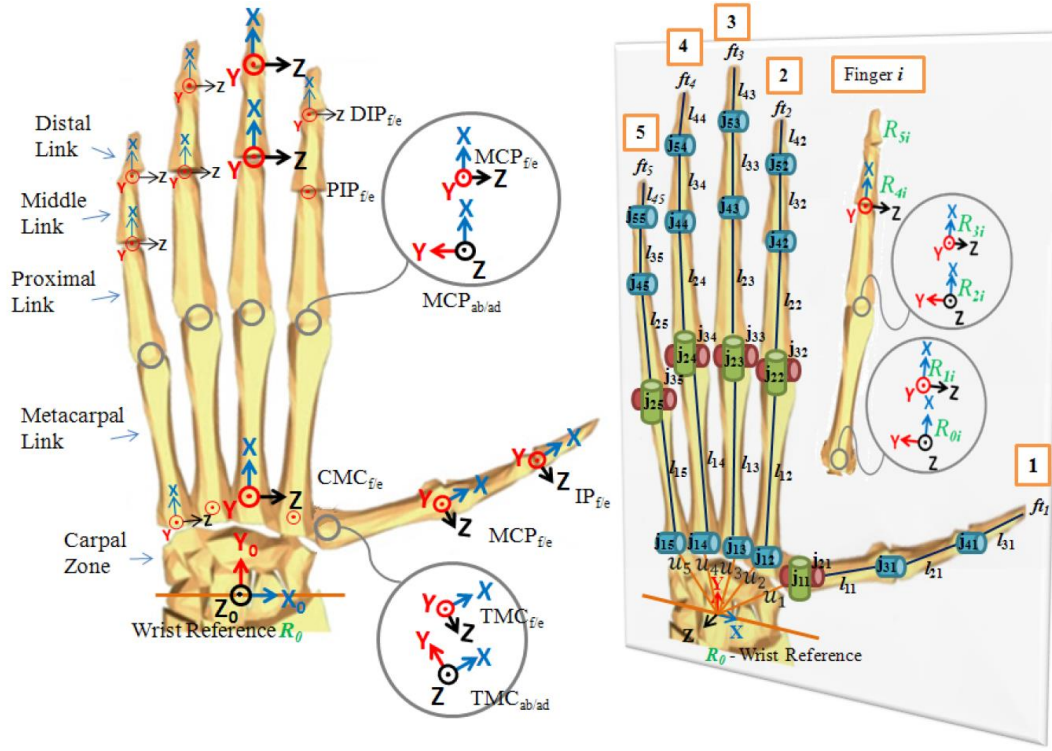


Figura 2.18: Configuración Cinemática de la mano humana con la definición de la orientación del sistema de coordenadas para cada una de las uniones

El modelo cinemático directo utiliza la representación de Denavit Hartenberg [20] para calcular la posición, los parámetros utilizados en el dedo índice son:

Tabla 2.4: Parámetros de Denavit Hartenberg utilizados para realizar el esquema cinemático directo del dedo índice

Unión	α_{i-1}	a_{i-1}	d_i	θ_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	$\theta_{DIP_{f/e}}$

Es interesante observar que la unión j_{12} y la unión j_{32} no tienen una longitud de eslabón asociada, esto es debido a que se utilizan solo para representar otro grado de libertad en ese punto.

Utilizando 3 sensores MPU9250 conectados a Arduino se realiza la lectura de datos, posteriormente estos son compartidos a través del puerto serial con el Software Matlab que es donde se grafican 4 eslabones que simulan las tres falanges de un dedo y la palma (Figura 2.19).

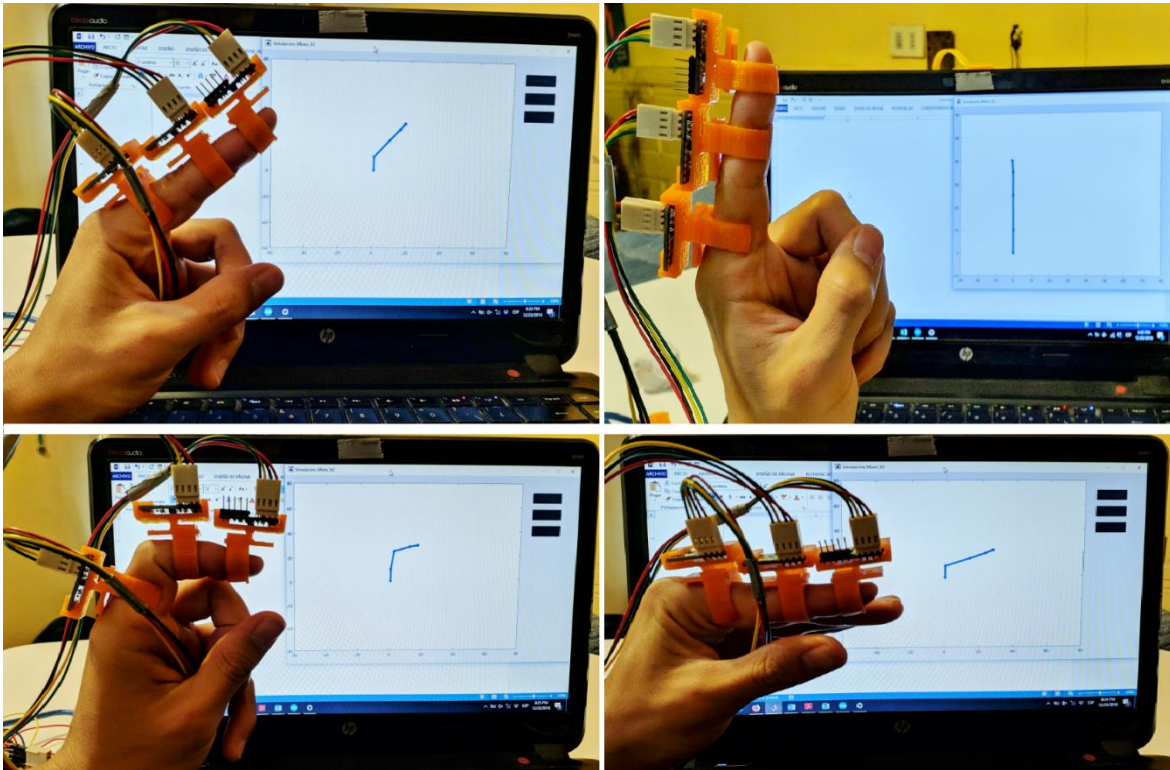


Figura 2.19: Distintas posiciones del sistema de captación de movimientos instalado en las falanges del dedo índice para hacer validación de este por medio de un método gráfico utilizando cinemática directa

Con este método gráfico se valida que los datos entregados por los sensores son correctos ya que pueden emular el movimiento de un dedo. Tomando en cuenta que el objetivo del sistema de captación de movimientos es el control de la mano robótica presentada en el **capítulo 1.5.2** hay que tomar en cuenta las limitaciones de movimiento que tiene, es por ello que el sistema de captación de

movimientos tiene que contemplar una falange correspondiente al DoF controlado, (falange medial), y el movimiento de la muñeca.

2.7 Experimentos Realizados

El control PID requiere encontrar los valores de las constantes para obtener una óptima respuesta, para ello se realizan diversas pruebas para el movimiento de la falange proximal y la muñeca. A continuación, se lista la metodología utilizada para encontrar las constantes esto es realizado tanto para la muñeca como para la falange del dedo índice de forma independiente:

- 1) Sintonización constantes PID dado un valor de referencia
 - a) Búsqueda de constante Proporcional (Control Proporcional)
 - b) Adición de constante Integral (Control PI)
 - c) Adición de constante Derivativa (Control PID)
- 2) Análisis de control PID frente a respuesta variable

Los primeros experimentos de la lista son realizados frente a una referencia constante, lo que se traduce en que el programa realiza una única trayectoria desde una posición inicial a una final considerando o el movimiento de la muñeca o la falange del dedo. Para el caso de la falange del dedo índice, la posición angular es $\theta = 50^\circ$, y para la muñeca $\theta = 20^\circ$ producto de la posición perpendicular a la mano que tiene el sensor. (Figura 2.20 y Figura 2.21)

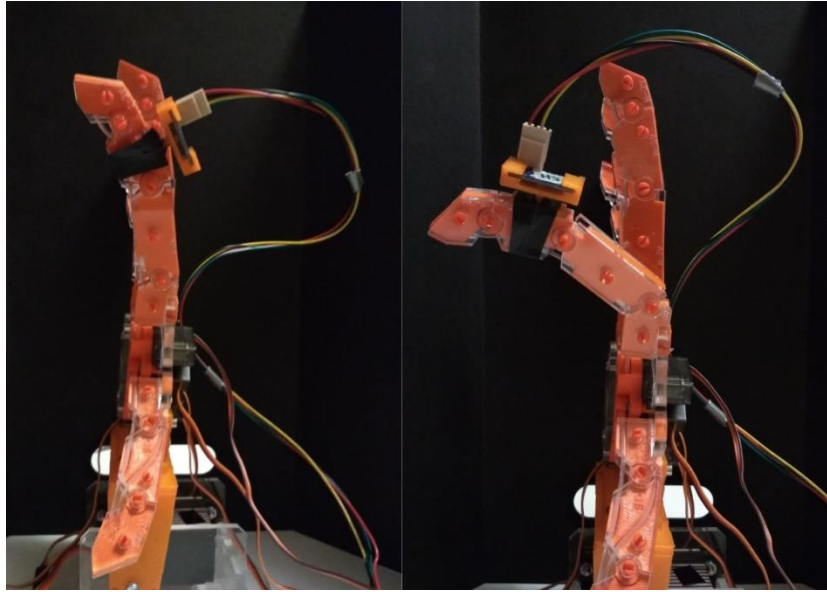


Figura 2.20: Posición inicial (izquierda) y final (derecha) a la que debe llegar dedo índice cuando se realizan los experimentos para encontrar valores de constantes PID frente a una referencia constante ($\theta = 50^\circ$).

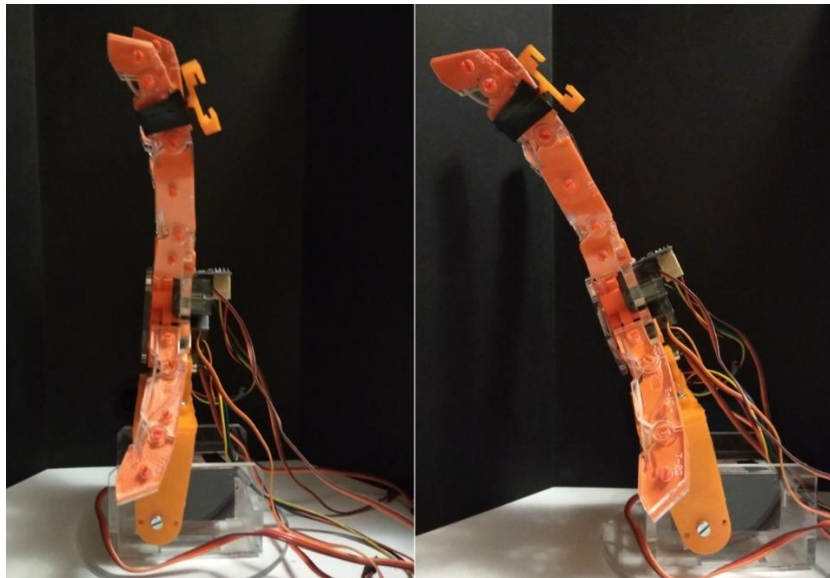


Figura 2.21: Posición inicial (izquierda) y final (derecha) a la que debe llegar la mano cuando se realizan los experimentos para encontrar valores de las constantes PID del movimiento de muñeca frente a una referencia constante ($\theta = 20^\circ$).

Resulta interesante ver la respuesta del sistema al seguir una referencia con otro tipo de control, para ello el sistema de control relaciona directamente la posición angular del dedo con la posición del servomotor. De forma de relacionar las variables mencionadas anteriormente se hacen mediciones de la posición angular del dedo para distintas posiciones angulares del servomotor con ayuda de sensores en la mano robótica, estos resultados son graficados para posteriormente obtener la ecuación que relaciona ambas variables. Posteriormente se realiza una prueba de rendimiento en donde se instala un sensor en el dedo índice para obtener información de la posición angular real del dedo. La prueba consiste en ver el comportamiento del dedo de la mano robótica frente a una referencia variable entregada por el sistema de captación de movimientos, esto es, seguir el movimiento de la mano humana.

3 Resultados

3.1 Sintonización constantes PID dado un valor de referencia

3.1.1 Sintonización control proporcional (P) dedo índice

Para la sintonización de la constante de control proporcional se realizaron ensayos con distintos valores de k_p , en la **Figura 3.1** se presentan gráficos de la posición angular del dedo índice en el tiempo para distintos valores de k_p , en ellos se observa que los valores de la constante son menores a 1 por lo que el tiempo de respuesta del actuador es relativamente alto, es por ello que el valor de la constante debe ser pequeño. Con un valor de $k_p = 0.1$ el dedo tiende a oscilar en el tiempo, al disminuir el valor de la constante la respuesta del dedo se amortigua a medida que avanza el tiempo, pero llega un punto en el que un valor más bajo de la constante produce que la respuesta del dedo sea más lenta, como es el caso de $k_p = 0.04$.

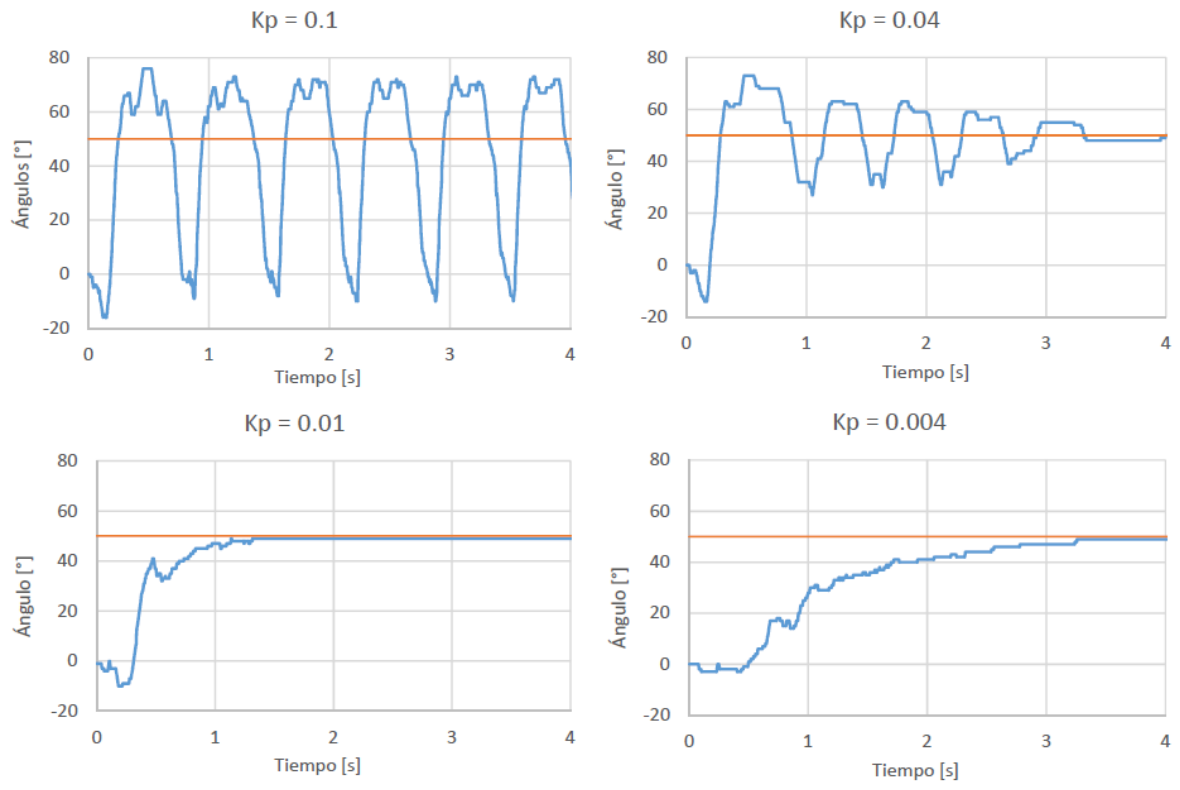


Figura 3.1: Tiempo de respuesta del sistema con control Proporcional para distintos valores de constante k_p , implementado en dedo índice con una señal de referencia $\theta = 50^\circ$.

3.1.2 Sintonización control PI (Proporcional - Integral) dedo índice

Se considera un buen resultado el obtenido con $k_p = 0.01$ con control proporcional ya que el tiempo de respuesta es de aproximadamente 1 [s]. Para disminuir el error que tiene en el tiempo se añade la componente integral al lazo de control. Los gráficos de la **Figura 3.2** muestran el desempeño a lo largo del tiempo con distintas combinaciones de constantes proporcional e integral (k_p, k_I). Al utilizar un valor similar de k_p y k_I el sistema no es estable, oscila con una gran amplitud entre los valores máximo y mínimo. La combinación $k_p = 0.016, k_I = 0.0001$ disminuye el tiempo de respuesta anterior a 0.5 [s] con un overshoot de 10 [°]. Para eliminar este overshoot, el valor de la constante

proporcional disminuye y el de la constante integral aumenta, el resultado es un aumento de 0.2 [s] pero sin sobrepasar el valor de referencia.

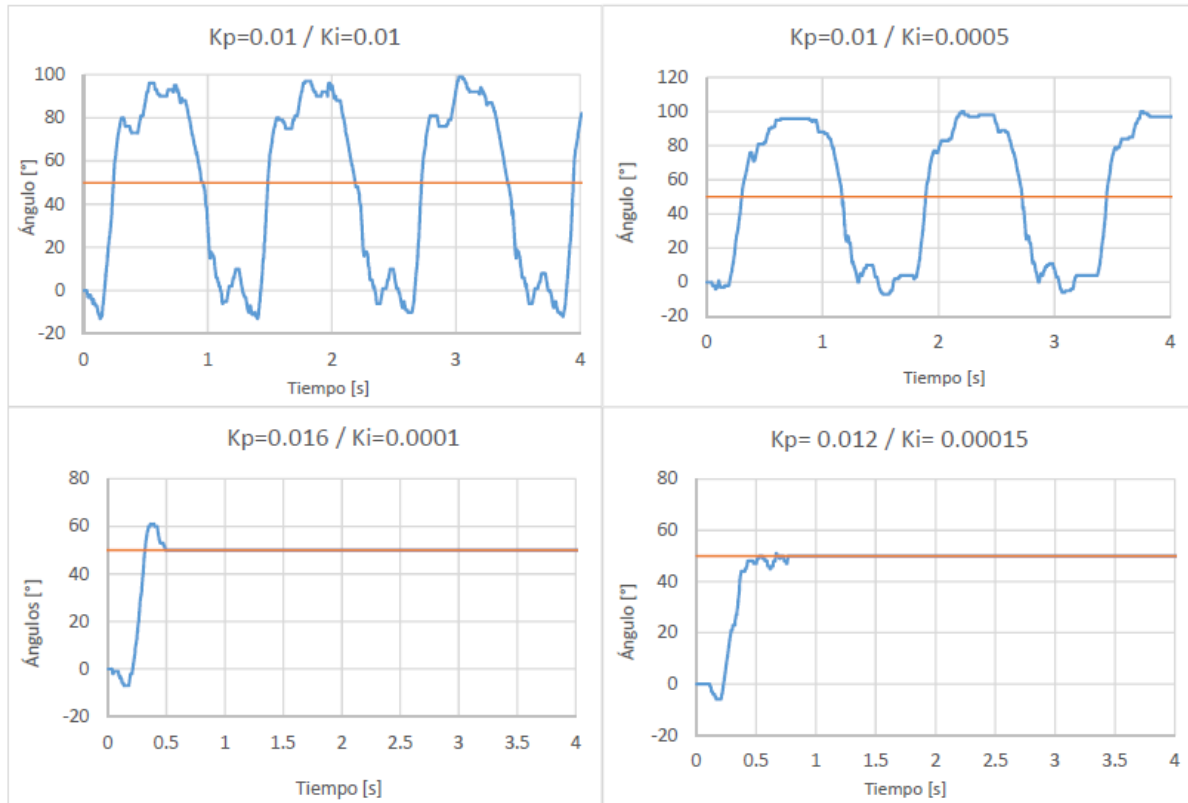


Figura 3.2: Tiempo de respuesta del sistema con distintos valores de constantes k_p y k_i en un control PI en un dedo para una señal de referencia $\theta = 50^\circ$

3.1.3 Sintonización control PID (Proporcional-Integral-Derivativo) dedo índice

Tomando como base la combinación de valores de constantes PI ($k_p = 0.012, k_i = 0.00015$) se agrega la componente derivativa y se analiza el desempeño con distintas combinaciones (**Figura 3.3**) La constante derivativa debe tener un valor mayor en orden de magnitud que las otras constantes para que sea notorio el cambio. Con un valor $k_d = 0.1$ la respuesta es parecida a la del control PI, pero al modificar la constante proporcional el tiempo disminuye a 0.5 [s] pero se observa un error de 1 [°]

que se mantiene. Buscando disminuir el overshoot que se genera para $k_p = 0.013, k_I = 0.00015$ es que disminuye el valor de la parte proporcional y aumenta el de la parte integral. Con esto se obtiene un resultado muy similar al anterior a diferencia de que elimina el error, pero el overshoot aumenta. Debido a lo anterior es que se selecciona como mejor combinación de valores de constante $k_p = 0.012, k_I = 0.00015, k_D = 0.1$ ya que un error de 1 [°] es aceptable en el sistema.

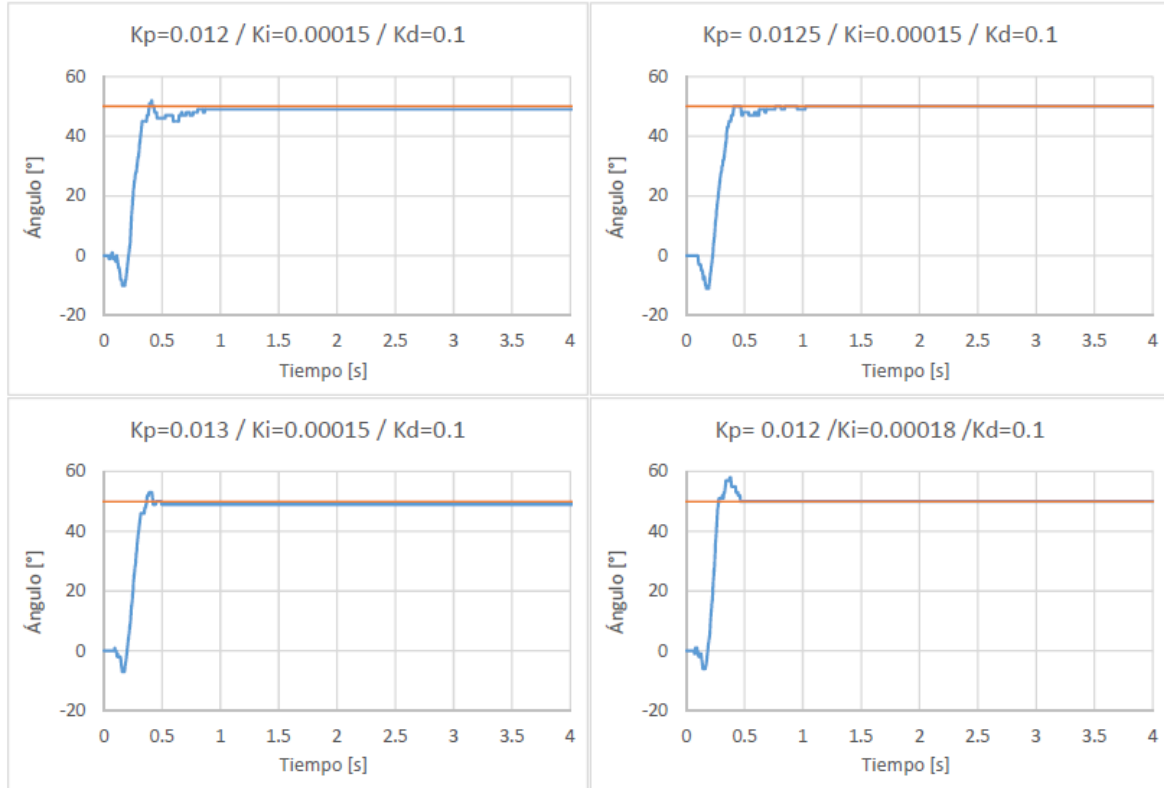


Figura 3.3: Tiempo de respuesta del sistema para distintas combinaciones de valores de constantes proporcional, integral, derivativa al seguir una referencia de $\theta = 50^\circ$ en el dedo medio de la mano robótica.

3.1.4 Sintonización constantes PID para Muñeca

Se realiza el mismo análisis anterior para el control de la Muñeca, esto es, ver el tiempo de respuesta con distintos valores de constante de control **P**, **PI**, **PID**, en las siguientes imágenes se muestran la mejor combinación de constantes para cada una de ellas.

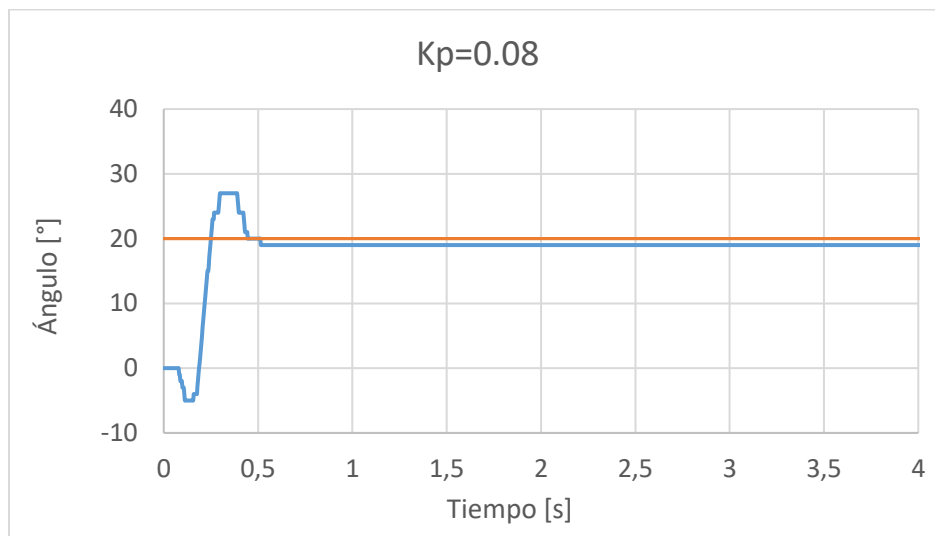


Figura 3.4: Tiempo de respuesta de la muñeca con control proporcional al seguir una señal de referencia $\theta = 20^{\circ}$

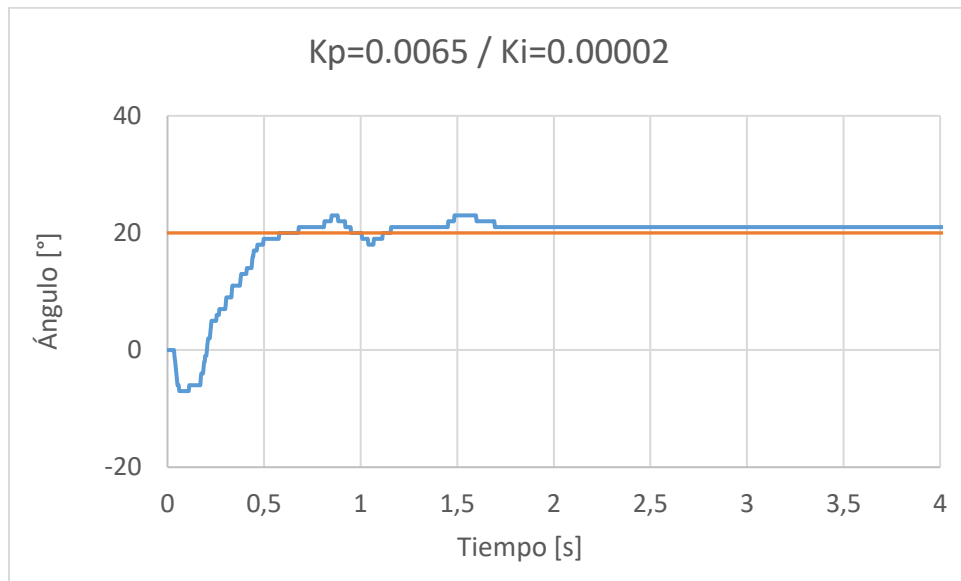


Figura 3.5: Tiempo de respuesta de la muñeca con control PI , al seguir una referencia de $\theta = 20$ [°]

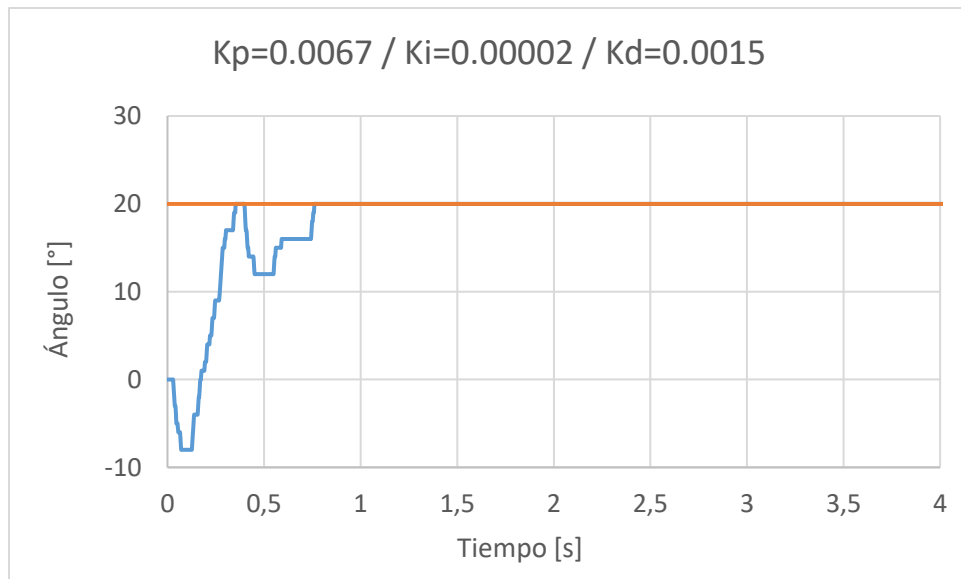


Figura 3.6: Tiempo de respuesta de la muñeca con control PID al seguir una referencia de $\theta = 20^\circ$

Los valores de las constantes no son iguales a los del caso de los dedos debido a que el mecanismo de accionamiento es distinto producto del montaje sobre el servomotor.

3.2 Respuesta PID frente a referencia variable

Una vez sintonizadas las variables del PID para el dedo y la muñeca, es observada la respuesta que tiene el sistema cuando debe seguir una referencia variable que viene dada por el sistema de captación de movimientos de la mano humana.

En la **Figura 3.7** están graficados los datos de la respuesta del dedo respecto a la referencia variable. Como primera diferencia se observa un desfase entre ambas curvas de aproximadamente 1 [s], este desfase se debe a la velocidad de respuesta en la ganancia del lazo del lazo de control. En los primeros instantes del programa existe un desfase mayor producto del mecanismo de actuación. Cuando la referencia se detiene la diferencia entre los ángulos de ambas manos tiende a minimizarse y no produce mayor desfase que el mencionado anteriormente.

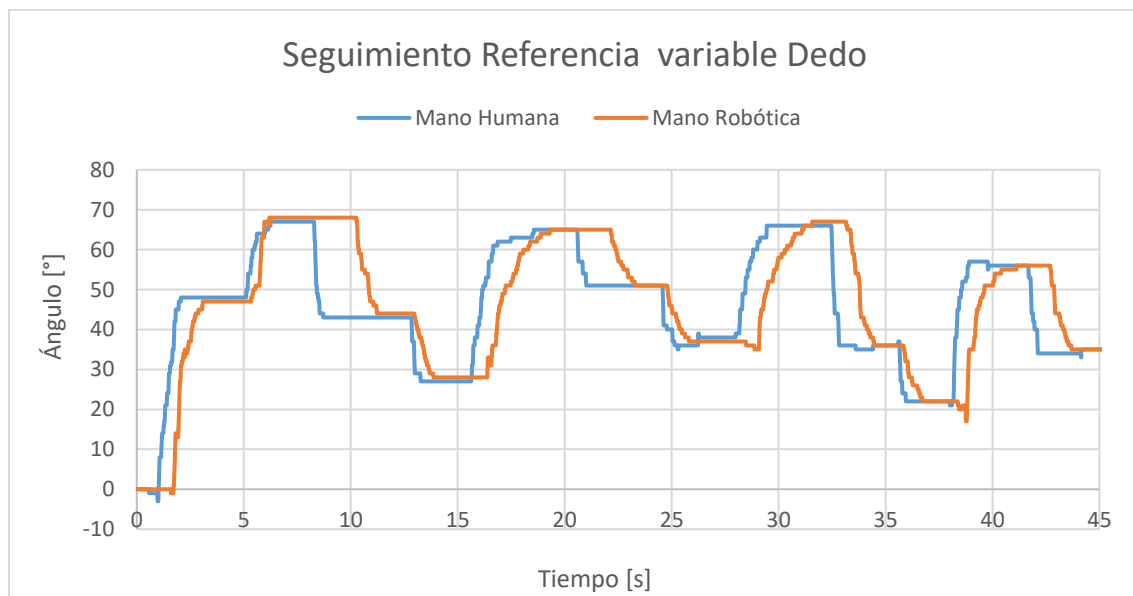


Figura 3.7: Respuesta del dedo de la mano robótica frente a una referencia variable proveniente del sistema de captación de movimientos de la mano humana, con valor de constantes PID: $k_p = 0.013$, $k_I = 0.00015$, $k_D = 0.1$

Al aumentar al doble el valor de la constante derivativa esperando disminuir el tiempo de respuesta, se obtiene un resultado que no dista de manera significativa del anterior. (**Figura 3.8**).

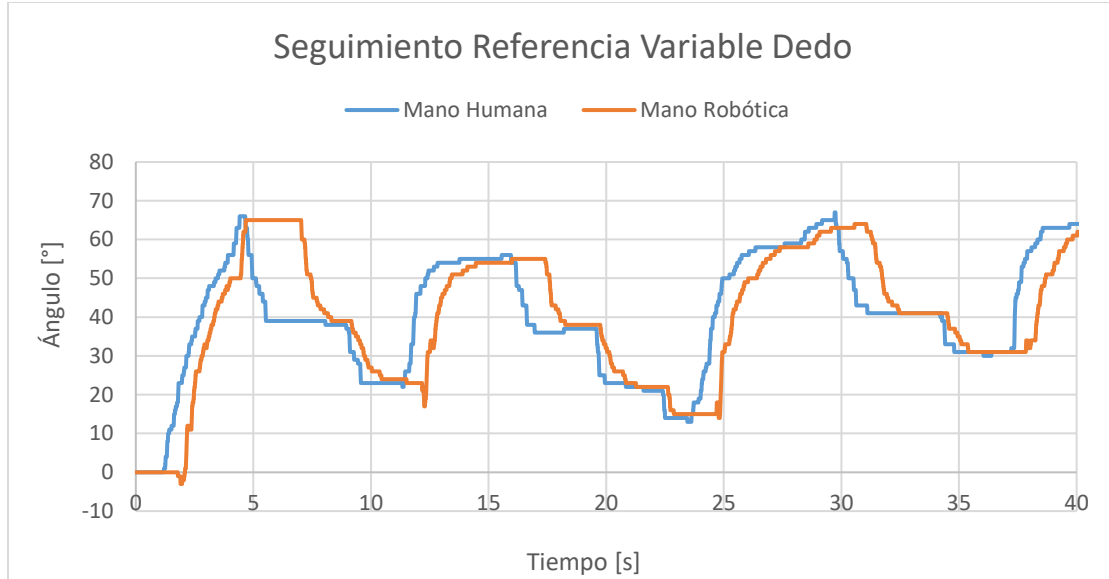


Figura 3.8: Respuesta del dedo de la mano robótica frente a una referencia variable proveniente del sistema de captación de movimientos de la mano humana, con valor de constantes PID: $k_p = 0.013$, $k_I = 0.00015$, $k_D = 0.2$

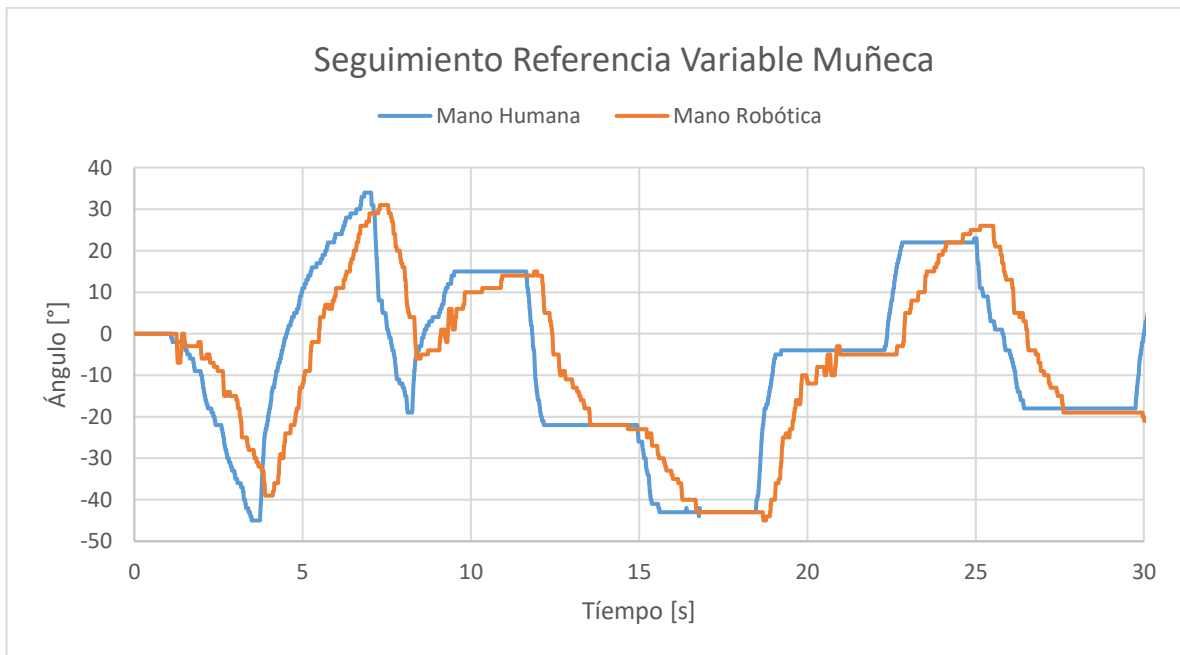


Figura 3.9: Respuesta de la muñeca de la mano robótica frente a una referencia variable proveniente de la muñeca del sistema de captación de movimiento de la mano humana, con valor de constantes de PID: $k_p = 0.065$, $k_I = 0.000015$, $k_D = 0.0015$.

La respuesta de la muñeca es más rápida que la de los dedos, las curvas en la **Figura 3.9** tienen un desfase aproximado de 0.7 [s] que es producto de la conexión directa al eje del servomotor. El tiempo de desfase en la muñeca es importante ya que el funcionamiento del control de los dedos está directamente relacionado con la posición de la muñeca por lo que, si demora más en el control de la muñeca va a generar perturbaciones al PID del dedo.

3.3 Control proporcional entre posición de Servomotor y ángulo de dedo

Se estudia la respuesta del sistema al seguir una referencia con otro tipo de control, para ello se considera un control que relaciona directamente la posición angular del dedo con la posición del servomotor. Se toman varias muestras del valor del ángulo del dedo para distintas posiciones del servomotor, con esto se caracteriza la relación entre ambas variables, posteriormente se grafica la posición del servomotor en función del ángulo del dedo (**Figura 3.10**)

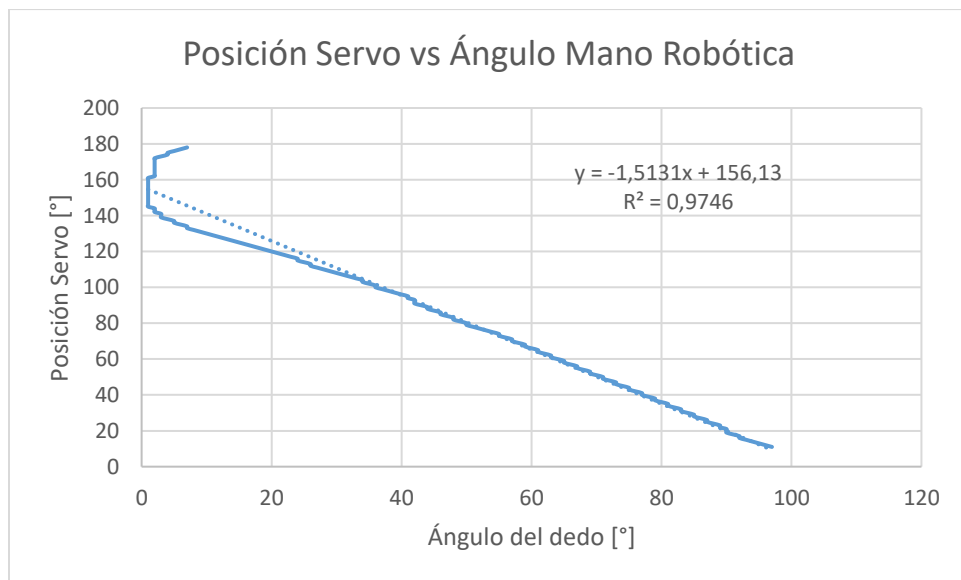


Figura 3.10: Gráfico de Posición del servomotor en función del ángulo del dedo. En la esquina superior derecha se presenta la ecuación de la recta junto con el coeficiente de correlación entre ambas variables

La línea de tendencia se define para un ajuste lineal, lo que da una ecuación de recta:

$$Pos_{Servo} = -1.5131 * \theta_{dedo} + 156.13 \quad (10)$$

Esta ecuación es introducida en el código, entonces, dado el ángulo obtenido por el sistema de captación de movimientos se calcula a través de la ecuación (10) la posición que debe tener el servomotor para llegar a ese ángulo. Para analizar el rendimiento de este sistema de control, se coloca un sensor en el dedo medio para ir comparando la medición real con el ángulo teórico entregado por el sistema de captación de movimientos (señal de referencia). Se observa en la **Figura 3.11** que el sistema tiene una respuesta rápida, pero en el flanco de bajada se produce un desfase de aproximadamente 1 [s] entre la señal de referencia y el ángulo del dedo en la mano robótica. También se observa que si la señal de referencia se mantiene fija el dedo no llega al valor correcto de ángulo, incluso en los puntos más bajos de la curva.

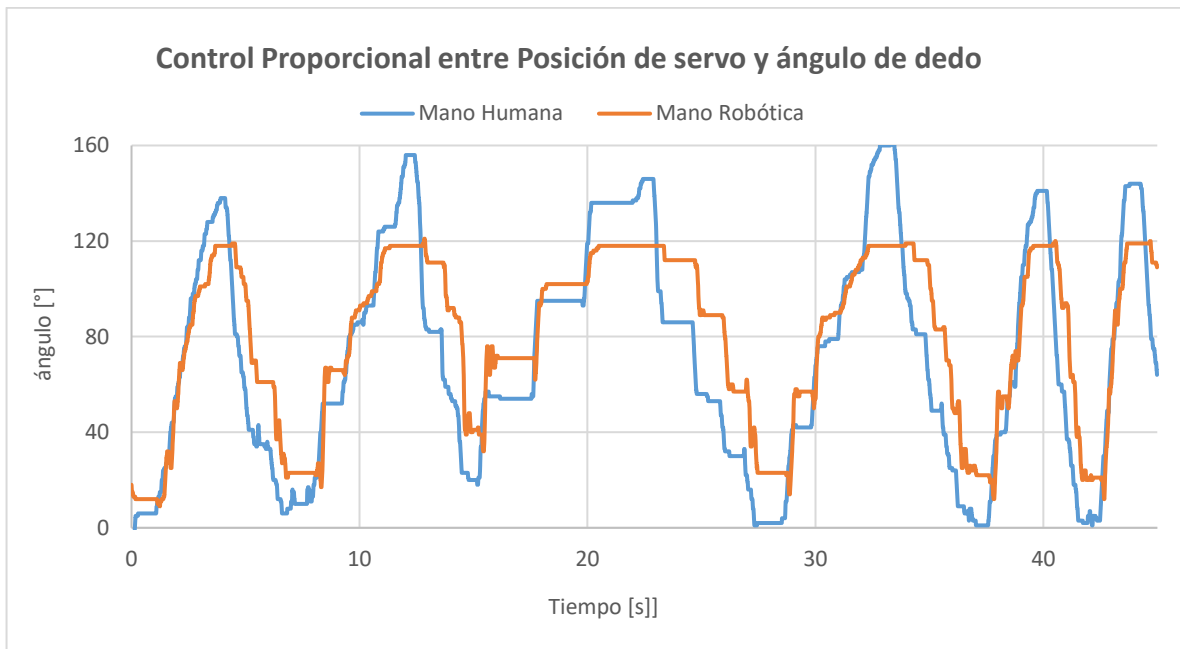


Figura 3.11: Ángulo de la mano robótica al hacer seguimiento de la referencia variable dada por la mano humana, utilizando un control que relaciona la posición del servomotor y el ángulo del dedo de forma proporcional.

4 Conclusiones

La mano humana presenta 24 pares de movimiento activo que corresponden a movimientos de extensión - flexión, aducción - abducción y movimientos de oposición. El movimiento de flexión-extensión está presente en todas las falanges de cada uno de los dedos a diferencia del movimiento de aducción-abducción que se realiza sobre la articulación metacarpofalángica que actúa sobre la falange proximal. Referente al movimiento de oposición, este es realizado solo por el dedo meñique y el pulgar y aunque su rango de movimiento es acotado estos son importantes ya que permiten realizar posturas básicas para el agarre de objetos. La muñeca se considera parte de la mano y puede realizar los movimientos de flexión-extensión y aducción-abducción. Los rangos de movimiento de cada parte de la mano varían entre ellos, pero en general los movimientos de flexión-extensión son los que tienen mayor rango seguidos de los movimientos de abducción – aducción y por último los de oposición.

A diferencia de la mano humana la mano robótica sobre la que se hace control es más simple debido a que la cantidad de movimientos que puede realizar se reduce a movimientos de flexión-extensión en los dedos los que son generados por la tracción de un cable de nylon conectado a la falange medial. El movimiento de la muñeca no lo puede realizar la mano robótica por si sola, para llevar a cabo el movimiento de flexión-extensión de la muñeca es desarrollado un mecanismo que conecta la mano robótica con un motor.

El dispositivo de captación de movimientos desarrollado está compuesto por sensores IMU con acelerómetros y giroscopios, cada uno de estos sensores es acoplado a una falange de los dedos y al dorso de la mano para captar los movimientos de flexión -extensión de los dedos y de la muñeca. Este dispositivo toma los datos del giroscopio y del acelerómetro de cada sensor para calcular el ángulo de inclinación respecto a una superficie horizontal en el plano de flexión-extensión del dedo o muñeca. Para la estimación de la orientación del sensor se utiliza un filtro complementario que combina los datos del acelerómetro y del giroscopio, logrando lecturas más estables del ángulo de inclinación. Este dispositivo fue validado a través de un método gráfico en donde se simulaban las tres falanges de un dedo comprobando que el dedo emulado siguiera el movimiento del dedo humano.

Para ubicar los sensores en los dedos fue diseñada una pieza con forma de anillo en donde se ensambla un porta sensor, de este modo se pueden colocar varios sensores en un solo dedo para captar los movimientos de todas las falanges. Los movimientos que el sistema es capaz de obtener son extensibles a la aducción – abducción de los dedos calculando el ángulo de inclinación en ese plano teniendo cuidado de que el ángulo a medir quede en los ejes X o Y de los sensores. Para extender la captura de movimientos a los movimientos de oposición es necesario añadir los datos de otro sensor, como lo es el magnetómetro, de forma de obtener la inclinación en el eje Z y trabajar con otra representación de sistema de coordenadas para no tener el problema del Gymbal Lock, como es el caso de la representación a través de cuaterniones.

Una vez que el sistema de captación de movimientos estuvo operativo se analizó como acoplar la información que entrega el dispositivo con el control de la mano robótica. Al implementar un control a lazo abierto por medio de una relación proporcional entre el ángulo del dedo y la posición del servomotor se obtuvo una rápida respuesta del dedo de la mano robótica pero imprecisa en el seguimiento del ángulo del dedo la mano humana. El control a lazo cerrado permite mayor precisión ya que siempre actúa sobre el error del sistema. Es por ello que se llevó a cabo una estrategia de control tipo PID en donde la mano robótica y la humana tienen sensores en cada dedo y en su muñeca. Al comparar el ángulo de inclinación de un sensor en la mano humana y su correlativo en la mano robótica se obtiene el error y con este se calcula la corrección de posición angular que debe hacer el servomotor de la mano robótica para llegar a la posición correcta. Al ajustar los parámetros de las constantes del PID se logra un desfase de solo 0.5 [s] al hacer seguimiento de una señal de referencia variable con un error menor a 1 [°].

Como el sistema de control es a lazo cerrado los sistemas de cada mano (robótica y humana) deben ser calibrados para que el movimiento entre ellos sea coordinado. Esto se logra por medio de dos pasos, el primero corresponde a la calibración de los sensores para que estos tengan el mismo punto de referencia y su rango de lectura sea el mismo, lo que se logra a través del cálculo de los offsets de las lecturas en cada eje del acelerómetro y del giroscopio. El segundo paso se realiza al comenzar el programa de control haciendo que los ángulos de todos los sensores de ambas manos sean cero cuando la mano robótica y la humana están con los dedos extendidos en posición vertical, de este modo se corrobora que las mediciones de ambas manos tenga el mismo origen.

5 Trabajo a futuro

Para darle sentido tanto a las memorias ya realizadas como a las que vengan a futuro y contextualizarlas en el desarrollo a nivel mundial, es necesario retomar estos trabajos con la consideración de que la finalidad de estos es obtener un robot “actuador final antropomorfo”, basándonos en [21], desarrollar “Dextrous Hands” es más amplio que sólo desarrollar apariencia estética y propiedades cinemáticas, los índices indicados por [21] se refieren al grado de “antropomorfismo”, que estudia el parecido con la mano humana (en relación al aspecto y estructura mecánica) y el nivel de destreza resultante de la combinación de la configuración cinemática, el aparato sensorial y el sistema de control. Además de esto, otra línea de trabajo es el “telecontrol”, basado en un sistema de captación de todos los movimientos de la mano humana para, posteriormente, ser utilizado en el control del actuador final.

Otro punto importante a considerar es la transmisión mecánica, según [22] el diseño del dispositivo robótico debe ser tan simple y confiable como sea posible, y a pesar de que es deseable un alto número de DoF este no debe obtenerse si involucra aumentar sustancialmente la complejidad en la interfaz human-machine. Además, **Gama, Avilés y Amaya**, enuncian la importancia de permitir un bloqueo mecánico con el objetivo de ahorrar energía al ejecutar el agarre, ya que este debe mantenerse estable.

Respecto a la captación de movimiento uno de los trabajos a realizar a futuro, para extender el uso de este sistema de Spatial Tracking a todos los movimientos de la mano humana para ser utilizado en el control del prototipo de mano robótica [6], es habilitar la lectura del ángulo en el eje Z y analizar el rango de los sensores. El movimiento de oposición que realiza el pulgar y el meñique tiene una componente rotativa del dedo sobre el eje vertical, entonces el movimiento que se produce es en tres dimensiones. Para captar este movimiento es necesario habilitar las lecturas del magnetómetro por el efecto explicado en la **sección 2.5.3.1**

Con los sensores MEMS utilizados en este trabajo y por medio de las ecuaciones de la **sección 2.5.3** el rango de los ángulos que puede captar se reduce de 0 a 180 [°], este rango no alcanza a abarcar los

movimientos de los elementos más distales a la muñeca, como se muestra en la **Figura 5.1**, si se toma en cuenta que el movimiento de la muñeca afecta a los demás componentes.

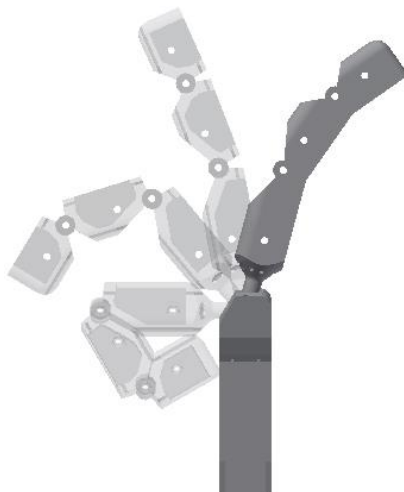


Figura 5.1: Diferentes posiciones del dedo índice del prototipo de mano mecánica desarrollada en tesis anterior [6] para visualizar el rango de movimientos que tiene sus elementos.

Debido a lo anterior es que se presentan dos líneas de trabajo a seguir para aumentar el rango de medición y optimizar el Spatial Tracking. Una de ellas consiste en hacer la representación de la orientación a través de Cuaterniones y con ello evitar el efecto de *gymbal lock* que se da con la representación de Euler y Tait-Bryan (**Sección 2.5.3**). La otra opción es cambiar los sensores utilizados por otros, como por ejemplo encoders, sensores de flexión o un sensor IMU de orientación absoluta como el desarrollado por Bosch que tiene un algoritmo de fusión de datos para los 9 ejes [23]. Así mismo hay muchos otros métodos de seguimiento del movimiento, ya sean indirectos (Magnéticos o Inerciales), Directos (Encoders y potenciómetros) o Espaciales (visuales, magnéticos u ópticos) ([7])

Para este trabajo se fabricaron piezas con forma de anillo en donde van ensamblados los sensores, si bien sirve para su propósito no es cómodo para el usuario. La placa electrónica en la que viene el sensor MPU9250 tiene varios pines que no son utilizados para este trabajo, por lo que se podría

reducir su tamaño de forma considerable y así ser acoplado a un guante, en cada una de las falanges de forma de tener el movimiento de todas las partes de la mano.

La velocidad con que reacciona el sistema es otro punto que se podría mejorar, para ello se sugieren varios caminos a seguir. El primero de ellos y quizás el más obvio, es aumentar la potencia del microcontrolador que se utiliza, que sea capaz de trabajar a una mayor frecuencia lo que influye en la velocidad de ejecución del código. Otro camino es hacer un cambio de protocolo de comunicación con los sensores, cambiar de un bus I2C a bus SPI aumentaría la velocidad de adquisición de datos. Como se mencionó en la sección de resultados, el tiempo de respuesta con un control proporcional entre el ángulo del dedo y la posición del servomotor es casi instantáneo con un desfase muy pequeño por lo que una mezcla entre este tipo de control y un PID podría tener un bajo tiempo de respuesta y a la vez ser preciso. Por último, se da la idea de hacer un estudio de si la paralelización por dedo de las lecturas y control de los actuadores mejoraría el desempeño teniendo en cuenta que el tiempo de ejecución del código aumenta proporcional al número de dedos a controlar.

El prototipo de mano mecánica desarrollado en el trabajo de tesis anterior fue utilizado en un comienzo para realizar el control, pero este presentó detalles al momento de implementarlo. Uno de ellos es la tensión de los cables, estos debían estar tensos para lograr la fuerza necesaria para mantener la posición de la parte de la mano tanto por el lado de flexión como por el de extensión. El problema es que el hilo de pescar utilizado como cable tiene una alta deformación cuando el servomotor aplica fuerza por lo que el sistema no está siempre tenso provocando que las fuerzas de un lado y otro sean distintas. El comportamiento observado al realizar las pruebas con la falange proximal de la mano mecánica fue que la falange alcanzaba correctamente los valores mínimos y máximos, sin embargo resultaba mucho más complejo alcanzar los valores de posiciones angulares intermedias y a medida que se añadían movimientos, más oscilatorias se volvían las respuestas. Se sugiere cambiar el material de los cables a otro que no tenga tanta deformación como, por ejemplo, cuerdas metálicas de guitarra. Como se mencionó anteriormente es importante que los cables estén tensos para mantener las posiciones de los elementos, para asegurar la tensión se sugiere diseñar una pieza que mantenga esta tensión y que además permita hacer un ajuste de esta.

Respecto a la electrónica de la mano mecánica la fuente de poder que se utiliza para alimentar los servomotores emite bastante ruido en la señal de los servomotores, se recomienda utilizar una con menos ruido o hacer un arreglo de fuentes de alimentación de notebook que tienen integrados filtros para eliminar el ruido.

6 Referencias

- [1] «Anatomía Mano,» [En línea]. Available: <https://kiaenzona.com/wp-content/uploads/2014/02/Anatom%C3%ADa-mano.gif>. [Último acceso: 10 12 2018].

- [2] «Articulaciones de la Mano,» [En línea]. Available: <http://www.centralx.es/p/printImage/sistema-musculoesqueletico/esqueleto/articulaciones/articulaciones-de-la-mano.jpg>. [Último acceso: 10 Diciembre 2018].

- [3] R. L. Drake, A. W. VOGL y A. W. M. G. Mitchell, «Gray Anatomía para estudiantes,» Elsevier España, pp. 683-834.

- [4] «Movimiento Dedos,» [En línea]. Available: https://accessphysiotherapy.mhmedical.com/data/books/965/ham22541_f0620.png. [Último acceso: 10 Diciembre 2018].

- [5] «Thumb movements,» [En línea]. Available: <https://boneandspine.com/wp-content/uploads/2016/09/thumb-movements.png>. [Último acceso: 10 Diciembre 2018].

- [6] J. E. Salas, «Desarrollo de mano mecánica para emulación y determinación de movimientos,» p. 80, Octubre 2017.

- [7] S. A. Powell, «A Review of Anthropomorphic Robotic Hand Technology and Data Glove,» pp. 1-2, 2016.
- [8] Z. Yu y J. Gu, «A survey on real-time controlled multi-fingered robotic hand,» pp. 975-980.
- [9] «HackerHand,» [En línea]. Available: http://backyardbrains.cl/products/img/HackerHand2_web.jpg. [Último acceso: 10 Julio 2019].
- [10] «Control lazo Cerrado-Abierto,» [En línea]. Available: <https://hooshmand.net/content/images/size/w1600/2020/05/hooshmand.net---open-vs-closed-loop-exhaust.jpg>. [Último acceso: Abril 2018].
- [11] «Esquema PID,» [En línea]. Available: <https://dewesoft.pro/images/uploads/PIDfront.PNG>. [Último acceso: Abril 2018].
- [12] «Distintos valores constante Proporcional,» [En línea]. Available: https://s3.amazonaws.com/embeddedrelated/user/1/pid_fig9_59578.png. [Último acceso: Abril 2018].
- [13] «Overshoot, Undershoot Oscillation diagram,» [En línea]. Available: <https://oscarliang.com/ctt/uploads/2018/02/pid-overshoot-undershoop-oscillation-diagram-explain.jpg>. [Último acceso: Junio 2018].
- [14] «Distintos Valores constante integral,» [En línea]. Available: https://s3.amazonaws.com/embeddedrelated/user/1/pid_fig12_86567.png. [Último acceso: Junio 2018].

- [15] Invensense, *Invensense MPU9250 Datasheet*, p. 38.
- [16] Wikipedia, «Ángulos de Euler,» Wikipedia, [En línea]. Available: <https://upload.wikimedia.org/wikipedia/commons/8/82/Euler.png>. [Último acceso: 10 Septiembre 2018].
- [17] «Ángulos de navegación,» Wikipedia, [En línea]. Available: https://es.wikipedia.org/wiki/%C3%81ngulos_de_navegaci%C3%B3n. [Último acceso: 10 septiembre 2018].
- [18] «Tutorial MPU6050, Acelerómetro y Giroscopio,» Naylamp Mechatronics, [En línea]. Available: https://naylampmechatronics.com/blog/45_Tutorial-MPU6050-Aceler%C3%B3metro-y-Giroscopio.html. [Último acceso: 10 Septiembre 2018].
- [19] «Complementary Filter,» [En línea]. Available: http://www.pietter-jan.com/images/Complementary_Filter.png. [Último acceso: 25 Agosto 2018].
- [20] F. Chen Chen, A. Favetto, M. Mousavi, E. P. Ambrosio, S. Appendino, A. Battezzato, D. Manfredi, F. Pescarmona y B. Bona, «Human Hand: Kinematics, Statics and Dynamics,» 41st International Conference on Environmental Systems, Portland, Oregon, 2011.
- [21] L. Biagiotti, F. Lotti, C. Melchiorri y G. Vassura, «¿How Far Is the Human Hand? A Review on Anthropomorphic Robotic End-effectors».
- [22] E. Gama, O. Avilés y D. Amaya, «Anthropomorphic robotic hands: a review,» vol. 32, 2014.

- [23] «Adafruit BNO055 Absolute Orientation Sensor -Overview,» [En línea]. Available: https://cdn-shop.adafruit.com/datasheets/BST_BNO055_DS000_12.pdf. [Último acceso: Diciembre 2018].
- [24] «Tecnología MEMS,» [En línea]. Available: <http://www.cmm.org.mx/index.php/microsistemas/tecnologia-mems>. [Último acceso: 6 Diciembre 2017].
- [25] «A sample of MEMS device,» [En línea]. Available: https://www.researchgate.net/profile/Grzegorz_Pittner/publication/244995422/figure/fig1/AS:298266869682177@1448123771167/A-sample-of-MEMS-technology-device.png. [Último acceso: 20 noviembre 2017].
- [26] «Accelerometer,» OMEGA, [En línea]. Available: <https://www.omega.com/literature/transactions/volume3/forcefig3.html>.
- [27] «Esquema acelerómetro capacitivo,» Ingeniería Mecafenix, [En línea]. Available: <https://i1.wp.com/www.ingmecafenix.com/wp-content/uploads/2017/10/Esquema-acelerometro-capacitivo.jpg>. [Último acceso: 20 Mayo 2018].
- [28] M. Pedley, «Tilt Sensing Using a Three-Axis,» *Freescale Semiconductor, Application note*, p. 3, 2013.
- [29] D. o. M. Engineering, «Department of Mechanical Engineering IISC Bangalore,» [En línea]. Available: <http://www.mecheng.iisc.ernet.in/~suresh/me237/Projects2014/Shounak.pdf>. [Último acceso: 14 Junio 2018].

- [30] «Sensor Wiki,» [En línea]. Available: <https://sensorwiki.org/sensors/gyroscope>. [Último acceso: 14 Junio 2018].
- [31] «Cinemática Inversa Directa,» [En línea]. Available: <http://wiki.robotica.webs.upv.es/wiki-de-robotica/cinematica/>. [Último acceso: Diciembre 2017].
- [32] Wikipedia, «Planar three link manipulator,» [En línea]. Available: <https://upload.wikimedia.org/wikipedia/commons/c/c5/Planar-three-link-manipulator.svg>. [Último acceso: Diciembre 2018].
- [33] A. Barrientos y L. F. Peñin, Fundamentos de Robótica, 2da edición ed., pp. 94 - 107.
- [34] «Componentes Arduino Uno,» [En línea]. Available: <https://www.robomart.com/image/catalog/RM0058/02.jpg>. [Último acceso: Mayo 2018].
- [35] «Sensor MPU9250,» [En línea]. Available: <https://i.ebayimg.com/images/g/szEAAOSwJ2pbuwO7/s-l640.jpg>. [Último acceso: Mayo 2018].
- [36] «Placa TCA9548A,» [En línea]. Available: https://cdn10.bigcommerce.com/s-vt19phz/product_images/uploaded_images/adafruit-tca9548a-i2c-multiplexer-breakout-board-from-tronixlabs-australia-500.jpg?t=1442454117. [Último acceso: Mayo 2018].
- [37] «Servo CYS S8218,» [En línea]. Available: <https://ae01.alicdn.com/kf/HTB1nhd5IXXXXXaAXVXXq6xXFXXXG/223482262/HTB1nhd5IXXXXXaAXVXXq6xXFXXXG.jpg>. [Último acceso: Diciembre 2018].

- [38] «Esquema conexion Bus I2C,» [En línea]. Available: <http://microcontrollers.pro/wp-content/uploads/2017/11/Tutorial-I2C.jpg>. [Último acceso: Diciembre 2017].
- [39] «Bus I2C Funcionamiento,» [En línea]. Available: <https://www.luisllamas.es/wp-content/uploads/2016/05/arduino-bus-i2c-funcionamiento.png>. [Último acceso: Diciembre 2017].
- [40] «DfRobot,How to Use a Three-Axis Accelerometer for Tilt Sensing,» DFRobot, [En línea]. Available: https://www.dfrobot.com/wiki/index.php/How_to_Use_a_Three-Axis_Accelerometer_for_Tilt_Sensing. [Último acceso: Abril 2018].
- [41] «Digital Implementation of Complementary Filter for Attitude Estimation with Low-Cost Accelerometer and Gyroscope,» FK Engineering blog, [En línea]. Available: <https://fkeng.blogspot.com/2018/05/digital-implementation-of-complementary.html>. [Último acceso: Septiembre 2018].
- [42] «A short survey of sensor fusion solutions,» Kriswiner GitHub Wiki, [En línea]. Available: https://github.com/kriswiner/EM7180_SENtral_sensor_hub/wiki/A.-A-Short-Survey-of-Sensor-Fusion-Solutions. [Último acceso: Junio 2018].

Anexos

A. Descripción Prototipo de Mano Mecánica

La mano robótica desarrollada en el proyecto de tesis anterior [6] y sobre el cual se iba a realizar el control en un comienzo, se encuentra sobre una estructura de 8 bases de madera situadas de forma vertical, una sobre la otra como se muestra en la **Figura A.1****Error! Reference source not found.**. En cada una de estas bases se disponen, de forma circular, servomotores que tensionan hilos de pescar, simulando los tendones. En total, son 47 servomotores que a través de la tensión sobre los tendones permiten los movimientos en cada una de las partes de la mano a la que están conectadas.

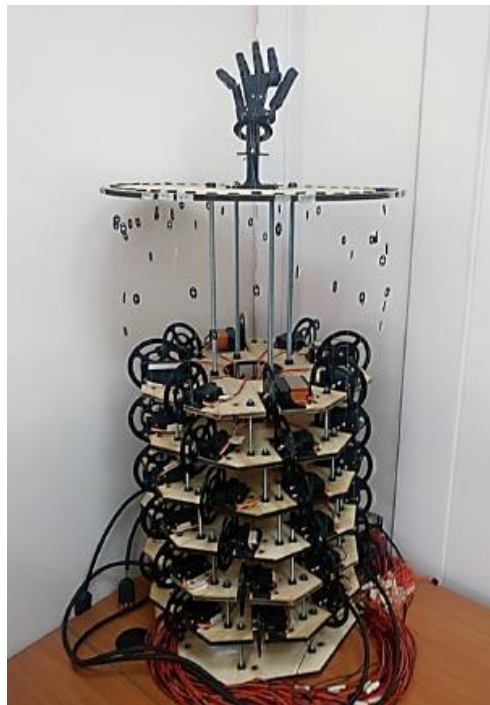


Figura A.1: Prototipo Mano Robótica desarrollada por Jorge Salas como memoria del departamento de ingeniería mecánica de la Universidad Técnica Federico Santa María [6]

La mano robótica se puede dividir en 5 secciones: Palma, dedo índice, medio, anular, meñique y pulgar (**Figura A.2**). Los metacarpos del dedo índice, medio y anular están representados por una sola pieza que representa la palma, en el caso del meñique, el hueso metacarpiano es una pieza aparte conectada a la palma, ya que permite hacer el movimiento de oposición.

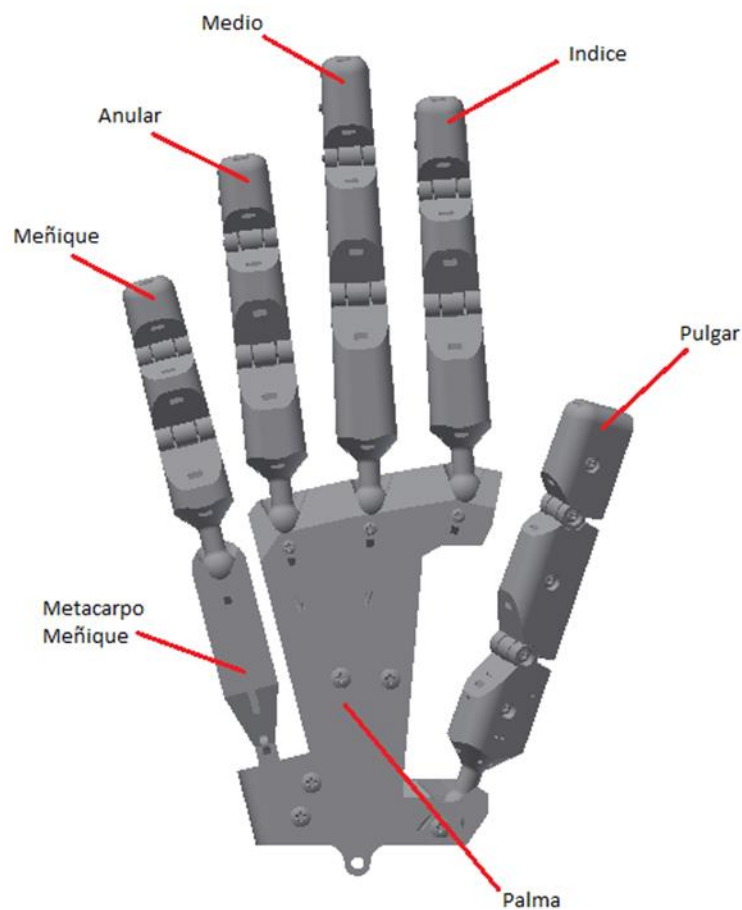


Figura A.2: Ensamble de las partes de la mano robótica sin los elementos de sujeción.

Cada dedo está compuesto por la falange proximal, medial y distal, en el caso del pulgar también se considera este mismo diseño ya que “*el hueso metacarpiano, falange proximal y falange distal se pueden aproximar a las falanges proximal, medial y distal*” [6].

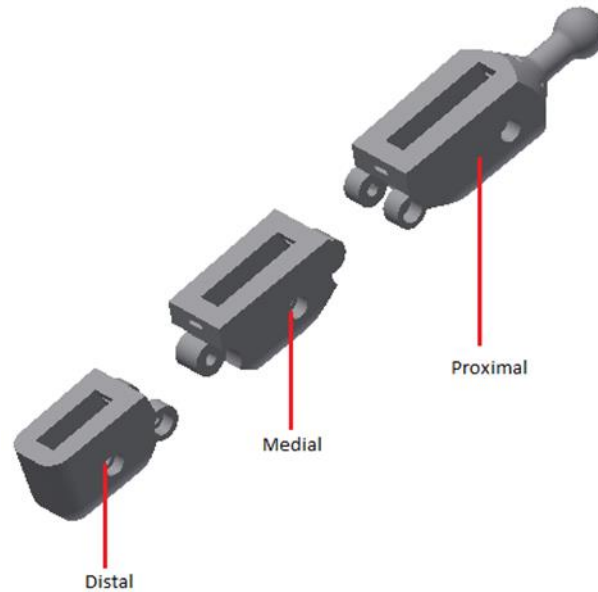


Figura A.3: Vista explosionada del ensamble de los dedos de la mano robótica

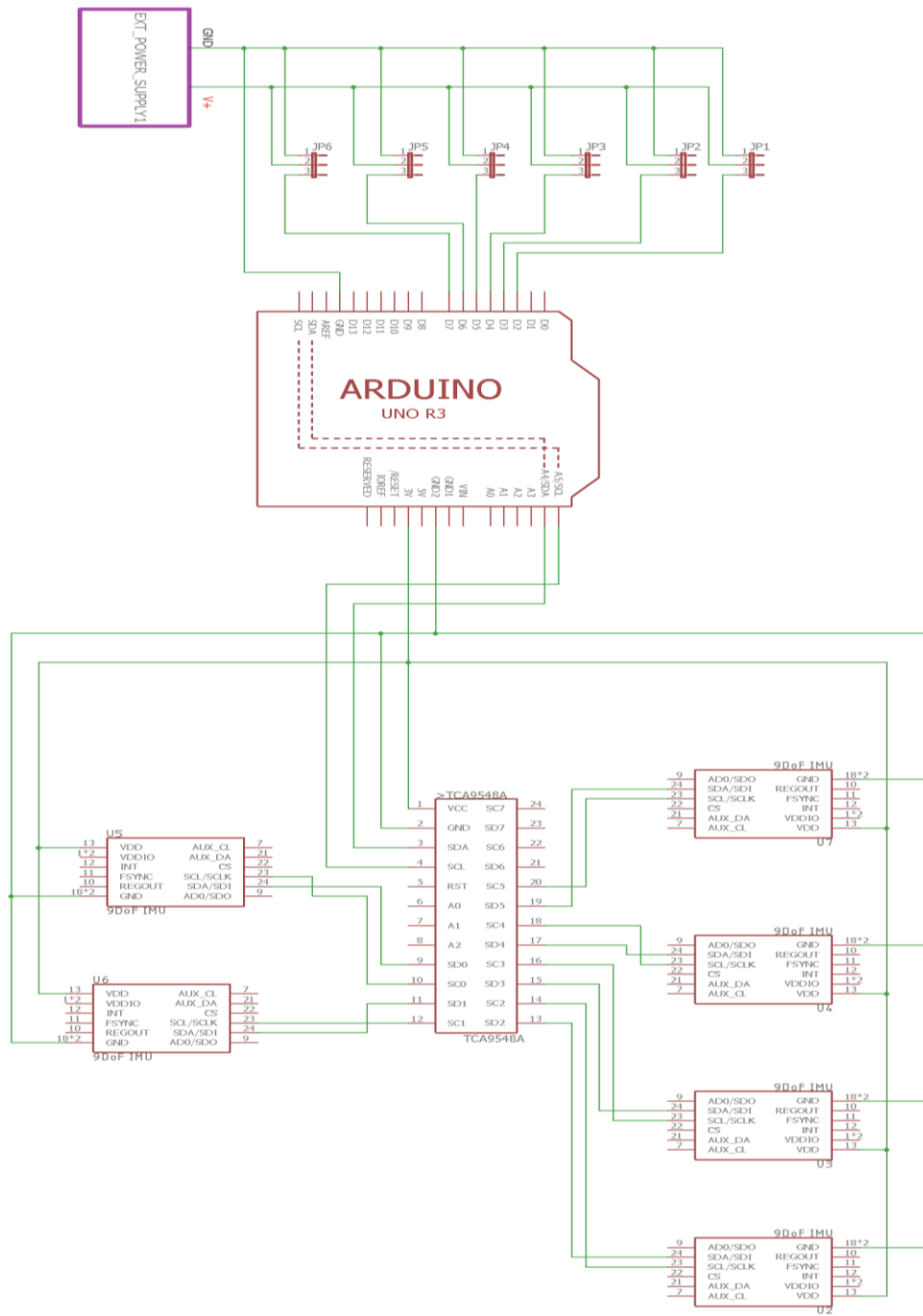
Como se puede observar en la **Figura A.3** la falange proximal de los dedos presenta una esfera en la parte inferior, esta esfera es la que permite hacer el ensamble del dedo en la parte superior del metacarpo y la que permite los movimientos de flexión, extensión, abducción y aducción.

Finalmente, la mano es sostenida por una estructura que cuenta con dos discos por donde pasan los tendones como se muestra en la **Figura A.4**. El disco superior tiene la capacidad de flexionarse en conjunto con el resto de la mano mientras que el inferior no tiene movimiento y sirve de guía para los tendones que van desde la base superior de madera hasta las partes de la mano.



Figura A.4: Vista isométrica del ensamble de la estructura que sostiene la muñeca con la mano

B. Diagrama de conexión para mano robótica completa



C. Definición sensores micro electro-mecánicos

Las tecnologías MEMS (Micro-Electro-Mechanical Systems) hacen referencia a “*dispositivos mecánicos y electromecánicos miniaturizados que se realizan con técnicas de micro fabricación*” [24]. Las dimensiones de estos elementos pueden variar bastante, pero tienen un rango aproximado entre 1 a 100 [μm], y están compuestos por estructuras simples de un elemento, o estructuras más complejas con varios elementos. El criterio principal para que un dispositivo sea considerado MEMS es que alguno de sus componentes tenga alguna funcionalidad mecánica, independiente de si se pueden mover o no.

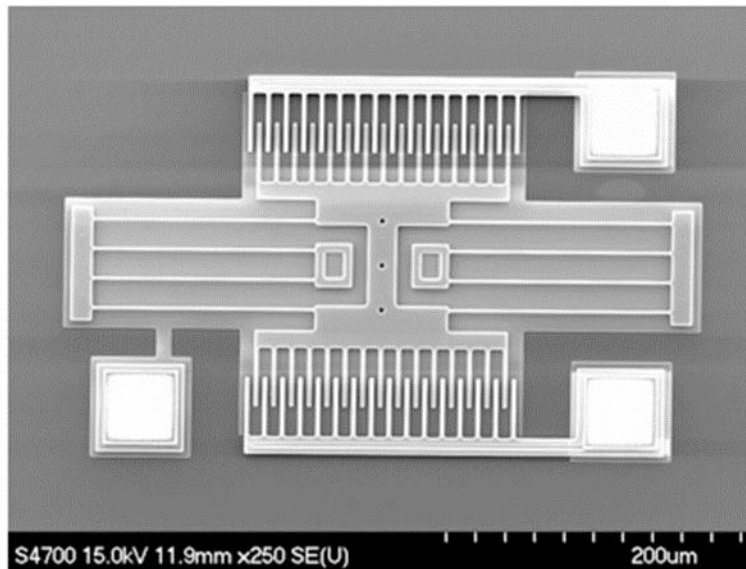


Figura C.1: Resonador de superficie micro mecanizada desarrollado por la empresa MNX. Este dispositivo puede ser utilizado como un microsensor y como un micro actuador [25]

D. Definición Acelerómetro

Un acelerómetro es un dispositivo electromecánico que mide la aceleración de las fuerzas, o, dicho de otro modo, el cambio de velocidad en el tiempo. Estas fuerzas pueden ser estáticas, como la fuerza de gravedad, o pueden ser dinámicas producto del movimiento o vibración del acelerómetro. Existen distintos tipos de tecnologías para realizar la medición de la aceleración, pero la selección de una u otra depende de la aplicación a la que esté destinado.

D1. Acelerómetro Mecánico

Existen distintos tipos de acelerómetros, el más básico es el acelerómetro mecánico constituido por una masa unida a un dinamómetro cuyo eje está en la misma dirección que la aceleración que se desea medir. Al actuar una fuerza sobre la masa, se puede obtener el módulo de aceleración al dividir la fuerza entre la masa, ambos, datos conocidos.

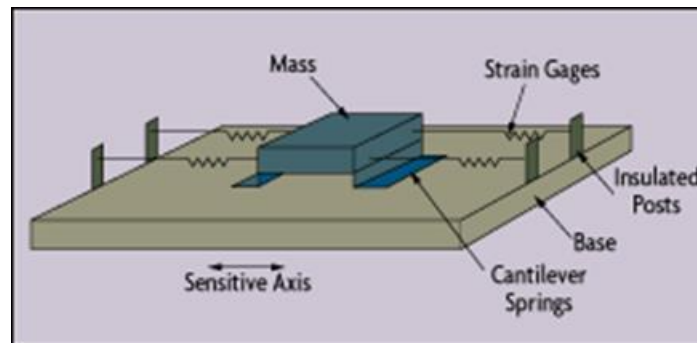


Figura D.1: Esquema Acelerómetro mecánico [26]

D2. Acelerómetro Piezoeléctrico

Otro tipo de acelerómetro es el basado en el efecto piezoeléctrico, utiliza un material que al ser sometido a tensiones mecánicas genera una diferencia de potencial en su superficie, con lo que se obtiene una señal eléctrica proporcional a la presión aplicada en el material. En la **Figura D.2** se muestra un esquema de un acelerómetro piezoeléctrico.

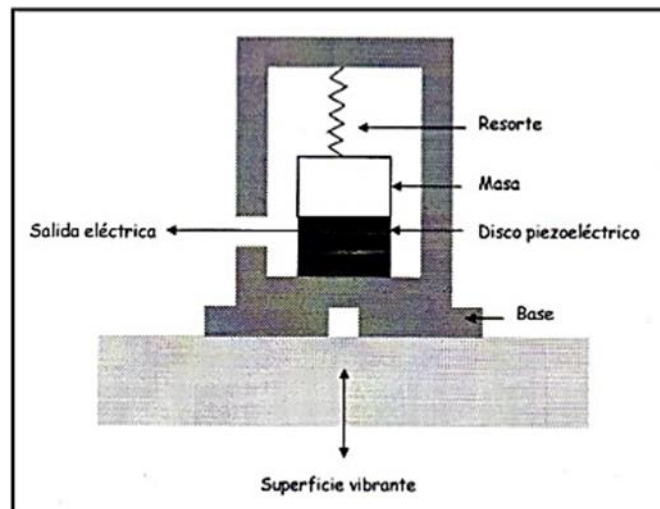


Figura D.2: Esquema general de componentes de un acelerómetro basado en materiales piezoeléctricos

D3. Acelerómetro capacitivo

Está compuesto por pequeñas placas que actúan como condensadores. Cuando el dispositivo es sometido a aceleraciones estas placas modifican su distancia entre ellas, y con ello, varían su capacidad. Los acelerómetros se construyen de forma tal que una de sus placas se encuentra fija y la otra puede desplazarse, como se muestra en la **Figura D.3**.

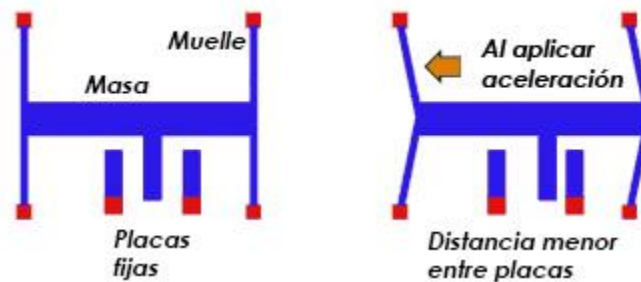


Figura D.3: Esquema de funcionamiento de un acelerómetro capacitivo. Este funcionamiento es el que se utiliza en sensores MEMS [27]

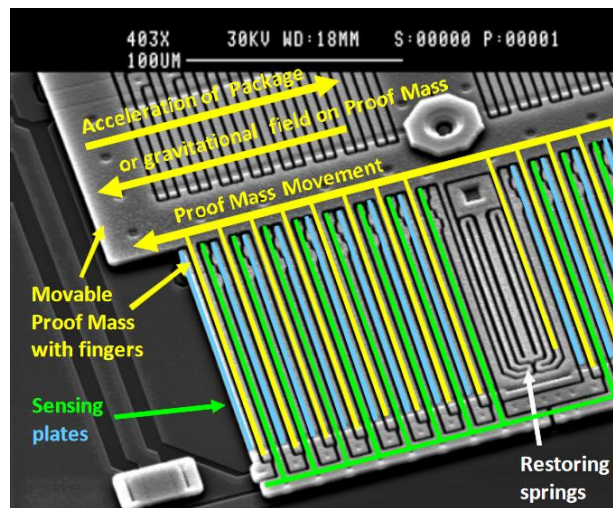


Figura D.4: Esquema de funcionamiento de un acelerómetro capacitivo. Este funcionamiento es el que se utiliza en sensores MEMS [28]

E. Giroscopios

Los giroscopios son dispositivos capaces de medir la velocidad angular de la estructura sobre la que están montados. Esta velocidad es entregada por lo general en valores de [$^{\circ}/s$] o en RPS (revoluciones por segundo) y es posible encontrar giroscopios capaces de medir la velocidad angular en 1, 2, y 3 ejes. Existen distintos tipos de giroscopios:

E1. Giroscopio Rotativo

Es el diseño de giroscopio más básico y utiliza el principio de conservación del momento angular. Este sistema mecánico consiste en un disco giratorio en un eje que es montado sobre una estructura que tiene libertad para rotar sobre uno o dos ejes, dependiendo de la cantidad de ejes que pueden rotar es la cantidad de ejes en los que va a poder hacer medición de la velocidad angular.

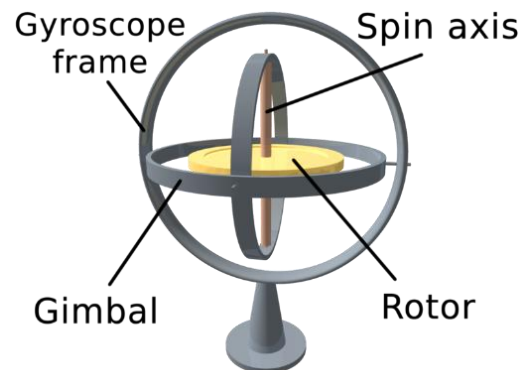


Figura E.1: Representación el mecanismo de un giroscopio rotativo [29]

E2. Giroscopio de estructura vibratoria

Estos giroscopios contienen una masa micro mecanizada conectada a una cubierta externa mediante resortes. A través del principio de que un objeto que está vibrando tiende a seguir vibrando en el mismo plano, si la plataforma sobre la que se encuentra rota, el objeto experimentara una aceleración de Coriolis.

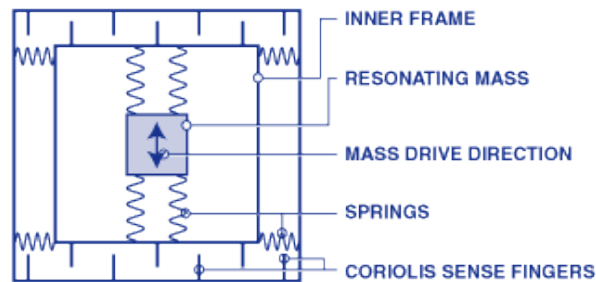


Figura E.2: Esquema de la estructura vibratoria que va sobre la plataforma que gira [30]

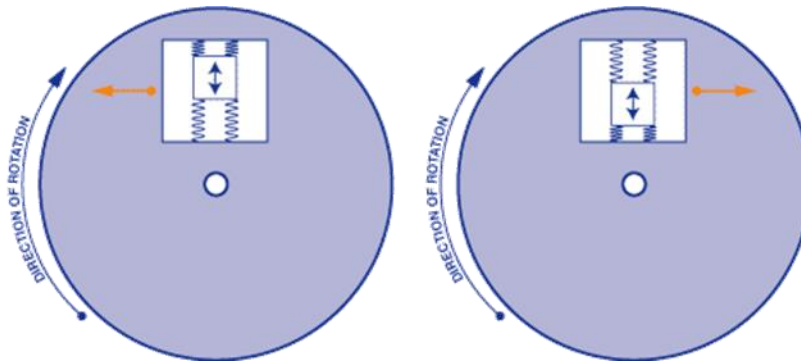


Figura E.3: Representación de la fuerza experimentada por la estructura vibratoria cuando gira la plataforma [30]

E3. Giroscopio Óptico

Fueron desarrollados después del descubrimiento de la tecnología láser y una de sus grandes diferencias con los otros tipos de giroscopios, es que no utiliza partes móviles además de no basarse en la conservación del momentum angular, en vez de ello, se basan en la luz y en el efecto Sagnac. Se puede explicar este fenómeno de forma fácil para el caso de un círculo. Una fuente de luz es posicionada en un círculo emitiendo dos haces de luz en direcciones opuestas. Si la fuente de luz permanece quieta, ambos haces de luz requieren la misma cantidad de tiempo para recorrer el perímetro del círculo. Si la fuente está rotando en el círculo, entonces el tiempo que demoran ambos haces de luz en recorrer el perímetro del círculo será distinto tal como se muestra en la **Figura E.4**, lo anterior también se cumple para cualquier superficie cerrada, como el sistema de la **Figura E.5**.

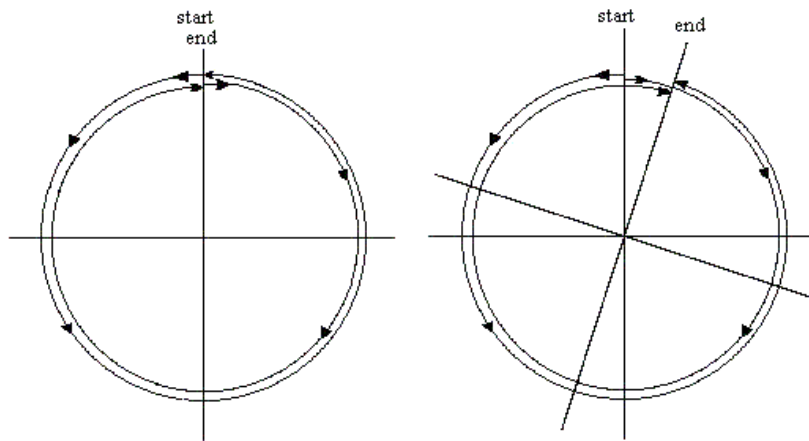


Figura E.4: Diagrama explicativo efecto Sagnac en un círculo [30]

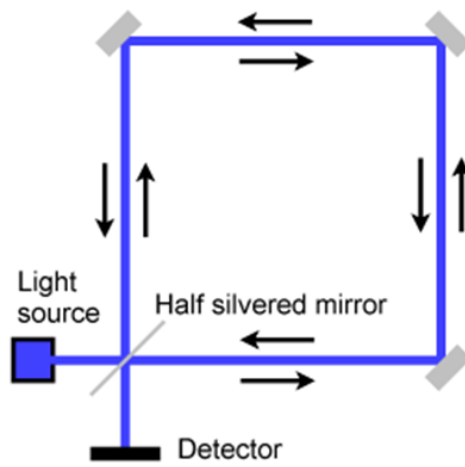


Figura E.5: Diagrama funcionamiento giroscopio óptico [25]

F. Cinemática de un robot

La cinemática de un robot es el estudio de los movimientos respecto de un sistema de referencia, sin tomar en cuenta las fuerzas/torques que causan el movimiento. En este estudio se determinan las relaciones entre la posición, velocidad y aceleración de cada una de sus uniones (links) para planear y controlar el movimiento de cada actuador. La cinemática relaciona la posición y orientación del extremo del robot (end effector) con los valores de sus coordenadas articulares, dependiendo de los datos disponibles para hacer el movimiento será necesario realizar un esquema cinemático directo o uno inverso.



Figura F.1: Esquema del problema cinemático directo e inverso. Si se tiene el valor del movimiento que realizan los elementos del robot, se puede obtener la posición y orientación del extremo del robot (cinemática directa), al revés, si se cuenta con la información de la posición, se pueden obtener los movimientos que tiene que realizar cada uno de sus elementos para llegar a esa posición [31]

F1. Cinemática Directa

La resolución del problema cinemático directo consiste en encontrar la posición y orientación del extremo final del robot, con respecto a un sistema de referencia, conocidos los valores de los movimientos de traslación o rotación que realiza cada uno de sus links, y los parámetros geométricos de los elementos del robot. En una cadena cinemática serial la solución para este tipo de problema siempre es única, es decir, dado un conjunto de vectores estos siempre corresponderán a una única posición del actuador.

Como se mencionó anteriormente, un robot está formado por eslabones (links) unidos por articulaciones. A cada eslabón se le puede asignar un sistema de referencia respecto a él mismo y, utilizando transformaciones homogéneas, es posible representar las rotaciones y traslaciones relativas entre los distintos eslabones del robot. Para ello se define previamente el concepto de coordenadas y matrices homogéneas.

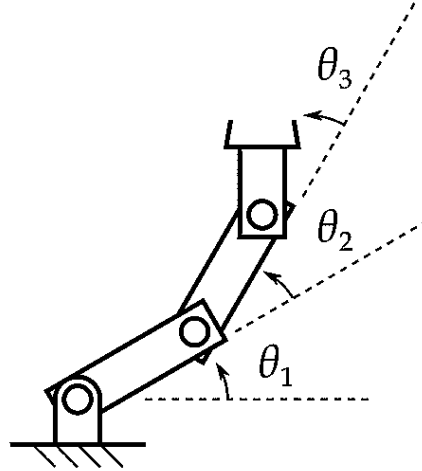


Figura F.2: Representación de robot plano de 3 grados de libertad con los valores de movimiento en sus articulaciones [32]

F2. Coordenadas y matrices homogéneas

La representación conjunta de la posición y orientación (localización) de un sólido en el espacio se puede realizar mediante coordenadas homogéneas. Un espacio n -dimensional se encuentra representado en coordenadas homogéneas por $(n+1)$ dimensiones, de tal forma que un vector $\mathbf{p}(x, y, z)$ vendrá representado por $\mathbf{p}(wx, wy, wz, w)$, donde w tiene un valor arbitrario y representa un factor de escala. Una matriz de transformación homogénea \mathbf{T} es una matriz de 4×4 que representa la transformación de un vector de coordenadas homogéneas de un sistema de coordenadas a otro.

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

En robótica generalmente solo interesara conocer el valor de la submatriz \mathbf{R} y \mathbf{p} , considerándose las componentes de \mathbf{f} nulas y la de \mathbf{w} la unidad. [33]

F3. Resolución problema cinemático directo mediante matriz de transformación homogénea

Considérese la matriz de transformación homogénea ${}^{i-1}\mathbf{A}_i$ que representa la posición y orientación relativa entre los distintos sistemas de ejes coordenados, asociados a dos eslabones consecutivos. Entonces, la matriz ${}^0\mathbf{A}_1$ describe la posición y orientación del sistema de referencia solidario al primer eslabón con respecto al sistema de referencia solidario a la base, así mismo ${}^1\mathbf{A}_2$ describe la posición y orientación del segundo eslabón respecto del primero, etc. Cuando se consideran todos los grados de libertad del robot, la matriz ${}^0\mathbf{A}_n$, que es el producto de las matrices de transformación de cada eslabón, se le denomina \mathbf{T} matriz de transformación que relaciona la posición y orientación del extremo final del robot respecto del sistema fijo situado en la base del mismo.

Para el caso de un robot de 6 grados de libertad, la matriz \mathbf{T} vendría dada de la forma:

$$\mathbf{T} = {}^0\mathbf{A}_n = {}^0\mathbf{A}_1 \cdot {}^1\mathbf{A}_2 \cdot {}^2\mathbf{A}_3 \cdot {}^3\mathbf{A}_4 \cdot {}^4\mathbf{A}_5 \cdot {}^5\mathbf{A}_6$$

F4. Representación de Denavit-Hartenberg (D-H)

Se utiliza para establecer la ubicación de los sistemas de referencia de los eslabones permitiendo describir la relación que existe entre dos sistemas de referencia asociados a eslabones. La representación de D-H permite pasar de un sistema de coordenadas a otro mediante 4 transformaciones básicas que dependen exclusivamente de las características geométricas del eslabón.

Estas transformaciones básicas consisten en una sucesión de rotaciones y traslaciones que permitan relacionar el sistema de referencia del elemento i con el sistema del elemento $i-1$. Estas transformaciones, denominadas parámetros D-H del eslabón i , son:

- 1) Rotación alrededor del eje Z_{i-1} un ángulo θ_i
- 2) Traslación a lo largo de Z_{i-1} una distancia d_i
- 3) Traslación a lo largo de x_i una distancia a_i
- 4) Rotación alrededor del eje x_i un ángulo α_i

*Nótese que los parámetros θ_i y d_i están asociados a las características de la articulación, mientras que los parámetros a_i y α_i están asociados a la geometría del brazo.

El producto de estas transformaciones entrega la matriz de transformación que relaciona los sistemas de referencia $\{S_{i-1}\}$ y $\{S_i\}$. Cabe destacar que, como el producto entre matrices no es conmutativo, las transformaciones se deben realizar en el orden indicado, por lo que se tiene:

$${}^{i-1}A_i = T(z, \theta) \cdot T(0, 0, d_i) \cdot T(a_i, 0, 0) \cdot T(x, \alpha_i)$$

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

Algoritmo de resolución de D-H

- 1) Numerar los eslabones comenzando con 1 (primer eslabón móvil de la cadena) y acabando con n (último eslabón móvil). Se numerará como eslabón 0 a la base fija del robot.
- 2) Numerar cada articulación comenzando por 1 (la correspondiente al primer grado de libertad) y acabando en n .

- 3) Localizar el eje de cada articulación. Si esta es rotativa, el eje será su propio eje de giro. Si es prismática, será el eje a lo largo del cual se produce desplazamiento.
- 4) Para i de 0 a $n - 1$ situar el eje \mathbf{z}_i sobre el eje de la articulación $i + 1$.
- 5) Situar el origen del sistema de la base $\{S_0\}$ en cualquier punto del eje \mathbf{z}_0 . Los ejes \mathbf{x}_0 e \mathbf{y}_0 se situarán de modo que formen un sistema de dextrógiro (en sentido de las manillas del reloj).
- 6) Para i de 1 a $n - 1$, situar el sistema $\{S_i\}$ (solidario al eslabón i) en la intersección del eje \mathbf{z}_0 con la línea normal común a \mathbf{z}_{i-1} y \mathbf{z}_i . Si ambos ejes se cortasen se situaría $\{S_i\}$ en el punto de corte. Si fuesen paralelos $\{S_i\}$ se situaría en la articulación $i + 1$.
- 7) Situar \mathbf{x}_i en la línea normal común a \mathbf{z}_{i-1} y \mathbf{z}_i .
- 8) Situar \mathbf{y}_i de modo que forme un sistema dextrógiro con \mathbf{x}_i y \mathbf{z}_i .
- 9) Situar el sistema $\{S_n\}$ en el extremo del robot de modo que \mathbf{z}_n coincida con la dirección de \mathbf{z}_{n-1} y \mathbf{z}_n .
- 10) Obtener θ_i como el ángulo que hay que girar en torno a \mathbf{z}_{i-1} para que \mathbf{x}_{i-1} y \mathbf{x}_i queden paralelos.
- 11) Obtener d_i como la distancia, medida a lo largo de \mathbf{z}_{i-1} , **que habría que desplazar $\{S_{i-1}\}$** para que \mathbf{x}_{i-1} y \mathbf{x}_i quedasen alineados.
- 12) Obtener a_i como la distancia medida a lo largo de \mathbf{x}_i (que ahora coincidiría con \mathbf{x}_{i-1}) que habría que desplazar el nuevo $\{S_{i-1}\}$ para que su origen coincidiese con $\{S_i\}$.
- 13) Obtener α_i como el ángulo que habría que girar en torno a \mathbf{x}_i (que ahora coincidiría con \mathbf{x}_{i-1}) para que el nuevo $\{S_{i-1}\}$ coincidiese totalmente con $\{S_i\}$.
- 14) Obtener las matrices de transformación ${}^{i-1}\mathbf{A}_i$ para cada elemento.
- 15) Obtener la matriz de transformación que relaciona el sistema de la base con el del extremo del robot $\mathbf{T} = {}^0\mathbf{A}_1 \cdot {}^1\mathbf{A}_2 \dots {}^{n-1}\mathbf{A}_n$
- 16) La matriz \mathbf{T} define la orientación (submatriz de rotación) y posición (submatriz de traslación) del extremo referido a la base en función de las n coordenadas articulares.

G. Hardware Utilizado

- Microcontrolador Arduino UNO

Arduino UNO es una placa electrónica “Open Source” basada en el microcontrolador ATmega328p. Esta placa permite interactuar con sensores y controlar actuadores a través de la programación del microcontrolador por medio del IDE de Arduino. La placa UNO cuenta con 14 entradas/salidas digitales y 6 entradas analógicas, puede establecer comunicación por bus I2C, Serial, y SPI con otros periféricos, y la comunicación con el computador se realiza a través de puerto USB. A continuación, se muestra una tabla con sus características:

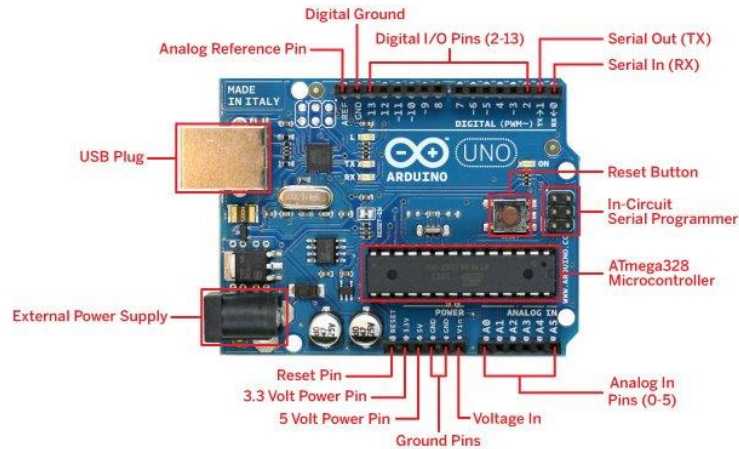


Figura G.1: Vista superior de la placa Arduino UNO con sus componentes [34]

Tabla G.1: Características placa Arduino UNO

Arduino UNO	
Voltaje de operación	5 [V]
Voltaje de alimentación	7 – 12 [V]
I/O Digitales	14
I/O PWM Digitales	6
Entradas Analógicas	6
Corriente DC por I/O Pin	20 [mA]
Velocidad de Reloj	16 [MHz]
Puertos de Comunicación	USB, Serial, I2C y SPI

- Sensor MEMS MPU9250

El MPU9250 es un sensor IMU de 9 ejes, compuesto de un acelerómetro, un giroscopio y un magnetómetro. Combina el sensor MPU6050 (acelerómetro y giroscopio) con el sensor AK8963 que es un magnetómetro de 3 ejes. Este sensor contiene una unidad DMP (digital Motion Processor) que es capaz de realizar la fusión de datos de los sensores. La comunicación con el sensor puede ser realizada por bus SPI o I2C y la placa viene integrada con resistencias pull up para utilizar directamente cualquiera de estos dos protocolos.



Figura G.2: Sensor MPU9250 [35]

- Multiplexor I2C TCA9548A

Permite conectar al bus I2C varios dispositivos que tienen la misma dirección, se pueden conectar hasta 8 dispositivos. El multiplexor puede variar su dirección entre 0x70 y 0x77 soldando los pines A0, A1 y A2, también trae resistencias pull up incluidas las que pueden ser cortadas en caso de que se desee colocar valores distintos de resistencias.



Figura G.3: Multiplexor I2C TCA9548A, distribuido por Adafruit [36]

- Servomotor CYS8218

Servomotor de engranajes metálicos, de gran tamaño y con un torque máximo de $38 \left[\frac{kg}{cm} \right]$ utilizado para agregar el movimiento de muñeca a la mano robótica. Sus características son:

- Peso: 164g
- Tamaño: 59.5 X 29.0 x 55.2 mm
- Velocidad: 0.20seg/60° (6V)
- 0.18seg/60° (7.4V)
- Torque: 36kg*cm (6V)
- 36-38kg*cm (7.4V)



Figura G.4: Servomotor CYS S8218 utilizado para agregar movimiento flexión-extensión en la muñeca [37]

H. Códigos de Programación

H1. Código Control Arduino

```
#include <WSWire.h>

#include <Servo.h>

//===== Variables Servos =====

Servo Indice;
#define SERVOMIN 10 // this is the 'minimum' angle
#define SERVOMAX 178 // this is the 'maximum' angle that a servo can
reach

#define SERVOMIN_MUNECA 50 // this is the 'minimum' angle
#define SERVOMAX_MUNECA 130 // this is the 'maximum' angle that a servo
can reach

float PosServo;

//===== Variables comunicación I2C =====

#define MPU9250_ADDRESS_MR 0x68 //Define valores de
variables para luego llamarlas
#define MPU9250_ADDRESS_MH 0x69

#define GYRO_FULL_SCALE_250_DPS 0x00
#define GYRO_FULL_SCALE_500_DPS 0x08
#define GYRO_FULL_SCALE_1000_DPS 0x10
#define GYRO_FULL_SCALE_2000_DPS 0x18

#define ACC_FULL_SCALE_2_G 0x00
#define ACC_FULL_SCALE_4_G 0x08
#define ACC_FULL_SCALE_8_G 0x10
#define ACC_FULL_SCALE_16_G 0x18

//===== Variables Globales =====

float dato_MR[6]; // Array donde se guardan los datos del acelerómetro y
del giroscopio para que sean devueltos por la función
float dato_MH[6];

float accel_MR[3] ;
float gyro_MR[3];

float accel_MH[3] ;
```

```

float gyro_MH[3];

float ref=50;
float error=0;
float U;

boolean primeravezAng_MH=true;
boolean primeravezAng_MR=true;
boolean primeravezPID=true;
boolean primeravezLoop=true;

int n=0; //contador
unsigned long tiempo_f;

//// Variables Función Angle \\\

float NewAngle_MR=0;
float DeltaAngle_MR=0;
float DisplayAngle_MR=0;
float Angulo_MR=0;

float NewAngle_MH=0;
float DeltaAngle_MH=0;
float DisplayAngle_MH=0;
float Angulo_MH=0;

//=====Funcion auxiliar lectura=====

void I2Cread(uint8_t Address, uint8_t Register, uint8_t Nbytes, uint8_t*
Data)
{
    Wire.beginTransaction(Address);
    Wire.write(Register);
    Wire.endTransmission();

    Wire.requestFrom(Address, Nbytes);
    uint8_t index = 0;
    while (Wire.available())
        Data[index++] = Wire.read();
}

//=====Funcion auxiliar de escritura, configura
acelerómetro y giroscopio=====

void I2CwriteByte(uint8_t Address, uint8_t Register, uint8_t Data)
{
    Wire.beginTransaction(Address);
    Wire.write(Register);
    Wire.write(Data);
}

```

```

    Wire.endTransmission();
}

//===== Funcion WaitForInput =====

#define WAITFORINPUT() { \
    while(!Serial.available()){ \
        lectura_MR(); \
        NewAngle_MR=Calculo_ang_MR(gyro_MR, accel_MR); \
        lectura_MH(); \
        NewAngle_MH=Calculo_ang_MH(gyro_MH, accel_MH); \
        Angle_MR(); \
        Angle_MH(); \
    }; \
    while(Serial.available()){ \
        Serial.read(); \
    }; \
}

//===== SETUP=====

void setup()
{
    Wire.begin();
    Serial.begin(2000000);

    //===== Configurar acelerometro =====

    I2CwriteByte(MPU9250_ADDRESS_MR, 28, ACC_FULL_SCALE_16_G);
    //Configurado para tener la menor sensibilidad y mayor rango
    I2CwriteByte(MPU9250_ADDRESS_MH, 28, ACC_FULL_SCALE_16_G);

    //===== Configurar giroscopio =====

    I2CwriteByte(MPU9250_ADDRESS_MR, 27, GYRO_FULL_SCALE_2000_DPS);
    //Configurado para tener la menor sensibilidad y mayor rango
    I2CwriteByte(MPU9250_ADDRESS_MH, 27, GYRO_FULL_SCALE_2000_DPS);

    //===== Inicialización Servos =====

    Indice.attach(5);
    motormoove0();
}

void loop()
{
    while (n<=100)
    {
        lectura_MR(); //Devuelve datos de giroscopio y acel después de
        sensibilidad

```

```

NewAngle_MR=Calculo_ang_MR(gyro_MR, accel_MR);
lectura_MH();
NewAngle_MH=Calculo_ang_MH(gyro_MH, accel_MH);
Angle_MR();
Angle_MH();

// Serial.print("Ref: ");
// Serial.print(ref);
// Serial.print(" \t ");
// Serial.print("DispAng_MR= ");
// Serial.print(DisplayAngle_MR);
// Serial.print(" \t ");
// Serial.print("DispAng_MH= ");
// Serial.print(DisplayAngle_MH);
// Serial.print(" \t ");
n=n+1;
// Serial.println();
}

if (primeravezLoop==true)
{
    //Serial.println("Reset display Angle MR");
    WAITFORINPUT();
    //delay(5000);
    DisplayAngle_MR=0;
    DisplayAngle_MH=0;
    primeravezLoop=false;
}

lectura_MR(); //Devuelve datos de giroscopio y acel después de
sensibilidad
NewAngle_MR=Calculo_ang_MR(gyro_MR, accel_MR);
lectura_MH();
NewAngle_MH=Calculo_ang_MH(gyro_MH, accel_MH);
Angle_MR();
Angle_MH();

error=DisplayAngle_MH-DisplayAngle_MR;

// Serial.print("Ref: ");
// Serial.print(ref);
// Serial.print(" \t ");
// Serial.print("DispAng_MR= ");
// Serial.print(DisplayAngle_MR);
// Serial.print(" \t ");
// Serial.print("DispAng_MH= ");
// Serial.print(DisplayAngle_MH);
// Serial.print(" \t ");
// Serial.print("error= ");
// Serial.print(error);
// Serial.print(" \t ");

```

```

        U=PID();

//      Serial.print("U: ");
//      Serial.print(U);
//      Serial.print(" \t ");

        tiempo_f=millis();

        if (abs(error)>1)
        {
            motormoove1();
        }

        //Serial.println();
    }

void motormoove0()
{

    Indice.write(SERVOMAX);
    PosServo=SERVOMAX;
}

void Angle_MH()

{

    DeltaAngle_MH=Angulo_MH-NewAngle_MH;
    DisplayAngle_MH=(DisplayAngle_MH+DeltaAngle_MH);
    Angulo_MH=NewAngle_MH;
}

void Angle_MR()

{

    DeltaAngle_MR=Angulo_MR-NewAngle_MR;
    DisplayAngle_MR=(DisplayAngle_MR+DeltaAngle_MR);
    Angulo_MR=NewAngle_MR;
}

float Calculo_ang_MH(float G[3],float A[3])
{
    float ang;
    float gain=0.92;
    static unsigned long PreviousTimeAng_MH;
    static float PreviousAngle_MH;
    unsigned long CurrentTime_MH;

```



```

float deltaT_MH;

if (primeravezAng_MH==true)
{
    PreviousTimeAng_MH=0;
    PreviousAngle_MH=0;
    primeravezAng_MH=false;
}

CurrentTime_MH=millis();
deltaT_MH=(CurrentTime_MH-PreviousTimeAng_MH)/1000; //Tiempo está en
milisegundos

float ang_gyro=(G[0]*deltaT_MH+PreviousAngle_MH);
float ang_accel=atan(A[1]/sqrt(pow(A[0],2) +
pow(A[2],2)))*(180.0/3.14); //Para llevar a grados

ang=round(gain*(ang_gyro)+(1-gain)*ang_accel);

PreviousTimeAng_MH=CurrentTime_MH;
PreviousAngle_MH=ang;

ang=ang*(-1);
return ang;
}

float Calculo_ang_MR(float G[3],float A[3])
{
    float ang;
    float gain=0.92;
    static unsigned long PreviousTimeAng_MR;
    static float PreviousAngle_MR;
    unsigned long CurrentTime_MR;
    float deltaT_MR;

    if (primeravezAng_MR==true)
    {
        PreviousTimeAng_MR=0;
        PreviousAngle_MR=0;
        primeravezAng_MR=false;
    }

    CurrentTime_MR=millis();
    deltaT_MR=(CurrentTime_MR-PreviousTimeAng_MR)/1000; //Tiempo está en
milisegundos

    float ang_gyro=(G[0]*deltaT_MR+PreviousAngle_MR);
    float ang_accel=atan(A[1]/sqrt(pow(A[0],2) +
pow(A[2],2)))*(180.0/3.14); //Para llevar a grados

    ang=round(gain*(ang_gyro)+(1-gain)*ang_accel);

```

```

    PreviousTimeAng_MR=CurrentTime_MR;
    PreviousAngle_MR=ang;

    ang=ang*(-1);
    return ang;
}

float PID()
{
    static unsigned long PreviousTimePID;
    unsigned long CurrentTimePID;
    static long error_anterior;
    static float sum_error;
    unsigned long deltaTPID;
    float ki=0.00015;
    float kp=0.013;
    float kd=0.5;
    float U;
    float Up;
    long Ui;
    double Ud;

    if(primeravezPID==true)
    {
        PreviousTimePID=0;
        error_anterior=0;
        primeravezPID=false;
    }

    CurrentTimePID=millis();

    deltaTPID=(CurrentTimePID-PreviousTimePID);

    Up=kp*error;
    Ui=ki*(error*deltaTPID+sum_error);

    Ud=((error-error_anterior)/deltaTPID);
    Ud=kd*Ud;

    U=(Up+Ui+Ud);
    U=U*(-1);

    PreviousTimePID=CurrentTimePID;

    sum_error=sum_error+error;
    error_anterior=error;

    return U;
}

```

```

int lectura_MH()
{
    //Serial.println("entro a lectura");

    // --- Lectura acelerometro y giroscopio ---
    uint8_t Buf[14]; // Define array donde se guardn los bits que envía
    el sensor
    memset(Buf, 0, sizeof(Buf)); // Vacía cache del Buffer
    I2Cread(MPU9250_ADDRESS_MH, 0x3B, 14, Buf); // Llama a función I2C
    Read creada más arriba

    // Convertir registros acelerometro
    int16_t ax = -(Buf[0] << 8 | Buf[1]);
    int16_t ay = -(Buf[2] << 8 | Buf[3]);
    int16_t az = (Buf[4] << 8 | Buf[5]);

    // Convertir registros giroscopio
    int16_t gx = -(Buf[8] << 8 | Buf[9]);
    int16_t gy = -(Buf[10] << 8 | Buf[11]);
    int16_t gz = Buf[12] << 8 | Buf[13];

    dato_MH[0]=ax-54; // Se le suman los offsets correspondientes
    dato_MH[1]=ay+119;
    dato_MH[2]=az+218;
    dato_MH[3]=gx-36;
    dato_MH[4]=gy-14;
    dato_MH[5]=gz+26;

    accel_MH[0]=(dato_MH[0]/2048)*9.81;
    accel_MH[1]=(dato_MH[1]/2048)*9.81;
    accel_MH[2]=(dato_MH[2]/2048)*9.81;
    gyro_MH[0]=(dato_MH[3]/16.4);
    gyro_MH[1]=(dato_MH[4]/16.4);
    gyro_MH[2]=(dato_MH[5]/16.4);

    return dato_MH;
}

int lectura_MR()
{
    //Serial.println("entro a lectura");

    // --- Lectura acelerometro y giroscopio ---
    uint8_t Buf[14]; // Define array donde se guardn los bits que envía
    el sensor
    memset(Buf, 0, sizeof(Buf)); // Vacía cache del Buffer
    I2Cread(MPU9250_ADDRESS_MR, 0x3B, 14, Buf); // Llama a función I2C
    Read creada más arriba

    // Convertir registros acelerometro
    int16_t ax = -(Buf[0] << 8 | Buf[1]);
    int16_t ay = -(Buf[2] << 8 | Buf[3]);

```

```

int16_t az = (Buf[4] << 8 | Buf[5]);

// Convertir registros giroscopio
int16_t gx = -(Buf[8] << 8 | Buf[9]);
int16_t gy = -(Buf[10] << 8 | Buf[11]);
int16_t gz = Buf[12] << 8 | Buf[13];

dato_MR[0]=ax-54;
dato_MR[1]=ay+119;
dato_MR[2]=az+218;
dato_MR[3]=gx-36;
dato_MR[4]=gy-14;
dato_MR[5]=gz+26;

accel_MR[0]=(dato_MR[0]/2048)*9.81;
accel_MR[1]=(dato_MR[1]/2048)*9.81;
accel_MR[2]=(dato_MR[2]/2048)*9.81;
gyro_MR[0]=(dato_MR[3]/16.4);
gyro_MR[1]=(dato_MR[4]/16.4);
gyro_MR[2]=(dato_MR[5]/16.4);

return dato_MR;
}

void motormoovel()
{
    float Pos_Servo_new=PosServo+U;

    //===== IF DE SEGURIDAD =====

    if (Pos_Servo_new>SERVOMAX) //If de seguridad para que no sobrepase
valores max y min de servos
    {
        Pos_Servo_new=SERVOMAX;
    }
    if (Pos_Servo_new<SERVOMIN)
    {
        Pos_Servo_new=SERVOMIN;
    }

    //===== Movimiento Servo =====

    Indice.write((Pos_Servo_new));
    //delay(20);

    PosServo=Pos_Servo_new;
    // Serial.print("PosServo");
    // Serial.print(PosServo);
    // Serial.print("\t");
}

```

```

void motormoove2()
{
    boolean flex;

    float Pos_Servo_new=PosServo+U;

    //===== IF DE SEGURIDAD =====

    if (Pos_Servo_new>SERVOMAX_MUNECA) //If de seguridad para que no
sobrepase valores max y min de servos
    {
        Pos_Servo_new=SERVOMAX_MUNECA;
    }
    if (Pos_Servo_new<SERVOMIN_MUNECA)
    {
        Pos_Servo_new=SERVOMIN_MUNECA;
    }

    //===== Movimiento Servo =====

    Indice.write((Pos_Servo_new));
    //delay(20);

    PosServo=Pos_Servo_new;
    //    Serial.print("PosServo");
    //    Serial.print(PosServo);
    //    Serial.print("\t");
}

```

H2. Código Visualización dedos Matlab

```
clc;
close all;
closeSerial();

global s %dato serial de arduino que mueve los servos
%% Datos iniciales POR DEDO

% Matriz en base a tabla 5 página 5 paper Humand Hand Kinematics and Dynamics

alpha=[radtodeg(pi/2);radtodeg(-pi/2);radtodeg(pi/2);0;0;0]; %6 articulaciones
considerando mov abducción + fingertip
a=[0;10;0;15;10;5];
d=[0;0;0;0;0;0];
global theta;
theta=zeros(6,1); %theta=[CMC_fe; MPC_ab ; MPC_fe ; PIP_fe ; DIP_fe; ft];

%% Cantidad de Sensores a considerar

global nsensores;
nsensores=4;

%% Setup arduino Manos

comPort1='COM15'; %Sensores Mano Humana

if (~exist('serialFlag1','var'))
    [gyroConnection1.s,serialFlag1]=setupSerial(comPort1);
end

disp('paso primer arduino');

%% Ciclo repetición

global Angle1; %Datos relacionados a MH
global displayAngle1;
global newAngle1;

global Angle2; %Datos relacionados a MR
global displayAngle2;
global newAngle2;

Angle1=zeros(3,4); %Se toman en cuenta solo mediciones de 3 primeros sensores
%%Modificar por theta correspondiente
displayAngle1=zeros(3,4); % Idem anterior
newAngle1=zeros(6,8);

Angle2=zeros(3,3); %Se toman en cuenta solo mediciones de 3 primeros sensores
%%Modificar por theta correspondiente
displayAngle2=zeros(3,3); % Idem anterior
```

```

newAngle2=zeros(6,8);

global error;
error=zeros(4,1); %Matriz de error para ángulo en X e Y, vector columna
error_anterior=zeros(4,1);

handle_to_part2 = Panel_Mov_Palma_Falange_V2();
handles_part2 = guidata(handle_to_part2);

handle_to_part3 = Simulacion_Mano_V2();
handles_part3 = guidata(handle_to_part3);

global FijaMH; %Variable que determina si boton para dejar fijo angulo de MH esta
activado

FijaMH=0;
n=0; %Para verificar que estan tensos tendones

global Dir_MH; % Matriz para cambio de dirección ejes de sensores
global Dir_MR;

global Start;
Start=0;

Dir_MH=ones(3,4); %Columnas designan sensores de falanges, filas designan ejes de
rotación
Dir_MR=ones(3,3);

%K=zeros(2,4); %Matriz que contiene constantes para PID, en fila son Kp, fila 2
Ki, columna son Palma,Prox,Medial,Distal
global K;
K=[0.2, 0.1 , 0.1 ,0.1;
    0 , 0 , 0 , 0];

U=zeros(4,1);

while (1)

    lectura(gyroConnection1,handles_part2,theta);

    if (Start==1)

        if n==0
            mbox=msgbox('¿Estan tensos tendones Flexión??');
            uiwait(mbox);
            n=1;
        end

        lectura(gyroConnection1,handles_part2,theta);

%        U=PID(error); %%Matriz de 4x1
%i=2;

```

```

        for i=1:1:nsensores
            Simulacion(handles_part3,alpha,a,d,U(i,1),i)
        end %end for
        %pause(0.05)
    end
end

function []=lectura(gyroConnection1,handles_part2,theta)

    global FijaMH;
    global newAngle1;
    global Angle1;
    global displayAngle1;
    global Dir_MH;
    global error;

    if FijaMH==0
        [newAngle1]=AnguloV10_1(gyroConnection1); %Toma de datos del sensor de
MH / AnguloV4_1 y 2 es porque tienen variables persistentes,
        end                                     ...para que no se mezclen se
hacen dos programas

    if FijaMH==0

[Angle1,displayAngle1]=Dedo_V7_Sim(Angle1,displayAngle1,newAngle1,Dir_MH); %
Datos para MH / Suma angulo antebrazo
        end

        drawnow; %Velocidad del Loop es muy rapida, por lo que drawnow asegura que
cambie texto

    end

function []=Simulacion(handles_part3,alpha,a,d,U,i)
    %disp(U);
    global displayAngle1;
    global theta;

    theta(3,1)=displayAngle1(1,1);
    theta(4,1)=displayAngle1(1,2)-displayAngle1(1,1);
    theta(5,1)=displayAngle1(1,3)-displayAngle1(1,2);

    A01=DHmod(alpha(1,1),a(1,1),d(1,1),theta(1,1)); %% DH Modificado
    A12=DHmod(alpha(2,1),a(2,1),d(2,1),theta(2,1));
    A23=DHmod(alpha(3,1),a(3,1),d(3,1),theta(3,1));
    A34=DHmod(alpha(4,1),a(4,1),d(4,1),theta(4,1));

```



```

A45=DHmod(alpha(5,1),a(5,1),d(5,1),theta(5,1));
A56=DHmod(alpha(6,1),a(6,1),d(6,1),theta(6,1));

A02=A01*A12;
A03=A02*A23;
A04=A03*A34;
A05=A04*A45;
A06=A05*A56;

P0=[0;0;0];
Pos=[P0 A01(1:3,4) A02(1:3,4) A03(1:3,4) A04(1:3,4) A05(1:3,4)
A06(1:3,4)];

%getframe;
plot(handles_part3.axes1,Pos(3,1:7),Pos(1,1:7),'-*','LineWidth',4);
axis(handles_part3.axes1,[-30,80,-10,60]); %Límites de los ejes

drawnow();

end

```

H3. Código visualización dedos Arduino

```
#include <Wire.h>

#define TCAADDR 0x70

////////////////////////////////////// Config IMU
//////////////////////////////////////

#define MPU9250_ADDRESS           0x68    //Define valores de
variables para luego llamarlas

#define GYRO_FULL_SCALE_250_DPS   0x00
#define GYRO_FULL_SCALE_500_DPS   0x08
#define GYRO_FULL_SCALE_1000_DPS  0x10
#define GYRO_FULL_SCALE_2000_DPS  0x18

#define ACC_FULL_SCALE_2_G        0x00
#define ACC_FULL_SCALE_4_G        0x08
#define ACC_FULL_SCALE_8_G        0x10
#define ACC_FULL_SCALE_16_G       0x18

//////////////////////////////////////
//////////////////////////////////////

int16_t dato[6]; // Array donde se guardan los datos del acelerómetro y
del giroscopio para que sean devueltos por la función

/* Assign a unique ID to this sensor at the same time */
int IMU0=0;
int IMU1=1;
int IMU2=2;
int IMU3=3;
int IMU4=4;
int IMU5=5;
int IMU6=6;
int IMU7=7;

int n=0;

////////////////////////////////////// Funcion auxiliar lectura MPU9250
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

void I2Cread(uint8_t Address, uint8_t Register, uint8_t Nbytes, uint8_t*
Data, int i)
{
    //Serial.println("entro a I2CRead");
    //tcasselect(i);
    Wire.beginTransmission(Address);
```

```

    //Serial.println("paso begintransmission");
    //tcselect(i);
    Wire.write(Register);
    //Serial.println("paso wire write");
    //tcselect(i);
    Wire.endTransmission();

    //tcselect(i);
    Wire.requestFrom(Address, Nbytes);
    uint8_t index = 0;
    while (Wire.available())
        //tcselect(i);
        Data[index++] = Wire.read();
}

////////////////////////////////////7 Funcion auxiliar de escritura,
configura acelerómetro y giroscopio //////////////////////////////////

void I2CwriteByte(uint8_t Address, uint8_t Register, uint8_t Data , int
i)
{
    tcselect(i);
    Wire.beginTransmission(Address);
    Wire.write(Register);
    Wire.write(Data);
    Wire.endTransmission();
}

//////////////////////////////////// Funcion Selección puerto Multiplexor
////////////////////////////////////

void tcselect(uint8_t i) { // Función que selecciona el puerto y el
sensor, luego abre la comunicación con ese sensor
    if (i > 7) return; // Hasta 8 sensores máximo por cada
multiplexor, si da un puerto mayor que 7 (recordar que empieza de 0),
devuelve cero.

    Wire.beginTransmission(TCAADDR);
    Wire.write(1 << i); // Envía bit con el número del puerto
(sensor) a abrir.
    Wire.endTransmission();
}

////////////////////////////////////
////

void setup()
{
    Wire.begin();
    Serial.begin(115200);
    //Serial.setTimeout(10);

```



```

        Serial.println();
        break;
    } //Cierre If Serial.Available
}

int lectura(int IMU)
{
    //Serial.println("entro a lectura");

    // --- Lectura acelerometro y giroscopio ---
    uint8_t Buf[14]; // Define array donde se guardn los bits que envía
    el sensor
    memset(Buf, 0, sizeof(Buf)); // Vacía cache del Buffer
    tcselect(IMU);
    I2Cread(MPU9250_ADDRESS, 0x3B, 14, Buf, IMU); // Llama a función I2C
    Read creada más arriba

    // Convertir registros acelerometro
    int16_t ax = -(Buf[0] << 8 | Buf[1]);
    int16_t ay = -(Buf[2] << 8 | Buf[3]);
    int16_t az = (Buf[4] << 8 | Buf[5]);

    // Convertir registros giroscopio
    int16_t gx = -(Buf[8] << 8 | Buf[9]);
    int16_t gy = -(Buf[10] << 8 | Buf[11]);
    int16_t gz = Buf[12] << 8 | Buf[13];

    dato[0]=ax;
    dato[1]=ay;
    dato[2]=az;
    dato[3]=gx;
    dato[4]=gy;
    dato[5]=gz;

    return dato;
}

void printvalues2(int x[])
{
    for (int i=0;i<6;i++)
    {
        Serial.print(x[i]);
        //Serial.print("/");
        if (i!=5)
        {
            Serial.print("/");
        }
    }
}

```

I. Cálculo deformación cable

La deformación del cable utilizado como tendón en el prototipo de mano mecánica desarrollado por [6] es importante debido a que es parte del sistema de transmisión de movimiento. Para el prototipo fueron utilizados dos hilos de pescar de Nylon de distintos diámetros (0.2 [mm] y 0.3 [mm]) esto debido a que el cable de menor diámetro es conectado directamente al servomotor y el de mayor diámetro es conectado a las partes de la mano para que en caso de problemas con la programación de los actuadores el prototipo de mano no sea dañado por el tensado excesivo de los cables.

El servomotor que tiene el prototipo de mano mecánica para los movimientos de las falanges de los dedos es capaz de entregar un torque máximo de 11 [kg * cm] además, las poleas conectadas a los servomotores tienen un diámetro de 100 [mm] , con esta información se calcula la fuerza máxima a la que estará sometida el cable:

$$\tau = F \cdot r$$

$$F_{max} = \frac{\tau}{r} = \frac{11}{5} = 21,582 [N]$$

Para el cálculo de la deformación del cable es considerado el cable de mayor diámetro (0.3 [mm]), por lo que el área de sección transversal tiene un valor de

$$A = \frac{\pi \cdot d^2}{4} = 0,07068 [mm^2]$$

Con los datos anteriores se obtiene el valor del esfuerzo máximo al que está sometido el cable:

$$\sigma_{max} = \frac{F_{max}}{A} = \frac{21,528}{0,07068} = 305,322 [MPa]$$

El módulo de elasticidad se puede escribir en términos del esfuerzo y del coeficiente de deformación como:

$$E = \frac{F/A}{\Delta L/L} = \frac{\sigma}{\varepsilon}$$

De la literatura, el valor del módulo de elasticidad del hilo de pescar tiene un valor de

$$E_{hilo\ pescar} = 200.000 [MPa]$$

Con ello el coeficiente de deformación es:

$$\varepsilon = 0,001526 [-]$$

Tomando en cuenta un largo de cable $L = 500 [mm]$, que es una longitud aproximada de la distancia desde la parte inferior de la mano hasta una polea que se encuentra en la base de más arriba, la deformación del cable da un resultado de:

$$\Delta L = 6 [mm]$$

Esta deformación no permite asegurar precisión de los movimientos teniendo en cuenta que cada vez que se induce una deformación hay que ajustar el conjunto de tendones.

J. Rango de captación de sensores

El rango de captación de la posición angular del sensor MPU9250, utilizando el modelo matemático presentado en la sección 2.5.3, es de 180° , los que van desde -90° a $+90^\circ$. Debido a lo anterior es importante la posición en la que es colocado el sensor en las distintas partes de la mano, suponga el caso en que se tienen dos sensores, uno para captar los movimientos de la muñeca y otro para captar los movimientos de la falange del dedo. Si ambos son instalados de forma perpendicular a la superficie, la posición angular final máxima que serán capaces de captar es -90° y $+90^\circ$ (Figura J.1). Si se desea realizar un movimiento de 30° con la muñeca, la posición inicial de la falange del dedo también será 30° , por lo que la posición máxima que podrá alcanzar en sentido positivo será $+60^\circ$.

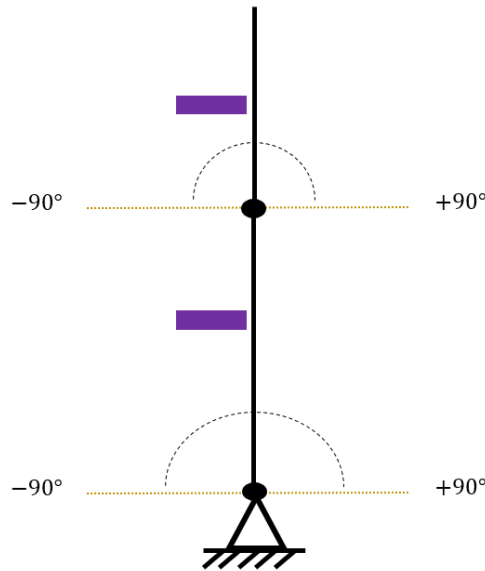


Figura J.1: Representación del rango de movimiento angular que pueden captar los sensores en la muñeca y en una falange proximal cuando son instalados perpendicular a la superficie. La muñeca y la falange están representadas por un eslabón o link cada uno, los rectángulos morados corresponden a los sensores.

De forma de aumentar el rango de medición del sensor ubicado en la falange los sensores son instalados con orientaciones distintas, el sensor en la muñeca es ubicado de forma perpendicular a la superficie ($\theta = 0^\circ$) y el de la falange es instalado de forma horizontal ($\theta = -90^\circ$), lo que incrementa el rango de medición en la falange puesto que desde la posición anatómica ahora se obtiene un rango de 180° como se muestra en la **Figura J.2**, considerando que la mayoría de los movimientos de las falanges se realizan en el sentido de las manecillas del reloj.

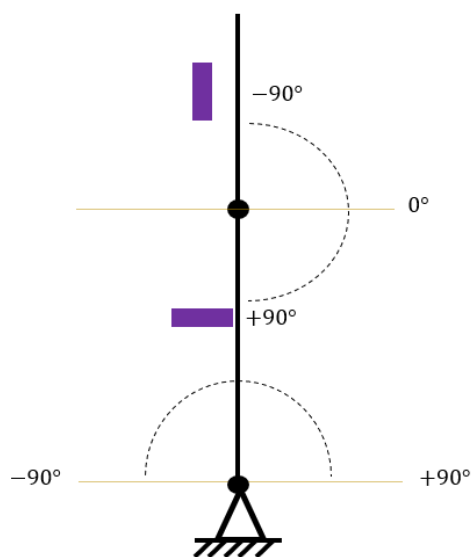


Figura J.2: Representación del rango de medición angular si el sensor de la muñeca es instalado de forma perpendicular a la superficie, y el de la falange de forma horizontal.

Lo anterior no afecta las mediciones producto de cómo son trabajados los datos en el código de programación, esto es, trabajar con las diferencias de ángulos y no con el dato del ángulo directamente. Esto se realiza a través de una variable llamada `DisplayAngle`, la que modifica su valor en base a las diferencias de los ángulos actual y anterior, además, esta variable permite realizar el Reset de los ángulos de los sensores de ambas manos para que partan en la misma posición.

