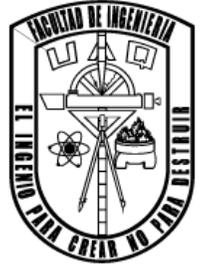




Universidad Autónoma de Querétaro
Facultad de Ingeniería
Campus San Juan del Río



Ingeniería Electromecánica
Línea Terminal en Mecatrónica

Diseño y construcción de una mano robótica de ocho grados de libertad

Tesis:

Que como parte de los requisitos para obtener Título de Ingeniero Electromecánico con Línea terminal en Mecatrónica

Presenta:

Jorge Agustín Velázquez Villegas

Dirigido por:

M. en C. Manuel García Quijada

San Juan del Río, Qro. Marzo, 2022



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
FACULTAD DE INGENIERÍA
CAMPUS SAN JUAN DEL RÍO



DISEÑO Y CONSTRUCCIÓN DE UNA MANO ROBÓTICA DE OCHO GRADOS DE
LIBERTAD

T E S I S

Como parte de los requisitos para obtener el título de

INGENIERO ELECTROMECAÁNICO

Línea Terminal

MECATRÓNICA

Presenta

JORGE AGUSTÍN VELÁZQUEZ VILLEGAS

Dirigido por:

M. EN C. MANUEL GARCÍA QUIJADA

SINODALES:

M. en C. Manuel García Quijada (6279)
Presidente

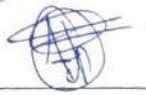
Dr. Jesús Rooney Rivera Guillen (8943)
Secretario

Dr. J. Jesús de Santiago Pérez (6589)
Vocal

Dr. Miguel Trejo Hernández (6443)
Suplente









San Juan del Río, Qro., 25 de marzo de 2022.

RESUMEN

En este trabajo primeramente se presenta la idea de lo que es un robot, menciones de lo que se puede considerar un robot a través de la historia, como ciertas ideas y eventos fueron fundamentales para desembocar en lo que fue la construcción del que se considera uno de los primeros robots y el posterior asentamiento de las ideas en los llamados brazos robóticos, en los cuales su arquitectura es un aspecto fundamental en la construcción de un robot, después de esta introducción se da a conocer los objetivos de este trabajo y por qué se realizó. En el capítulo dos se presentan los conceptos ya establecidos de lo que es un robot, las partes que lo componen, los diferentes tipos de clasificación de los robots, el análisis matemático en el diseño de la arquitectura que da forma un robot, y continuando con lo que permite tener movimiento al robot, que son los actuadores, además mencionar algunos trabajos en los que se puede ver similitudes con este mismo trabajo que se hizo en esta tesis. El capítulo tres contiene la información del diseño del robot construido en este trabajo, primeramente, se asienta el diseño conceptual de este robot, después el análisis de su arquitectura, siguiendo con un diseño realizado en computadora para ciertos ajustes en el diseño, y para antes de terminar se menciona la fabricación de sus piezas, el ensamble del robot y las restricciones que este mismo posee. En el cuarto capítulo se explica una simulación del funcionamiento de robot, después sigue con otra explicación sobre una interfaz que permite controlar al robot y hacerlo funcionar, después se explican las pruebas de funcionamiento del robot, y para terminar se redacta una conclusión del este trabajo.

Palabras clave: manipulador paralelo, mano robótica, cadena cinemática.

SUMMARY

In this work, the idea of what a robot it is first presented, mentions of what could be considered a robot throughout history, how certain ideas and events were fundamental to lead to what was the construction of what is considered one of the first robots and the subsequent settling of ideas in the so-called robotic arms, in which their architecture is a fundamental aspect in the construction of a robot, after this introduction, the objectives of this work and why it was carried out are disclosed. In chapter two, the already established concepts of what a robot is are presented, the parts that compose it, the different types of classification of robots, the mathematical analysis in the design of the architecture that shapes a robot, and continuing with which allows the robot to have movement, which are the actuators, in addition to mentioning some works in which you can see similarities with this same work that was done in this thesis. Chapter three contains the information on the design of the robot built in this work, firstly, the conceptual design of this robot is established, then the analysis of its architecture, following with a design made in a computer for certain adjustments in the design, and before Finally, the manufacture of its parts, the assembly of the robot and the restrictions that it has are mentioned. In the fourth chapter a simulation of the robot operation is explained, then it continues with another explanation about an interface that allows to control the robot and make it work, then the robot operation tests are explained, and finally a conclusion of this work is written.

Keywords: parallel manipulator, robotic hand, kinematic chain.

Agradecimientos

Agradezco a quienes dedicaron parte de su tiempo a leer lo que está escrito en este trabajo, los sinodales que acreditaron este trabajo: el Dr. Miguel, el Dr. Rooney, el Dr. J. Jesús y el Maestro Manuel, siendo este último mi tutor de tesis. Le agradezco especialmente a mi tutor, porque con su ayuda en algunas partes de este trabajo fui capaz de resolver los problemas que se presentaron al momento de estar realizando en este proyecto, además de ser paciente con las ciertas veces que yo me comunicaba con él. Quienes me conocen saben bien que soy callado y tiendo a hablar muy poco con los demás, no por ser descortés, sino porque busco la mejor forma de decir lo que pienso y eso hace que me tarde en decir lo que pienso a las demás personas con las que converso, esto es porque el hecho de comunicar una idea o algo a otra persona me fascina y es algo que yo valoro mucho.

Agradezco a mi familia, a mi mamá por ser alguien que me cuida y ayudo mucho cuando buque irme a estudiar a otro estado, en este caso fue el de estudiar en Querétaro; agradezco mi hermano mayor Alfredo por mostrarme mis errores cuando trataba con otras personas y la confianza que él me daba para seguir adelante cuando esto no iba bien; agradezco mis dos hermanos menores y hermana menor por recordarme que siempre se puede tener una ilusión por seguir; y agradezco a mi padre por apoyarme para poder estudiar, aunque nuestra relación padre e hijo es distante, él es alguien a quien respeto mucho y por eso le agradezco por terminar este trabajo.

INDICE

RESUMEN.....	iii
SUMARY	iv
Agradecimientos	v
1 INTRODUCCIÓN.....	1
1.1 Antecedentes históricos	1
1.2 Objetivos	6
1.3 Justificación.....	7
2 REVISIÓN DE LITERATURA	8
2.1 Estado del arte.....	8
2.1.1 Definición de Robot	8
2.1.2 Clasificación de los robots	10
2.1.3 Robots industriales.....	13
2.1.4 Robot manipulador	14
2.1.5 Robot Paralelo.....	17
2.2 Cinemática de un mecanismo.....	20
2.2.1 Cadena cinemática.....	20
2.2.2 Notación de Denavit-Hartenberg.....	24
2.3 Actuadores.....	27
2.3.1 Motores eléctricos.....	28
2.3.2 Mecanismos hidráulicos y neumáticos	30
2.4 Precedentes de este proyecto	30
3 DISEÑO Y CONSTRUCCIÓN DE LA MANO ROBÓTICA	37
3.1 Diseño conceptual.....	37
3.2 Metodología	38
3.3 Cálculo de las cadenas cinemáticas	49
3.4 Programa de simulación	57
3.5 Manufactura	64
3.5.1 Fabricación	64
3.5.2 Ensamblaje.....	69
3.6 Restricciones de diseño.....	76
4 RESULTADOS Y DISCUSIÓN	83
4.1 Simulación	83

4.2	Pruebas generales de funcionamiento en físico	86
4.2.1	Interfaz de usuario	87
4.2.2	Pruebas de manipulación	95
4.3	Conclusiones y prospectivas	107
5	BIBLIOGRAFÍA	109
6	APÉNDICES	111
6.1	Apéndice 1: programa principal de la interfaz de control.....	111
6.2	Apéndice 2: funciones secundarias del programa principal	123

1 INTRODUCCIÓN

1.1 Antecedentes históricos

En el principio de la especie humana está siempre busco la manera de realizar tareas muy particulares mediante el uso de herramientas específicas para trabajar. Estas herramientas siempre eran requeridas por individuos para cualquier tarea en específico y que a través del tiempo fueron aumentando en cantidad las herramientas con diferentes propósitos cada una, y con ello lo que una persona podría lograr hacer, pero para lograr hacer una mayor cantidad de artículos en específico también se requería de una mayor cantidad de mano de obra. En la época de la revolución industrial las maquinas sustituyeron a gran parte de la mano de obra para fabricar productos, ya que estas lo podían crear de manera más rápida y eficiente, pero gran parte de las máquinas de la época seguían requiriendo de un operario para que logran hacer el trabajo adecuadamente. Con los siguientes avances en el siglo XX estas máquinas se volvieron cada vez más complejas y tuvieron una cierta autonomía debido a la inteligencia adquirida por los avances tecnológicos (Challacombe, Khan, Murphy, & Dasgupta, 2006). Los robots que se pueden observar actualmente en fábricas con tecnología más avanzada, son controlados por una inteligencia artificial que les provee de un mejor control sobre lo que hacen, para prevenir accidentes y percances en la fabricación; pero la inteligencia artificial se ha adaptado a robots que no se encuentran dentro de un espacio cerrado como las fábricas, sino que también puede vérsela en vehículos de conducción autónoma y en aplicaciones en el campo de la medicina, y con todo esto se les ha clasificado como robots, pero la idea de lo que es un robot tiene un origen que proviene desde las antiguas civilizaciones.

La concepción de la idea puede encontrarse en mitos de la antigua civilización griega, en donde se relataban mitos de seres mecánicos construidos por el dios forjador Hefesto para ser sirvientes de los dioses. El mito griego origina una percepción de ver a estos seres como sirvientes, pero no aclara lo que podrían llegar a realizarse de manera más pragmática y lógica, y esto último esta mejor expresado por el filósofo griego Aristóteles en una frase: *“Si cada instrumento pudiera realizar su propio trabajo, cumplir o anticipar la voluntad de los demás... si lanzadera pudiera tejer, y el pico toca la lira, sin ninguna mano para guiarlos; el jefe de trabajadores no necesitaría sirvientes”* (Challacombe, Khan, Murphy, & Dasgupta, 2006).

En el siglo XX un escritor checo plantearía una historia de seres que cumplirían tareas similares a las que Aristóteles mencionaba, además de que utilizo una palabra en específico referirse a estos seres, tal escritor fue Karel Capek y la palabra

fue uso fue “*robot*”; las ideas de Capek fueron plasmadas en dos de sus obra de teatro, “*Opilec*” de 1917 y “*Rossum’s Universal Robots*” de 1921, la segunda obra habla de trabajadores humanoides los cuales son fabricados por miles, sin embargo, estos seres humanoides eran orgánicos no mecánicos, pero la idea de crear seres y que estos realicen trabajos específicos se mantiene como anteriormente se menciona, ya que estos seres desempeñan las mismas tareas y actividades que un robot hace en una fábrica: realizar trabajos que para el ser humano serian difíciles, monótonos e incluso peligrosos. En su novela Karel Capek recurrió de la palabra “robot” que en checo significa de manera brusca “trabajador” o “esclavo asalariado”, esta palabra fue aceptada para referirse a seres que hacen trabajos repetitivos, además la idea también se extendió entre escritores de ciencia ficción, los cuales contenían en sus obras a máquinas con autonomía e inteligencia propia (Nocks, 2007).

La novela de Capek pudo haber influenciado la novela de ciencia ficción del escritor Isaac Asimov y principalmente su obra más conocidas “*Runaround*” de 1942, donde el escritor idea a un ser creado a través de diferentes piezas mecánicas y que tienen como propósito realizar trabajos que ser humano le ordene, pero también la capacidad de concebir ideas que le darán la aptitud de revelarse ante su creador, en este caso el ser humano; aunque para evitar la rebelión de este ser, Asimov instaura en su obra tres leyes que le permitirán al ser humano controlar al robot y evitar que se revele contra la humanidad, estas tres leyes son conocidas como las tres leyes de la robótica.

En cuanto al aspecto de los robots, los primeros indicios surgieron en el siglo XVI con los humanoides mecánicos creados para ejecutar labores específicas, estos eran llamados autómatas y eran figuras con la forma de personas, su principal trabajo era tocar las campanas cada hora. Con el paso del tiempo y concretamente en el siglo XVIII grandes inventores contribuyeron con avances en este tipo de mecanismo, destacando el invento de Pierre Jaquet-Droz, quien era un creador de marionetas. Una de sus creaciones llamada “*The Scribe*” o El escritor, el cual tenía la capacidad de escribir un texto ya previamente escrito y el cual debía tener menos de cuarenta letras; lo más destacable de este autómata es la capacidad ser programado debido a los diferentes textos que este podía escribir. Los autómatas construidos posteriormente solamente tenían el objetivo de entretener y su tecnología no era aplicable a procesos industriales, pero llegaría a verse hasta en la década de 1950 donde la tecnología avanzo para efectuar tareas peligrosas, estos grandes avances tecnológicos se originaron principalmente en el segundo conflicto bélico más importante la primera mitad del siglo XX. En la Segunda Guerra Mundial el significado de robot se extendió hasta incluirlo en el piloto automático, encontrándolo en bombas robot, también durante ese momento de la historia aparecieron en desarrollo las computadoras, al igual que los primeros manipuladores remotos, conocidos como robots tele-operados. Por lo que la Segunda Guerra Mundial fue un periodo donde los servosistemas, el control

numérico, los procesos de control y avances en los robots empezaron a sobresalir, para el año 1954 el primer robot industrial fue construido y dos patentes para robots industriales fueron solicitadas (Poole, 2012)

En el año de 1954 una compañía fabricante de radiadores solicito a Planet Corporation of Lansing, que diseñara un sistema capaz de manipular la fundición caliente mientras esta última es enfriada con agua, a este robot se le dio el nombre de Planobot, mostrado en la Figura 1.1, el desarrollo de este robot tardó unos tres años y sus avances tecnológicos fueron mostrados en la feria internacional de automatización de 1957, en Estocolmo. También en 1954, en el mes de marzo el inventor británico Cyril W. Kenward solicitó una patente británica para un robot, este dispositivo tenía dos brazos manipuladores con cuatro grados de libertad; la patente de este dispositivo fue publicada en 1957 y se tituló *“Manipulating Apparatus”*. En el mismo año en el que Kenward solicitó su patente, pero dos meses después, el inventor americano George C. Davol solicitó una patente para robot, la cual se tituló *“Programmer Article Transfer”*, y esta fue otorgada en 1961. La patente de George C. Davol ayudó en la fundación de la compañía Unimation (Universal Automation), y los primeros robots de esta compañía fueron construidos en 1961, su primer robot fue entregado a la planta automotriz de General Motors en Nueva Jersey.

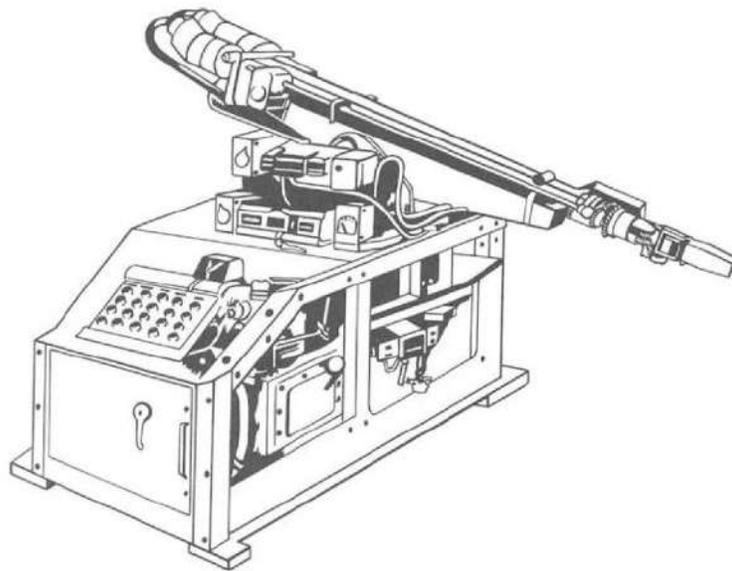


Figura 1.1 Planobot (Poole, 2012).

El interés en los robots se expandiría alrededor del mundo en 1965, con más compañías interesadas en avances tecnológicos en este campo del conocimiento. En 1966 Trallfa, una firma noruega de maquinaria agrícola, construiría un robot para pintar carretillas, en 1969 Unimation introdujo un robot de soldadura de punto. A inicios de 1970, Francia apoyaría la investigación en robótica, la Unión Soviética desarrollaría su primer robot durante 1972, y países como Alemania, Italia y Suecia tendrían una gran relevancia en la robótica durante la década de los setenta. Pero

antes, a finales de 1960 ciertas universidades se enfocaron en la investigación de la robótica, el Stanford Research Institute se embarcó en este campo mejorando la movilidad de los robots, los sistemas de visión y la inteligencia artificial con un robot denominado “*Shakey*”; también el MIT (Massachussets Institute of Thechnology) fue otra de las entidades académicas que aporó con el desarrollo de ciertos de leguajes robóticos, los cuales eran lenguaje MHI y MINI.

Japón inicio el desarrollo e investigación de la robótica en el año 1969 por medio de una licencia otorgada por Unimation a Kawasaki Heavy Industries. La entrada de Japón en campo de la robótica fue muy agresiva, desarrollando en 1970 un robot adaptativo con un sistema de visión que podía distinguir entre nueve colores, y además también podía corregir sus errores. En 1973 ASEA (Allmänna Svenska Elektriska Aktiebolage (General Swedish Electric Company)) de Suecia, construyó el primer robot totalmente eléctrico, ya que los primeros robots previos al de ASEA funcionaban con sistemas hidráulicos. Para 1978 la compañía Unimation construyó el robot PUMA (Programmable Universal Machine of Assembly), Figura 1.2, el cual puede manipular objetos pequeños y de esta formar abarcar una gran cantidad de tareas a realizar.



Figura 1.2 Robot PUMA (Poole, 2012).

Un gran avance en tecnología y diseño vino por parte de la Universidad Yamanashi, Japón, la cual desarrollo en 1979 el brazo robótico articulado de cumplimiento selectivo (SCARA), Figura 1.3, el cual realizaba tareas de ensamblaje fácilmente debido a que sus movimientos los ejecutaba en el plano horizontal, y no en el vertical (Poole, 2012).

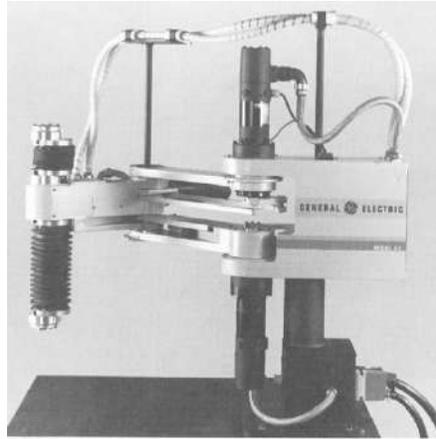


Figura 1.3 Robot SCARA (Poole, 2012).

Los avances tecnológicos posteriores, ya sea en las mejoras de los sensores, diferente tipo de actuadores, una mayor precisión debido a unas mejores tarjetas de programación capaces de procesar cálculos más complejos, permitieron que los robots obtuvieran una mayor precisión al ejecutar sus trabajos; como también un mayor rango de alcance y un mejor control de lo que podían realizar para prevenir accidentes. Todo lo anteriormente mencionado ha generado la capacidad de crear robots más reminiscentes al ser humano, y uno de los robots que más evoca el comportamiento físico de una persona es el robot Atlas de la compañía Boston Dynamics, imagen de la derecha Figura 1.4; pero también existen otro tipo robots, que aunque no se asemejen a un ser humano, son muy útiles, como es el caso del robot Versius, de la compañía CRM Surgical, imagen izquierda Figura 1.4 el cual tiene la capacidad de efectuar procedimientos quirúrgicos, pero lo que compete a esta tesis es el desarrollo de una mano robótica como las que se muestran en la Figura 1.5, y añadirla esta mano robótica a un brazo robótico.



Figura 1.4 Robot Versius en el lado izquierdo (Versius Surgical Robotic System - CRM Surgical, 2020) y el robot Atlas en el lado derecho (Atlas | Boston Dynamics, 2020).

El enfoque de este trabajo es hacer una mano como las mostradas en la Figura 1.5, porque aún con los grandes avances tecnológicos que se han generado a través de la historia, los ejemplos mostrados en la figura de arriba son robots con dos enfoques distintos y con un coste muy elevado: siendo el robot de la derecha de la Figura 1.4, un robot enfocado en el campo de la medicina para realizar operaciones quirúrgicas; el de la izquierda un prototipo en desarrollo que se sigue perfeccionando, y aun teniendo que definir en donde se puede aplicar dicho robot. En cambio, las manos robóticas mostradas en la Figura 1.5 son uno de los robots en los que se enfoca esta tesis, debido a las facilidades que otorga su forma y las posibilidades de manipulación que esta ofrece sobre un objeto, ya que al ser esta una mano robótica más simple que una mano robótica antropomórfica, poder llegar a sujetar, sostener, e incluso mover objeto llega a ser muy útil, además el abanico de posibilidades es más amplio cuando esta mano robótica de tres o cuatro dedos forma parte como efector final de un brazo robótico.

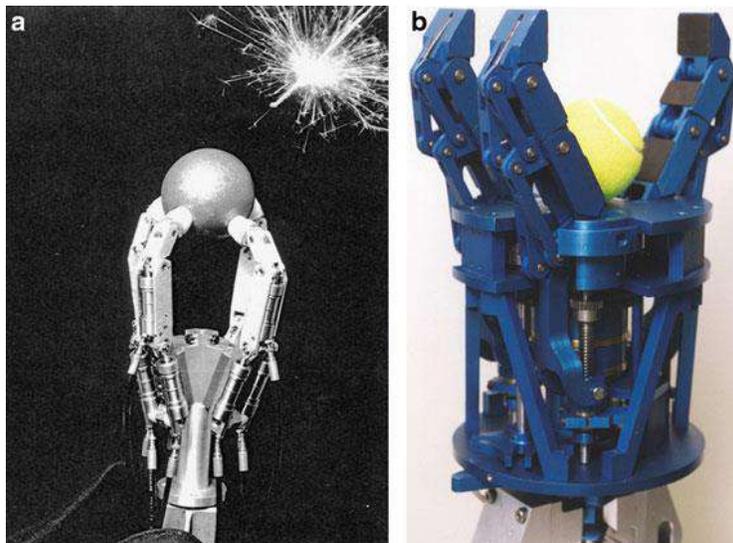


Figura 1.5 Dos instancias de manos robóticas: (a) la mano de cuatro dedos accionada hidráulicamente del TU Múnich; (b) la mano SARAH de tres dedos de la Université Laval (Angeles, 2014).

1.2 Objetivos

Objetivo General

El diseño y fabricación de una mano robótica de tres dedos y con ocho grados de libertad para realizar operaciones de manipulación, todo esto mediante el uso de programas CAD y de la tecnología de impresión en 3D, y que además sirva como extensión del brazo robótico Yaskawa MH5LF instaurado en las instalaciones de la UAQ.

Objetivos particulares:

- Modelar y analizar el diseño de la mano robótica por medio de los programas de diseño asistido por computadora (CAD), para de esta forma obtener las medidas adecuadas de cada una de las piezas y evitar fabricarlas erróneamente.
- Hacer uso de la notación de Denavit-Hartenberg para obtener las ecuaciones de posicionamiento directo e inverso de la mano robótica, y de esta forma predecir o dictar el comportamiento de la misma.
- Simular en un entorno virtual las ecuaciones obtenidas, para de esta forma hacer correcciones al diseño arquitectónico de la mano robótica, y además de servir como verificación de funcionamiento de las ecuaciones.
- Manufacturar el prototipo mediante el uso de la tecnología de impresión en 3D y ensamblar todas las partes que dan forma de manera física a la mano robótica.
- Programar una interfaz que permita controlar los actuadores de la mano robótica mediante un sistema computo, y de esta forma poder realizar pruebas de manera física con la mano robótica.
- Realizar las pruebas de manipulación de objetos esféricos de manera física para de esta forma comprobar el funcionamiento de la mano robótica.

1.3 Justificación

El uso de manos robóticas es empleando por su flexibilidad de manipular objetos con diferentes formas geométricas. Los principales propósitos del desarrollo de sistemas robóticos son las habilidades de agarre y manipulación de objetos (Wang, Zhao, Xiao, Zhu, & Wang, 2018). El uso de manos robóticas con dedos múltiples permite realizar operaciones que demanden o requieran un mayor grado de complejidad, además que el estudio sobre este tipo de robots se esta extendido, en gran parte por los múltiples diseños que se puede llegar a crear y los diferentes propósitos en los que pueden emplear.

Con lo anteriormente mencionado sobre los manipuladores y las manos robóticas, en este trabajo se pretende de desarrollar de un robot paralelo, el cual puede ser una mano robótica; y que además llega a ser más complejo que un robot simple o en serie, como lo son los brazos robóticos. Este trabajo también pretende servir como una muestra palpable de cómo llegar a crear un robot paralelo en la materia de robótica de la UAQ, debido a que este robot manipulador puede ser adaptado a un brazo robótico Yaskawa MH5LF que se encuentran en las instalaciones de la misma.

2 REVISIÓN DE LITERATURA

2.1 Estado del arte

En esta sección se explican los conceptos básicos sobre lo que es un robot, las partes que los conforman, los diferentes tipos de robots que pueden llegar a nombrar dependiendo de la clasificación que se quiera utilizar, además de explicar los parámetros de Denavit-Hartenberg, que son utilizados en el análisis cinemático del robot a construir; también se mostrara los diferentes tipos de actuadores que pueden ser utilizados en un robot, y por último se hace mención a algunos trabajos que preceden a este, ya sea manera local o internacional.

2.1.1 Definición de Robot

La definición de lo que es un “robot” actualmente está definido por la Organización Internacional de Normalización (ISO, *International Standard Organization*) y la Federación Internacional de Robótica (IFR), la definición se explica en el siguiente párrafo:

“Mecanismo accionado programable en dos o más ejes con un grado de autonomía, moviéndose dentro de su entorno para realizar tareas previstas: un robot incluye un sistema de control y la interfaz de control, la clasificación de un robot industrial o robot de servicio se realizará de acuerdo con su aplicación prevista” (Robots and robotic devices - Vocabulary, 2020).

Mecanismo

Un mecanismo es un dispositivo que transforma el movimiento en un patrón deseable, que por lo general despliega muy poca fuerza y la transmisión de potencia es baja. Hunt definió un mecanismo como un medio de transmisión, control o restricción del movimiento relativo (Norton, 2009).

Los robots independientemente de propósito final exhiben tres elementos claves que ayuda a diferenciarlos de un simple mecanismo, estos elementos son:

- **Programabilidad.** Implica capacidades computacionales o de manipulación de símbolos que un diseñador puede combinar como se desee.
- **Capacidad mecánica.** Es la habilidad que el permite actuar en un entorno en lugar de funcionar simplemente como un dispositivo de procesamiento de datos o computacional.

- **Flexibilidad.** En el sentido de que puede operar utilizando una variedad de programas de computacionales, al igual de tener posibilidad de manipular y transportar materiales de diversas maneras.

Las características de un robot industrial difieren de las de un robot de servicios o propósito específico, como es el caso del robot para operaciones quirúrgicas. Las características de los robots industriales son en gran parte la de ser: más robustos, rápidos en las tareas que ejecutan y de realizar trabajos repetitivos; en cambio, los quirúrgicos requieren de una mayor precisión y por lo tanto tener una velocidad más controlada que les permita realizar una operación o movimientos delicados, con lo que debe proporcionar una mayor seguridad a la hora de ejecutar dicha tarea (Challacombe, Khan, Murphy, & Dasgupta, 2006).

Arquitectura general de un sistema mecánico robótico

Un sistema mecánico robótico se compone de algunos subsistemas, los cuales son los siguientes: sistema mecánico compuesto por cuerpos rígidos y deformables, un sistema de detección, un sistema de actuación, sistema de control, sistema de procesamiento de información. Todos estos subsistemas necesitan comunicarse entre ellos y lo hacen a través de interfaces, la cuales decodifican la información que se transmite de un medio a otro, en la Figura 2.1 se muestran algunos subsistemas de los que se compone un sistema mecánico robótico, y en la Figura 2.2 se muestra la arquitectura general de un sistema mecánico robótico.

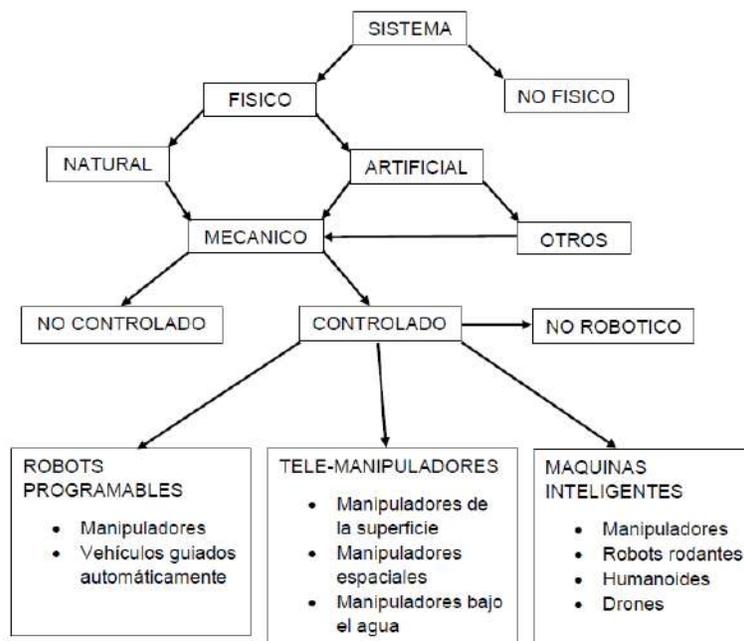
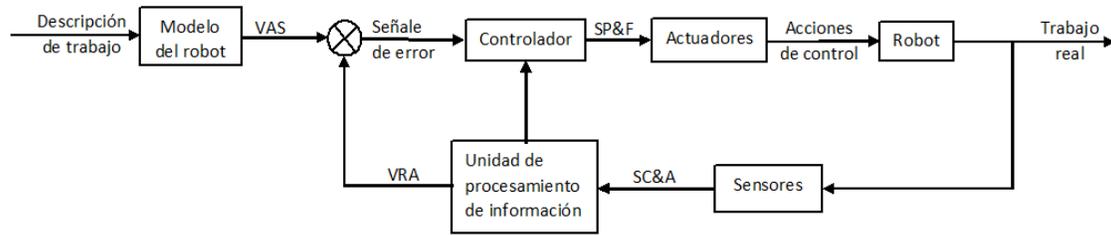


Figura 2.1 Genealogía de los sistemas mecánicos robóticos (Angeles, 2014).



VAS: Variable articulada sintetizada (ángulos pares y de torsión)
 SP&F: Señales de posición y de fuerza
 SC&A: Señales cartesianas y de articulaciones
 VRA: Variables reales de articulaciones (ángulos y pares de torsión)

Figura 2.2 Arquitectura general de un sistema mecánico robótico (Angeles, 2014).

En el diagrama de la Figura 2.2 la señal de entrada es una tarea prescrita para que el robot la ejecute; en bloque donde se encuentra el modelo del robot, se puede decir que es donde se encuentra la inteligencia artificial del sistema, pero en otros casos donde el robot no es lo suficientemente capaz para tomar una decisión, este lugar lo ocupa un operador. En las ocasiones cuando el sistema mecánico robótico sea capaz de tomar decisiones, en este caso el controlador, los actuadores serán manejados por la inteligencia artificial de este; los sensores adquieren las señales del trabajo real y las comparan con la señal prescrita, el error que se obtiene de la comparación se agrega al controlador, y éste toma la acción corregir la señal enviada a los actuadores. En el caso de que un operador maneje el sistema mecánico robótico, esto se deberá a que el sistema no posee una retroalimentación o la inteligencia del robot no es capaz de tomar decisiones ante situaciones complicadas, o simplemente no requiere un gran sistema de control.

2.1.2 Clasificación de los robots

La clasificación de los robots se puede ampliar acorde al propósito del propio robot, pero también a características de fabricación, el entorno en donde el robot se encuentra, etc. En las siguientes clasificaciones se muestran las consideraciones que tiene cada una:

Clasificación acorde a su uso. Esta clasificación utiliza el criterio de que los robots son apropiados para realizar trabajos en ambientes donde al ser humano se le complicaría realizar determinada tarea:

- a) Fabricación automática.
- b) Exploración remota.
- c) Aplicaciones biomédicas.
- d) Manejo de materiales peligrosos.
- e) Robot de servicio.

Clasificación acorde a sus capacidades de operación. Esta clasificación se enfoca al tipo de robots industriales que se vende comercialmente:

- a) Robot de control de secuencia limitada.
- b) Robot de reproducción.
- c) Robot de ruta controlada.
- d) Robot de control adaptativo.
- e) Robot de inteligencia artificial.

Clasificación acorde a su inteligencia. En esta clasificación se contemplan solamente dos grupos, los robots con una **baja tecnología** y los de **alta tecnología**. El primer grupo son robots que no tiene control de servomotores, y cuya tecnología hace que sus sistemas de control estén predeterminados a un ciclo de movimientos único, tal como el de elegir y colocar los componentes en donde cada eje esta normalmente restringido a dos puntos finales; con todo lo mencionado se entiende que este tipo de robots solo ejecuta operaciones simples que no requieren de tecnología tan avanzada. Los robots con alta tecnología se caracterizan por tener un gran equipamiento: ya sea en sensores, sistemas de control más robustos, o simplemente los avances tecnológicos más recientes; esto les permite realizar operaciones en tiempo real que pueden ayudar a prevenir accidentes o tener una mayor variedad de usos (Kuttan K. K., 2007).

Clasificación acorde a sus capacidades estructurales. Esta clasificación toma en cuenta la capacidad estructural del robot, sea este capaz de moverse en el espacio o estar asentado en un lugar designado, por lo tanto, esta clasificación solamente tiene dos tipos de robots: **los fijos y los móviles**. El robot fijo siempre se encontrará unido a una base fija y su comportamiento será la de una cadena cinemática abierta. El robot móvil usa un sistema locomotor que le permite moverse alrededor de su entorno, pero este se divide en otras dos subclases de robots: **el robot de rueda y el robot caminante**; el primero tipo de robot es llamado vehículo de guía automático (AGVs, *Automatic Guide Vehicles*), y tiene un sistema de energía incorporado y otro de rastreo que les permite moverse en un entorno controlado con un sistema locomotor rodante; el segundo tipo de robot son más reminiscentes a una ser vivo debido a que pueden mover en la forma de cómo lo haría un animal en cuatro patas o en dos como el ser humano y reminiscentes (Kuttan K. K., 2007).

Clasificación acorde a su generación de robot. En esta clasificación se dividen en torno a las capacidades computacionales del robot y la forma de interactuar en su entorno:

- **Primera generación.** En esta generación se encuentran los robots que son un simple dispositivo y que operan conforme a estrictas secuencias, estos robots no pueden detectar los cambios a su alrededor, por lo que no pueden rectificar sus acciones.
- **Segunda generación.** En esta generación los robots están equipados con sensores y con un procesador de cómputo para reaccionar ante pequeños cambios en su entorno, de esta forma pueden tomar acciones de corrección, y además resultan ser más complejos que de la primera generación.
- **Tercera generación.** Los robots de esta generación son conocidos como robots inteligentes, tiene una retroalimentación a través de sensores y un de procesamiento de datos basado en técnicas de inteligencia artificial, esto les permite tomar decisiones, resolver problema y realizar tareas de manera inteligente.
- **Cuarta generación.** Los robots que se encuentran en esta generación son más reminiscentes a un ser humano, aunque sin la necesidad de serlo, estas máquinas son similares en inteligencia al ser humano debido a que poseen una gran variedad de sensores, mejores métodos de procesamiento que les permite tomar decisiones rápidas a problemas complicados, por lo que todo esto los vuelven más complejos.
- **Quinta generación.** En esta generación los robots obtienen una mayor complejidad que la anterior, en previas generaciones los robots claramente pueden ser más fuertes que una persona, pero en esta generación los robots poseen una inteligencia y capacidades físicas superiores a las del ser humano.

La cuarta generación actualmente se puede considerar que está en investigación como lo muestra los desarrollos de la empresa Boston Dynamics con su robot Atlas, en cambio, la quinta generación aún se encuentra en un estado teórico, en libros y películas de ciencia ficción (Kuttan K. K., 2007).

Clasificación acorde a la automatización industrial. Esta clasificación responde a como los robots pueden llegar a realizar trabajos en un entorno de producción industrial, los robots manipuladores son comúnmente utilizados en este tipo de ambiente industrial.

- **Automatización fija.** Robot que es usado cuando el volumen de producción es muy alto, entonces se diseña un equipo o sistema apropiado para el procesamiento de un producto.
- **Automatización programable:** Este robot es utilizado en el procesamiento de un producto cuando este es muy bajo y además se tiene una mayor variedad de productos a procesar, estos productos son procesados en lotes y cuando uno es terminado la programación del robot cambia para poder trabajar con el siguiente tipo de producto.

- **Automatización flexible.** En esta clasificación los robots pueden procesar diferentes tipos de productos al mismo tiempo, dándole un mayor alcance en diferentes tipos de productos, pero que puede que sean en cantidades pequeñas en comparación a lo que haría un robot de automatización fija.

2.1.3 Robots industriales

Estos robots tienen diferentes definiciones planteadas por diferentes asociaciones de robots a nivel internacional, en los siguientes párrafos se hace mención a ellas y los organismos que las dictaminan.

La definición planteada por ISO (*International Standard Organization*) es amplia y evita confusiones en cuanto a los diferentes tipos de robots que llegan a existir debido a la función que cada uno desempeña. Existen otras definiciones planteadas por diferentes asociaciones robóticas en diferentes países, pero están más acorde con lo que vendrían siendo los robots industriales, dichas asociaciones son: la Asociación Británica de Robot (BRA), Instituto de Robot de América (RIA) y la Asociación Japonesa de Robot Industriales (JIRA).

La definición planteada por el BRA es la siguiente: un dispositivo programable diseñado para manipular y transportar partes, herramientas o implementos de fabricación especializados, a través de variables de movimientos programables para el desempeño de específicas tareas de fabricación. La definición del RIA es muy similar a la previamente mencionada: un robot industrial es un dispositivo programable, manipulador multifuncional diseñado para mover material, partes, herramientas o dispositivos especializados, a través de variables de movimiento programables para el desempeño de variadas tareas. Ambas definiciones contienen elementos clave que son la programabilidad y la cualidad multifuncional de los robots (Kuttan K. K., 2007).

La JIRA, en cambio, plantea cinco niveles de robots industriales, los cuales son:

- a) **Manipulador.** El cual es directamente operado por un ser humano.
- b) **Robot de secuencia.** El cual entra en más de dos categorías: el robot de secuencia fija y el robot de secuencia variable.
- c) **Robot de reproducción.** Ejecuta instrucciones fijas.
- d) **Los robots de Control Numérico (NC).** Estos ejecutan información cargada numéricamente.
- e) **Robots inteligentes.** Estos robots tienen su propio sistema basado en sensores lo cual ayuda a los programas a tomar decisiones en tiempo real.

2.1.4 Robot manipulador

En los robots industriales se encuentran los robots manipuladores, que son sistemas mecánicos robóticos los cuales han merecido una gran atención, y existen razones por las cuales son gran interés: la primera de ellas es porque aparecen muy frecuentemente en el sector industrial debido a su simpleza, otra se debe a su arquitectura robótica, la cual es la más simple de todas las arquitecturas robóticas, por lo tanto permite a esta arquitectura de ser encontrada continuamente en otro tipo de mecanismos robóticos más complejos. En si un manipulador es: un sistema mecánico con el objetivo de manipular objetos; 'manipular' significa mover algo con la mano, esta palabra proveniente del latín 'manu' que significa 'mano', además este tipo de robots se divide en dos subsistemas: **el brazo robótico y la mano robótica** (Angeles, 2014).

Brazo robótico

En los manipuladores se encuentra un subsistema de robots que son los brazos robóticos que tiene una similitud con los brazos humanos. Los brazos robóticos se constituyen de parejas de enlaces estructuralmente robustas, ya sean articulaciones rotacionales o de traslación. También este tipo de estructuras son una concatenación de enlaces que dan forma a una cadena cinemática abierta, en el cual cada enlace predecesor se está enlazado con un sucesor, excepto el primer y último enlace, los cuales son enlazados con un enlace sucesor y predecesor respectivamente, en la Figura 2.3 se puede observar un ejemplo de este tipo de manipulador robótico (Angeles, 2014). Este mecanismo es el mismo que da forma al robot Yaskawa MH5LF y en donde se acoplara la mano robótica que compete a esta tesis.



Figura 2.3 Brazo robótico con arquitectura en serie (Angeles, 2014).

Mano robótica

La mano robótica es un sistema más sofisticado que el brazo robótico, porque está diseñada para trabajos de manipulación, ya sean simples o que requieren una mayor habilidad en la ejecución.

La manipulación simple. En esta forma de manipulación los dedos de una mano robótica ejercen como estructuras que mantienen al objeto en una posición determinada, y permite colocarlo sobre la palma de la mano robótica, por lo que los dedos pueden llegar a envolver al objeto que estén sujetando.

La manipulación hábil. En este método la manipulación de un objeto conlleva un movimiento más controlado con respecto de la palma. El objeto es agarrado con las puntas de los dedos y el movimiento de este será con respecto a la palma de la mano robótica, este tipo de manipulación es requerido trabajos donde la precisión debe ser alta para un movimiento más suave o sutil.

Las manos robóticas hábiles frecuentemente poseen múltiples dedos y una cadena cinemática abierta altamente redundante. Una mano robótica con múltiples dedos consta de una palma y un conjunto de dedos que se originan desde esta, dando como resultado una topología de árbol. La arquitectura de los dedos es la de un simple manipulador, que pueden tener de dos a cuatro enlaces acoplados a revoluciones, y que cumplen la función de una falange. El agarre ocurre cuando todos los dedos de una mano están sujetando el objeto (Angeles, 2014).

Los diferentes tipos de manos robóticas que se pueden llegar a fabricar constarán de la habilidad requerida para determinada tarea; si un trabajo requiere una simple sujeción, la mano robótica que puede ser utilizada es una pinza, pero si este requiere de una manipulación con más destreza y movimientos complejos, entonces se necesitara una mano robótica diestra. La pinza y mano robótica diestra son dos tipos de manos robóticas que existen, pero hay un tercer tipo que se encuentra entre las dos y se llama mano robótica sub-actuada, los tres tipos de mano robótica se explican aquí abajo:

Pinza

La pinza es un mecanismo que tienen una junta en cada uno de los dedos que posee, pero esta es incapaz de realizar trabajos con una mayor complejidad y además de estar limitada al no ser capaz de ajustarse a diferentes tipos de objetos (Song & Zhang, 2018). Con la función de únicamente abrir y cerrar para presionar un cierto objeto, la pinza solo puede llegar a realizar manipulaciones simples, pero con un mantenimiento muy sencillo que le permite cambiar piezas fácilmente. Las tareas asignadas a la pinza estarían entre la automatización fija y la programable.



Figura 2.4 Robot Hand RH-P12-RN-UR (ROBOTIS HAND, 2021)

Mano robótica diestra

Son las manos robóticas que se basan mayor mente en la imitación del comportamiento de una mano humana, y suelen tener mayor versatilidad que las pinzas industriales, pero debido a la complejidad de este tipo de mecanismo, esta mano robótica resulta ser más complicada fabricar, ya que las piezas son más difíciles de conseguir y el sistema de control es más complejo (Song & Zhang, 2018). La mano robótica diestra puede llegar a sujetar objetos de diferentes formas, eso le permite tener una manipulación hábil sobre el objeto, pero debido a su complejidad el mantenimiento de dicho sistema es más difícil de llevar a cabo que una pinza, aunque los trabajos que tendría este sistema se prestarían en la automatización flexible, programable y fija, sin embargo, en esta última no se estaría dando un uso de todo el potencial del sistema y los trabajos que requieran más fuerza que habilidad podrían llegar a ser contraproducentes con la mano robótica diestra, un ejemplo de esta mano es mostrado en la Figura 2.5.



Figura 2.5 Mano robótica diestra con actuadores lineales (Kim, Lim, & Lee, 2010)

Mano sub-actuada

La mano sub-actuada posee menos actuadores que grados de libertad, pero por lo general tienen tres dedos y no más de nueve grados de libertad. Este tipo de mano es más flexible que las pinzas industriales, aunque no lleguen al grado de flexibilidad de una mano robótica diestra, pero tiene menos complejidad en la fabricación en sus piezas y puede ser controlada más fácilmente (Song & Zhang, 2018). La mano sub-actuada puede ejecutar trabajos de manipulación similares que la mano robótica diestra, pero siendo un mecanismo más simple que esta, y al ser un punto intermedio entre la pinza y la mano robótica diestra puede tener un alcance intermedio en cuanto a la automatización fija, programable y flexible, llegando a tener una mayor fuerza para los trabajos, pero con una menor habilidad.

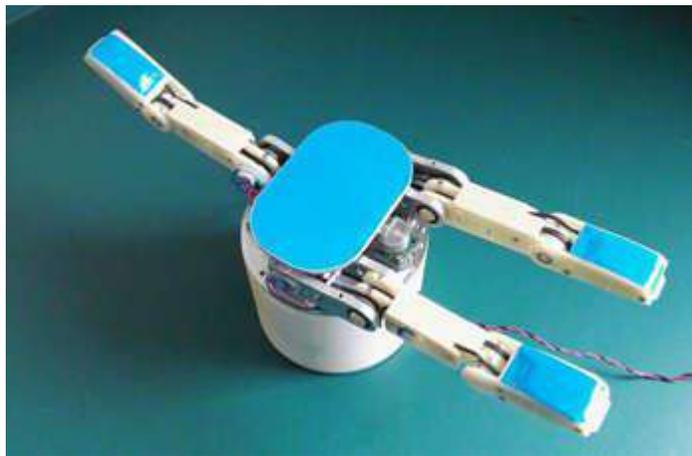


Figura 2.6 Mano robótica sub-actuada (Wang, Zhao, Xiao, Zhu, & Wang, 2018).

2.1.5 Robot Paralelo

Definición

“Un robot paralelo se define como un efector final con n grados de libertad, con una base fija y unida con al menos dos cadenas cinemáticas independientes. La actuación se ejecuta a través de n actuadores simples” (Merlet, 2006). La Figura 2.7 muestra una estructura general de un robot paralelo.

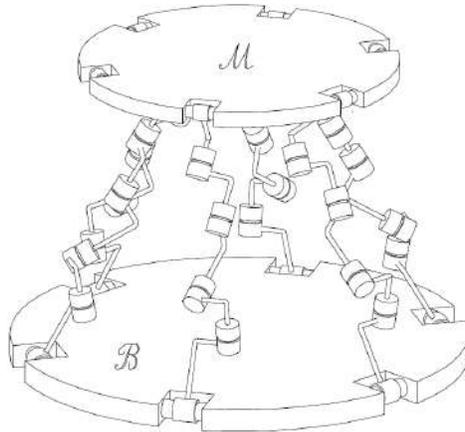


Figura 2.7 Estructura general de un robot paralelo de seis grados de libertad en cada cadena cinemática (Angeles, 2014).

Manipulador paralelo

“Un manipulador paralelo es un mecanismo de cadena cinemática de bucle cerrado en cuya base se unen diferentes cadenas cinemáticas” (Merlet, 2006).

La cadena cinemática de un mecanismo permite representar a los robots paralelos de dos formas: los robots planos y los robots espaciales. Los robots planos se caracterizan por sus cadenas cinemáticas que solamente permiten un movimiento en dos dimensiones, la Figura 2.8 muestra la disposición de un robot plano. En cambio, los robots espaciales son capaces de moverse en las tres dimensiones y la Figura 2.9 ilustra la distribución de cadenas cinemáticas de este en contraparte de las del robot plano.

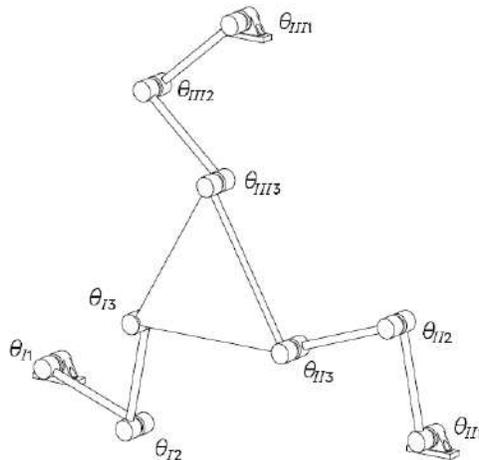


Figura 2.8 Robot paralelo plano (Angeles, 2014).

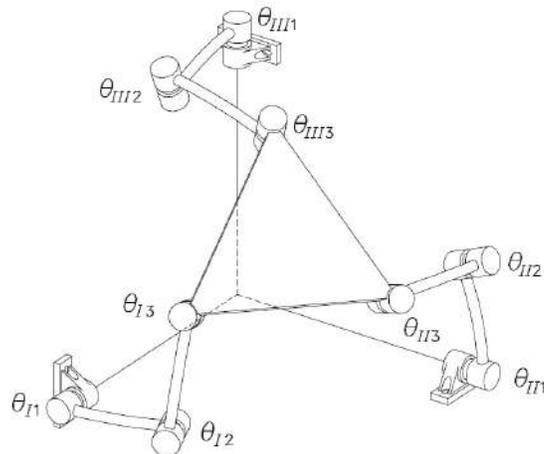


Figura 2.9 Robot paralelo espacial (Angeles, 2014).

El robot paralelo plano tiene diferentes cadenas cinemáticas independientes colocadas sobre una misma base, cada una con su propio actuador, pero los actuadores deben estar en el mismo plano bidimensional para que pueda ejecutar una acción. Los robots paralelos planos pueden encontrarse como diferentes tipos de robots dependiendo de su propósito, por ejemplo: los vehículos de guía automática poseen una base con actuadores independientes lo cual les permite moverse en una superficie plana, aunque también se pueden mover por espacios tridimensionales los actuadores están efectuando sus acciones en un plano bidimensional, la Figura 2.10 muestra un ejemplo.



Figura 2.10 Robot RHEX (Angeles, 2014).

La configuración de un robot paralelo espacial le permite movimiento en tres dimensiones a las cadenas cinemáticas de este, por lo que se puede encontrar en robots que pueda asemejarse en movimiento a un ser viviente que también realicen una gran variedad de tareas que puedan ser simples o complejas. La disposición de los robots paralelos espaciales se asemeja a la de un robot paralelo plano, en cuanto a poseer una base donde todas las cadenas cinemáticas emerjan, pero con la posibilidad de moverse en tres dimensiones, además de tener la capacidad de interactuar con un objeto de su entorno en una forma en la que el robot paralelo plano no podría. Los robots paralelos espaciales que pueden ser un ejemplo son los

manipuladores, en concreto: las manos robóticas, los robots andantes y otros varios tipos de robots que tengan más de una cadena cinemática en una base. En la Figura 2.11 se ilustra un ejemplo de robot paralelo espacial.

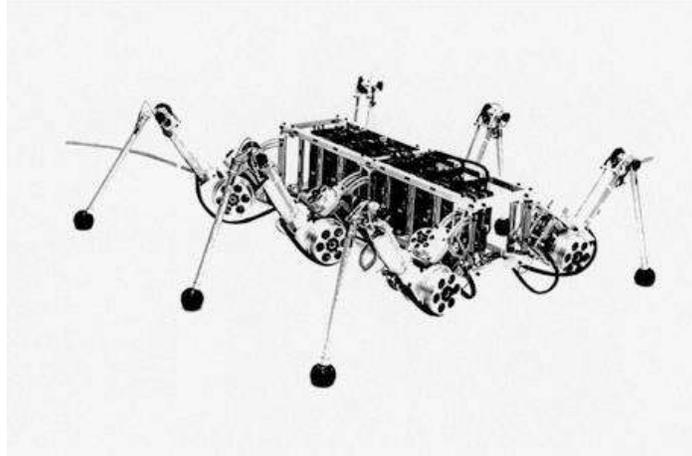


Figura 2.11 Hexápodo (Angeles, 2014).

2.2 Cinemática de un mecanismo

Cinemática

“La cinemática es el estudio del movimiento sin considerar las fuerzas” (Norton, 2009)

2.2.1 Cadena cinemática

Definición

“Una cadena cinemática en un grupo de cuerpos rígidos, también denominados eslabones, acoplados por pares cinemáticos, también llamados articulaciones o juntas” (Angeles, 2014). También se define como: “Un ensamble de eslabones y juntas con un eslabón acoplado al marco de referencia o base de la cadena” (Norton, 2009).

Los pares cinemáticos se pueden dividir en dos categorías: la primera es en el **par cinemático superior**, el cual se origina cuando dos cuerpos entran en contacto y de este se origina en una línea o punto, este par cinemático se pueden encontrar en las levas, trenes de engranaje y en rodamientos; la segunda categoría es el **par cinemático bajo**, el cual ocurre cuando dos cuerpos entran en contacto

en una superficie en común, este par cinemático se compone de seis tipos diferentes: la revoluta, el prismático, el helicoidal, el cilíndrico, el esférico y el plano; pero todos ellos pueden ser originados por la revoluta y el prismático, en la Figura 2.12 se observa los pares cinemáticos de la revoluta (R) y el prismático (P), donde la revoluta permite un movimiento de rotación y el prisma un movimiento de traslación o lineal.

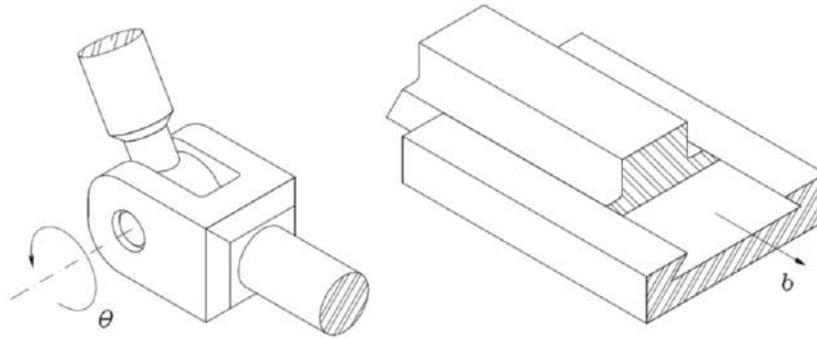


Figura 2.12 Pares cinemáticos bajos: la revoluta y el prismático (Angeles, 2014).

La articulación está formada por al menos dos eslabones: los eslabones son cuerpos rígidos que poseen dos o más nodos, que son puntos de unión con otro eslabón (Norton, 2009). Los eslabones son las piezas que dan forma al mecanismo y lo mantienen unido con los nodos de cada eslabón, también estos pueden poseer más de un solo nodo, permitiendo que un eslabón pueda articularse con dos, tres o los nodos que el eslabón posea, la Figura 2.13 muestra diferentes tipos de eslabones con diferente número de nodos.

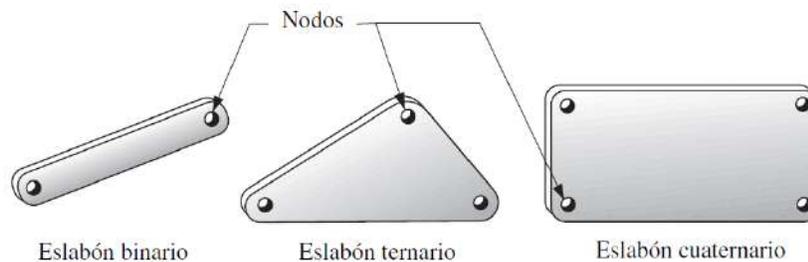


Figura 2.13 Eslabones con diferente número de nodos (Norton, 2009).

Movilidad

El movimiento que el mecanismo tiene, depende del tipo de par cinemático que se le otorgue este, debido a que los dos tipos de pares cinemáticos, la revoluta y el prismático, pueden reproducir dos movimientos diferentes cada uno, y con esto se llega a concluir que existen dos formas de conceder movimiento a una articulación;

un movimiento de rotación y el otro el de traslación, juntos formando un movimiento complejo.

- **Rotación.** En este movimiento se encuentra un punto específico, que es el eje de rotación, el cual no se mueve y todos los demás puntos del cuerpo describen un círculo alrededor del este eje de rotación.
- **Traslación.** En este movimiento todos los puntos del cuerpo trazan un recorrido paralelo o en línea recta.
- **Movimiento complejo.** Es una combinación del movimiento de rotación y traslación, en el cual una línea que se describa por el cuerpo podrá cambiar su posición lineal y orientación angular.

Grados de libertad

Los grados de libertad de son el número de entradas que se necesitan proporcionar para crear una salida predecible en un mecanismo, también son el número de coordenadas independientes necesarias para determinar su posición (Norton, 2009).

Las juntas o pares cinemáticos mencionadas anteriormente en la Figura 2.13 tienen intrínsecamente grados de libertad. La junta prismática, la revoluta y la helicoidal, otro tipo de par cinemático o junta, poseen un grado de libertad; en cambio la junta cilíndrica posee dos grados de libertad, la junta esférica y plana tres grados de libertad, y las combinaciones entre todas ellas crean juntas con más grados de libertad.

La cantidad de grados de libertad que un mecanismo tiene se determina por el tipo de cadena cinemática que este mismo forme. Una cadena cinemática simple, que es la forma más básica de arquitectura robótica, puede ser abierta o cerrada, y para determinar el número de grados de libertad de un mecanismo se utiliza el criterio de Grübler-KutzBach, el cual considera la interacción que sucede con los eslabones y juntas. Este criterio identifica los grados de libertad mediante el uso de dos ecuaciones: una ecuación para mecanismos que se muevan en el plano y otra para los que su movimiento es en tres dimensiones.

- **Ecuación de Grübler-Kutzbach en mecanismos planos**

$$M = 3(L - 1) - 2J_1 - J_2 \quad (2.1)$$

M = grado de libertad o movilidad

L = número de eslabones

J_1 = número de juntas de un grado de libertad

J_2 = número de juntas de dos grados de libertad

En esta ecuación M muestra la cantidad entradas necesaria para mover el mecanismo, teniendo como casos: a) en donde $M = 0$, el mecanismo es una estructura y no puede moverse; b) cuando $M = -1$, en donde el mecanismo tiene restricciones redundantes, por lo que esta indeterminado; c) si $M \geq 1$, el mecanismo debe ser impulsado por las entradas que resulten, ya sea una, dos o las entradas que necesite el mecanismo.

- **Ecuación de Grübler-Kutzbach en mecanismos espaciales**

$$M = 6(L - 1) - 5J_1 - 4J_2 - 3J_3 - 2J_4 - J_5 \quad (2.2)$$

M = grado de libertad o movilidad

L = número de eslabones

J_1 = número de juntas de un grado de libertad

J_2 = número de juntas de dos grados de libertad

J_3 = número de juntas de tres grados de libertad

J_4 = número de juntas de cuatro grados de libertad

J_5 = número de juntas de cinco grados de libertad

J_6 = número de juntas de seis grados de libertad

En esta ecuación al igual que en el mecanismo plano, M indica las entradas necesarias para que el mecanismo se mueva, pero con el agregado de que ahora la ecuación suma juntas de más de dos grados de libertad, llegando a tener las juntas de cinco grados de libertad. En muchos mecanismos las juntas que se utilizan son las de un grado de libertad, como la revoluta o el prismático, ya que ambas poseen un grado de libertad, pero la junta cilíndrica tiene dos grados de libertad, mientras que la esférica y plana poseen tres grados de libertad, y una combinación entre ellas permite generar una junta de cinco o seis grados de libertad.

Una mano robótica está compuesta por varias cadenas simples unidas a una base en común, por lo tanto, para determinar el número de grados de libertad de un mecanismo en paralelo se calcula los grados de libertad de cada cadena, se suman los grados de libertad de todas las cadenas, y esta suma dará la movilidad del mecanismo, sin embargo, esta forma de obtener la movilidad se aplica en casos donde las cadenas difieren en arquitectura, debido a los diferentes tipos de eslabones y nodos que pueden llegar a tener, pero en el caso de que todas las cadenas posean la misma arquitectura solamente es necesario obtener la movilidad de una cadenas y multiplicarla por el número el cadenas.

La forma en que se describirá la cinemática de cada uno de los dedos de la mano robótica de esta tesis, será mediante el uso de la notación de Denavit-Hartenberg. Esta notación permite crear cadenas cinemáticas que prescindan del eje Y en cada articulación, para que sea más fácil identificar el movimiento de cada una de las revoluciones o traslaciones en una cadena cinemática, y de esta forma identificar más fácilmente las medidas del robot en la misma cadena cinemática. La Figura 2.14 muestra un ejemplo de un robot y la cadena cinemática que permite determinar la posición final de órgano terminal.

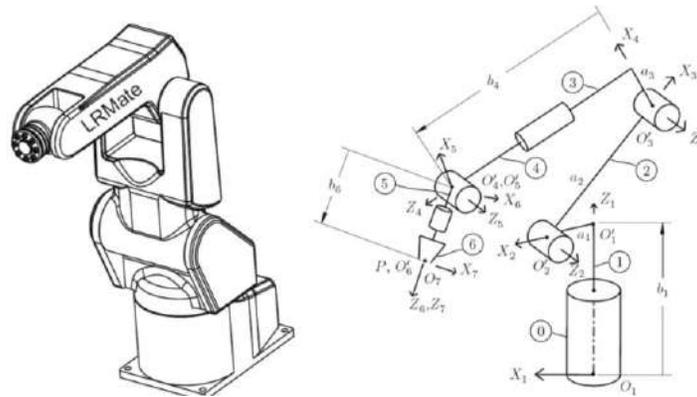


Figura 2.14 Proyección ortográfica del robot LRMate a la izquierda y la cadena cinemática del robot RLMate a la derecha (Angeles, 2014).

2.2.2 Notación de Denavit-Hartenberg

La nomenclatura de Denavit–Hartenberg se emplea a fin de describir de manera única la *arquitectura* de la cadena cinemática, es decir, definir la ubicación relativa y orientación de los ejes de los pares vecinos. Para este fin, los eslabones se enumeran con $0, 1, \dots, n$, el i –ésimo par se define como el que acopla el eslabón $(i - 1)$ con el eslabón i . De aquí, se asume que el manipulador está compuesto de $n + 1$ eslabones y n pares; éstos últimos pueden ser R o P , dónde el eslabón 0 es el que está fijo a la base, mientras que el eslabón n es el órgano terminal. Luego, un marco de coordenadas \mathfrak{S}_i está definido con el origen O_i y los ejes X_i, Y_i y Z_i . Este marco está ligado al eslabón $(i - 1)$ –¡no al eslabón i –ésimo! – para $i = 1, \dots, n + 1$. Para los primeros n marcos, se deben seguir las siguientes reglas.

1. Z_i es el eje del i –ésimo par, note que existen dos posibilidades de definir la dirección positiva de este eje, ya que cada eje del par es únicamente una línea, no un segmento dirigido. Además, el eje Z_i de un par prismático puede ser ubicado arbitrariamente, ya que únicamente su dirección está definida por el eje de este par.
2. X_i se define como la perpendicular común a Z_{i-1} y Z_i , dirigido del anterior al último como se muestra en la figura 4.2a., nótese que, si éstos dos ejes se

intersecan, la dirección positiva de X_i es indefinida y puede ser libremente asignada. En lo sucesivo, nos regiremos por la regla de la mano derecha en tal caso. Esto significa que si los vectores unitarios \mathbf{i}_i , \mathbf{k}_i y \mathbf{k}_i están ligados a los ejes X_i , Z_{i-1} y Z_i respectivamente como se indica en la figura 4.4b, entonces \mathbf{i}_i se define como $\mathbf{k}_{i-1} \times \mathbf{k}_i$. Además, si Z_{i-1} y Z_i son paralelos, la ubicación de X_i es indefinida. A fin de definirla de manera única, se especifica que X_i pase a través del origen del marco $(i - 1)$ como se muestre en la Figura 2.15c.

3. La distancia entre Z_i y Z_{i+1} se define como a_i y en consecuencia en no-negativa.
4. La coordenada Z_i de la intersección O'_i de Z_i con X_{i+1} se denota por b_i . Dado que esta cantidad es una coordenada, puede ser positiva o negativa. Su valor absoluto es la distancia entre X_i y X_{i+1} , también nombrada como la *compensación* entre perpendiculares comunes sucesivas.
5. El ángulo entre Z_i y Z_{i+1} está definido como α_i y se mide sobre la dirección positiva de X_{i+1} . Este término se conoce como *ángulo de torsión* entre ejes de pares sucesivos.
6. El ángulo entre X_i y X_{i+1} se define como θ_i y se mide sobre la dirección positiva de Z_i .

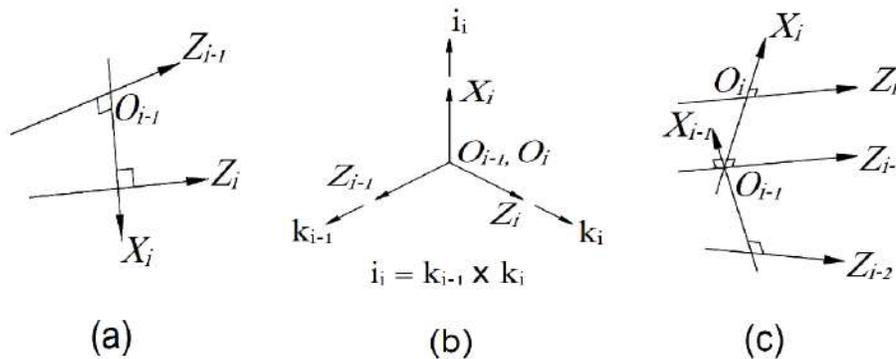


Figura 2.15 Definición de X_i cuándo Z_{i-1} y Z_i : (a) son oblicuos; (b) se intersecan; y (c) son paralelos (Angeles, 2014).

El marco de coordenadas $(n - 1)$ está ligado al último extremo del eslabón $n - \text{ésimo}$. Dado que el manipulado no tiene un eslabón $(n + 1)$, las reglas anteriores no sirven para la definición del último marco. El analista, de este modo, tiene la libertad de definir este marco como mejor se ajuste a la operación a realizar. Nótese que los $(n + 1)$ marcos, $\mathfrak{S}_1, \mathfrak{S}_2, \dots, \mathfrak{S}_{n+1}$, considerando a los eslabones numerados de 0 a n . En resumen, un manipulador de n -ejes está compuesto de $n + 1$ eslabones y $n + 1$ marcos de coordenadas (Angeles, 2014).

La notación de Denavit-Hartenberg proporciona la información de unas variables que están ligadas a las articulaciones del robot, independientemente de la forma de este, estas variables son de movimiento angular o de traslación

dependiendo el tipo de articulación. Cada una de estas variables representa una relación de los marcos de referencia en donde se encuentra la articulación, en la Figura 2.16 se observa un manipulador con articulaciones de revoluta, que brinda al manipulador de un movimiento angular en cada articulación de este, pero también significando que el manipulador puede llegar a tener seis variables.

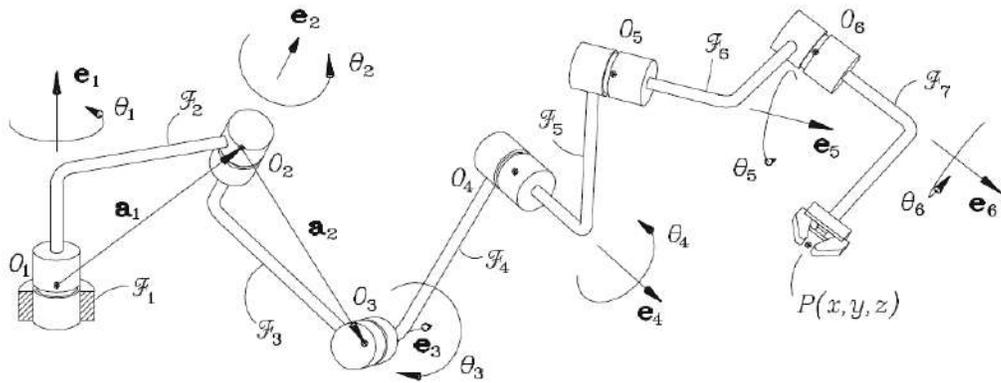


Figura 2.16 Manipulador en serie de seis ejes (Angeles, 2014).

En un caso como el de la figura anterior, el desplazamiento que este mecanismo realice dependerá de articulaciones en revoluta, pero para determinar el problema desplazamiento de este mecanismo u otro con formas diferentes o similares, se dispone de dos formas de afrontarlo: la primera es por medio del problema de desplazamiento directo (DDP, *Direct Displacement Problem*) y la segunda sería mediante el problema de desplazamiento inverso (IDP, *Inverse Displacement Problem*), y en donde cada una de esta forma de análisis describe la forma que adoptada por el mecanismo dependiendo de donde se encuentre el órgano terminal.

En el caso de la Figura 2.16, las ecuaciones obtenidas del mecanismo son las siguientes:

$$[Q_6]_1 [Q_5]_1 [Q_4]_1 [Q_3]_1 [Q_2]_1 [Q_1]_1 = [Q]_1 \quad (2.3)$$

$$[a_1]_1 + [a_2]_1 + [a_3]_1 + [a_4]_1 + [a_5]_1 + [a_6]_1 = [p]_1 \quad (2.4)$$

Las ecuaciones (2.3) y (2.4) muestran la orientación y posición del órgano terminal del mecanismo. La ecuación de orientación representa las seis rotaciones individuales de cada marco de referencia del mecanismo, mediante una matriz de orientación Q , cada marco girando con respecto al marco de coordenadas de la base, en este caso F_1 , y la ecuación de posición representando los vectores posición (a) de cada marco de referencia con respecto a marco de referencia de la base. Las ecuaciones de abajo muestran las relaciones de cada marco individual.

$$[Q_1]_1 [Q_2]_2 [Q_3]_3 [Q_4]_4 [Q_5]_5 [Q_6]_6 = [Q]_1 \quad (2.5)$$

$$a_1 + Q_1(a_2 + Q_2 a_3 + Q_2 Q_3 a_4 + Q_2 Q_3 Q_4 a_5 + Q_2 Q_3 Q_4 Q_5 a_6) = P \quad (2.6)$$

Las ecuaciones pueden ser resueltas de las dos formas DDP y IDP. En el caso de DDP la información de entrada son los ángulos de rotación y la salida es la posición del punto de operación P del órgano terminal, la forma de obtener dicho valor es simple, por el hecho de que solamente se necesita agregar los ángulos que cada revoluta gira. El inconveniente que existe con en DDP, es que no se puede saber con exactitud la posición en la que se llegara a posicionarse el punto de operación, causando que se realicen continuas pruebas, hasta lograr acertar con una combinación de movimientos angulares que den con una posición optima.

En el IDP la entrada es la posición de punto de operación y las salidas son los movimientos angulares de cada revoluta, pero la forma de obtener los valores es más complicada, debido al hecho de debe despejar más de una sola variable y de esa forma obtener cada valor. En esta forma la posición del punto de operación no resulta en continuas pruebas, si no que este es una simple entrada, sin embargo, la orientación del órgano terminal también debe ser determinada mediante la ecuación de orientación, aunque esto no es muy complicado debido a que solamente se debe obtener las rotaciones del todo el mecanismo con respeto al marco de referencia de la base.

2.3 Actuadores

Los actuadores son dispositivo que permiten generar el movimiento un sistema mecánico, impulsando las articulaciones necesarias para colocar el órgano terminal en la posición en la que se desea posicionar. Los dispositivos que generan el movimiento de las articulaciones son los motores, en los cuales la energía más común en ellos es la electricidad, aunque también pueden ser accionados por un fluido o un gas comprimido. Los motores generalmente proporcionan un movimiento angular, pero un movimiento lineal requiere de otro dispositivo, por lo general son los mecanismos hidráulicos o neumáticos los que producen tal movimiento con un cilindro.

2.3.1 Motores eléctricos

Los motores eléctricos se clasifican acorde a su configuración eléctrica o por la función de estos. La clasificación en base a una configuración eléctrica es debido al tipo de corriente eléctrica suministrada al motor, por lo tanto, esta clasificación inicia su división en motores de corriente directa (CD) y motores de corriente alterna (CA); sin embargo, la clasificación de los tipos de motor eléctrico en base a su función es: el motor de engranes, el servomotor y el motor a pasos.

Motores de CD y CA

Los motores de corriente directa tienen diferentes configuraciones las cuales son alimentadas por una señal eléctrica continua, la división de configuraciones se debe al hecho de que las bobinas del rotor están conectadas con respecto a las bobinas del estator o que el estator posea imanes en vez de bobinas. Las configuraciones del motor de CD son las siguientes: imán permanente, devanados en derivación, devanados en serie y devanados compuestos; cada una de ellas genera un diferente tipo de par de torsión.

En cambio, los motores de corriente alterna son alimentados generalmente por una señal eléctrica sinusoidal con una frecuencia de 60 Hz o de 50 Hz, una amplitud pico a pico de ± 120 , ± 240 , ± 480 , y la variedad de este tipo de motor en CA es más amplia que la de CD, pero las velocidades de los motores de CA están restringidas, debido a que están limitados en función de la frecuencia de alimentación. En la Figura 2.17 se muestra un diagrama con los diferentes tipos de motor de CA.

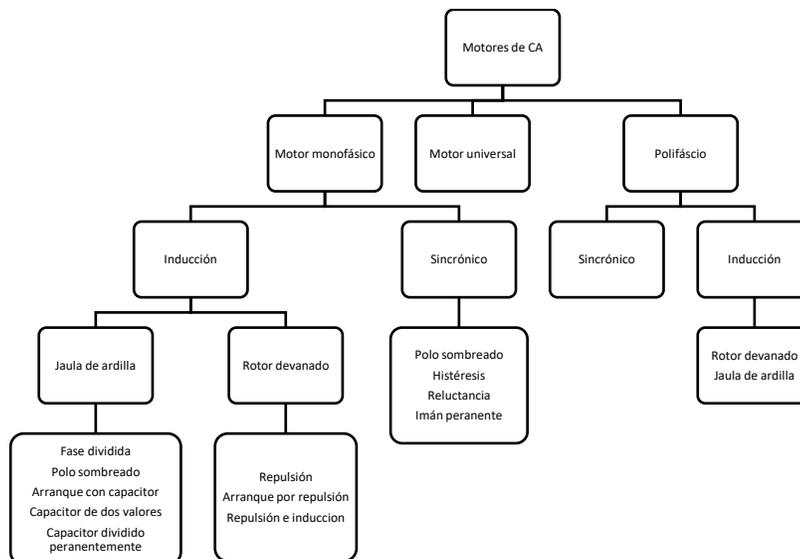


Figura 2.17 Tipos de motor de AC

El motor universal que se muestra en el diagrama de arriba también se encuentra en los tipos de motor de CD. Los motores de CD y CA son de un movimiento rotativo continuo, y si en algún caso se detuvieran, la corriente que los alimenta generaría un sobrecalentamiento en ellos, por lo que no pueden ser útiles en trabajos que requieran un paro, ya sea momentáneo o largo.

Motores de engranes

Este tipo de motor pueden ser de CD o CA, pero lo que hace destacar a estos motores son los engranajes que este posee. Los engranajes de este motor están colocados en la salida del motor, que es el eje del motor, los engranajes están colocados dentro una caja, la cual se puede encontrar en la parte interna de la carcasa del motor o afuera de ella. La caja de engranajes ocasiona una reducción de velocidad, pero en cambio proporciona una mayor potencia al motor, por lo que este tipo de motor se puede tener en una amplia variedad de configuraciones debido a las diferentes velocidades y potencias en la que se puede construir (Norton, 2009).

Servomotores

Los servomotores son motores capaces de tener una aceleración o cambio de velocidad, un control de posición y mantener la posición fijamente en contra de una corriente de alimentación, lo que significa que no sufren de sobrecalentamientos como los motores de CD o AC, pero los servomotores pueden ser alimentados de una corriente directa o alterna. Un servomotor posee sensores que permiten un control en lazo cerrado, los sensores transmiten la información de velocidad y posición del servomotor, de esta forma se puede cambiar la dirección de rotación del rotor, aumentar o disminuir el flujo de corriente con el que se alimenta, y cambiar la frecuencia de la señal de alimentación. Los sensores permiten una mayor precisión de posicionamiento, lo cual ocasiona que los servomotores se encuentren en aplicaciones de control de robots con gran frecuencia (Norton, 2009).

Motor de pasos

Los motores de pasos poseen imanes permanentes sin escobilla en el rotor en lugar de una bobina, estos motores mayoritariamente funcionan sin retroalimentación, es decir, en lazo abierto en vez del lazo cerrado como los servomotores, lo cual provoca que tengan desfases, pero al igual que estos últimos mantienen una posición fija durante un tiempo indefinido debido a que la alimentación constante que pueden recibir. Los motores a pasos son de movimiento intermitente y no tiene un movimiento continuo como los demás motores, poseen

un par de torsión bajo en comparación con los motores continuos de CD y AC (Norton, 2009).

2.3.2 Mecanismos hidráulicos y neumáticos

Motores hidráulicos y neumáticos

Este tipo de motores requieren de un gas comprimido o un fluido, su eficiencia es menor que la de los motores eléctricos, debido a la pérdida que existe en la conversión de la energía, pero generan al igual que los motores eléctricos un movimiento rotatorio. Los motores hidráulicos y neumáticos son más simples de construir sin considerar todo el sistema que los mantiene en funcionamiento, por eso se encuentran más frecuentemente en ambientes cerrados como talleres y fábricas, en donde un sistema suministra al motor un gas comprimido o un fluido en específico, también se encuentran en sistemas y maquinas tales como buques, aviones y grúas (Norton, 2009).

Cilindros hidráulicos y neumáticos

Estos dispositivos proporcionan un movimiento lineal en comparación con los motores, ya sean eléctricos, hidráulicos o neumáticos; también requieren de un gas comprimido o fluido, como los motores hidráulicos y neumáticos, al igual requieren un sistema que les proporcione gas o fluido que para su funcionamiento, aunque no necesariamente deben estar en un ambiente cerrado. Los cilindros tienen un límite de acción, que es la longitud de carrera del pistón en un cilindro, los lugares en donde se encuentran los cilindros hidráulico o neumático son los equipos de construcción o graciola, como los tractores y retroexcavadoras (Norton, 2009).

2.4 Precedentes de este proyecto

En este proyecto se pretende construir una mano robótica que tiene como inspiraciones o similitudes con otros trabajos encontrados en la Universidad Autónoma de Querétaro, en otras instituciones similares a la UAQ, e incluso son publicaciones de organismos de nivel internacional. La mención aquí abajo de cada uno de los trabajos precedentes tiene como propósito el destacar los aspectos que se tomaron en consideración para el desarrollo de este proyecto, además de aclarar el enfoque que tiene este proyecto. Estas menciones sirven para poder entender hasta dónde llega el trabajo en cuanto a desarrollo teórico y práctico, no para que se lleve más haya de estos trabajos previos y con ello se genere una idea de superar lo mencionado, esto último dicho es debido a que el proyecto tiene como objetivo el

ser un aditamento a un brazo robótico que se encuentra en las instalaciones de la UAQ y sirva más como equipo didáctico. Los trabajos previos a los que se hace mención son los siguientes:

Diseño, construcción y control de un robot didáctico de seis grados de libertad

Este trabajo tuvo como objetivo la construcción de un brazo robótico de seis grados de libertad para el uso didáctico en la UAQ, además de implementar el uso de un controlador para una mayor exactitud en el movimiento del brazo robótico. Cabe señalar que la caracterización de la cinemática y dinámica del brazo robótico son necesarias para de esta forma poder tener un modelo matemático del robot, el cual permita caracterizar una forma de control necesario para dicho brazo, con lo mencionado, lo que relaciona este trabajo con lo que está desarrollado en esta tesis es el uso de los parámetros de Denavit-Hartenberg para la caracterización de una cadena cinemática, pero no para un análisis dinámico, la Figura 2.18 muestra el prototipo al cual se hace referencia y en él se puede observar el uso de términos utilizados para el desarrollo de cadenas cinemáticas.



Figura 2.18 Robot de seis grados de libertad construido, con una vista de sus articulaciones (M.C. González Gutiérrez, 2013).

Diseño e implementación de una arquitectura abierta eléctrica y electrónica para un robot de seis grados de libertad

Este trabajo al igual que el anterior se centra en un brazo robótico de seis grados de libertad, pero con la particularidad que este trabajo utiliza un brazo robótico donado a la UAQ, además de que este proyecto se enfoca más en la reparación de dicho robot, ya sea la con la construcción de una red de suministro a brazo robótico, la caracterización de sus señales y la obtención de la arquitectura eléctrica; pero al igual que el trabajo anterior lo único que se relaciona con la tesis

desarrollada en este documento son los parámetros de Denavit-Hartenberg para la caracterización de la cadena cinemática, en la Figura 2.19 se puede observar el brazo robótico utilizado en el trabajo mencionado.



Figura 2.19 Robot industrial Nachi SA160F-01i (Correa Martínez, 2014)

Diseño mecatrónico óptimo de manipuladores paralelos a través del modelado y simulación

En este trabajo el principal objetivo es crear un diseño asistido por computadora (CAD) de un robot paralelo, el cual permita analizar el propio diseño y hacer ajustes al mismo. La forma en como está planteado este trabajo es más complicada de lo que fue previamente descrito, el diseño en CAD del robot paralelo abarca desde su cadena cinemática, dinámica, esquemas de control, la calibración de los robots paralelos, métodos de posicionamiento y orientación, técnicas de modelado y simulación. Este trabajo se basa en un robot paralelo ya construido, el cual le permite enfocarse en desarrollo de diseño en CAD para que de esta forma pueda analizarse los problemas que presenta el robot paralelo, la Figura 2.20 muestra dos diseños en CAD de un robot paralelo con la implementación de sus actuadores. En cuanto lo que sé que desarrolla en esta tesis es simplemente un análisis cinemático de un robot paralelo, y cuyas similitudes son la de simplemente ser robots paralelos, con un estudio de la cinemática directa e inversa de los robots, además de la simulación del robot, pero no en el mismo entorno de simulación y con profundidad de este trabajo, todo lo demás de este proyecto es algo que esta tesis no tiene como objetivo hacer.

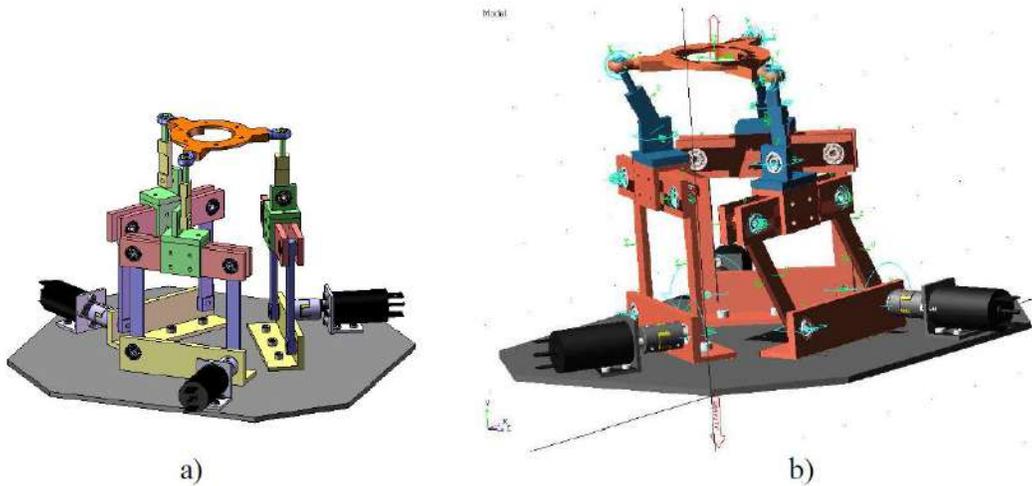


Figura 2.20 Modelo de CaPaMan2bis: a) en entorno SolidWorks 200; b) en entorno ADAMS (Hernández Martínez, 2010).

Mano robot sub-actuada para manipulación de dos brazos (Underactuated robot hand for dual-arm manipulation)

Este trabajo mencionado tiene más semejanzas con lo que se desarrolla en esta tesis, debido a que este trabajo consiste en una mano rotica la cual funciona como efector final para un brazo robótico, aunque con ciertas particularidades que diferencian esta mano robótica con la que se desarrolla en esta tesis. Lo que más destaca de esta mano es la capacidad para adaptarse a objetos con diferentes formas, permitiéndole sujetar con una mayor firmeza el objeto en cuestión, pero la particularidad de que solamente sujeta al objeto y no lo manipula, porque esta capacidad de manipular es posible cuando esta mano se encuentra añadida al brazo robótico. Lo que permite a esta mano robótica sujetar a un objeto es un mecanismo que da forma a cada dedo, este mecanismo se basa en un resorte que se encuentra en las articulaciones de la falange proximal y distal de cada dedo, este mecanismo mueve la falange distal al objeto y por último cierra con la falange distal, todo esto siendo posible mediante el uso de dos motores de corriente directa: uno para mover las falanges de los dedos simultáneamente mediante el mecanismo que da forma a cada dedo, y el otro motor para hacer rotar dos dedos en la base de la mano robótica, con esto último dándole la capacidad de poseer ocho grados de libertad, la Figura 2.21 muestra como está conformada la mano robótica de este trabajo. Las similitudes más cercanas de este trabajo con lo tesis que se desarrollada, es que ambos son un sistema robótico paralelo, pero con la diferencia que los ocho grados que posee esta mano no le permiten una gran maniobrabilidad para manipular un objeto solamente sujetarlo, aun con la misma cantidad de grados de libertad. La mano robótica desarrolla en esta tesis en comparación con esta mano robótica mostrada Figura 2.21 tiene la capacidad de mover cada articulación, sin tener un

mecanismo que haga depender directamente cada articulación de la otra, además de ser capaz de manipular el objeto.



Figura 2.21 Mano robótica (Yamaguchi, Hirata, & Ko, 2015).

Mano robótica con acoplamiento y adaptación en paralelo (The CPAM Hand: Coupling-Parallel-Adaptation Merged Robot Hand)

En este trabajo se presenta el desarrollo de un sistema mecánico en los dedos de la mano robótica, el sistema mecánico está compuesto de engranajes, que permiten a cada dedo tomar una posición dependiendo del objeto que se quiere sujetar. El trabajo presenta similitudes en el descrito anteriormente, como la posibilidad de hacer rotar dos de sus dedos en la base de la mano robótica, pero lo que los diferencia es que cada dedo posee un actuador que les permite moverse de manera individual y colocarse en posiciones requeridas para sujetar un determinado objeto. La particularidad de esta mano robótica es que puede sujetar un objeto no solamente con las dos falanges de cada dedo, sino que también lo hace utilizando únicamente la falange distal de cada dedo como se muestra en imagen del medio y derecha de la Figura 2.22. La capacidad de sujetar de esta mano robótica de la Figura 2.22 con solamente sus falanges distales es una particularidad que comparte con el trabajo desarrollado en esta tesis, pero esta mano robótica tiene un enfoque en la sujeción más que el de la manipulación.

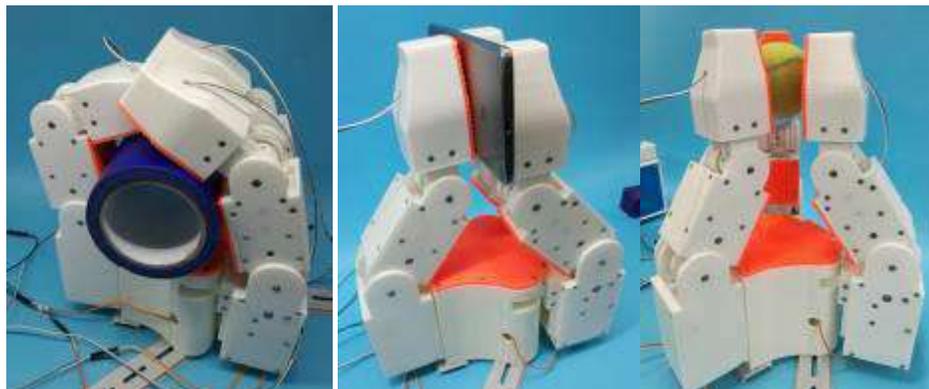


Figura 2.22 Experimento de agarre de la mano CPAM (Song & Zhang, 2018).

Diseño de una mano robótica sub-actuada con tres dedos articulados (Design of underactuated robotic hand with three articulated fingers)

El trabajo que se menciona a continuación posee similitudes con las dos manos robóticas mencionadas previamente; con la primera mano robótica su similitud es de la disposición de sus dedos en la base de mano robótica y un motor de corriente directa que hace rotar simultáneamente dos de sus dedos en su base; con la segunda mano robótica, la similitud es la de poseer para cada dedo un motor de corriente directa en la articulación de la falange proximal, además conectado con un mecanismo de transmisión individual a cada dedo el cual les permite tener la movilidad individual; pero lo más destacado es la de poseer dos sensores de fuerza en las falange distal de cada dedo y uno en la base de la mano robótica, en la imagen de la derecha de la Figura 2.23 se observa coloreado de azul el lugar en donde se encuentran los sensores de fuerza, estos últimos le dan a la mano robótica la capacidad de detectar el momento en que hace contacto con el objeto y permitiendo saber la fuerza que se necesita aplicar para sujetar dicho objeto. La característica que más se asemeja al trabajo desarrollado en la tesis desarrollada no es la de poseer sensores de fuerza o la disposición de sus dedos, sino la de poseer sensores en cada uno de sus articulaciones para detectar si la rotación de determinada articulación es correcta; en cuanto al sensor implementado en el desarrollo de esta tesis, este está integrado en cada actuador, los cuales son servomotores.

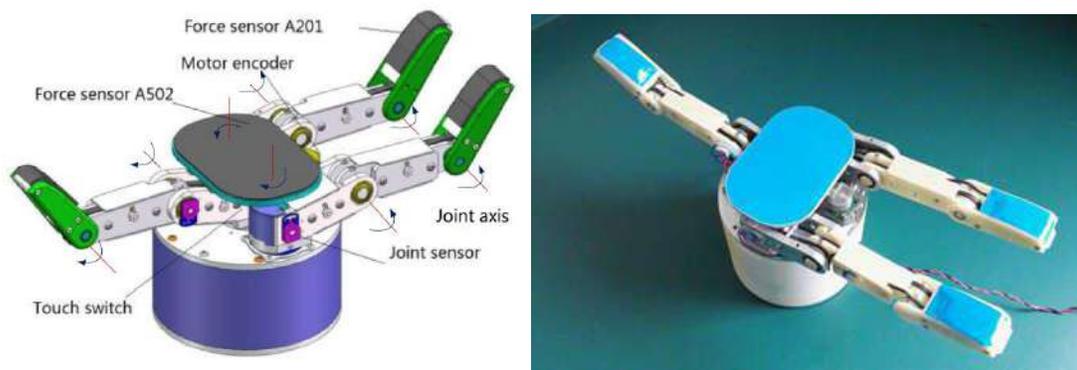


Figura 2.23 Modelo 3D en SolidWorks de una mano sub-actuada (imagen de la izquierda), prototipo de mano sub-actuada (imagen de la derecha) (Wang, Zhao, Xiao, Zhu, & Wang, 2018).

Todos los trabajos mencionados anteriormente proporcionan ciertas ideas para el proyecto que se desarrolla en esta tesis, particularmente los primeros trabajos tienen similitudes en cuanto al desarrollo de una cadena cinemática de un manipulador en el espacio tridimensional, pero en cuanto al trabajo de robot paralelo

se deja entreve que la idea de un manipulador de varias cadenas cinemáticas es lo que se pretende aplicar en esta tesis, y para cuando se muestra los tres trabajos de las manos robóticas la idea es más clara de lo que se pretende desarrollar en este tesis; la cual es crear una mano robótica que tenga la capacidad de sujetar, manipular un determinado objeto, y mediante el análisis de la cadena cinemática se diseñe los dedos de la mano robótica. El diseño la mano robótica y el desarrollo de las ecuaciones para cada dedo es explicada en el siguiente capítulo, el cual clarifica como es que implementa dicha idea de sujeción mediante cadenas cinemáticas, al igual que aclara como es que luce la mano robótica de esta tesis y de esta manera distinguirla de los trabajos mencionados previamente.

3 DISEÑO Y CONSTRUCCIÓN DE LA MANO ROBÓTICA

En este capítulo se verá el desarrollo del diseño de la mano robótica, como también el cálculo de su arquitectura cinemática, la simulación de la arquitectura cinemática, y por último la fabricación de todas las piezas que conforman dicha mano robótica. Para lograr lo anteriormente mencionado es necesario partir de unos cuantos puntos clave en el diseño de la mano robótica:

- **El peso total.** Este punto se refiere al peso de la mano robótica y del objeto juntos, ya que este dato esta delimito por el peso que el brazo robótico Yaskawa MH5LF puede sostener, que es de 5 kg máximo.
- **El objeto.** En este caso es el peso del objeto a sostener, su forma y las dimensiones que este posee. Una esfera de 3.5 kg máximo, con un diámetro mínimo de 1cm y uno máximo de 20 cm.
- **La estructura de la mano robótica.** La cantidad de dedos que tendrá la mano robótica, la distribución espacial de cada uno de ellos, las dimensiones de estos y el peso total de mano robótica. Tres dedos separados 120° entre cada uno de ellos sobre una base, capaces de rodear al objeto de 20 cm de diámetro, y un peso de 1.5 kg máximo.
- **La movilidad de la mano robótica.** Este punto hace referencia a la movilidad que posee toda la mano robótica, y como está distribuida en cada uno de dedos. Ocho grados de libertad en total, dos dedos con tres grados de libertad y uno dedo con dos grados de libertad.

3.1 Diseño conceptual

Consideraciones iniciales

En el diseño conceptual de la mano robótica se ha de tomar en cuenta las capacidades mecánicas de brazo robótico Yaskawa MH5LF, en específico la carga que este puede soportar, la cual es de 5 kg. Una vez conociendo la carga máxima que soporta el brazo robótico, esta medida se toma como referencia en el momento de seleccionar los tipos de actuadores que la mano robótica necesitara, así como el diseño la base de la mano robótica, las uniones entre los actuadores, y el órgano terminal de cada cadena cinemática, que en este caso son los dedos. En la carga útil del brazo robótico se ha de contemplar que el peso la mano robótica, por lo que se propone una carga de 3.5 kg, el cual es el peso máximo que la mano robótica manipulara, dejando 1.5 kg de carga para todo el mecanismo que conforma la mano robótica.

En el diseño de la mano robótica se ha de contemplar el tamaño del objeto a sujetar, al igual que la forma del objeto, sé propone que la forma del objeto sea una

esfera la cual permite que el cálculo de cadena cinemática sea más sencillo. El órgano terminal de la cadena cinemática también serán esferas para un cálculo sencillo de la cadena cinemática, en la Figura 3.1 se muestra un boceto de como lucirá un dedo de la mano robótica de manera lateral además de como interactuarían la parte final del órgano terminal y el objeto.

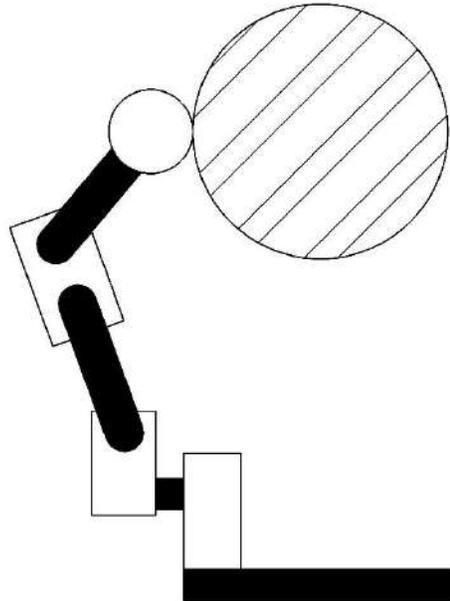


Figura 3.1 Boceto de un dedo de la mano robótica

La mano robótica debe ser capaz de sujetar un objeto de 3.5 kg máximo, pero esta posee tres dedos, por lo que la fuerza que ejerza al objeto será repartida entre los tres dedos. Calcular la medida que el objeto va a tener ayuda a determinar las dimensiones de las partes de la mano robótica, en el cálculo se debe considerar la máxima masa de carga que la mano robótica manipulará, de esta forma se obtendrá las medidas de la base de los dedos: la separación adecuada que debe haber entre los actuadores, el material utilizado en las piezas de unión entre los actuadores, y la del órgano terminal.

3.2 Metodología

Dado que se conocen las características del brazo robótico Yaskawa MH5HL para el que se pretende construir esta mano robótica, se planteó una estrategia para el diseño y fabricación del dispositivo, la Figura 3.2 que se muestra aquí abajo describe la metodología a seguir.

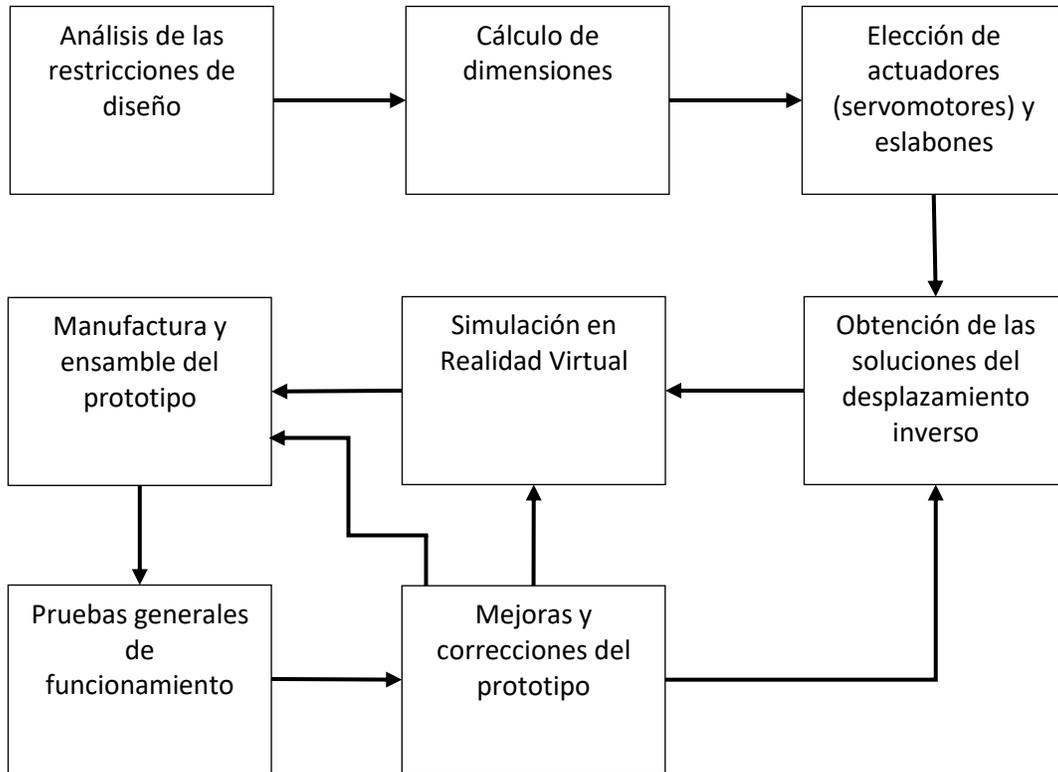


Figura 3.2 Diagrama de la metodología del desarrollo y manufactura del prototipo.

Cálculo de dimensiones

En el cálculo de las dimensiones del objeto, el peso máximo del material elegido es de 3.5 kg, este peso dividido en tres es lo que un dedo deberá soportar, y es que el peso objeto se distribuye entre los tres dedos contemplados, por lo que el resultado la operación siguiente es acorde a esta consideración:

$$m = \frac{3.5 \text{ kg}}{3} = \frac{7}{6} \text{ kg}$$

una vez calculado la masa que un dedo de la mano robótica deberá soportar, la fuerza que ejerza dicha masa se calcula con la aceleración gravitacional de la tierra:

$$F = m a \quad (3.1)$$

por lo que,

$$F = \left(\frac{7}{6} \text{ kg}\right) \left(9.81 \frac{\text{m}}{\text{s}^2}\right) = 11.445 \text{ N}$$

después con el dato de la masa se calcula el volumen que puede tener el objeto, para ello se calcula un volumen considerando un determinado material, en este caso el aluminio con una densidad de 2.7 g/cm³ y una masa máxima de 3.5 kg:

$$\rho = \frac{m}{V} \quad (3.2)$$

el volumen que se obtiene despejando la ecuación (3.2),

$$V = \frac{m}{\rho} = \frac{3500 \text{ g}}{2.7 \frac{\text{g}}{\text{cm}^3}} = 1296.29 \text{ cm}^3$$

obteniendo el volumen de objeto, se puede obtener el radio, este último dato muestra la distancia que existen desde el centro del objeto hasta la superficie del órgano terminal de cada dedo, en este caso una esfera, y con esto se puede esbozar la localización de los dedos en la base:

$$V = \frac{4}{3}\pi r^3 \quad (3.3)$$

despejando el radio de la ecuación (3.3) se puede obtener este último dato.

$$r = \sqrt[3]{\frac{3V}{4\pi}} = 6.76 \text{ cm}$$

Los actuadores que se utilizarán son servomotores, por lo que es necesario calcular el torque para determinar cuáles serán los servomotores adecuados para cada articulación, la Figura 3.3 muestra la disposición de los dedos para calcular el torque necesario de cada servomotor, y la forma en que se calcula es la siguiente:

$$\tau = Fd \quad (3.4)$$

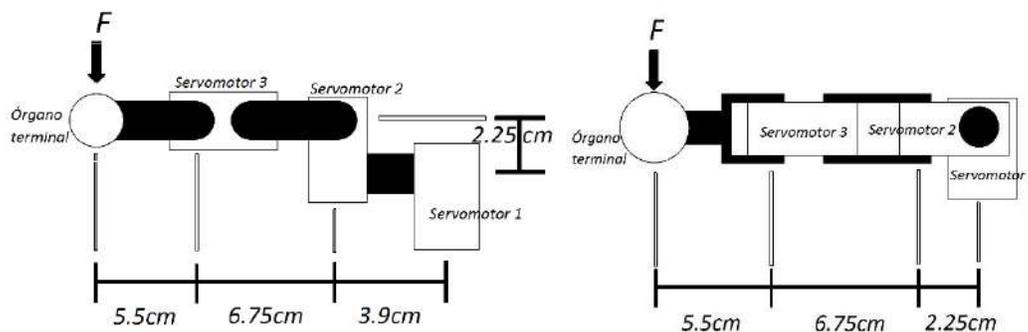


Figura 3.3 Bocetos de las disposiciones horizontales de un dedo de la mano robótica.

Las dos imágenes que se encuentran en la Figura 3.3 muestran las formas en las que se encontraría un dedo horizontalmente, con la carga de dividida entre tres del objeto. Esto permite plantear el cálculo de torque para cada servo motor, por lo tanto, el torque del servomotor número uno es el siguiente:

$$\tau_1 = Fd = (11.445 \text{ N})(5.5 \text{ cm} + 6.75 + 2.25) = 1.72 \text{ Nm}$$

el torque del servomotor número dos es:

$$\tau_2 = Fd = (11.445 N)(5.5 \text{ cm} + 6.75) = 1.4 \text{ Nm}$$

el torque del servomotor número tres es:

$$\tau_3 = Fd = (11.445 N)(5.5 \text{ cm}) = 0.63 \text{ Nm}$$

Las distancias mostradas en la Figura 3.3 son una inspiración para el diseño preliminar de un dedo de la mano robótica mostrada en la Figura 3.4, en ella se observan dos imágenes que muestran ya el diseño de un dedo, la imagen de izquierda es una vista lateral, en cambio la imagen de la derecha es una vista frontal en donde se percibe la flecha del servomotor, el servomotor número uno en la Figura 3.3 tiene una disposición perpendicular a los otros dos servomotores, el número dos y tres se encuentran en paralelo.

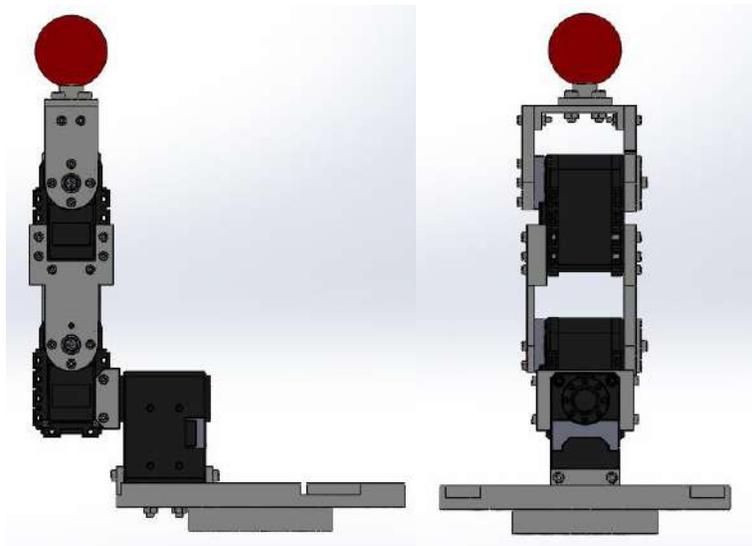


Figura 3.4 Diseño preliminar de un dedo de la mano robótica.

En la Figura 3.4 también se puede observar una plataforma sobre la cual se encuentra montado el servomotor número uno, esta es la base de la mano robótica, al igual se puede observar las piezas de cada eslabón que mantienen unido todo el mecanismo y le dan forma. Este diseño será descrito con más detalle en los siguientes subcapítulos.

Actuadores

Los actuadores de la mano robótica deben ser capaces de generar un torque mayor al calculado previamente para cada servomotor, $\tau_1 = 1.72 \text{ Nm}$, $\tau_2 = 1.4 \text{ Nm}$, $\tau_3 = 0.62 \text{ Nm}$. Los actuadores que se observan en la Figura 3.4, son los seleccionados para proporcionar el movimiento en cada articulación, estos servomotores son Dynamixel y se utilizan dos modelos, el servomotor número uno

es el modelo Dynamixel XM430-W210-T, y para los servomotores dos y tres se utiliza el modelo Dynamixel AX-12A.



Figura 3.5 Servomotor Dynamixel XM430-W210-T.

En la Figura 3.5 se puede observar el servomotor XM430-W210-T con más detenimiento, las especificaciones técnicas de este servomotor son las siguientes:

- Dimensiones: 28.5 mm en anchura, 46.5 mm de altura y 34 mm de profundidad.
- Peso: 82 g.
- Par de torque: 3.8 Nm con una alimentación de 11.1 V y 2.1 A.
- Alimentación: de 10 a 14.8 V, recomendando 12 V.
- Resolución: 4096 pulsos/r.
- Relación de transmisión: 353.5:1.
- Grados de carrera: 0° a 360°.
- Protocolos de comunicación: serial asincrónico TTL de 8 bits, 1 parada, ó serial sincrónico RS485 de 8 bits, 1 parada.
- Velocidad de transmisión: 9600 bps a 4.5 Mbps.
- Temperatura de operación: -5 °C a 80 °C.
- Retroalimentación: posición, velocidad, corriente, trayectoria, temperatura, voltaje de entrada, etc.



Figura 3.6 Servomotor Dynamixel AX-12A.

La Figura 3.6 muestra con detalle el servomotor AX-12A, las especificaciones técnicas de este servomotor son las siguientes:

- Dimensiones: 32 mm de anchura, 50 mm de altura y 40 mm de profundidad.
- Peso: 54.6 g
- Par de torque: 1.5 Nm a 12 V y 1.5 A.
- Alimentación: 9 a 12 V, recomendando 11.1 V.
- Resolución: 0.29, 1034 pulsos/r.
- Relación de transmisión: 254:1.
- Grados de carrera: 0° a 300°.
- Protocolo de comunicación: serial asincrónica semidúplex de 8 bits 1 parada
- Velocidad de transmisión: 7843 bps a 1 Mbps.
- Grados de carrera: -5 °C a 70 °C.
- Retroalimentación: posición, temperatura, voltaje de entrada, etc.

Con las características mostradas de cada servomotor, se tiene la idea que son lo suficientemente capaces de soportar la carga dividida entre 3.5 kg, y esto debido al torque que pueden llegar a ejercer cada servomotor es superior al torque calculado. Algunas especificaciones de los servomotores que muestra cada uno de los dos no son iguales, pero puede apreciarse como se puede utilizar cada uno.

En cuanto a las piezas que mantendrán unidas a los servomotores, y para dar forma a un solo dedo de la mano robótica, se trazan acorde a las medidas de los servomotores y de las facilidades de acoplamiento que brindan cada uno, además los tornillos que se pueden utilizar son medidas estandarizadas y pueden ofrecer posibilidad de acoplamiento sencillo. El diseño de cada pieza es creado en el programa de modelado SolidWorks, cada una de ellas es modelada con el material Acido Poliláctico (PLA), con excepción de una de dos piezas; una es un rodamiento 6701 ZZ y la otra es una esfera de caucho butílico. En cada pieza se muestra una figura tres vistas: una lateral, otra frontal y una última isométrica; además de su peso calculado por el programa.

Base. Figura 3.7, la base de la mano robótica con un peso de 135.97 g.

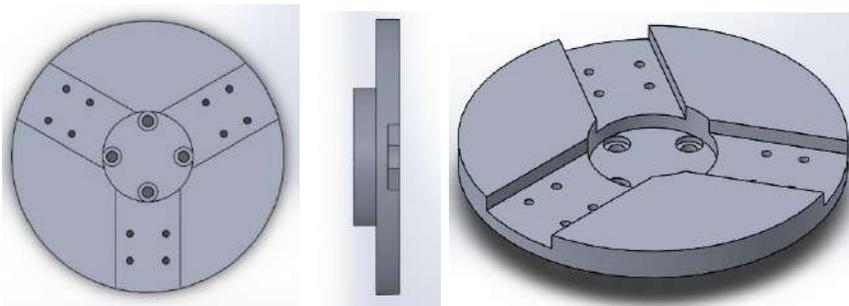


Figura 3.7 Base.

Pieza de ensamble 1.1. Figura 3.8, esta es una base que ayuda a una sujeción de la parte inferior de servomotor número uno y su peso es 8.04 g.

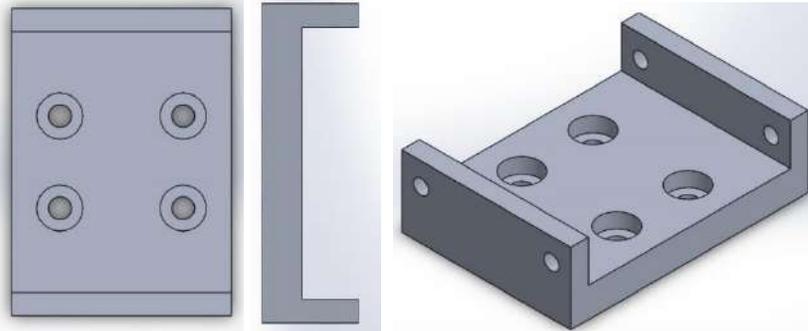


Figura 3.8 Base de servomotor número uno.

Pieza de ensamble 1.2. Figura 3.9, esta pieza sustituye a la pieza de ensamble 2 y un servomotor XM430 en uno de los tres dedos, debido a que la mano robótica tiene dos dedos con tres servomotores y uno con dos servomotores, y esta pieza mantiene las mismas distancias que tendrían con el servomotor XM430, el peso de esta pieza es de 44.71 g.

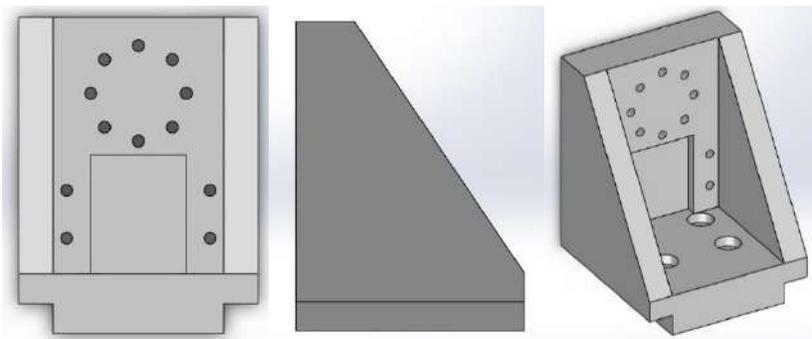


Figura 3.9 Pieza sustituta de un servomotor XM430.

Pieza de ensamble 2. Figura 3.10, esta placa permite al servomotor dos estar unido con el servomotor uno, su peso es 5.86 g.

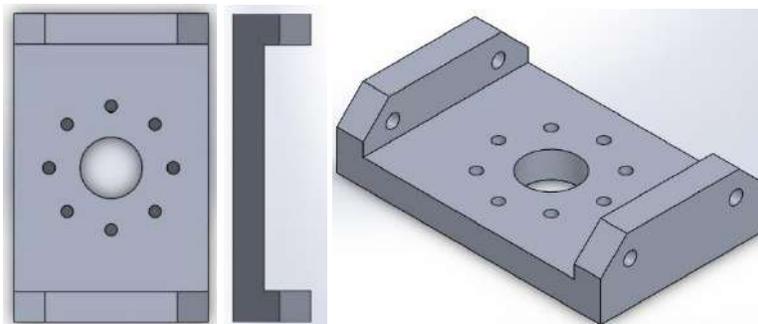


Figura 3.10 Placa de ensamble del servomotor uno y dos.

Pieza de ensamble 3. Figura 3.11, esta placa mantiene unido al servomotor dos y tres del lado izquierdo, donde se encuentra la flecha del servomotor dos, su peso es 8.04 g.

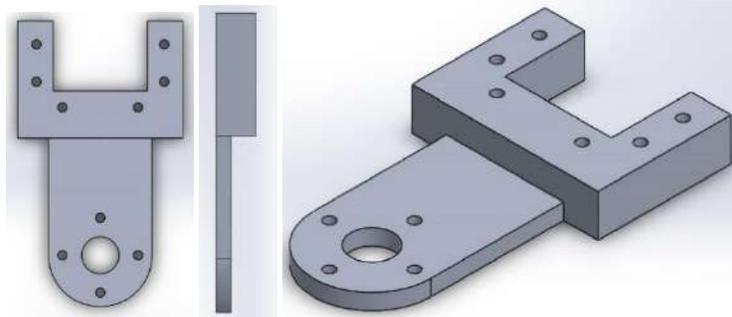


Figura 3.11 Placa de ensamble del lado izquierdo de servomotor dos y tres.

Pieza de ensamble 4. Figura 3.12, esta placa mantiene unido los servomotores dos y tres por el lado derecho, su peso es de 6.99 g.

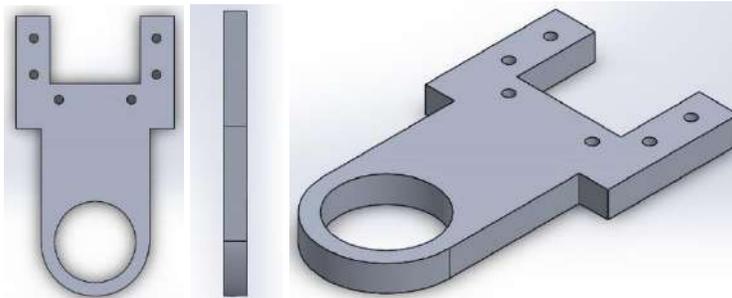


Figura 3.12 Placa de ensamble del lado derecho de servomotor dos y tres.

Pieza de ensamble 5. Figura 3.13, esta placa une el lado izquierdo del servomotor número tres con la base para el órgano terminal, su peso es de 5.11 g.

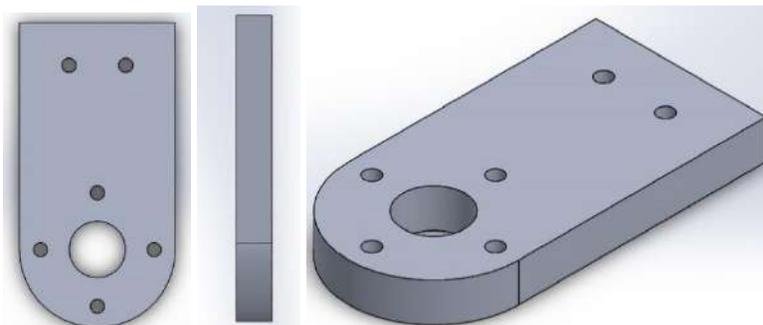


Figura 3.13 Placa de ensamble del lado izquierdo de servomotor tres con la base del órgano terminal.

Pieza de ensamble 6. Figura 3.14, esta placa une el lado derecho del servomotor número tres con la base para el órgano terminal, su peso es de 4.31 g.

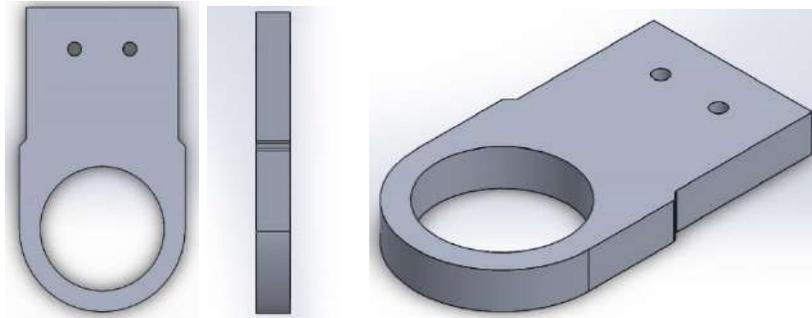


Figura 3.14 Placa de ensamblaje del lado derecho de servomotor tres con la base del órgano terminal.

Rodamiento 6701 ZZ. Este tipo de rodamiento se encuentra ensamblado con las piezas de ensamblaje del lado derecho 1 y 2 para permitirles rotar sin fricción, su peso es de 3g y la Figura 3.15 muestra a este rodamiento en el programa de diseño.



Figura 3.15 Rodamiento 6701 ZZ.

Base del rodamiento. Esta pieza se utiliza para ajustar el rodamiento 6701 ZZ con el lado posterior del motor AX-A12, su peso es de 0.19 g y esta pieza se muestra en la Figura 3.16.

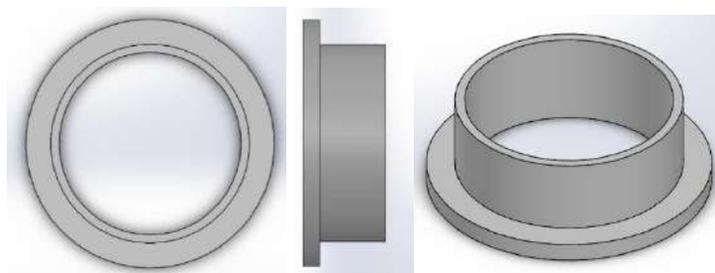


Figura 3.16 Base del rodamiento.

Tapa del rodamiento. Esta pieza permite evita que las piezas que se ensamblan con el rodamiento se desacoplen unas de otras, la Figura 3.17 muestra esta pieza, y su peso es de 1.19 g,

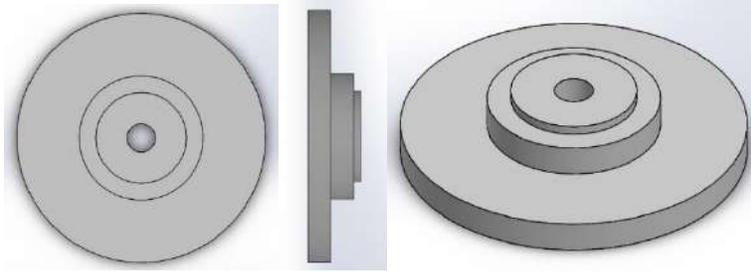


Figura 3.17 Tapa del rodamiento.

Pieza de ensamble 7. Figura 3.18, esta placa sirve como soporte para la base de la esfera, su peso es de 3.98 g.

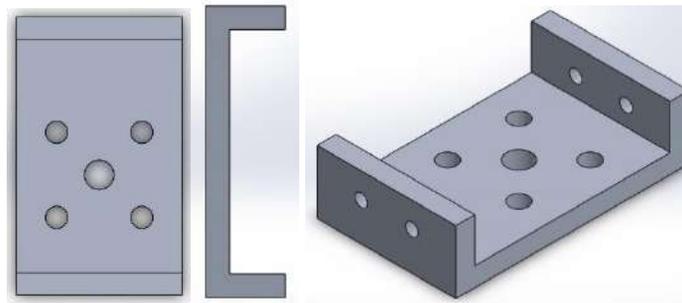


Figura 3.18 Base de órgano terminal.

Base de la esfera. Figura 3.19, esta pieza ayuda a sostener la parte final del dedo, la cual es una esfera, su peso 3.26 g.

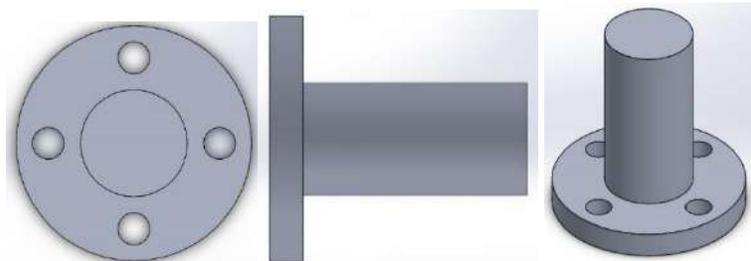


Figura 3.19 Extensión de la base del órgano terminal.

Esfera. Figura 3.20, es la parte final del dedo, y es la que hace contacto con el objeto para poder sostenerlo, su peso es 15.06 g, aunque el material de esta pieza es el caucho butílico o similar.

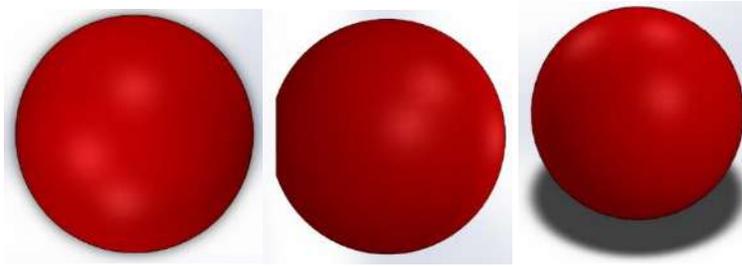


Figura 3.20 Esfera del órgano terminal.

Todas las piezas ensambladas dan forma a un dedo como el que se muestra en la Figura 3.4, pero en la Figura 3.21 se observa la mano robótica con sus tres dedos ensamblados. La imagen denota que cada pieza es utilizada tres veces, con excepción de la base que es una sola pieza; el rodamiento, su base y tapa las cuales son seis piezas; por lo que el cálculo de todo el peso de la mano robótica, es la masa de la base, más la suma de la masa de cada servomotor y las piezas, todo esto multiplicado por tres, y por último se le suma de las seis piezas de rodamiento, su base y tapa; con todo lo anteriormente calculado se obtiene una masa estimada sin considerar tornillos de 917.95 g, la cual es menor a la masa máxima permitida para el peso de la mano robótica, la cual es de 1.5 kg máximo.

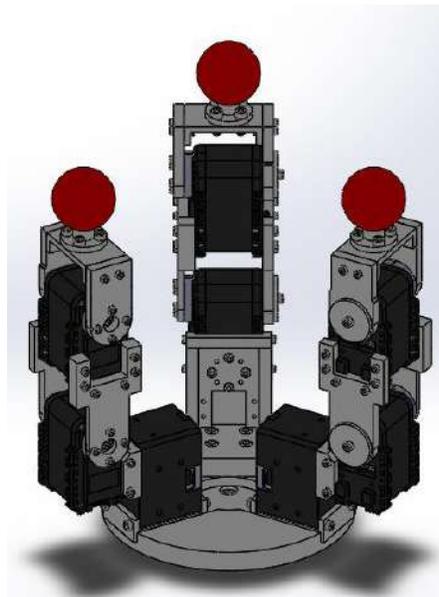


Figura 3.21 Diseño en CAD de la mano robótica.

3.3 Cálculo de las cadenas cinemáticas

La descripción del movimiento para la mano robótica se realizará mediante el cálculo de desplazamiento inverso de la cadena cinemática de un dedo de la, esto hace que solamente se describa el movimiento del objeto como una entrada para las variables de la cadena cinemática, las cuales son las articulaciones de un dedo robótico. La Figura 3.22 muestra un diseño en CAD de la mano robótica sosteniendo un cuerpo esférico, el cual es el objeto a manipular, además de que la sucesión de puntos amarillos, que son los marcos de referencia, da forma a una cadena que abarca desde la base de la mano robótica hasta el centro del objeto, y por ende dando la arquitectura cinemática de un dedo de la mano robótica.

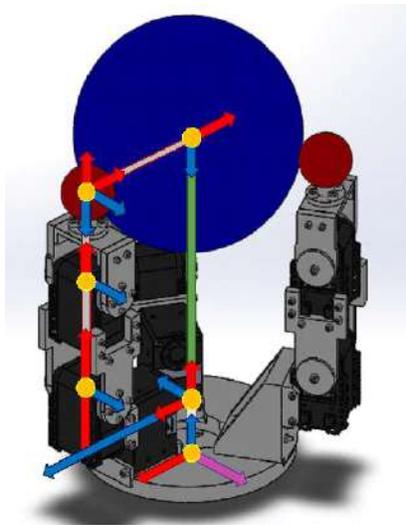


Figura 3.22 Cadena cinemática de un dedo de la mano robótica.

La Figura 3.22 solamente muestra la cadena de un solo dedo, esto se debe a que todos los dedos tienen la misma cadena cinemática, pero la cadena de un dedo tiene un ángulo de 120° respecto uno del otro; también se muestran los marcos de referencia, los cuales están representados por los puntos amarillos, las flechas azules y rojas representan los ejes Z y X respectivamente de cada uno de los marcos de referencia; la flecha verde que se observa representa la posición del centro del objeto con respecto al primer marco de referencia, el cual está en la base de la mano robótica, además de ser el marco de referencia para todos los dedos, la Figura 3.23 esboza la cadena cinemática de un dedo la mano robótica con más claridad.

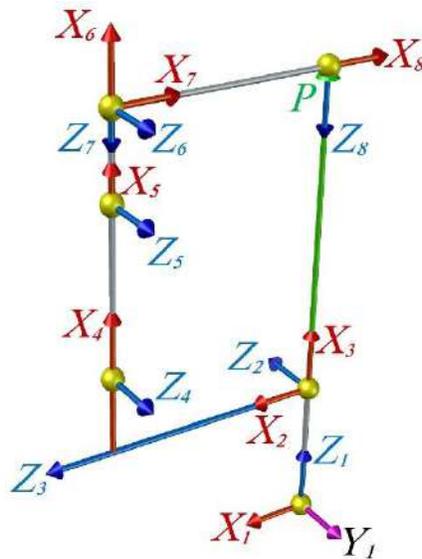


Figura 3.23 Marco de coordenadas de un dedo de la mano robótica.

Con la Figura 3.23 se identifica más fácilmente los parámetros de Denavit-Hartenberg para la cadena cinemática, tanto para las variables como los parámetros fijos, en este caso las distancias entre los marcos de referencia son fijas pero la orientación de un marco con respecto al otro varía, siendo los marcos cuatro, cinco y seis el lugar donde se encuentran en los ejes de los actuadores, aunque también se le puede agregar el radio del objeto, porque este objeto es una esfera cualquiera y solamente se tiene que conocer su radio para determinar la distancia entre los marcos de referencia que la componen. La Tabla 3-1 contiene los parámetros de un dedo de la mano robótica.

Tabla 3-1 Tabla de los parámetros de la notación Denavit-Hartenberg para un dedo de la mano robótica.

i	a_i	b_i	α_i	θ_i	$\lambda_i = \cos(\alpha_i)$	$\mu_i = \sin(\alpha_i)$
1	0	b_1	90°	θ_1	0	1
2	0	0	90°	90°	0	1
3	a_3	b_3	90°	θ_3^*	0	1
4	a_4	0	0°	θ_4^*	1	0
5	a_5	0	0°	θ_5^*	1	0
6	0	0	90°	-90°	0	1
7	a_7	0	0°	0°	1	0

En la tabla de arriba se representan los parámetros que se encuentran fijos con una letra enumerada, mientras las variables mantienen una letra enumerada con un asterisco. La letra a_i y b_i corresponden a las distancias entre los ejes Z y X respectivamente, el símbolo α_i representa un ángulo con respecto al eje X, y θ_i un ángulo con respecto al eje Z.

Las variables que aparecen en la Tabla 3-1 con un asterisco son las variables a encontrar, las cuales son θ_3 , θ_4 y θ_5 , y para lograr encontrar estos valores se tiene que recurrir al problema de desplazamiento inverso (IDP, *Inverse Displacement Problem*), el cual permite saber el comportamiento angular de dichas variables por medio de la posición del órgano terminal, en este caso el centro del objeto. La ecuación (3.5) es la representación de una cadena cinemática para un dedo, por lo que se tiene que despejar los ángulos θ_3 , θ_4 y θ_5 de dicha ecuación, y siendo esto a su vez la solución por medio del IDP.

$$[\mathbf{P}]_1 = [\mathbf{a}_1]_1 + [\mathbf{a}_2]_1 + [\mathbf{a}_3]_1 + [\mathbf{a}_4]_1 + [\mathbf{a}_5]_1 + [\mathbf{a}_6]_1 + [\mathbf{a}_7]_1 \quad (3.5)$$

Después de escribir la ecuación de un dedo es necesario expandir la ecuación (3.5), para poder encontrar la matriz de orientación de los marcos de referencia de cada uno de los ángulos buscados.

$$\begin{aligned} \mathbf{P} &= \mathbf{a}_1 + \mathbf{Q}_1 \mathbf{a}_2 + \mathbf{Q}_1 \mathbf{Q}_2 \mathbf{a}_3 + \mathbf{Q}_1 \mathbf{Q}_2 \mathbf{Q}_3 \mathbf{a}_4 + \mathbf{Q}_1 \mathbf{Q}_2 \mathbf{Q}_3 \mathbf{Q}_4 \mathbf{a}_5 + \mathbf{Q}_1 \mathbf{Q}_2 \mathbf{Q}_3 \mathbf{Q}_4 \mathbf{Q}_5 \mathbf{a}_6 \\ &\quad + \mathbf{Q}_1 \mathbf{Q}_2 \mathbf{Q}_3 \mathbf{Q}_4 \mathbf{Q}_5 \mathbf{Q}_6 \mathbf{a}_7 \\ \mathbf{P} &= \mathbf{a}_1 + \mathbf{Q}_1 \mathbf{a}_2 + \mathbf{Q}_1 \mathbf{Q}_2 \mathbf{a}_3 + \mathbf{Q}_1 \mathbf{Q}_2 \mathbf{Q}_3 \mathbf{a}_4 + \mathbf{Q}_1 \mathbf{Q}_2 \mathbf{Q}_3 \mathbf{Q}_4 \mathbf{a}_5 + \mathbf{Q}_1 \mathbf{Q}_2 \mathbf{Q}_3 \mathbf{Q}_4 \mathbf{Q}_5 \mathbf{a}_6 \\ &\quad + \mathbf{Q}_1 \mathbf{Q}_2 \mathbf{Q}_3 \mathbf{Q}_4 \mathbf{Q}_5 \mathbf{Q}_6 \mathbf{Q}_7 \mathbf{b}_7 \end{aligned}$$

Las distancias de los marcos de referencia dos y seis con respecto al a su marco anterior no existe, por lo que sus vectores son igual a cero, además que los movimientos angulares del último marco se pueden reducir a una matriz de orientación \mathbf{Q} :

$$\mathbf{Q}_1 \mathbf{Q}_2 \mathbf{Q}_3 \mathbf{Q}_4 \mathbf{Q}_5 \mathbf{Q}_6 \mathbf{Q}_7 = \mathbf{Q}; \quad \mathbf{a}_2 = \mathbf{0}; \quad \mathbf{a}_6 = \mathbf{0};$$

La ecuación (3.5) se simplifica a la forma de la ecuación (3.6), y esta última se divide en dos ecuaciones (3.7) y (3.8):

$$\mathbf{P} = \mathbf{a}_1 + \mathbf{Q}_1 \mathbf{Q}_2 \mathbf{a}_3 + \mathbf{Q}_1 \mathbf{Q}_2 \mathbf{Q}_3 \mathbf{a}_4 + \mathbf{Q}_1 \mathbf{Q}_2 \mathbf{Q}_3 \mathbf{Q}_4 \mathbf{a}_5 + \mathbf{Q} \mathbf{b}_7$$

$$\mathbf{P} - \mathbf{a}_1 - \mathbf{Q} \mathbf{b}_7 = \mathbf{Q}_1 \mathbf{Q}_2 \mathbf{a}_3 + \mathbf{Q}_1 \mathbf{Q}_2 \mathbf{Q}_3 \mathbf{a}_4 + \mathbf{Q}_1 \mathbf{Q}_2 \mathbf{Q}_3 \mathbf{Q}_4 \mathbf{a}_5 \quad (3.6)$$

$$\mathbf{C} = \mathbf{P} - \mathbf{a}_1 - \mathbf{Q} \mathbf{b}_7 \quad (3.7)$$

$$\mathbf{C} = \mathbf{Q}_1 \mathbf{Q}_2 \mathbf{a}_3 + \mathbf{Q}_1 \mathbf{Q}_2 \mathbf{Q}_3 \mathbf{a}_4 + \mathbf{Q}_1 \mathbf{Q}_2 \mathbf{Q}_3 \mathbf{Q}_4 \mathbf{a}_5 \quad (3.8)$$

En la ecuación (3.7) se encuentran los parámetros conocidos, y esta última permite describir la orientación, en cambio la ecuación (3.8) contiene las variables, por lo que se despeja esta última conteniendo lo valores de la ecuación (3.7) en \mathbf{C} :

$$\begin{aligned} Q_1^T C &= Q_2 a_3 + Q_2 Q_3 a_4 + Q_2 Q_3 Q_4 a_5 \\ Q_2^T Q_1^T C &= Q_3 b_3 + Q_3 a_4 + Q_3 Q_4 a_5 \end{aligned}$$

$$Q_3^T Q_2^T Q_1^T C = b_3 + Q_4(b_4 + a_5) \quad (3.9)$$

La ecuación (3.9) se utiliza para simplemente realizar las operaciones de suma y multiplicación entre las matrices y vectores. En la ecuación (3.10) se muestra las operaciones de la ecuación (3.7):

$$P = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix}; \quad a_1 = \begin{bmatrix} 0 \\ 0 \\ b_1 \end{bmatrix}; \quad b_7 = \begin{bmatrix} a_7 \\ 0 \\ 0 \end{bmatrix}; \quad Q = \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{bmatrix};$$

$$C = P - a_1 - Q b_7 = \begin{bmatrix} P_x - a_7 q_{11} \\ P_y - a_7 q_{21} \\ P_z - b_1 - a_7 q_{31} \end{bmatrix} \quad (3.10)$$

La ecuación (3.11) muestra las operaciones del lado derecho de la ecuación (3.9), además de que las funciones seno y coseno se acortan como $\sin\theta_i = s\theta_i$ y $\cos\theta_i = c\theta_i$, para que esta forma las funciones seno y coseno sean sencillas de escribir en las ecuaciones:

$$b_3 = \begin{bmatrix} a_3 \\ b_3 \\ 0 \end{bmatrix}; \quad b_4 = \begin{bmatrix} a_4 \\ 0 \\ 0 \end{bmatrix}; \quad a_5 = \begin{bmatrix} a_5 c\theta_5 \\ a_5 s\theta_5 \\ 0 \end{bmatrix}; \quad Q_4 = \begin{bmatrix} c\theta_4 & -s\theta_4 & 0 \\ s\theta_4 & c\theta_4 & 0 \\ 0 & 0 & 1 \end{bmatrix};$$

$$\begin{aligned} b_3 + Q_4(b_4 + a_5) &= \begin{bmatrix} a_3 \\ b_3 \\ 0 \end{bmatrix} + \begin{bmatrix} c\theta_4 & -s\theta_4 & 0 \\ s\theta_4 & c\theta_4 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_4 + a_5 c\theta_5 \\ a_5 s\theta_5 \\ 0 \end{bmatrix} \\ b_3 + Q_4(b_4 + a_5) &= \begin{bmatrix} a_3 \\ b_3 \\ 0 \end{bmatrix} + \begin{bmatrix} (a_4 + a_5 c\theta_5) c\theta_4 - (a_5 s\theta_5) s\theta_4 \\ (a_4 + a_5 c\theta_5) s\theta_4 + (a_5 s\theta_5) c\theta_4 \\ 0 \end{bmatrix} \end{aligned}$$

$$b_3 + Q_4(b_4 + a_5) = \begin{bmatrix} a_3 + (a_4 + a_5 c\theta_5) c\theta_4 - (a_5 s\theta_5) s\theta_4 \\ b_3 + (a_4 + a_5 c\theta_5) s\theta_4 + (a_5 s\theta_5) c\theta_4 \\ 0 \end{bmatrix} \quad (3.11)$$

en cambio, para el lado izquierdo de la ecuación (3.9) el resultado de las operaciones se muestra en la ecuación (3.12):

$$Q_1^T = \begin{bmatrix} c\theta_1 & s\theta_1 & 0 \\ 0 & 0 & 1 \\ s\theta_1 & -c\theta_1 & 0 \end{bmatrix}; \quad Q_2^T = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}; \quad Q_3^T = \begin{bmatrix} c\theta_3 & s\theta_3 & 0 \\ 0 & 0 & 1 \\ s\theta_3 & -c\theta_3 & 0 \end{bmatrix}; \quad C = \begin{bmatrix} c_x \\ c_y \\ c_z \end{bmatrix};$$

$$\begin{aligned}
Q_3^T Q_2^T Q_1^T C &= \begin{bmatrix} c\theta_3 & s\theta_3 & 0 \\ 0 & 0 & 1 \\ s\theta_3 & -c\theta_3 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} c\theta_1 & s\theta_1 & 0 \\ 0 & 0 & 1 \\ s\theta_1 & -c\theta_1 & 0 \end{bmatrix} \begin{bmatrix} c_x \\ c_y \\ c_z \end{bmatrix} \\
Q_3^T Q_2^T Q_1^T C &= \begin{bmatrix} c\theta_3 & s\theta_3 & 0 \\ 0 & 0 & 1 \\ s\theta_3 & -c\theta_3 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} c_x c\theta_1 + c_y s\theta_1 \\ c_z \\ c_x s\theta_1 - c_y c\theta_1 \end{bmatrix} \\
Q_3^T Q_2^T Q_1^T C &= \begin{bmatrix} c\theta_3 & s\theta_3 & 0 \\ 0 & 0 & 1 \\ s\theta_3 & -c\theta_3 & 0 \end{bmatrix} \begin{bmatrix} c_z \\ c_x s\theta_1 - c_y c\theta_1 \\ c_x c\theta_1 + c_y s\theta_1 \end{bmatrix} \\
Q_3^T Q_2^T Q_1^T C &= \begin{bmatrix} (c_x s\theta_1 - c_y c\theta_1) s\theta_3 + c_z c\theta_3 \\ c_x c\theta_1 + c_y s\theta_1 \\ -(c_x s\theta_1 - c_y c\theta_1) c\theta_3 + c_z s\theta_3 \end{bmatrix} \tag{3.12}
\end{aligned}$$

Con las todas las operaciones realizadas para ambos lados de la ecuación (3.9), estas operaciones permiten una simplificación de toda la ecuación de desplazamiento inverso, y esto a su vez hace que sea fácilmente visualizar el despeje las variables angulares θ_3 , θ_4 y θ_5 en la ecuación (3.13).

$$\begin{aligned}
Q_3^T Q_2^T Q_1^T C &= b_3 + Q_4(b_4 + a_5) \\
\begin{bmatrix} (c_x s\theta_1 - c_y c\theta_1) s\theta_3 + c_z c\theta_3 \\ c_x c\theta_1 + c_y s\theta_1 \\ -(c_x s\theta_1 - c_y c\theta_1) c\theta_3 + c_z s\theta_3 \end{bmatrix} &= \begin{bmatrix} a_3 + (a_4 + a_5 c\theta_5) c\theta_4 - (a_5 s\theta_5) s\theta_4 \\ b_3 + (a_4 + a_5 c\theta_5) s\theta_4 + (a_5 s\theta_5) c\theta_4 \\ 0 \end{bmatrix} \tag{3.13}
\end{aligned}$$

La solución para el ángulo θ_3 se puede realizar mediante el despeje en la ecuación (3.13), la solución para este ángulo se muestra en la ecuación (3.14) y su despeje es el siguiente:

$$\begin{aligned}
-(c_x s\theta_1 - c_y c\theta_1) c\theta_3 + c_z s\theta_3 &= 0 \\
c_z s\theta_3 &= (c_x s\theta_1 - c_y c\theta_1) c\theta_3 \\
\frac{s\theta_3}{c\theta_3} &= \frac{c_x s\theta_1 - c_y c\theta_1}{c_z} \\
\tan \theta_3 &= \frac{c_x s\theta_1 - c_y c\theta_1}{c_z} \\
\theta_3 &= \tan^{-1} \left(\frac{c_x s\theta_1 - c_y c\theta_1}{c_z} \right) \tag{3.14}
\end{aligned}$$

en cambio, las otras dos variables θ_4 y θ_5 , se pueden despejar de la ecuación (3.13) como un sistema de dos ecuaciones. La primera solución de una de estas dos variables será para θ_5 , cuyas soluciones se muestra en las dos ecuaciones (3.18) y (3.19), ambas soluciones parten del sistema de ecuaciones que se encuentra aquí abajo:

$$\begin{aligned}(c_x s\theta_1 - c_y c\theta_1)s\theta_3 + c_z c\theta_3 &= a_3 + (a_4 + a_5 c\theta_5)c\theta_4 - (a_5 s\theta_5)s\theta_4 \\ c_x c\theta_1 + c_y s\theta_1 &= b_3 + (a_4 + a_5 c\theta_5)s\theta_4 + (a_5 s\theta_5)c\theta_4\end{aligned}$$

$$\begin{aligned}(c_x s\theta_1 - c_y c\theta_1)s\theta_3 + c_z c\theta_3 - a_3 &= (a_4 + a_5 c\theta_5)c\theta_4 - (a_5 s\theta_5)s\theta_4 \\ c_x c\theta_1 + c_y s\theta_1 - b_3 &= (a_4 + a_5 c\theta_5)s\theta_4 + (a_5 s\theta_5)c\theta_4\end{aligned}$$

y simplificando el lado izquierdo del sistema de ecuaciones se obtienen las siguientes dos ecuaciones:

$$k_1 = (c_x s\theta_1 - c_y c\theta_1)s\theta_3 + c_z c\theta_3 - a_3 \quad (3.15)$$

$$k_2 = c_x c\theta_1 + c_y s\theta_1 - b_3 \quad (3.16)$$

la primera solución de θ_5 se obtiene despejando de la siguiente forma:

$$k_1 = (a_4 + a_5 c\theta_5)c\theta_4 - (a_5 s\theta_5)s\theta_4$$

$$k_2 = (a_4 + a_5 c\theta_5)s\theta_4 + (a_5 s\theta_5)c\theta_4$$

$$k_1^2 = (a_4 + a_5 c\theta_5)^2 c^2\theta_4 - 2(a_4 + a_5 c\theta_5)(a_5 s\theta_5)c\theta_4 s\theta_4 + (a_5 s\theta_5)^2 s^2\theta_4$$

$$k_2^2 = (a_4 + a_5 c\theta_5)^2 s^2\theta_4 + 2(a_4 + a_5 c\theta_5)(a_5 s\theta_5)c\theta_4 s\theta_4 + (a_5 s\theta_5)^2 c^2\theta_4$$

$$k_1^2 + k_2^2 = (a_4 + a_5 c\theta_5)^2 + (a_5 s\theta_5)^2 = a_4^2 + 2a_4 a_5 c\theta_5 + a_5^2 c^2\theta_5 + a_5^2 s^2\theta_5$$

$$k_1^2 + k_2^2 - a_4^2 - a_5^2 = 2a_4 a_5 c\theta_5$$

y se simplifica el lado izquierdo como:

$$k_3 = k_1^2 + k_2^2 - a_4^2 - a_5^2 \quad (3.17)$$

con esto último la primera solución para θ_5 es la siguiente ecuación:

$$\theta_5 = \cos^{-1}\left(\frac{k_3}{2a_4 a_5}\right) \quad (3.18)$$

la segunda solución de θ_5 se obtiene despejando de la siguiente forma:

$$\begin{aligned} \mathbf{k}_1 &= (\mathbf{a}_4 + \mathbf{a}_5 \mathbf{c}\theta_5) \mathbf{c}\theta_4 - (\mathbf{a}_5 \mathbf{s}\theta_5) \mathbf{s}\theta_4 \\ \mathbf{k}_2 &= (\mathbf{a}_4 + \mathbf{a}_5 \mathbf{c}\theta_5) \mathbf{s}\theta_4 + (\mathbf{a}_5 \mathbf{s}\theta_5) \mathbf{c}\theta_4 \end{aligned}$$

$$\begin{aligned} \mathbf{k}_1 &= \mathbf{a}_4 \mathbf{c}\theta_4 + \mathbf{a}_5 \mathbf{c}\theta_4 \mathbf{c}\theta_5 - \mathbf{a}_5 \mathbf{s}\theta_4 \mathbf{s}\theta_5 = \mathbf{a}_4 \mathbf{c}\theta_4 + \mathbf{a}_5 \mathbf{c}(\theta_4 + \theta_5) \\ \mathbf{k}_2 &= \mathbf{a}_4 \mathbf{s}\theta_4 + \mathbf{a}_5 \mathbf{c}\theta_4 \mathbf{s}\theta_5 + \mathbf{a}_5 \mathbf{s}\theta_4 \mathbf{c}\theta_5 = \mathbf{a}_4 \mathbf{s}\theta_4 + \mathbf{a}_5 \mathbf{s}(\theta_4 + \theta_5) \end{aligned}$$

$$\begin{aligned} \mathbf{k}_1^2 &= \mathbf{a}_4^2 \mathbf{c}^2 \theta_4 + 2\mathbf{a}_4 \mathbf{a}_5 \mathbf{c}\theta_4 \mathbf{c}(\theta_4 + \theta_5) + \mathbf{a}_5^2 \mathbf{c}^2(\theta_4 + \theta_5) \\ \mathbf{k}_2^2 &= \mathbf{a}_4^2 \mathbf{s}^2 \theta_4 + 2\mathbf{a}_4 \mathbf{a}_5 \mathbf{s}\theta_4 \mathbf{s}(\theta_4 + \theta_5) + \mathbf{a}_5^2 \mathbf{s}^2(\theta_4 + \theta_5) \end{aligned}$$

$$\begin{aligned} \mathbf{k}_1^2 + \mathbf{k}_2^2 &= \mathbf{a}_4^2 + \mathbf{a}_5^2 + 2\mathbf{a}_4 \mathbf{a}_5 [\mathbf{c}\theta_4 \mathbf{c}(\theta_4 + \theta_5) + \mathbf{s}\theta_4 \mathbf{s}(\theta_4 + \theta_5)] \\ \mathbf{k}_1^2 + \mathbf{k}_2^2 - \mathbf{a}_4^2 - \mathbf{a}_5^2 &= 2\mathbf{a}_4 \mathbf{a}_5 \cos[\theta_4 - (\theta_4 + \theta_5)] \end{aligned}$$

$$\mathbf{k}_3 = 2\mathbf{a}_4 \mathbf{a}_5 \cos(-\theta_5) ;$$

al igual que en la primera solución de θ_5 se utiliza la misma simplificación, que es la ecuación (3.17), por lo tanto, la segunda solución de θ_5 se muestra en la siguiente ecuación:

$$\theta_5 = -\cos^{-1}\left(\frac{\mathbf{k}_3}{2\mathbf{a}_4 \mathbf{a}_5}\right) \quad (3.19)$$

En cuanto a la variable θ_4 , esta también posee dos soluciones las cuales se muestran en las ecuaciones (3.20) y (3.21), e igualmente proviene del sistema de dos ecuaciones:

$$\begin{aligned} \mathbf{k}_1 &= (\mathbf{a}_4 + \mathbf{a}_5 \mathbf{c}\theta_5) \mathbf{c}\theta_4 - (\mathbf{a}_5 \mathbf{s}\theta_5) \mathbf{s}\theta_4 \\ \mathbf{k}_2 &= (\mathbf{a}_4 + \mathbf{a}_5 \mathbf{c}\theta_5) \mathbf{s}\theta_4 + (\mathbf{a}_5 \mathbf{s}\theta_5) \mathbf{c}\theta_4 \end{aligned}$$

La primera forma de obtener una solución para θ_4 es descrita de la siguiente manera:

$$\begin{aligned} \mathbf{k}_1 &= (\mathbf{a}_4 + \mathbf{a}_5 \mathbf{c}\theta_5) \mathbf{c}\theta_4 - (\mathbf{a}_5 \mathbf{s}\theta_5) \mathbf{s}\theta_4 \\ \mathbf{k}_1 &= \sqrt{(\mathbf{a}_4 + \mathbf{a}_5 \mathbf{c}\theta_5)^2 + (\mathbf{a}_5 \mathbf{s}\theta_5)^2} \left(\frac{(\mathbf{a}_4 + \mathbf{a}_5 \mathbf{c}\theta_5)}{\sqrt{(\mathbf{a}_4 + \mathbf{a}_5 \mathbf{c}\theta_5)^2 + (\mathbf{a}_5 \mathbf{s}\theta_5)^2}} \mathbf{c}\theta_4 \right. \\ &\quad \left. + \frac{(\mathbf{a}_5 \mathbf{s}\theta_5)}{\sqrt{(\mathbf{a}_4 + \mathbf{a}_5 \mathbf{c}\theta_5)^2 + (\mathbf{a}_5 \mathbf{s}\theta_5)^2}} \mathbf{s}\theta_4 \right) \\ \gamma &= \tan\left(\frac{\mathbf{a}_5 \mathbf{s}\theta_5}{\mathbf{a}_4 + \mathbf{a}_5 \mathbf{c}\theta_5}\right) \end{aligned}$$

$$\begin{aligned} \frac{\mathbf{k}_1}{\sqrt{(\mathbf{a}_4 + \mathbf{a}_5 \mathbf{c}\theta_5)^2 + (\mathbf{a}_5 \mathbf{s}\theta_5)^2}} &= \cos \gamma \cos \theta_4 - \sin \gamma \sin \theta_4 \\ \frac{\mathbf{k}_1}{\sqrt{(\mathbf{a}_4 + \mathbf{a}_5 \mathbf{c}\theta_5)^2 + (\mathbf{a}_5 \mathbf{s}\theta_5)^2}} &= \cos(\gamma + \theta_4) \end{aligned}$$

$$\theta_4 = \cos^{-1} \left(\frac{k_1}{\sqrt{(a_4 + a_5 c \theta_5)^2 + (a_5 s \theta_5)^2}} \right) - \tan^{-1} \left(\frac{a_5 s \theta_5}{a_4 + a_5 c \theta_5} \right) \quad (3.20)$$

La segunda forma de obtener una solución para θ_4 , se describe de la siguiente forma:

$$k_2 = (a_4 + a_5 c \theta_5) s \theta_4 + (a_5 s \theta_5) c \theta_4$$

$$k_2 = \sqrt{(a_4 + a_5 c \theta_5)^2 + (a_5 s \theta_5)^2} \left(\frac{(a_4 + a_5 c \theta_5)}{\sqrt{(a_4 + a_5 c \theta_5)^2 + (a_5 s \theta_5)^2}} s \theta_4 + \frac{(a_5 s \theta_5)}{\sqrt{(a_4 + a_5 c \theta_5)^2 + (a_5 s \theta_5)^2}} c \theta_4 \right)$$

$$\gamma = \tan^{-1} \left(\frac{a_5 s \theta_5}{a_4 + a_5 c \theta_5} \right)$$

$$\frac{k_2}{\sqrt{(a_4 + a_5 c \theta_5)^2 + (a_5 s \theta_5)^2}} = \cos \gamma \sin \theta_4 + \sin \gamma \cos \theta_4$$

$$\frac{k_2}{\sqrt{(a_4 + a_5 c \theta_5)^2 + (a_5 s \theta_5)^2}} = \sin(\gamma + \theta_4)$$

$$\theta_4 = \sin^{-1} \left(\frac{k_2}{\sqrt{(a_4 + a_5 c \theta_5)^2 + (a_5 s \theta_5)^2}} \right) - \tan^{-1} \left(\frac{a_5 s \theta_5}{a_4 + a_5 c \theta_5} \right) \quad (3.21)$$

En cuanto a los valores de la matriz de orientación Q , se pueden obtener por medio de las matrices de orientación de cada marco de referencia, esto se debe a que esta matriz de orientación Q es el resultado de productos entre matrices de orientación de cada marco de referencia, y el resultado de estas operaciones se muestra en la ecuación (3.22).

$$Q = Q_1 Q_2 Q_3 Q_4 Q_5 Q_6 Q_7 = \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{bmatrix}$$

$$Q = \begin{bmatrix} s \theta_1 s \theta_3 s(\theta_4 + \theta_5) - c \theta_1 c(\theta_4 + \theta_5) & -s \theta_1 c \theta_3 & -s \theta_1 c \theta_3 c(\theta_4 + \theta_5) - c \theta_1 s(\theta_4 + \theta_5) \\ -c \theta_1 s \theta_3 s(\theta_4 + \theta_5) - s \theta_1 c(\theta_4 + \theta_5) & c \theta_1 c \theta_3 & c \theta_1 c \theta_3 c(\theta_4 + \theta_5) - s \theta_1 s(\theta_4 + \theta_5) \\ c \theta_1 s(\theta_4 + \theta_5) & s \theta_3 & -c \theta_1 c(\theta_4 + \theta_5) \end{bmatrix} \quad (3.22)$$

El resultado mostrado en la ecuación (3.22), se debe a que en esta ecuación los ángulos Alpha y Theta poseen valores definidos, con excepción de los θ_3 , θ_4 y θ_5 mostrados en la Tabla 3-2, además todos valores de son Alpha fijos y la mitad de los ángulos Theta una propuesta para una posición determinada.

Tabla 3-2 Tabla de parámetros de la notación Denavit-Hartenberg para un dedo de la mano robótica con valores constantes.

i	$a_i(\text{cm})$	$b_i(\text{cm})$	α_i	θ_i
1	0	4.525	90°	$0, \pm 120$
2	0	0	90°	90°
3	2.25	7.9	90°	$\theta_3^* + 0^\circ$
4	6.75	0	0°	$\theta_4^* + 0^\circ$
5	5.5	0	0°	$\theta_5^* + 0^\circ$
6	0	0	90°	$\theta_6 - 90^\circ$
7	$1.5 + r_2$	0	0°	$\theta_7 + 0^\circ$

En la Tabla 3-2, los valores de las distancias a_i y b_i se muestran en segunda y tercera columna respectivamente, estos valores se encuentran en centímetros, además la distancia a_7 está conformada por dos valores, debido que el primer valor es el radio del a esfera del órgano terminal del dedo y el segundo radio es el del objeto, el cual variara según el tamaño de este; en cambio, las últimas dos columnas son los valores de los ángulos Alpha y Theta, además de que en la columna de Theta se puede observar que a los valores de cada uno de estos se le suma un ángulo, esto es porque dichos valores son la posición inicial de θ_i con excepción de θ_1 y θ_2 , los cuales poseen valores fijos: uno valor para θ_2 , el cual es 0° ; y tres valores para θ_1 , los cuales son 0° , $+120^\circ$ y -120° para cada uno de los tres dedos. Los tres valores θ_1 son debido a que la orientación con respecto al marco de referencia en la que se encuentra en la base, es el mismo para los tres dedos y cada uno de ellos se encuentra con un ángulo de 120° en relación uno con el otro. Los valores de θ_6 y θ_7 también pueden cambiar, pero estos valores deben ser colocados dependiendo la posición en la que se encuentre el objeto.

3.4 Programa de simulación

Con la obtención de las variables θ_3 , θ_4 y θ_5 , por medio del problema de desplazamiento invertido (IDP, *Inverse Displacement Problem*), se puede realizar una simulación de todas las cadenas cinemáticas que conforman la mano robótica manipulando un objeto. Para llevar esto a cabo, se ha de usar un programa en donde se visualice el comportamiento de la mano robótica, ubicando los límites de sujeción y manipulación de esta.

El programa de Matlab hace posible realizar la simulación, y para llevarlo a cabo, se necesita de un modelo tridimensional de la mano robótica, además de un enlace del modelo tridimensional con las ecuaciones, las cuales provienen de las variables θ_3 , θ_4 y θ_5 . En la visualización del modelo tridimensional se implementa la herramienta de V-Realm Builder, el cual permite diseñar y editar escenas en tres

dimensiones. En cambio, en la implementación de las ecuaciones con el modelo se opta por utilizar el programa Simulink, el cual permite programar un ambiente virtual.

Modelo tridimensional de la mano robótica

El modelo virtual de la mano robótica respeta las medidas del diseño que se realizó en SolidWorks, pero se simplifica las piezas hasta una forma geométrica tridimensional debido a que solamente se necesita conocer los puntos en donde la mano robótica no podría manipular el objeto, y de esta forma trazar trayectorias de manipulación para esta. La Figura 3.24 muestra el modelo tridimensional de toda la mano robótica en el programa V-Realm Builder.

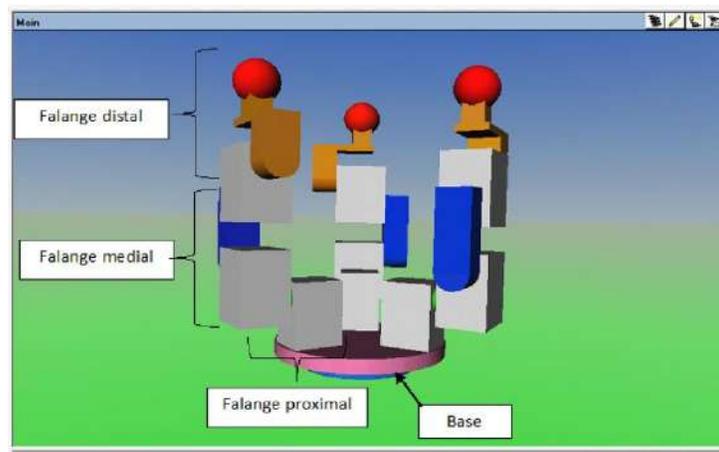


Figura 3.24 Modelo de la mano robótica en el programa V-Realm Builder.

En la Figura 3.24 se logra observar que las piezas son simplemente figuras geométricas: la base se contemplan como dos cilindros: uno azul y el otro rosa; los servomotores son prismas rectangulares color gris, conservando las dimensiones generales de estos mismos; las articulaciones son simples cilindros con extensiones en forma de prisma rectangulares, los cuales conectan con el siguiente servomotor o eslabón, pero con un color diferente para cada articulación del dedo: la primera siendo de color verde, el azul para la segunda y el café para la tercera articulación; por último se observa que la esfera del efector terminal del dedo es de color rojo.

Programación de las ecuaciones de la mano robótica

El modelo creado en el programa V-Realm Builder se enlaza con el programa de Simulink en Matlab, para de esta forma programar las ecuaciones que describen el movimiento de las articulaciones en la mano robótica. En dicha herramienta se programarán las ecuaciones de desplazamiento y orientación para cada dedo, en la Figura 3.25 se observa de manera general la programación de las ecuaciones en el simulador.

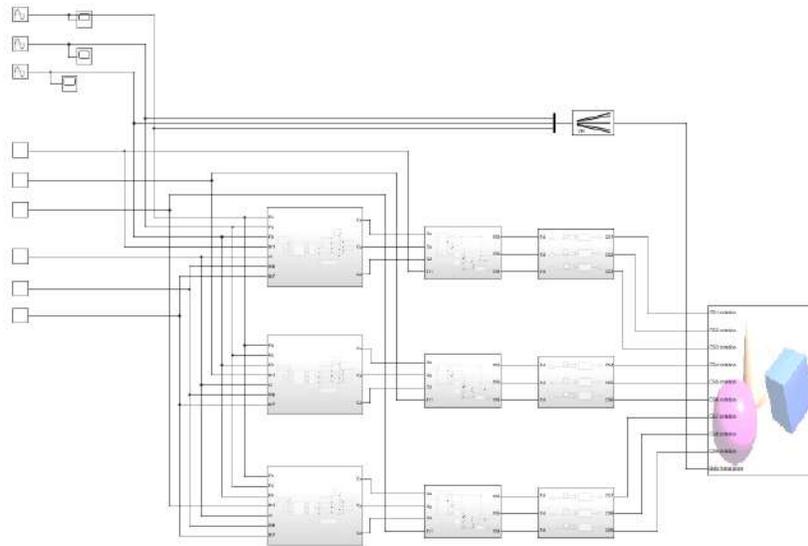


Figura 3.25 Diagrama general de programación de la mano robótica en el simulador.

El diagrama de la Figura 3.25 muestra en la parte superior izquierda tres señales de entrada, las cuales son la trayectoria del vector ' \mathbb{P} ' en las ecuaciones, y en la Figura 3.26 se muestra con más detalladamente que son los puntos P_x , P_y y P_z de la trayectoria.

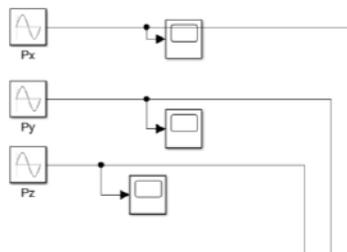


Figura 3.26 Trayectoria de simulación del punto central de la esfera, es decir del vector \mathbb{P} .

En la misma parte de la Figura 3.25, pero debajo de las señales de la trayectoria del vector \mathbb{P} , se encuentran los ángulos θ_1 de cada dedo, en la Figura 3.27 se muestra que cada ángulo tiene un valor diferente, y además son de 120° de diferencia entre cada uno, debido que los dedos de la mano están dispuestos en 120° uno del otro, y por lo tanto también las cadenas cinemáticas de cada uno.

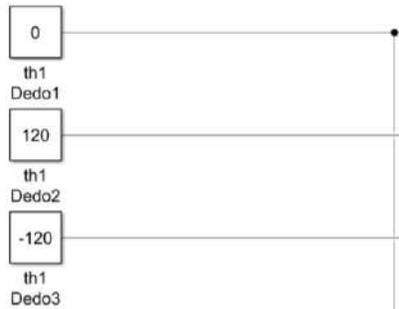


Figura 3.27 Ángulos θ_1 para cada uno de los dedos de la mano robótica.

Por último, en la parte izquierda de la Figura 3.25 se puede ver tres recuadros, los cuales son el radio del objeto, el cual es el radio de una esfera, y los ángulos θ_6 y θ_7 respectivamente, la Figura 3.28 muestra estos tres últimos recuadros que representan las entradas de la simulación.

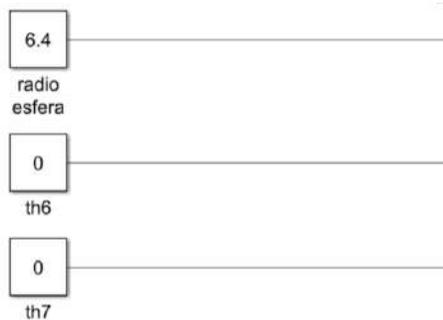


Figura 3.28 Radio propuesto del objeto y los ángulos θ_6 y θ_7 .

Estas tres últimas imágenes muestran las entradas que se requieren para que la simulación de la mano robótica funcione. En la primera parte se proporciona un camino por el que el punto P recorrerá, que tal punto es el centro del objeto; en este caso son las tres señales de entrada que permitirán dibujar el recorrido en las tres dimensiones, además siendo los seis recuadros constantes de la simulación: primero con los tres ángulos θ_1 cimentando la disposición de cada dedo en la base; segundo el radio del objeto, que aunque pueda variar debido al tamaño del objeto, su valor no lo hará durante la manipulación; por último los ángulos θ_6 y θ_7 determinan la forma en que las esferas de los dedos sujetan al objeto, estos últimos datos pueden tener un valor diferente dependiendo en donde se encuentre sujetando el objeto o simplemente se puede optar por mantener un valor fijo para estos ángulos, en este caso se mantiene en un valor fijo para facilitar la acción de manipular.

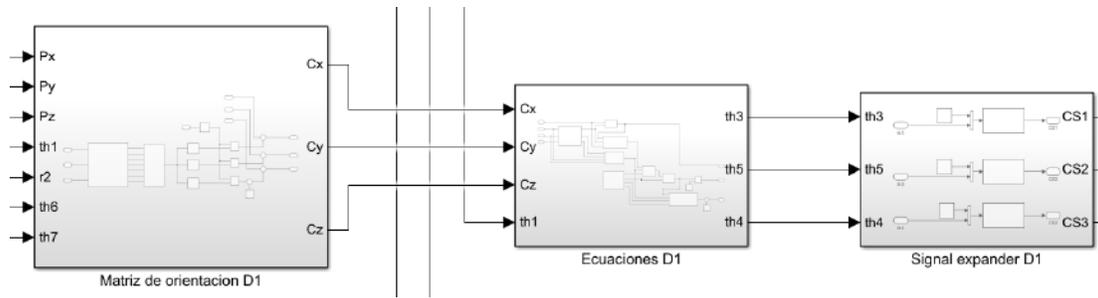


Figura 3.29 Bloques de la matriz de orientación, de las ecuaciones y de las señales que surgen de estas últimas dirigidas hacia el modelo tridimensional de la mano robótica.

Con las entradas definidas, estas entran en otros bloques, en donde se encuentra la programación de las ecuaciones que permiten el movimiento de un dedo. En la Figura 3.29 se observan los recuadros, en donde el primero de ellos es la programación de la matriz de orientación Q y el segundo el de las ecuaciones de desplazamiento, debajo del primer recuadro se lee “Matriz de orientación D1”, el ‘D1’ significa que esta matriz de orientación pertenece a dedo número uno, esto último también se encuentra en el segundo recuadro.

La programación de la matriz de orientación se puede ver en la Figura 3.30, en donde se observa como las señales P_x , P_y , P_z , el ángulo θ_1 , el radio del objeto y la señal de los ángulos θ_6 y θ_7 , son las señales de entrada para este recuadro.

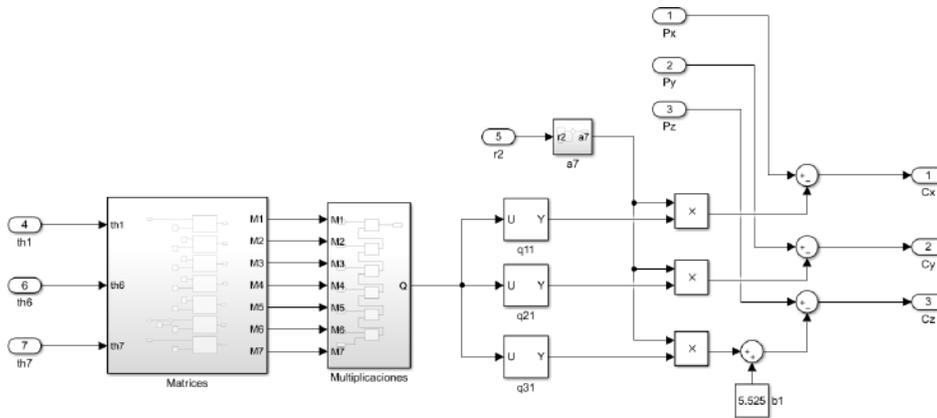


Figura 3.30 Programación de la matriz de orientación.

Con las entradas para realizar el cálculo de la matriz de orientación, las señales de los ángulos Theta se internan en el primer recuadro de la Figura 3.30, el cual se llama “Matrices”, en este recuadro se le asigna a cada una de las matrices de orientación de cada marco de referencia los valores de los ángulos de Alpha y, de ángulos los faltantes Theta que se encuentran en la Tabla 3-2. Después en esa misma imagen se observa un siguiente recuadro en donde se realiza las operaciones de multiplicación y lleva por nombre “Multiplicaciones”, de ese recuadro sale una señal, la cual contiene los tres valores q_{11} , q_{21} y q_{31} que se necesitan para multiplicar y sumar los valores de los recuadros a_7 y b_1 , que son las distancias a

las que se encuentran en la Tabla 3-2. Por último, se suman las operaciones anteriores con las señales P_x , P_y y P_z , por lo que toda esta programación no solamente corresponde a la matriz de orientación, sino también a las operaciones mostradas en la ecuación (3.10), y con ello salen tres señales de este recuadro de programación llamadas C_x , C_y y C_z .

Con la programación de la matriz de orientación para un dedo, se continua con la programación de las ecuaciones para poder mover las articulaciones del modelo, se utilizan las señales de salida del recuadro de “Matriz de orientación D1” como señales de entrada para el bloque “Ecuaciones D1” además de la señal del ángulo θ_1 , y todo lo anteriormente mencionado se muestra en la Figura 3.31.

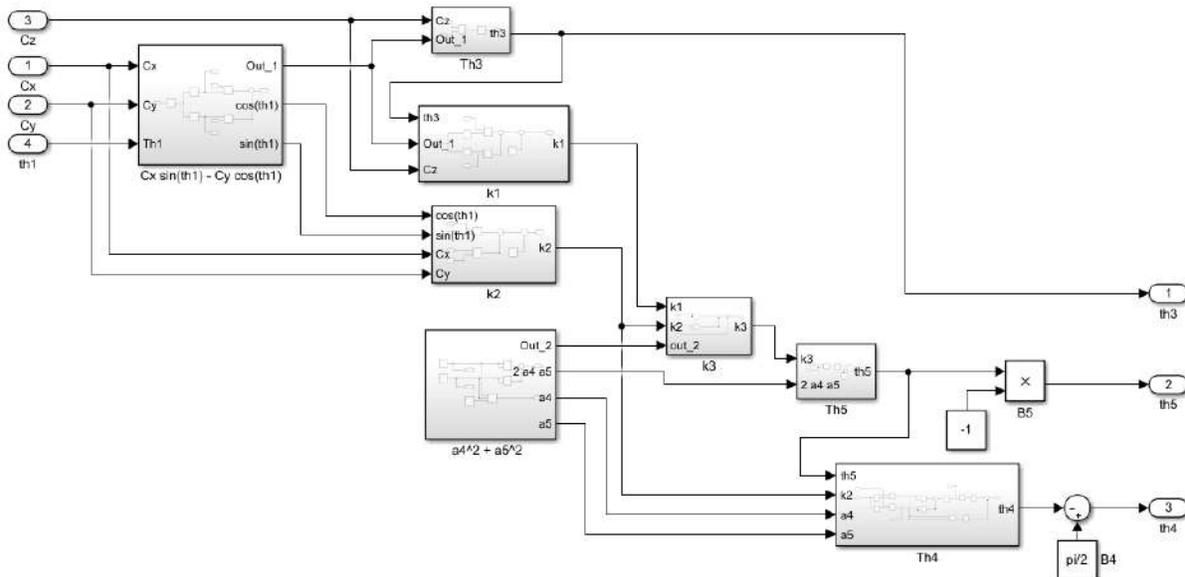


Figura 3.31 Programación de las ecuaciones de desplazamiento.

La imagen de arriba muestra como entran las señales: C_x , C_y , C_z y θ_1 en el recuadro de programación movimiento de las articulaciones. Lo primero que se observa es que C_x , C_y y θ_1 entran en el primer recuadro, el cual es la siguiente operación $C_x \sin(\theta_1) - C_y \cos(\theta_1)$, y que cuya salida Out_1 entra en el recuadro “Th3”, en este mismo recuadro se utiliza C_z para poder realizar las operaciones de la ecuación (3.14) y se obtiene el valor de θ_3 . Con el valor de θ_3 calculado, se utiliza su señal en el recuadro “k1” que representa la ecuación (3.15), la cual es una simplificación utilizada en un sistema de ecuaciones; el bloque “k2” es la ecuación (3.16) la cual también es una simplificación para el sistema de ecuaciones, este último recuadro “k2” se emplea las señales C_x , C_y y dos señales del primer recuadro, debido que utiliza la misma operación $C_x \sin(\theta_1) - C_y \cos(\theta_1)$; con los valores obtenidos del recuadro “k1” y “k2” estos pasan al recuadro de “k3”, el cual representa las operaciones de la ecuación (3.17), pero también se le agrega otra señal que son valores $a4$ y $a5$ que se utilizan en dicha ecuación, por lo que la señal de “k3” y estos últimos valores permiten obtener el valor de θ_5 , siendo la ecuación (3.19) la que se programa en este recuadro “Th5”. En la obtención del valor de θ_4

se usa la ecuación (3.20) para la programarla en el recuadro “Th4”, en donde se obtiene el valor de dicho ángulo, en este recuadro se utilizan la señal de $Th5$, que es valor del ángulo θ_5 , además de las señales de los valores $a4$, $a5$, y por último la señal que proviene de “k2”. En la parte de programación mostrada en la Figura 3.31 se observa una suma para las señales de los ángulos θ_4 y θ_5 , esta operación se origina porque en el modelo tridimensional de la simulación tiene los marcos de referencia desfasados con respecto a los teorizados en la arquitectura cinemática de un dedo robótico. Con los valores calculados de los ángulos, estos se utilizan en señales de salida del recuadro “Ecuaciones D1” y pasan al recuadro “Signal expander D1” de Figura 3.29, para ser implementadas adecuadamente en el modelo tridimensional.

Los tres dedos de la mano robótica implementan la programación explicada en el párrafo de arriba, porque utilizan las mismas ecuaciones con la diferencia de que algunas constantes poseen diferentes valores, todas las señales de cada uno de los dedos y del objeto se unen en un bloque, el cual conecta el mundo virtual de modelo tridimensional con las ecuaciones programadas, la Figura 3.32 muestra dicho bloque.

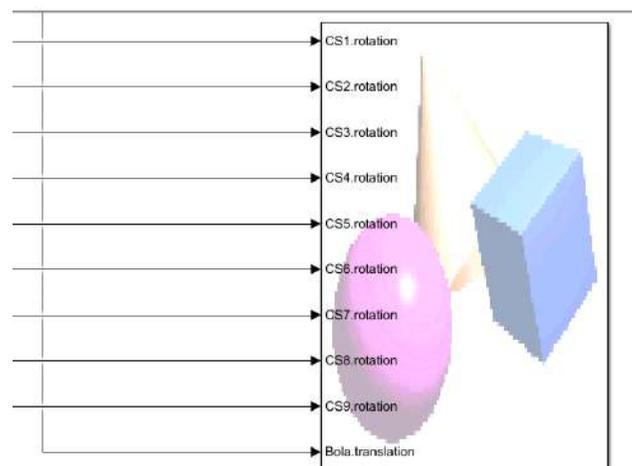


Figura 3.32 Bloque VR Sink del modelo de la mano robótica.

La imagen mostrada arriba es un acercamiento en la parte derecha inferior de la Figura 3.25, esta última imagen muestra el bloque VR Sink con las señales de entrada como CS, que son los ángulos de la articulación para cada dedo, siendo CS1, CS2 y CS3 para el primer dedo; CS4, CS5 y CS6 para el segundo dedo; CS8, CS7 y CS9 para el tercer dedo; y por último las tres señales de entrada P_x , P_y y P_z se agrupan una señal, la cual representa el centro del objeto o el punto ‘P’ en las ecuaciones. Todas las señales son visualizadas en el mundo virtual del bloque VR Sink, la Figura 3.33 muestra la mano robótica sosteniendo el objeto esférico en el mundo virtual.

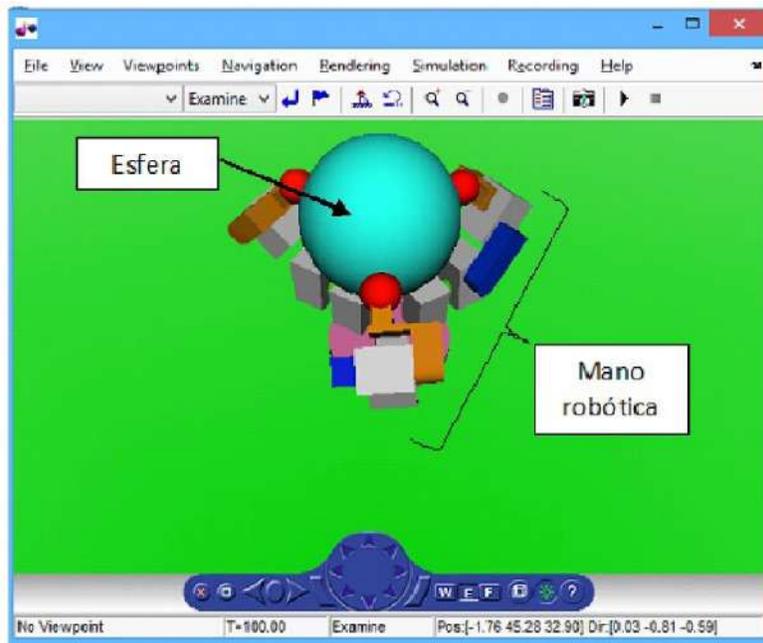


Figura 3.33 Sujeción del objeto esférico en el mundo virtual del bloque VR Sink.

3.5 Manufactura

Con la corroboración de las ecuaciones de la mano robótica mediante el programa de simulación, se retoma el diseño en SolidWorks, para poder obtener las medidas exactas de las piezas y de esta forma poder fabricarlas.

3.5.1 Fabricación

Las piezas son creadas en impresión 3D mediante el método de fabricación de filamento fundido (Fused Deposition Modeling (FDM)), el material utilizado es el Ácido Poliláctico (PLA) y es el mismo utilizado en SolidWorks para calcular la masa de cada una de las piezas, con excepción del rodamiento, las esferas y los actuadores. La forma en cómo se imprimieron las piezas fue considerando un 100% de densidad del material, para que las piezas fueran los más rígidas posibles y resistentes con los movimientos de los servomotores.

En las siguientes imágenes, desde la Figura 3.34 hasta la Figura 3.47 se muestran las piezas impresas, y debajo de ellas se encuentra una pequeña descripción de los tornillos utilizados en la misma pieza además de donde va ensamblada dicha pieza.



Figura 3.34 Base de la mano robótica.

Figura 3.34, en esta pieza se utiliza cuatro tornillos M5x15 para ensamblarse con el robot Yaskawa, y doce tornillos M3x10 para las piezas de la figura de abajo.

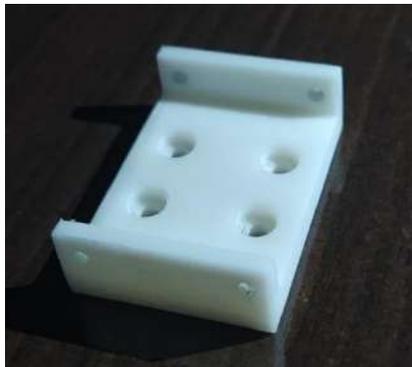


Figura 3.35 Pieza de ensamble 1.1

Figura 3.35, se utiliza los mismos cuatro tornillos M3x10 para ensamblarse con la base, además también utiliza cuatro tornillos M2.5x12 para sujetar el servomotor XM-430-W210.



Figura 3.36 Pieza de ensamble 1.2

Figura 3.36, utiliza cuatro tornillos M3x16 para ensamblarse con la base, además de cuatro u ocho M2x16 para unir la pieza de ensamble 2. Esta pieza sustituye a la pieza de ensamble 1.1 con el servomotor XM430-W210 para poder unir un servomotor AX-12A con la base.



Figura 3.37 Pieza de ensamble 2.

Figura 3.37, en esta pieza puede llegar a utilizar de cuatro a ocho tornillos M2x6 para ensamblarse con la base en el eje del servomotor XM-430-W210, y cuatro tornillos M2x8 para sujetar el servomotor AX-A12. Los tornillos de 6mm de largo son colocados alrededor del gran orificio central.



Figura 3.38 Pieza de ensamble 3.

Figura 3.38, esta pieza es ensamblada con cuatro tornillos M2x6 en la base para el eje del servomotor AX-A12, y se usan seis tornillos M2x12 para ensamblarse con el otro servomotor AX-A1.



Figura 3.39 Pieza de ensamble 4.

Figura 3.39, en esta pieza son requeridos solamente seis tornillos M2x10 para sujetar el siguiente servomotor AX-A12 por lado posterior de este, en su orificio es colocado un rodamiento 6701-ZZ.



Figura 3.40 Pieza de ensamble 5.

Figura 3.40, esta pieza utiliza cuatro tornillos M2x8, para ensamblarse con la basen en el eje del servomotor AX-A12, y dos tornillos M2x10 para sujetar la pieza de ensamble 7.



Figura 3.41 Pieza de ensamble 6.

Figura 3.41, esta pieza solamente utiliza dos tornillos M2x10 para sujetar la pieza de ensamble 7, esta pieza es acoplada por el lado posterior del servomotor AX-A12, su orificio también lleva un rodamiento 6701-ZZ.



Figura 3.42 Rodamiento 6701-ZZ.

Figura 3.42, el rodamiento 6701-ZZ solamente es utilizado para permitir que las piezas de ensamble 4 y 6 giren con una mayor facilidad, este es colocado en el gran orificio de estas dos piezas



Figura 3.43 Base del rodamiento.

Figura 3.43, la base del rodamiento es colocada en la parte posterior del motor AX-A12, permitiendo que el rodamiento se ajuste con una protuberancia cilíndrica de este servomotor.



Figura 3.44 Tapa del rodamiento.

Figura 3.44, esta pieza utiliza un tornillo M3x12 el cual es colocado en la parte posterior de motor AX-A12, evitando que el rodamiento y las piezas de ensamble 4 y 6 se desacoplen del servomotor.

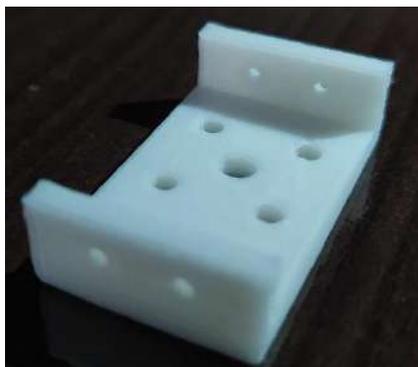


Figura 3.45 Pieza de ensamble 7.

Figura 3.45, esta pieza utiliza cuatro tornillos M3x10 para ensamblarse con la base de la esfera y, otros cuatro M2x10 para las piezas de ensamble 5 y 6, esta pieza conecta la penúltima parte del dedo con el segundo servomotor AX-A12.



Figura 3.46 Base de la esfera.

Figura 3.46, esta pieza utiliza cuatro tornillos M3x10 para ensamblarse con la pieza de ensamble 7, sujeta la esfera que es la parte final del dedo de la mano robótica.



Figura 3.47 Esfera.

Figura 3.47, esta pieza es una esfera que simplemente va embonada con su base, y por ende es el efector final para cada dedo de la mano robótica.

3.5.2 Ensamblaje

Con las piezas fabricadas por medio de la impresión en 3D, se prosigue con el ensamble de un dedo de la mano robótica, para de esta forma poder observar el acoplamiento de las piezas. La forma de ensamble se detalla aquí abajo en una secuencia de pasos a seguir:

1. El primer paso es simplemente tener la base de la mano robótica, y colocar sobre ella la pieza de ensamble 1 con cuatro tornillos M3x12 y sus tuercas, esto se puede visualizar en la Figura 3.48.



Figura 3.48 Pieza de ensamble 1 acoplada con la base de la mano robótica.

2. El siguiente paso es colocar el servomotor XM-430-W210, con tornillos M2.5x16 por el lado posterior y el frontal de servomotor, la Figura 3.49 muestra el acoplamiento.



Figura 3.49 Ensamble de servomotor XM-430-W210.

3. Lo siguiente es colocar la pieza de ensamble 2 en la base para el eje del servomotor XM-430-W210, en este ensamblaje se colocan los cuatro tornillos M2x6, aunque pueden llegar a ser ocho, la Figura 3.50 muestra el ensamble de la pieza.



Figura 3.50 Ensamble de la pieza de en ensamble 2 con el servomotor XM-430-W210.

4. En este paso se ensambla con cuatro tornillos M2x8 el servomotor AX-A12, con la pieza de ensamble 2, en la Figura 3.51 muestra el ensamble.



Figura 3.51 Ensamble del servomotor AX-A12 con la pieza de ensamble 2.

5. El siguiente paso es acoplar la pieza de ensamble 3, con la base para el eje del servomotor AX-A12, en el ensamblaje se utilizan cuatro tornillos M2x6, y esto se puede visualizar en la Figura 3.52.



Figura 3.52 Ensamble de la pieza de ensamble 3.

6. En este paso se ensambla con seis tornillos M2x12, el segundo servomotor AX-A12 con la pieza de ensamble 3, la Figura 3.53 muestra el ensamblaje.



Figura 3.53 Ensamble del segundo servomotor AX-A12 con la pieza de ensamble 3.

7. Para este paso la pieza de ensamble 4 se acopla con el primer servomotor AX-A12 en tres simples etapas: en la primero se coloca sobre el servomotor la base del rodamiento; después se coloca el rodamiento y la pieza de ensamble 4 previamente acoplados, se verifica que la pieza de ensamble 4 este alineada con el segundo servomotor AX-A12; y por último, se coloca la tapa de rodamiento sobre con un tornillo M3x12; todo este procedimiento se puede observar en la Figura 3.54.



Figura 3.54 Ensamble de la pieza de ensamble 4 con el primer servomotor AX-A12.

8. Después de ensamblar la pieza de ensamble 4 con el primer servomotor AX-A12, esta se ensambla con el segundo servomotor AX-A12 con seis tornillos M2x10, la Figura 3.55 muestra el ensamblaje.



Figura 3.55 Ensamble de la pieza de ensamble 4 con el segundo servomotor AX-A12.

9. En este paso solamente se une la pieza de ensamble 5 en la base para el eje del segundo servomotor AX-A12, en este ensamblaje son utilizados cuatro tornillos M2x8, la Figura 3.56 muestra el resultado.



Figura 3.56 Ensamble de la pieza de ensamble 5.

10. En este paso se realiza el mismo procedimiento de tres etapas que en el paso 7, solamente que es con la pieza de ensamble 6; se coloca la base de rodamiento sobre el segundo servomotor AX-A12 por el lado posterior, después el rodamiento se coloca sobre la base del de este y por último se coloca la tapa del rodamiento con un tornillo M3x12, todo esto se muestra en la Figura 3.57.



Figura 3.57 Ensamble de la pieza de ensamble 6 con el segundo servomotor AX-A12.

11. En este paso se ensambla con cuatro tornillos M3x10, la pieza de ensamble 7 con la base de la esfera, además de acoplar la esfera con la base de esta misma, todo esto se muestra en la Figura 3.58.



Figura 3.58 Ensamble de la pieza de ensamble 7, la base de la esfera y la esfera misma.

12. Este es el último paso y el ensamble utiliza cuatro tornillos M2x10, se unen las piezas del paso anterior con las piezas de ensamble 5 y 6 utilizando, el resultado se muestra en la Figura 3.59.



Figura 3.59 Ensamblaje de las piezas del paso 11 con las piezas de ensamble 5 y 6.

Con todas las piezas totalmente ensambladas se puede observar el resultado de la secuencia de pasos en la Figura 3.60, en ella se visualiza un dedo de la mano robótica en una posición vertical para las falanges medial y distal.



Figura 3.60 *Ensamble de un dedo de la mano robótica.*



Figura 3.61 *Ensamble de total de la mano robótica.*

La cantidad de tornillo necesaria para armar un solo dedo de la mano robótica es mostrada en la Tabla 3-3. En esta tabla se puede observar que los primeros dos dedos tienen los mismos datos, con excepción del tercero, el cual tiene en sus tres primeras filas datos con la misma cantidad de tornillos, pero las unidades son diferentes, esto es debido a que un servomotor XM430 se intercambia por la pieza de ensamblaje 1.1. El conteo final de cada dedo muestra una cantidad de 46 tornillos, y sumándolos el resultado es de 138 tornillos, pero aún son necesarios otros cuatro tornillos para unir la base de la mano robótica con el brazo robótico Yaskawa MH5LF, las medidas de estos últimos cuatro tornillos son M5 x 20, y en total serían 142 tornillos necesarios para poder ensamblar la mano robótica.

Tabla 3-3 Tabla de tornillos necesarios para cada dedo, con la cantidad de tornillos necesarios y sus respectivas medidas

Dedo 1. Tres servomotores		Dedo 2. Tres servomotores		Dedo 3. Dos servomotores	
No. de tornillos	Medida del tornillo (mm)	No. de tornillos	Medida del tornillo (mm)	No. de tornillos	Medida del tornillo (mm)
4	M3 x 12	4	M3 x 12	4	M3 x 16
4	M2.5 x 16	4	M2.5 x 16	4	M2 x 16
4 (8)	M2 x 6	4 (8)	M2 x 6	4 (8)	M2 x 16
4	M2 x 8	4	M2 x 8	4	M2 x 8
4	M2 x 6	4	M2 x 6	4	M2 x 6
6	M2 x 12	6	M2 x 12	6	M2 x 12
1	M3 x 12	1	M3 x 12	1	M3 x 12
6	M2 x 10	6	M2 x 10	6	M2 x 10
4	M2 x 8	4	M2 x 8	4	M2 x 8
1	M3 x 12	1	M3 x 12	1	M3 x 12
4	M3 x 10	4	M3 x 10	4	M3 x 10
4	M2 x 10	4	M2 x 10	4	M2 x 10
Total, de 46 tornillos		Total, de 46 tornillos		Total, de 46 tornillos	
Total, de los tres dedos: 138 tornillos					

La Tabla 3-4 muestra la cantidad total de los tornillos necesarios para poder ensamblar por completo la mano robótica. La tabla que se encuentra abajo sirve como un recuento de todos los tornillos que se están utilizando.

Tabla 3-4 Tabla de la cantidad de tornillos necesarios para ensamblar toda la mano robótica.

Cantidad de tornillos	Medidas del tornillo
4	M5 x 20
4	M3 x 16
14	M3 x 12
12	M3 x 10
8	M2.5 x 16
8	M2 x 16
18	M2 x 12
30	M2 x 10
24	M2 x 8
20	M2 x 6
Total: 142 tornillos	

En cuanto al peso total de la mano robótica junto con los tornillos difiere del obtenido mediante estimación. El peso estimando de la mano robótica ensamblada sin tornillos es de 917.95 g, en cambio el peso obtenido en con junto con los tornillos y los cables de comunicación es de 930 g, en la Figura 3.62 se muestra el peso mencionado; con lo visto en la figura se puede decir que las estimaciones no estaban muy lejos del peso total que llega a tener la mano robótica, además de que este peso este muy cercano al estimado, también se encuentra debajo del peso máximo que se le permitía tener, el cual era de 1.5 kg, dado que el brazo Yaskawa

MH5LF permite un peso máximo de 5 kg, y el de la mano robótica debía encontrarse en el mencionado previamente, dejando los 3.5 kg al objeto a sujetar y manipular .



Figura 3.62 Peso total de la mano robótica.

3.6 Restricciones de diseño

La manipulación es el principal propósito de esta mano robótica y, por ende, es importante considerar el rango de manipulación que esta tendrá, para ello encontrar los límites de acción será necesario, dicho esto las restricciones que esta mano robótica presenta estarán delimitadas por la dimensión del objeto y la posición que este alcance al estar sujetando por los tres dedos, en otras palabras, la posición que el objeto alcance variara con el tamaño de este último. También se considera el tamaño mínimo y máximo que la mano robótica puede sujetar, estos límites permiten saber si el objeto se puede sujetar o no, y por ende manipular.

Tamaño mínimo y máximo

Los valores para los tamaños mínimo y máximo que la mano robótica puede sujetar, se obtienen de dos diferentes formas. En cuanto al tamaño mínimo que se puede sujetar se obtiene mediante la geometría que forman las tres esferas de cada dedo al estar juntas, en la Figura 3.63 se muestra una vista de cómo se observarían las esferas de cada dedo una a lado de otra en 60° , esta imagen se ve desde una vista superior para poder distinguir el espacio que dejan las esferas al estar juntas, de esta manera se forman un triángulo equilátero, y con ello puede deducirse el tamaño del objeto que estará en el centro del triángulo y los círculos.

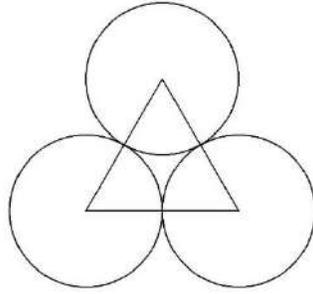


Figura 3.63 Vista superior de sujeción.

Con la información de que cada esfera en la Figura 3.63 tiene un radio de 1.5 cm, y que al estar juntas de esta manera forman un triángulo equilátero, lo único que se debe hacer es obtener el punto central del triángulo y la distancia de este al perímetro de uno de los tres círculos, la obtención de este último dato se hace mediante trigonometría, sacando la hipotenusa a 30° de cualquiera de los círculos y con el radio como cateto adyacente, y obteniendo un valor para la hipotenusa de 1.7320, al restar a este valor 1.5 se obtiene un valor de 0.232, que es el radio mínimo que el objeto debe tener para ser sujetado, pero debido a que este valor resulta muy pequeño y puede que sea complicado sujetar un objeto tal dimensión, se ha optado que el valor mínimo del radio sea de 0.5 cm, y por ende una diámetro de 1 cm mínimo.

En cuanto el valor del tamaño máximo que se puede sujetar, este es obtenido mediante la el programa de simulación, el cual se le asigna un diámetro al objeto y se prueba si este es capaz de ser manipulado por los dedos de la mano robótica o simplemente puede ser sujetado por ellos. El tamaño máximo en que los dedos son capaces sujetar sin que haya un problema, ya sea las dimensiones o la arquitectura de la mano robótica, es de 22 cm de diámetro; pero para tener un rango de mayor seguridad el valor se cambia a 20 cm de diámetro como tamaño máximo.

Rango de movimiento del objeto

El rango de movimiento que la mano robótica posee dependerá del rango que posee cada dedo, debido a que son estos los que al final sostienen al objeto y lo manipulan. Encontrar el rango de movimiento en tres dimensiones no es muy complicado, pero delimitar hasta donde se podrá mover cada dedo lo es, porque se debe considerar las capacidades de movimiento de los actuadores; las dimensiones de las partes que conforman a los dedos para que no colisionen uno con el otro; el tamaño del objeto que se está sosteniendo, y que el objeto no choque con alguna otra parte de los dedos indebidamente. Con lo anteriormente considerado, las limitaciones del rango de movimiento del objeto dependen del movimiento de los dedos y sus dimensiones, la obtención de dichos rangos es adquirida mediante el uso del modelo en CAD del programa de SolidWorks, la Figura 3.65 muestra el

modelo de la mano robótica con un plano atravesando en medio de uno de los dedos, para hacer una descripción visual de los límites de un dedo robótico.

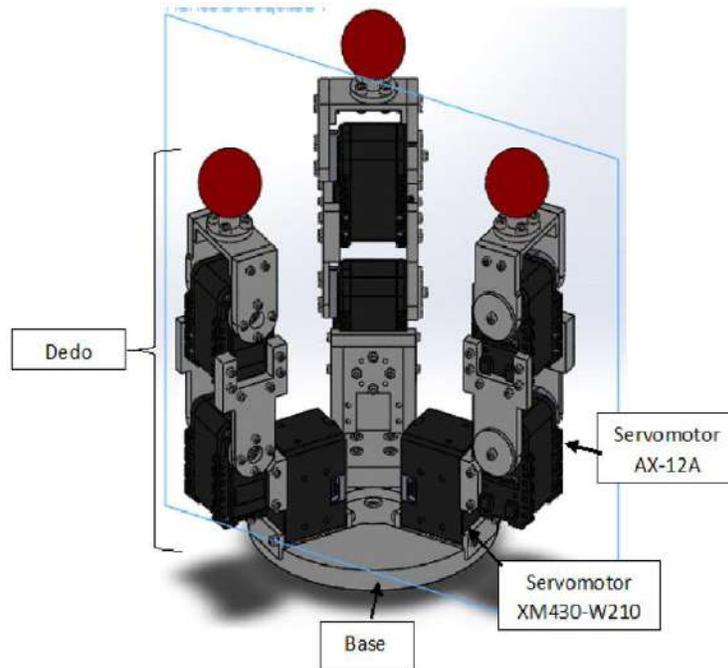


Figura 3.64 Modelo de la mano robótica en CAD.

El rango de máximo alcance de un dedo de la mano robótica se obtiene con las medidas que se muestran en la Figura 3.65, en donde la imagen de la izquierda se observa que por el centro de la base cruza una línea perpendicular a ella, la cual es el eje Z de la arquitectura de la mano robótica; en cambio, el eje X que se observa en la imagen de la derecha está a una altura 67.75 mm de la base, esta es altura del centro de rotación del segundo servomotor, el cual es el centro de rotación de la falange medial; esta figura muestra también una distancia de 79 mm de separación entre el eje Z y una línea que pasa por los dos centros de rotación de las falange media, distal y el centro de la esfera; en cuanto a la distancia de 122.5 mm, es el radio de máximo de acción de estas dos falanges, teniendo esto en cuenta todo lo anterior se puede crear una ecuación de posición en el eje Z y X, que es la distancia máxima que un dedo puede alcanzar a sujetar el objeto.

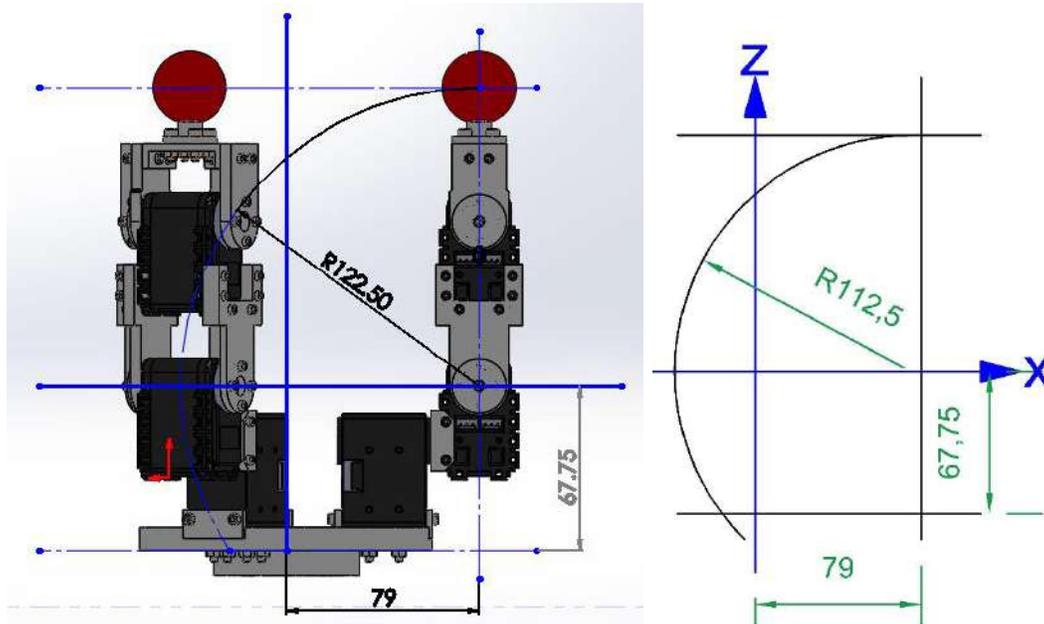


Figura 3.65 Imagen de perfil de un dedo de la mano robótica desplegado con las medidas en milímetros.

Para obtener la ecuación, se hace uso de la fórmula del círculo en donde el centro de rotación de la falange medial es el centro del círculo:

$$(x - a)^2 + (z - b)^2 = r^2$$

siendo $a = 79$ mm, $b = 0$ y $r = 122.5$:

$$(x - 79)^2 + z^2 = (122.5)^2$$

y despejando cada una de las variables, dan como resultado cuatro fórmulas de las que se utilizan solamente dos, las cuales son las siguientes:

$$z = \sqrt{(122.5)^2 - (x - 79)^2} \quad (3.23)$$

$$x = -\sqrt{(122.5)^2 - z^2} + 79 \quad (3.24)$$

La ecuación (3.23) es para la posición máxima en Z y la ecuación (3.24) para la posición máxima en X. Estas ecuaciones necesitan de otros datos, como el hecho de que la verdadera distancia en Z se mide desde el eje que se encuentra en la base de la mano robótica, y la ecuación (3.23) muestra cómo se adaptaría el dato de la altura; también, la distancia 79 mm, que va del centro de la esfera al centro del objeto, y en la cual se encuentra el valor del radio del objeto; la ecuación (3.26) muestra a ro (radio del objeto) sumado con 1.5 cm (radio de la esfera de un dedo).

$$z = h - 67.75 \quad (3.25)$$

$$x = ro + 1.5 \quad (3.26)$$

El uso de estas cuatro ecuaciones es ambiguo, debido a que se puede usar las primeras dos ecuaciones, (3.23) y (3.24), para después obtener de las ecuaciones (3.23) y (3.26) que son los valores del radio del objeto necesario y la altura desde la base requerida; pero lo preferible es utilizar primero las ecuaciones (3.23) y (3.26) para después de las ecuaciones (3.23) y (3.24) obtener los rangos máximos en los que se podrá posicionar el dedo, también esta segunda opción permitirá utilizar otras dos ecuaciones para obtener los rangos mínimos de posición de un dedo de la mano robótica, en la Figura 3.66 se observa la forma en que se obtendrán dichos valores.

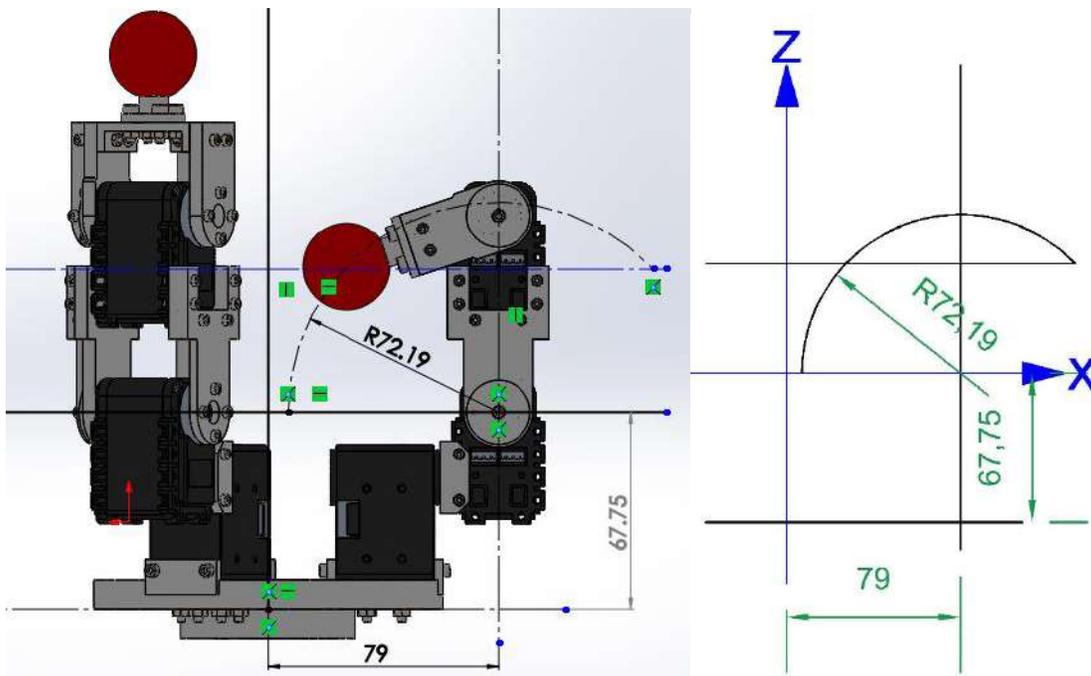


Figura 3.66 Imagen de perfil de un dedo de la mano robótica doblado, con las medidas en milímetros.

De la misma forma en que se obtuvieron las ecuaciones para los valores máximos de posicionamiento de un dedo, también se hace para los valores mínimos. La Figura 3.66 muestra un radio de 72.19 mm, con las mismas distancias al centro de rotación de la falange medial, con estos datos las ecuaciones que se obtienen son las siguiente:

$$z = \sqrt{(72.19)^2 - (x - 79)^2} \quad (3.27)$$

$$x = -\sqrt{(72.19)^2 - z^2} + 79 \quad (3.28)$$

Las ecuaciones de la (3.23) hasta (3.28), son parámetros que restringen el posicionamiento de los dedos de la mano robótica cuando estos se encuentran sujetando un objeto, pero esto no significa que es el rango total de posicionamiento de los dedos. Las ecuaciones mencionadas son límites que permite a la mano robótica sujetar el objeto sin que se este mismo se valla a soltar, y que los dedos sean capaces de amoldarse al objeto, cada dedo puede hacer uso de estas ecuaciones, pero con la consideración de que cada plano de los tres dedos se encuentra a 120° con respecto uno con el otro. La Figura 3.67 muestra cuál sería la zona en la que se le permitiría a un de dedo de la mano robótica moverse, mientras está sosteniendo un objeto de un radio entre 1 a 20 cm; en la imagen de la izquierda se observa que la distancia más grande que se puede alcanzar en el eje X desde el eje Z es de 115 mm (11.5 cm), esto debido a que el objeto más grande a sujetar es de 10 cm (100 mm) de radio y a este se le suma el radio de la esfera del dedo el cual es de 1.5 cm (15 mm); esto mismo se puede ver con la distancia de 20 mm (20 mm) en esa misma imagen, donde el objeto más pequeño a sujetar debe tener un radio de 0.5 cm (5 mm) y al sumarle el radio de la esfera del dedo da como resultado esa distancia. El centro la esfera de un dedo es el punto clave para poder determinar la zona de movimiento del mismo dedo; en ese punto en específico, como el eje de rotación, permiten al centro del objeto posicionarse en diferentes partes sin que el ángulo entre el objeto y el de la esfera de un dedo sea un inconveniente.

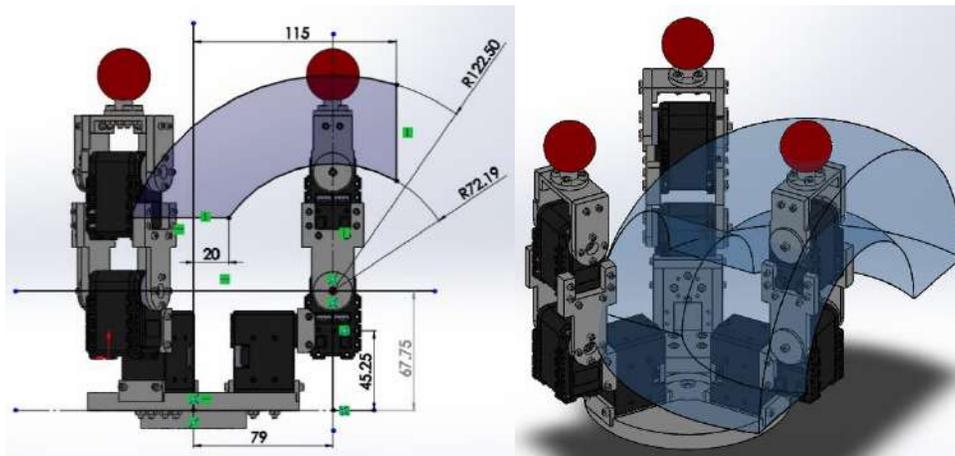


Figura 3.67 Zona de movimiento permitida para un dedo de la mano robótica.

El rango en el que un dedo se puede posicionar es mostrado en las Figura 3.68 (unidades de distancia se encuentran en mm), en la imagen de la izquierda se observa que su rango genera una figura similar a un semicírculo, esto se debe a que si el servomotor girara hasta formar el semicírculo la falange medial chocaría con el servomotor de la falange proximal, esta restricción le impide que suceda; esto mismo se puede observar en la imagen de la derecha en donde no se forma una semiesfera total, aunque con la consideración de que los otros dos dedos no encuentren ensamblados a la mano robótica, pero de estarlo se deberá tener en cuenta para que no colisionen un con otro. Si bien se observa en la imagen de la

derecha un rango en tres dimensiones, en el diseño de la mano robótica solo se contempla que dos dedos tengan ese rango, dejando a un dedo con un movimiento en dos dimensiones, que es la imagen de la izquierda; y todo esto debido a que la mano utiliza ocho servomotores, en donde dos dedos usan tres servomotores y relegando a un tercer dedo con dos.

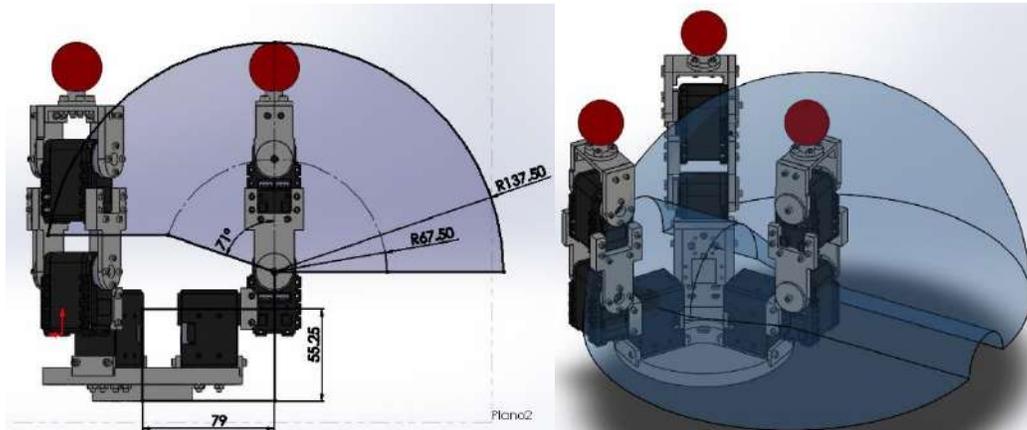


Figura 3.68 Zona de alcance de posicionamiento un dedo de la mano robótica en dos dimensiones (imagen de la izquierda) y en tres dimensiones (imagen de la derecha).

4 RESULTADOS Y DISCUSIÓN

En este capítulo se aborda lo que es el funcionamiento de la simulación para manipular un objeto, para de esta manera hacer pruebas con la mano robótica y corroborar si puede trabajar o moverse sin complicaciones, todo esto mediante una interfaz de control que ordenar a los servomotores como moverse. También se explica cómo se debe utilizar la interfaz y como es la comunicación con los servomotores, y por último se explicarán pruebas realizadas físicamente con la mano robótica para confirmar el funcionamiento de la misma.

4.1 Simulación

Con la creación de un entorno virtual para la simulación de la mano robótica, se prosigue con el hecho de poner a prueba la arquitectura que conforma la mano robótica. Esta prueba de la simulación tomará en cuenta que el objeto, en este caso podrá ser una pelota o un objeto esférico de 6.5 cm de radio, se encuentra sujetado por los tres dedos de la mano robótica. La Figura 4.1 muestra en el programa de Simulink a tres señales senoidales generadas en cada uno de los ejes X, Y y Z que corresponden a un espacio tridimensional, estas señales están identificadas como P_x , P_y y P_z para cada uno de los ejes respectivamente, pero solamente P_z se encuentra generando una señal de 3 mm de amplitud y 16 mm de offset mientras las otras dos señales tienen un valor de cero en su amplitud.

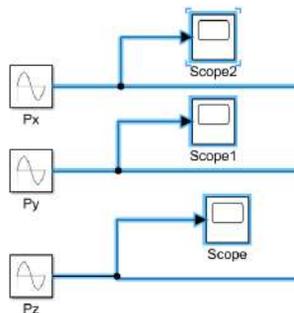


Figura 4.1 Señales de movimiento del punto 'P' en el programa de Simulink.

El valor de cada señal, en este caso P_x , P_y y P_z , se puede visualizar en la Figura 4.2, de color rojo para las dos primeras señales y amarilla para la última. En la figura se muestra una línea roja en el valor cero a través del tiempo, esto confirma que las dos primeras señales mantienen este valor durante toda la prueba, en cambio la señal amarilla que se muestra presenta un comportamiento senoidal con una amplitud de 3 mm y un offset de 16 mm, por lo que es la señal de P_z .

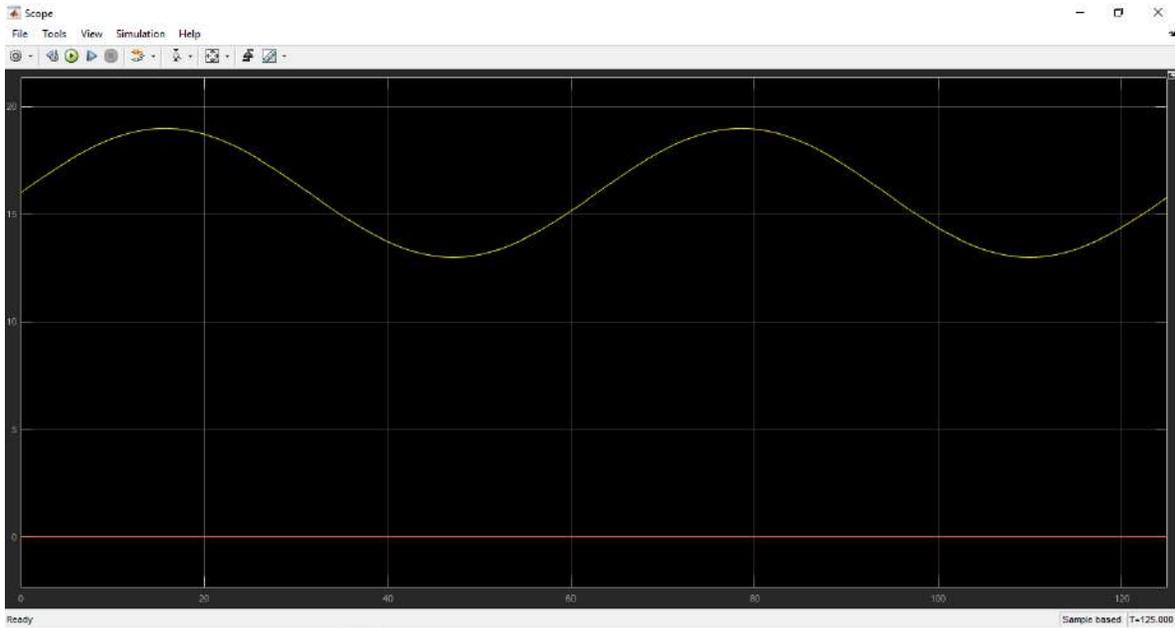


Figura 4.2 Señal senoidal P_x y P_y , roja; señal senoidal P_z , amarilla.

En el siguiente Figura 4.3 se muestran escaneadas las tres señales que salen del bloque de “Ecuaciones D1”, estas tres señales son los ángulos θ_3 , θ_4 y θ_5 de la simulación. Este escaneo ayuda corroborar el movimiento de cada una de las articulaciones del dedo de la mano robótica. Estas señales no son necesariamente iguales a las mostradas por la P_x , P_y y P_z , ya que su escaneo es utilizado para comprobar si la programación de las ecuaciones está realizando un cálculo.

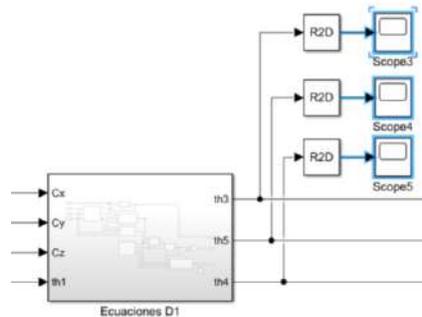


Figura 4.3 Señales de los ángulos θ_3 , θ_4 y θ_5 del dedo número 1.

En la siguiente imagen se muestra el escaneo de las señales de salida del bloque “Ecuaciones D1”. La Figura 4.4 muestra las señales de los ángulos θ_3 , θ_4 y θ_5 ; el movimiento del ángulo θ_3 se representa con una línea de color amarillo, además de que θ_3 describe el movimiento del primer servomotor en el dedo robótico, la señal mostrada se encuentra en cero constantemente por lo que se deduce que el servomotor 1 mantiene el objeto en el eje Z. En cambio, las señales de color rojo y azul en la gráfica representan los movimientos angulares de θ_4 y θ_5

respectivamente, estas dos señales dan constancia del movimiento en el eje Z de la mano robótica.

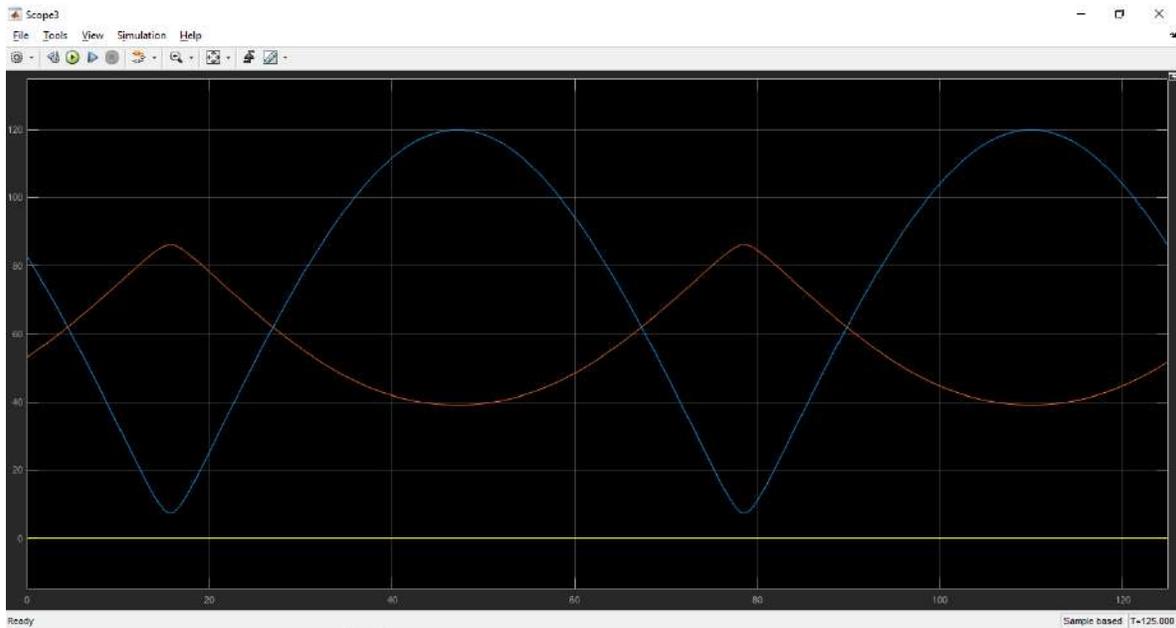


Figura 4.4 Señal den ángulo θ_3 , amarillo; señal del ángulo θ_4 , rojo; señal del ángulo θ_5 , azul.

Con las señales de los ángulos ya escaneadas solamente es necesario observar el movimiento de la mano robótica en un entorno virtual, la Figura 4.5 muestra el entorno virtual mientras que la Figura 4.6 muestra una secuencia movimientos de la mano robótica, la cual se está moviendo el con el objeto en el eje Z con la punta de sus dedos, en este caso las esferas de estos mismos. Estas imágenes confirman el movimiento que se puede realizar con las articulaciones de los dedos de la mano robótica.

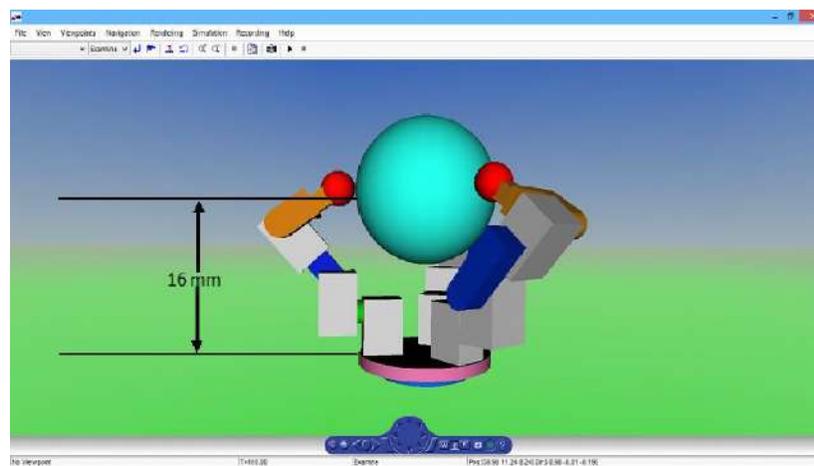


Figura 4.5 Mano robótica con la posición del punto P en 16 mm del eje Z.

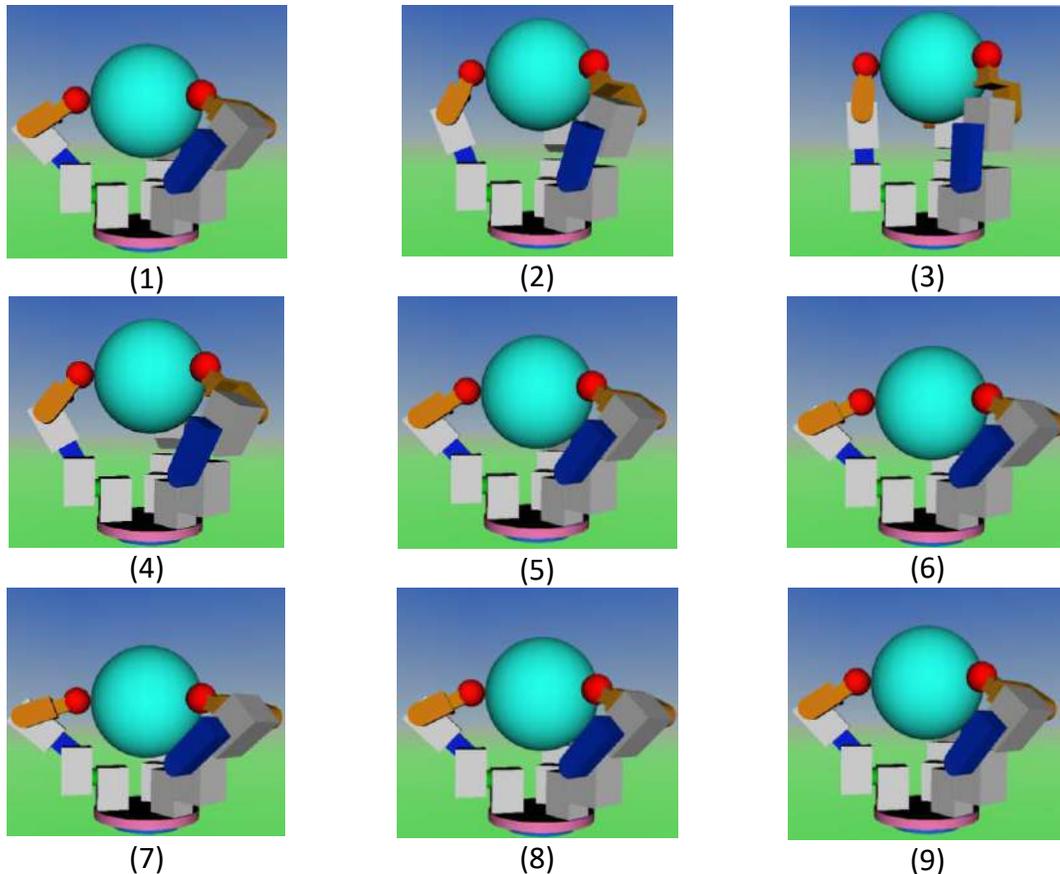


Figura 4.6 Imágenes en secuencia del movimiento de la mano robótica simulada, con una señal en cero los ejes X y Y, pero en el eje Z con una señal senoidal de amplitud de 3 mm y un offset de 16 mm

4.2 Pruebas generales de funcionamiento en físico

Con las pruebas ya realizadas en la simulación, y la comprobación de que el sistema de ecuaciones de la mano robótica funciona en un entorno virtual, es necesario llevar al mundo real las prueba de la mano robótica ya ensamblada con sus servomotores, por lo que para poder controlar los actuadores acorde con lo establecido en las ecuaciones de la cadena cinemática de cada dedo, es necesario crear una interfaz y esta misma ordene a los servomotores los movimientos que deberán hacer para que la mano robótica pueda manipular el objeto que está sosteniendo.

4.2.1 Interfaz de usuario

La interfaz es la forma en la que el usuario podrá mandar órdenes a los servomotores para que estos realicen la acción de manipular un objeto determinado, dependiendo de lo que la interfaz pueda realizar, esto mismo reflejara lo que la mano robótica puede hacer. En esta interfaz para la mano robótica se optó por un diseño en el cual las manipulaciones son sencillas, esto último no necesariamente quiere decir que la arquitectura de esta mano robótica no sea capaz de realizar manipulaciones más elaboradas, sino que, dependiendo de lo que la interfaz pueda crear será lo que esta mano robótica podrá hacer. La interfaz que se utiliza para controlar los servomotores de esta mano robótica esta creada en el entorno de desarrollo de Matlab, la Figura 4.7 muestra como es la interfaz de control.



Figura 4.7 Interfaz de control de la mano robótica.

El funcionamiento de la interfaz mostrada en la Figura 4.7 está dividido en dos fases, en la primera fase se encuentran involucradas unas cuantas cosas: las dos gráficas y el grupo de botones de la izquierda; en cuanto a la segunda fase, en esta se encuentran involucrados: la gráfica de la derecha, el grupo de botones de la derecha, y en ocasiones la gráfica izquierda que se relacionara con unos cuantos botones del grupo de la derecha.

El funcionamiento interno de la interfaz para calcular los movimientos que deben realizar los servomotores se basa en tener ya definidos parámetros iniciales, los cuales son: primeramente el tamaño del objeto, en este caso el objeto a manipular es esférico por lo que se tiene que asignar el diámetro del objeto; el segundo parámetro es la posición en la cual el objeto será sujetado antes de manipular, este dato siempre se encontrara en eje Z; y tercer y último dato es la cantidad de veces que se repetirá el proceso de manipulación. Las entradas de los valores de estos tres primeros datos se encuentran en el recuadro izquierdo, en un

subgrupo de datos llamado “datos iniciales”, esto mismo se observa en la Figura 4.7.

En cuanto al otro subgrupo de datos, en el recuadro izquierdo de la interfaz, son simplemente los valores de las amplitudes de señales senoidales, al observar la Figura 4.7 se puede ver que las amplitudes que pueden ser modificadas son las señales de los ejes X y Z, esto porque el diseño de la interfaz es simple en cuanto a la cantidad de movimientos que puede calcular, pero esto no quiere decir que la arquitectura de la mano robótica no sea capaz de adaptarse a movimientos más complejos. Con todo lo anteriormente mencionado, la idea detrás de esta interfaz es mostrar el funcionamiento de la mano robótica de la forma más sencilla, pero sin dejar en cuenta que esta misma tiene la posibilidad de hacer cosas más complejas.

En cuanto a los datos que la mano robótica que esta interfaz puede manejar de manera óptima o que se encuentran en un rango adecuado para ella, son los siguientes:

- Un diámetro máximo de 20 cm y un mínimo de 1 cm.
- Una posición inicial mínima de 11 cm y la máxima dependerá del diámetro del objeto.
- La amplitud en X y Z dependerá del diámetro del objeto, pero el recuadro izquierdo se mostrará un mensaje en rojo sin se necesita modificar alguno de estos dos valores.
- Una repetición como mínimo.

Instrucciones de la interfaz

La forma en que el usuario podrá hacer uso de la interfaz es la siguiente:

- 1) Los primero que el usuario deberá introducir es el *diámetro* del objeto esférico a manipular; la *posición inicial* en la que se encuentra el objeto, este dato tiene como referencia la base de la mano robótica, en la Figura 4.8 se muestra desde donde exactamente se obtendría dicho dato, el cual viene siendo una distancia desde la base hasta el centro del objeto; y por último las *repeticiones* del movimiento de manipulación, que es un dato en el cual siempre debe tener como valor mínimo un 1, de lo contrario no se podrá ejecutar el cálculo de los movimientos angulares de los servomotores.

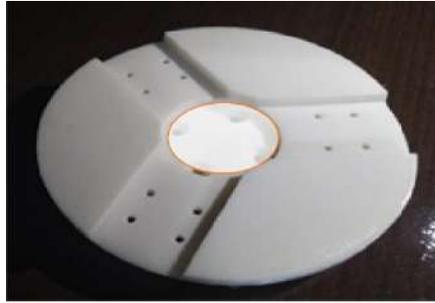


Figura 4.8 Base de la mano robótica con el señalamiento de un círculo naranja de donde se toma la referencia de la posición inicial en la interfaz.

- 2) En este paso el usuario deberá introducir un valor mayor a cero en la amplitud de la señal en X, Z o ambas, porque de lo contrario la mano robótica realizaría una simple sujeción del objeto, aunque sea una de las capacidades de la mano robótica, la interfaz ya posee un botón dedicado a simplemente sujetar.
- 3) Una vez con los datos iniciales y las amplitudes definidas, el siguiente paso es presionar el botón verde “GRAFICAR”, pero en momento que el usuario presione este botón pueden ocurrir dos escenarios: el primero es que muestre en la gráfica izquierda las señales con las amplitudes definidas y en la de derecha el movimiento del centro del objeto en tres dimensiones, y en el recuadro de abajo mostrando un mensaje en verde, en el cual hará alusión de que se logró graficar exitosamente; en el segundo escenario puede suceder que no muestre nada en ninguna de las gráficas, pero si un mensaje en rojo en el recuadro izquierdo indicando algún problema por el cual no se pudo graficar y brindando una posible solución para remediar el problema. El problema a solucionar siempre estará relacionado con un dato que no permitiría a la mano robótica comportarse adecuadamente, ya sea porque el objeto sea muy grande, que la posición inicial este fuera del rango de sujeción, o que el movimiento del objeto obligaría a los servomotores a tener valores que no pueden tener debido a las dimensiones de sus falanges; en la Figura 4.9 se muestra los dos escenarios que pueden suceder.

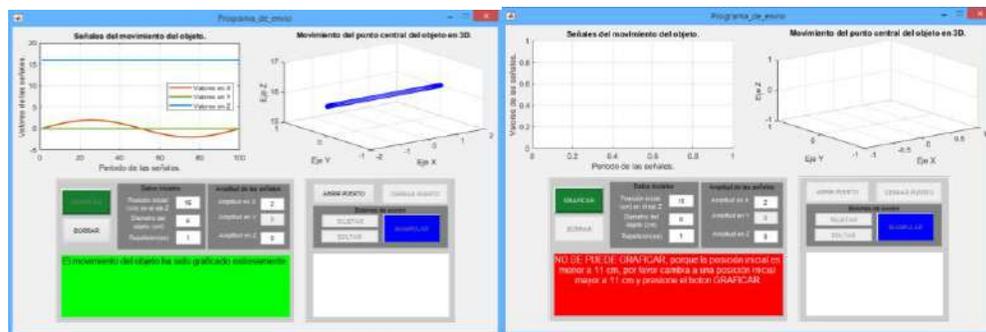


Figura 4.9 Resultados del uso de la interfaz, a la izquierda se muestra un ejemplo exitoso y en la derecha uno fallido.

- 4) Después de obtener las gráficas se habilitará el botón “ABRIR PUERTO” en el recuadro de la derecha, que es donde se encuentran los botones de acción de la mano robótica, abajo este grupo de botones se encontrara un recuadro, en el que se leerá un mensaje de que la computadora se estará comunicando con el controlador de los servomotores, si resulta que la comunicación es un éxito aparecerá un recuadro en color verde con un mensaje, de que la mano robótica se encuentra lista para realizar un acción, sino aparecerá un recuadro en rojo que indicara algún problema de comunicación con la interfaz y de los servomotores, y pedirá que se solucione el problema, en la Figura 4.10 se observa estas dos situaciones.

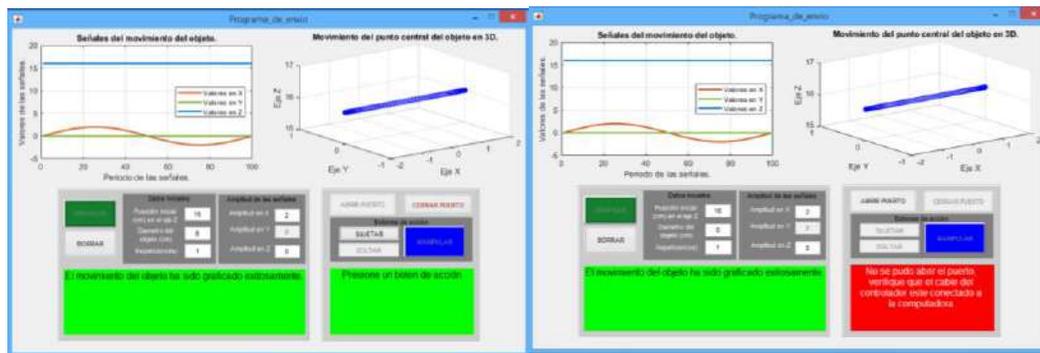


Figura 4.10 Resultados de la comunicación de la interfaz con el controlador de los servomotores, en el lado izquierdo se muestra una comunicación exitosa y en el derecho una fallida.

- 5) Con la comunicación exitosa entre la interfaz y los servomotores, se activarán los botones de acción de la interfaz. El primer paso es presionar el botón “SUJETAR”, el cual ordena a la mano robótica sujetar el objeto en la posición inicial indicada previamente, una vez sujetado el objeto se habilitará el botón “SOLTAR”, pero en caso de que no se logre sujetar correctamente el objeto él usuario puede presionar el botón “SOLTAR” e intentar hasta que el objeto este correctamente sujetado.
- 6) Con el objeto sujetado también se habilita el botón “MANIPULAR” el cual manda los valores a los servomotores para que realicen le movimiento de manipulación, que se encuentra representado en la gráfica de la derecha, tal y como se ve en la Figura 4.10.
- 7) Después de que la mano robótica realice la manipulación, se habilitara el botón de “SOLTAR”, pero también se encontrarán habilitados las amplitudes de las señales en X, Z y el número de repeticiones, todo esto si es que la mano robótica se encontrara sujetando el objeto, aunque el valor de la posición inicial estará deshabilitado.
- 8) Después de que se realice la manipulación y se presione el botón “SOLTAR”, ya sea para modificar los datos iniciales, cambiar de objeto, etc., aparecerá habilitado el botón “CERRAR PUERTO”, el cual detiene la comunicación

entre la interfaz y los servomotores, presionar este botón significaría que ya no se requiere realizar alguna manipulación y que el usuario desea apagar el programa.

Los pasos anteriormente mencionados son necesarios para hacer un adecuado uso de la interfaz, y para procurar un mantener los servomotores en óptimas condiciones sin tener que exigirles demasiado y de esta forma no dañarlos.

Convertidor de comunicación

Este instrumento es un elemento que permite comunicar la interfaz y los servomotores, se encarga de recibir las instrucciones de lo que deben hacer los servomotores y enviárselos a estos. El convertidor que se utiliza en la mano robótica es el U2D2, el cual es un convertidor de comunicación USB de pequeño tamaño que le permite a la interfaz controlar y operar servomotores DYNAMIXEL, en la Figura 4.11 se muestra el convertidor U2D2, en el cual se utiliza el puerto con tres pines para controlar los servomotores, este puerto utiliza el protocolo TTL (Time To Live), debió a que los servomotores seleccionados tienen este puerto de este protocolo para ser controlados.

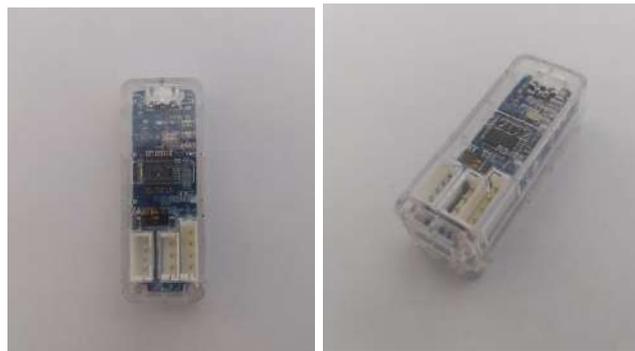


Figura 4.11 Convertidor de comunicación U2D2.

El convertidor no es lo único que se necesita para hacer funcionar los servomotores, estos también requieren de una fuente de alimentación para ser activado y de esta forma realicen su trabajo. Dado las características de los servomotores, estos requieren de un voltaje determinado para ser activados y estar en funcionamiento, el convertidor U2D2 hace uso de un centro de poder, el cuál es la placa mostrada en la Figura 4.12, esta tarjeta tiene la capacidad de suministrar un voltaje de 3.5 V ~ 24 V con una corriente máxima de 10.0 A.

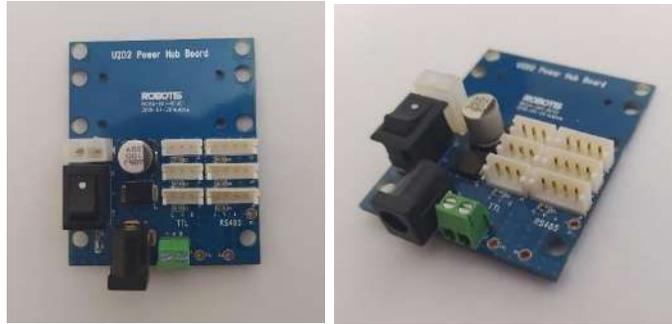


Figura 4.12 Tarjeta centro de poder U2D2.

En cuanto a la forma en que la deben estar conectados los servomotores con el centro de poder del U2D2, es necesario recurrir a una extensión de puertos de tres pines, debió a que los servomotores pueden ser conectados en serie, pero el centro de poder solo tiene tres puertos TTL, el primero para la comunicación con el convertidor U2D2 y los otros dos para comunicarse con los servomotores, y aunque de una cierta forma todos los servos puedan ser conectados uno tras otro, usando solo un puerto del centro de poder, esto resulta en una forma muy inconveniente y por eso se utiliza una extensión PCB 3P de tres pines con seis puertos para que cada dedo utilice un puerto de esta extensión, Figura 4.13 lo muestra la extensión, de esta forma es más fácil la conexión los dedos paralelamente a la centro de poder, además de que cada dedo tendrá conectados sus servos en forma serial.

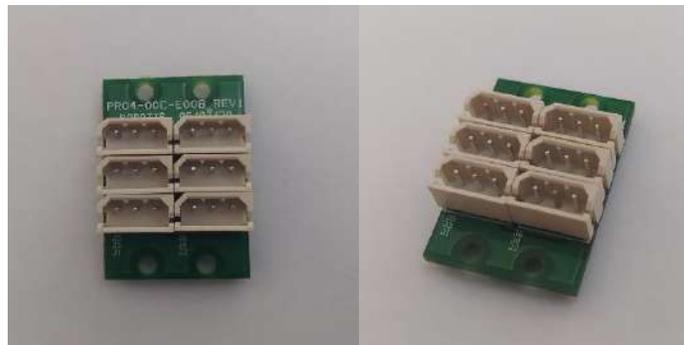


Figura 4.13 Extensión de PCB 3P. de tres pines con seis puertos.

La forma en como todo debe ser conectado es mostrado en diagrama de la Figura 4.14, en este diagrama primero muestra que la computadora, en donde se encuentra la interfaz, debe ser conectada con el controlador U2D2 por medio de un cable USB, después este se conecta con la tarjeta centro de poder U2D2 por medio de un cable de tres pines, esta tarjeta también requiere de que se presione un botón de encendido para que suministre energía por los puerto, además de que esta tarjeta debe estar conectada a la red eléctrica mediante una caja de reducción de voltaje; continuando con la conexión de todo el sistema, el siguiente componente es la extensión PCB 3P conectada con un cable de tres pines y, esta a su vez es seguida por todos los servomotores de la mano robótica conectados con un cable de tres pines. Todo lo anteriormente mencionado sería la forma en que se debe

conectar el sistema de la mano robótica para que funcione correctamente, sino se encuentra conectado de esta manera el sistema no funcionara.

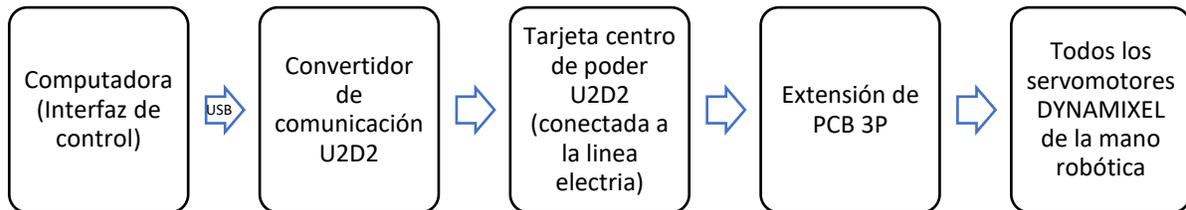


Figura 4.14 Diagrama de conexión de todo el sistema de control de la mano robótica.

Costo total de la mano robótica

En lo que concierne al costo de la mano robótica, este se muestra en la Tabla 4-1 que se encuentra aquí abajo, en ella se observa los nombres de los elementos que conforman la mano robótica con excepción de los tornillos, ya que estos pueden tener un costo que puede variar, y puede que no sea muy una variación muy costosa, además que el dólar es utilizado como moneda en cuanto al costo de cada elemento.

Tabla 4-1 Tabla de los costos de los elementos que conforman la mano robótica.

Elementos de la mano robótica	Costo en USD
Paquete de 6 Dynamixel AX-12	224.50
1 Dynamixel XM430-W2120-T	229.90
1 Dynamixel XM430-W2120-T 1PCS	229.90
1 convertidor de comunicación U2D2	26.90
1 tarjeta centro de poder U2D2	15.90
10 cables de conexión	13.40
Todas las piezas impresas en 3D	102.07
Costo total	845.47

El precio total de la mano robótica mostrado en la tabla de arriba es una estimación, que puede aclarar cual valor monetario puede llegar a tener. También se puede decir que las manos robóticas vendidas en el mercado, como las que se muestran en la Figura 4.15, pueden llegar a tener un precio mayor, el cual no significa que el costo de las manos comerciales demerite el valor de este proyecto o porque este tenga un precio menor se le dé una mayor estima, porque el enfoque de este proyecto es de un ámbito más didáctico.

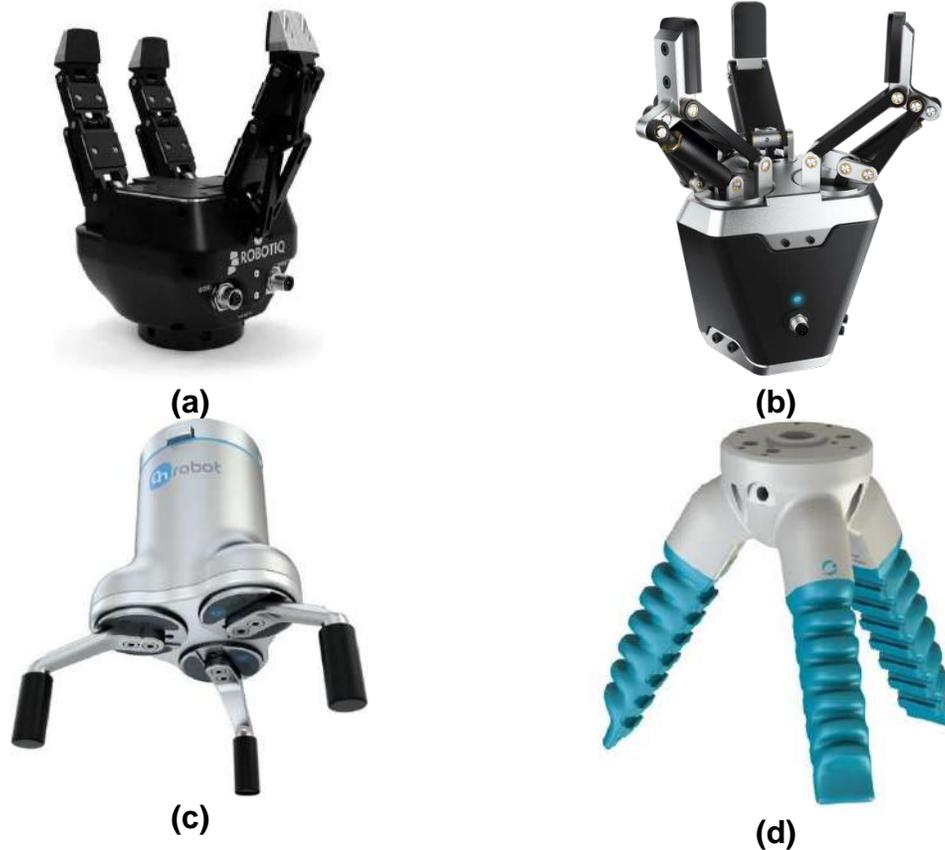


Figura 4.15 Manos robóticas comerciales: **a)** Robotiq 3-Finger Adaptive Gripper (Robotiq 3-Finger Gripper - Think Solutions LLC, 2020), **b)** DH Robotics 3 Finger Adaptive Gripper (DH Robotics 3 Adaptive Gripper, Lifting ... - IndiaMART, 2020), **c)** 8203 OnRobot Three Finger Gripper (8203 OnRobot Three Finger Gripper 3FG15 - EMI Corp, 2020), **d)** Three Finger Centric SoftGripper (Three Finger Centric SoftGripper - Developer - Soft Gripping, 2020).

Con lo anteriormente dicho, se puede proceder con una cierta aclaración sobre las manos robóticas mostradas en la Figura 4.15, la cuales tienen diferentes precios: comenzando por el inciso a), esta tiene un precio de 18 000.00 USD; el inciso b) un precio de 12 236.96 UDS estimado; el c) un precio de 4 360.00 USD; y por último el d) con un precio de 853.15 USD. Lo que se puede observar de estos precios, es que la mano robótica del inciso a) es la más cara, y con una mayor capacidad de maniobrabilidad en cada dedo, esta es seguida por la del inciso b), c) y d) respectivamente, aunque la que tiene la mayor capacidad de carga es la del inciso d) con 15 kg, después la a) con 10 kg, la b) con 2 kg y por último la d) con una capacidad de carga estimada de 1 kg. El precio de cada mano no determina cual es la mejor todas, ya que el diseño de una puede cumplir un propósito distinto a las otras, y el precio solamente pueda tenerse en consideración a la hora elegir si es asequible acorde al trabajo que va a realizar por determinada mano robótica.

4.2.2 Pruebas de manipulación

Esta sección abarca las pruebas de movimiento de la mano robótica construida físicamente, las pruebas se realizan mediante los movimientos del centro del objeto sobre un plano que atraviesa perpendicularmente la base de la mano robótica, tal y como se muestra en la Figura 4.16. En dicha imagen se observa que el plano también atraviesa perpendicularmente un dedo de la mano robótica, el cual es el que solamente puede moverse en el plano bidimensional, mientras que los otros dos dedos se pueden mover en tres dimensiones, con lo mencionado anteriormente el movimiento del centro del objeto se efectuara en el plano como el de dicha figura, pero las pruebas consisten en el monitoreo de la posición angular de los servomotores, ya que son los que permiten manipular a un objeto.

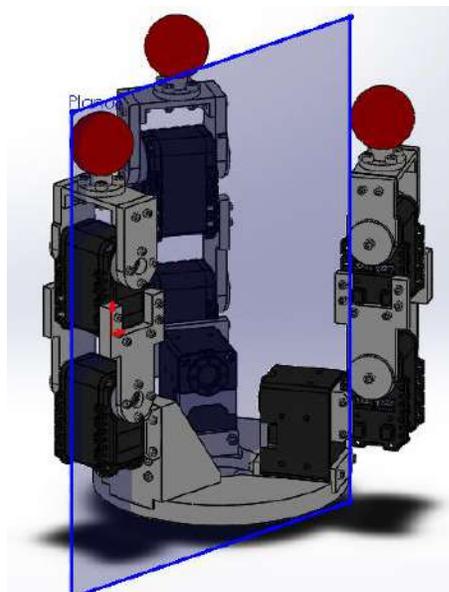


Figura 4.16 Plano de movimiento del objeto.

En las pruebas a realizar sobre los servomotores solamente se tomará en cuenta la posición que estos deben tener para realizar la manipulación, dejando de base una misma velocidad angular para cada uno de los servomotores, por lo que solamente se comparara los datos de posición que se le envíen a los servomotores y los datos que estos regresen. Los servomotores tienen la capacidad de recibir o regresar valores, siendo los valores para el servomotor AX-12A entre 0 a 1023, mientras que los valores para el servomotor XM430-W210 se encuentran entre 0 a 4095; dando como resultado una resolución de 10 bits para el AX-12A y 12 bits para el XM430-W210.

Pruebas del comportamiento de los servomotores de un dedo

En las pruebas de la mano robótica el monitoreo de los servomotores se hace solamente para los tres servos de un dedo de la mano robótica, en concreto el dedo de la derecha que se muestra en la Figura 4.17, el cual es que más cercano a la computadora en dicha figura. El monitoreo se hace mediante un envío de cadenas de datos sincronizados, esto se refiere a que todos los servomotores están siendo controlados al mismo tiempo debido a que cada uno está recibiendo los datos que le corresponde, pero esto mismo puede hacer que la respuesta de la mano robótica sea tardía o que el movimiento de manipulación tarde concretarse, debido a que los datos se están enviando desde la interfaz de la computadora a los servomotores y esto ralentiza el proceso de manipulación.

El análisis se hace solamente en un dedo, debido a lo mencionado en el párrafo de arriba, y que además en un solo dedo como el seleccionado en la Figura 4.17 poseen tres grados de libertad y sus actuadores tienen los dos tipos de servomotores utilizados en la mano robótica, con ello el monitoreo de un solo dedo puede brindar una cierta idea del comportamiento general de la mano robótica, debido a que las respuestas de los otros servomotores en los otros dos dedos pueden ser similares a los monitoreados en el dedo elegido.

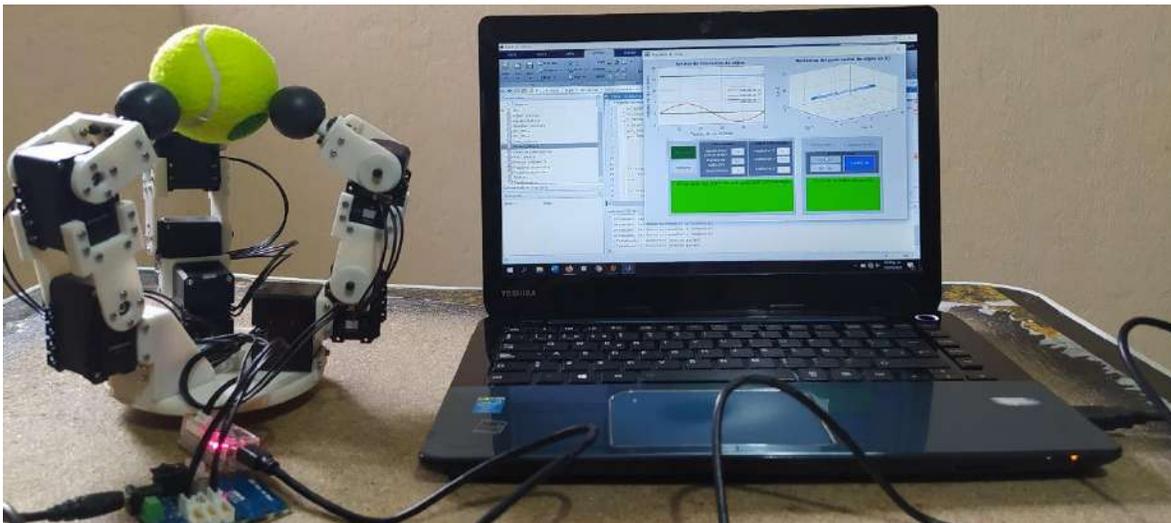


Figura 4.17 Pruebas de la mano robótica.

El análisis consta en poner a prueba los actuadores del dedo en diferentes movimientos de posición del centro del objeto, los movimientos que ponen a prueba la mano robótica son los siguientes: una línea en horizontal, una línea en vertical, una línea en diagonal y por último un movimiento en círculo o elipse. El monitoreo de cada movimiento se hará para los tres actuadores del dedo a seleccionado, y los datos obtenidos de cada dedo se comparan con los datos enviados mediante una gráfica. Las pruebas de movimiento son las siguientes:

- **Prueba del movimiento en una línea horizontal**

El movimiento de en una línea horizontal está representado en la interfaz de la Figura 4.18, este es el movimiento del centro del objeto de manera paralela a la base de la mano robótica, y en todas las pruebas de movimiento se utiliza una pelota de tenis con un diámetro de 6.3 cm, el cual es un dato de entrada en la interfaz.

En esta prueba el comportamiento que cada uno de los servomotores se muestra en la Figura 4.19, Figura 4.20 y Figura 4.21 para los servomotores 1, 2 y 3 respectivamente, los valores enviados a cada actuador se representan en las figuras con un línea verde y los valores con los que responden o los que llega a tener el servomotor se representan con una línea roja.

La grafica para el servomotor número 1 muestra que el comportamiento para dicho actuador es muy similar con el que debe tener, solamente mostrando algunos valores un poco fuera de lugar en los índices después del 10, 15, y concretamente en el 30 y 40, pero fuera de eso la línea roja sigue el mismo comportamiento que la verde. En cambio, para el servomotor 2 su línea roja de respuesta muestra un comportamiento distinto del que debería tener en los índices después del 10 y 15, además en los índices del 30 al 40 sus valores no son los mismos que los de la línea verde, aunque el comportamiento es muy similar. La grafica del servomotor 3 muestra en los índices 10 y 15 valores un poco alejados del que corresponde, y para los índices después del 30 los valores sufren un cierto desfase de los valores que deben tener, pero manteniendo la amplitud en dichos índices.

Los resultados del movimiento en línea horizontal muestran en algunos índices de cada grafica una cierta falta de precisión con los servomotores, pero mantienen el comportamiento que deben tener con respecto a los valores enviados a cada uno de los servomotores.

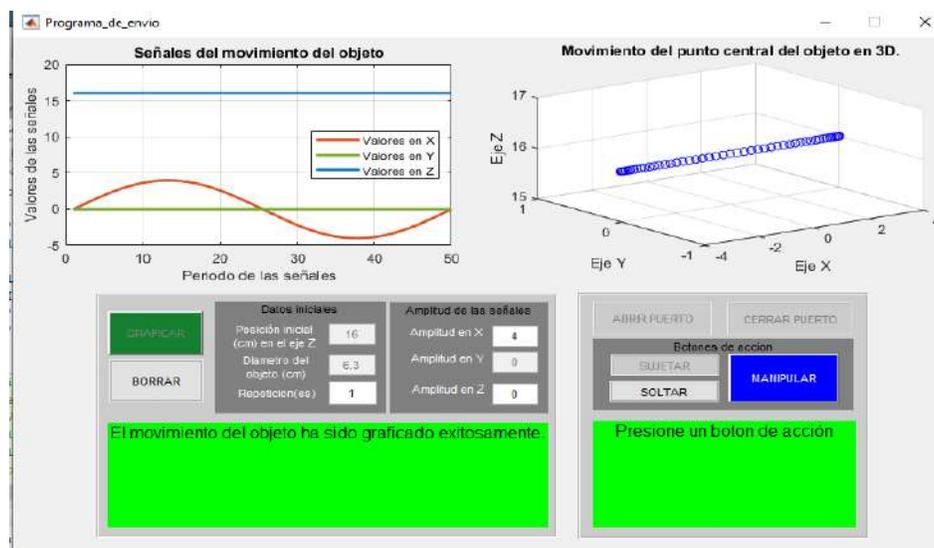


Figura 4.18 Interfaz de control con el movimiento programado en línea horizontal.

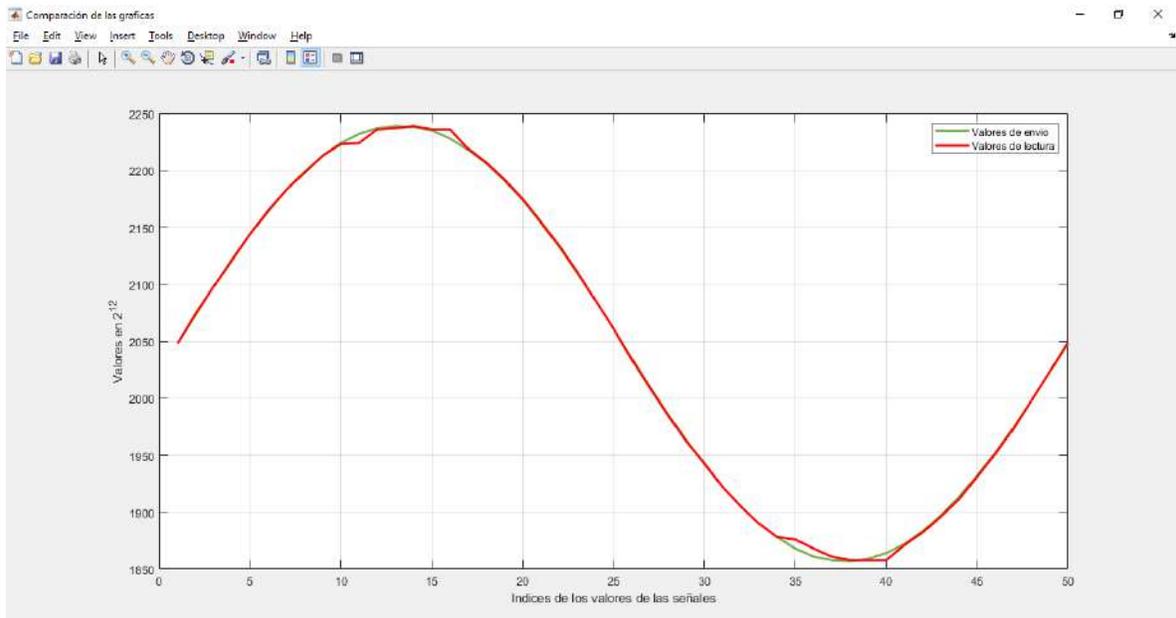


Figura 4.19 Grafica de comparación del servomotor número 1 para un movimiento en línea horizontal.

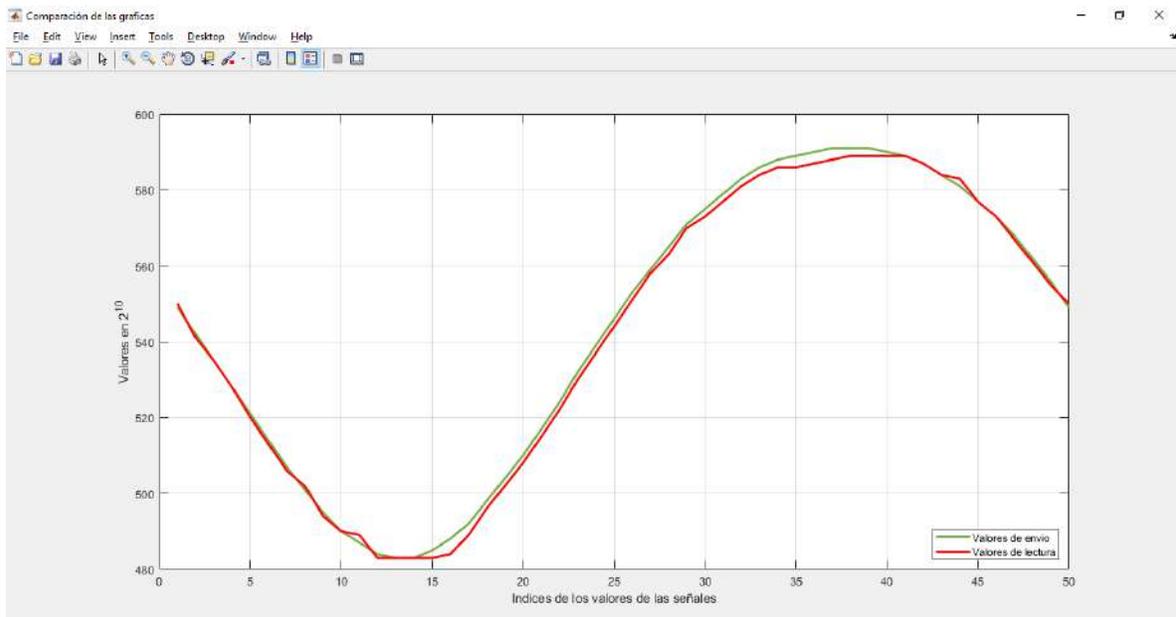


Figura 4.20 Grafica de comparación del servomotor número 2 para un movimiento en línea horizontal.

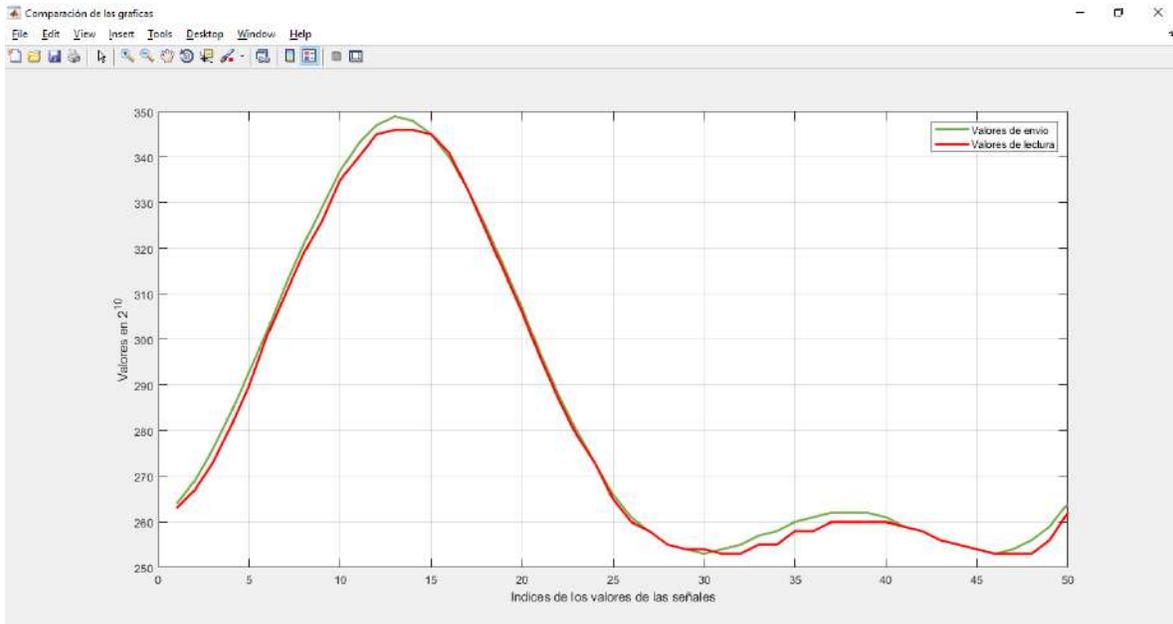


Figura 4.21 Grafica de comparación del servomotor número 3 para un movimiento en línea horizontal.

- **Prueba del movimiento en una línea vertical**

La prueba de comportamiento de la mano robótica para el movimiento de línea vertical es mostrada en la Figura 4.22. En la Figura 4.23 se observa la gráfica de comparación de los datos de envío y respuesta para el servomotor número 1, en dicha grafica se observan cambios de valor muy pequeños, debido que los valores en los que se encuentran dichos cambios son entre los valores 2048 y 2049, siendo estos los valores que debe tener este servomotor para mantenerse lo más inmóvil posible, ya que los servomotores 2 y 3 son los que está realizando la manipulación del objeto en este movimiento de línea vertical.

Con lo anteriormente dicho en el párrafo de arriba las gráficas para los servomotores 2 y 3 muestran un comportamiento muy similar entre los valores enviados con los conseguidos por dichos servomotores. La gráfica para la respuesta del servomotor 2 muestra algunos valores alejados un poco de los que deben tener, en concreto en los índices después del 15, y entre los 25 y 40. En cambio, la gráfica de respuesta del servomotor 3 solamente muestra algunos picos alejados de los valores que debe tener, por lo que su comportamiento es similar al que se espera tener.

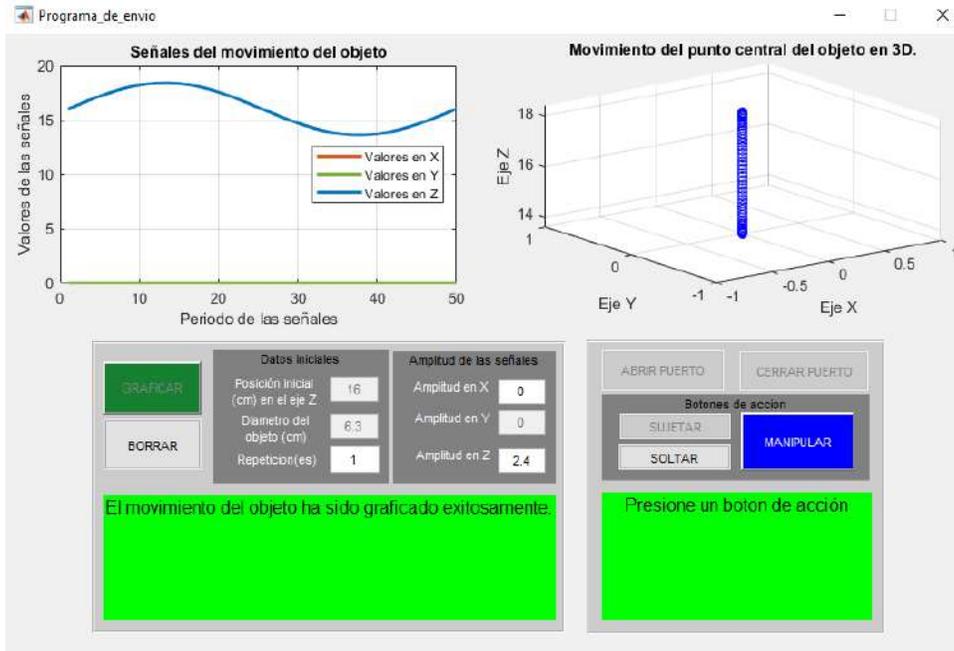


Figura 4.22 Interfaz de control con el movimiento programado en línea vertical.

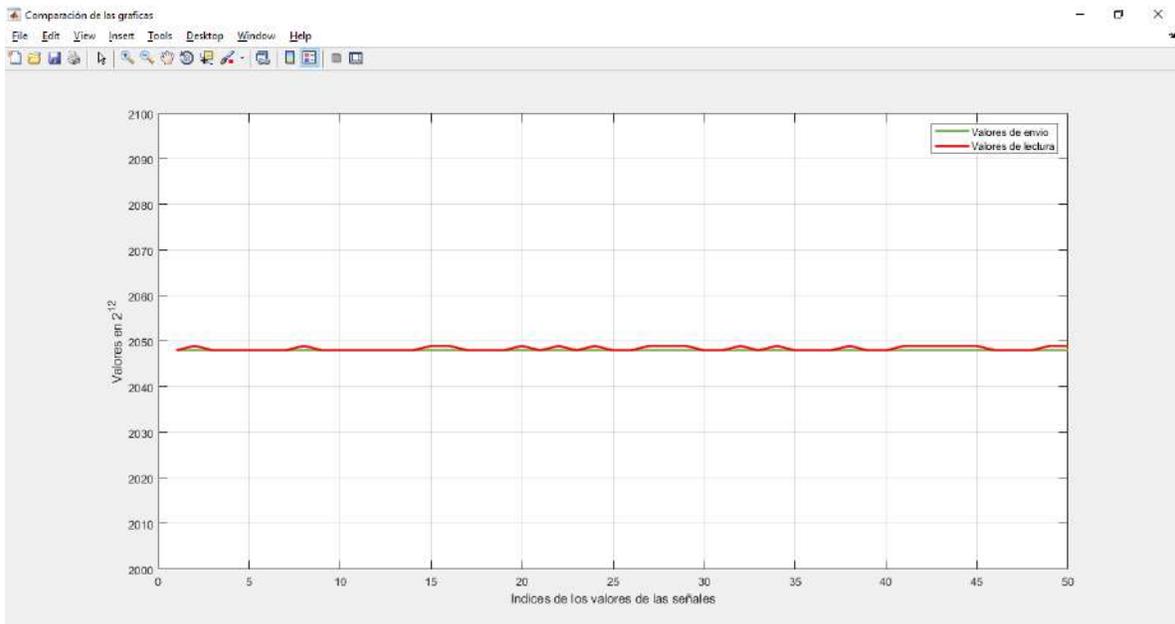


Figura 4.23 Grafica de comparación del servomotor número 1 para un movimiento en línea vertical.

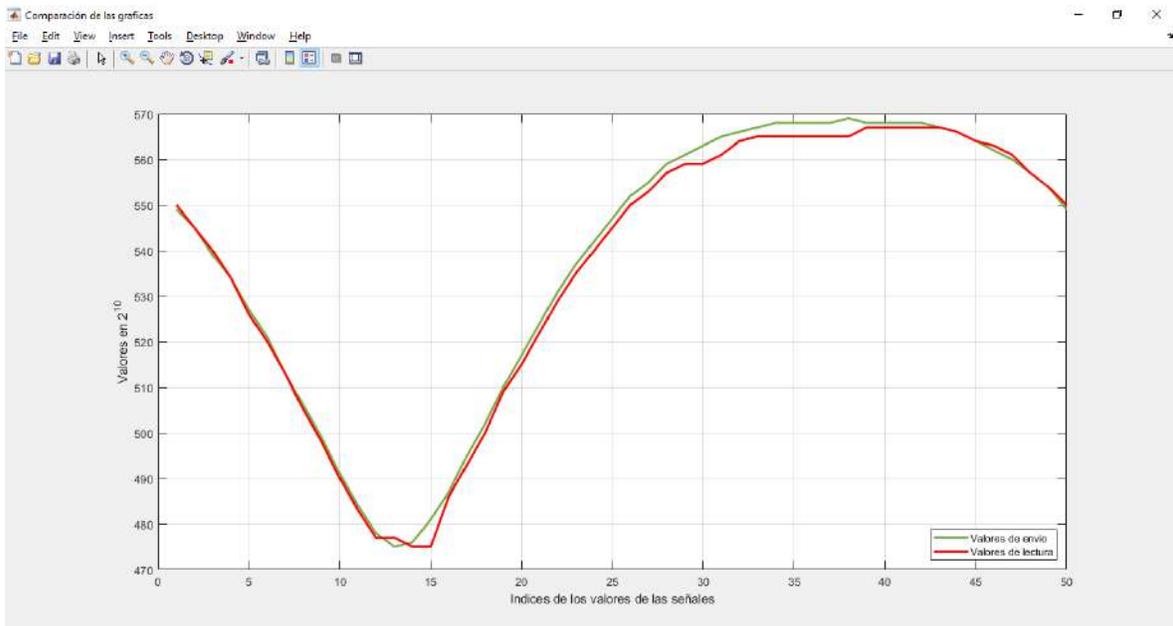


Figura 4.24 Grafica de comparación del servomotor número 2 para un movimiento en línea vertical.

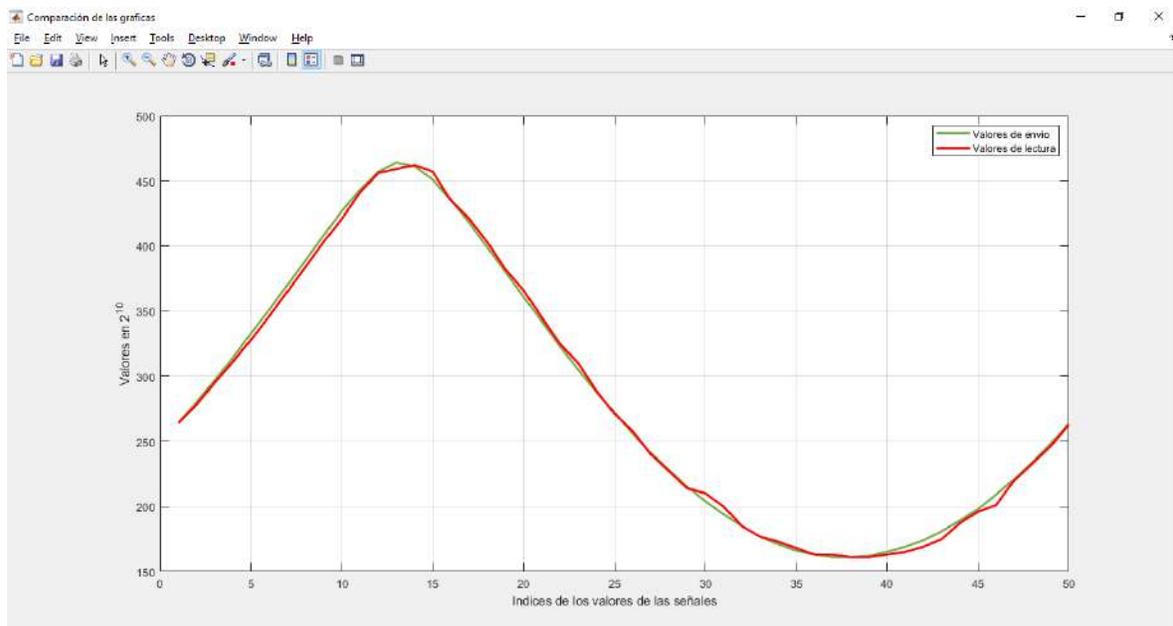


Figura 4.25 Grafica de comparación del servomotor número 3 para un movimiento en línea vertical.

- **Prueba del movimiento en una línea en diagonal**

En esta prueba de movimiento los tres servomotores tienen un movimiento o cambio de posición a diferencia de la línea vertical, las gráficas mostradas aquí bajo muestra el comportamiento con el que responden los servomotores. La gráfica para la respuesta del servomotor 1 muestra un comportamiento muy similar a la que debe tener, con excepción de los valores entre los índices 30 a 35, y 40 a 45. La gráfica en el servomotor 2, es la que muestra sus valores de respuesta un periodo más largo alejados del que deben tener en los índices entre 30 y 40, pero conservando la similitudes iniciales y finales. La grafica de respuesta para el servomotor 3 es muy similar a la que debe ser, solamente entre el índice 10 y 15 se encuentran unos valores que el servomotor no alcanza a obtener.

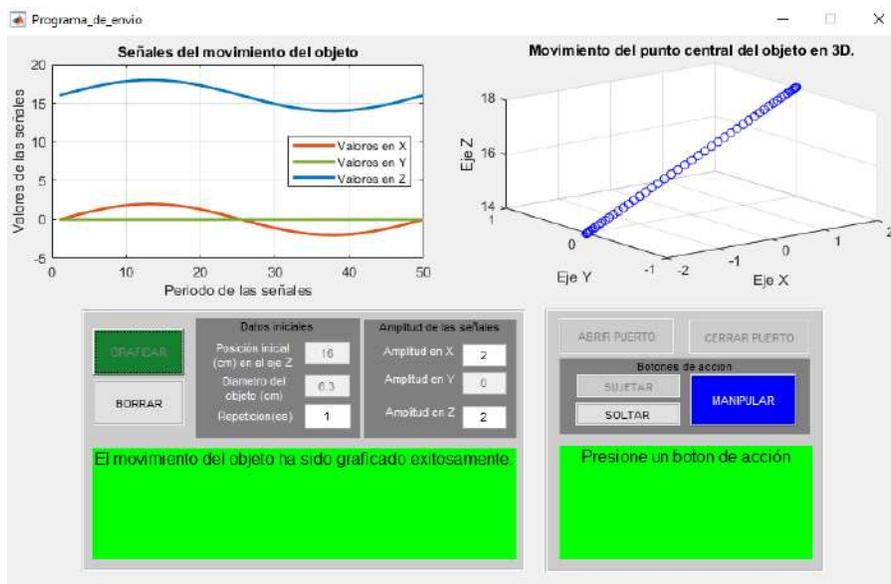


Figura 4.26 Interfaz de control con el movimiento programado en línea diagonal.

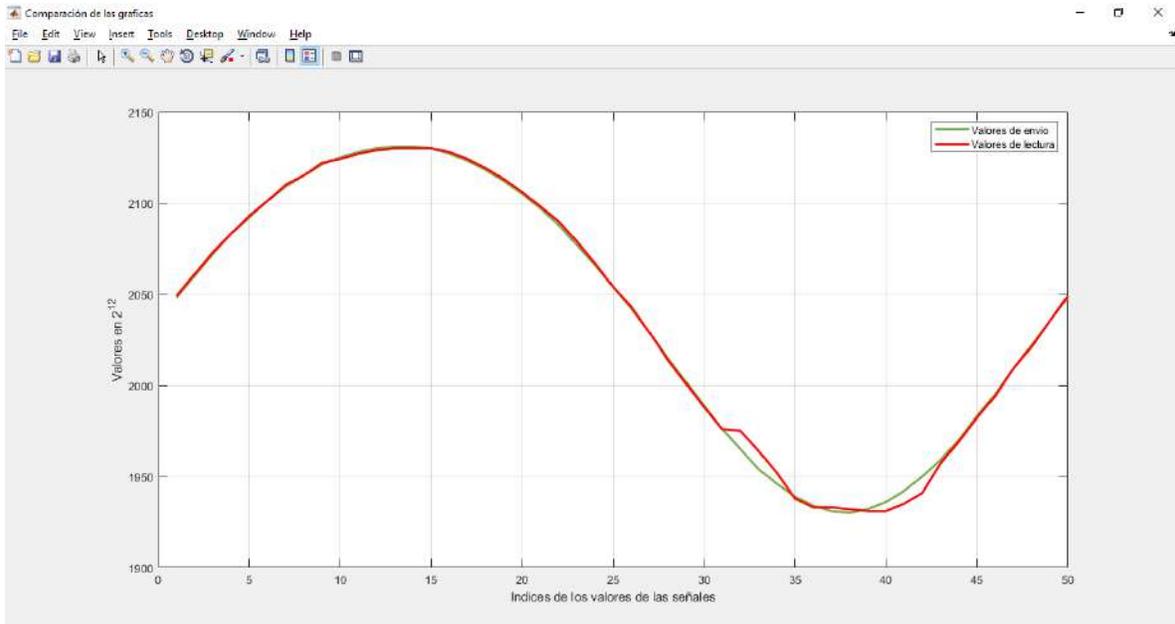


Figura 4.27 Grafica de comparación del servomotor número 1 para un movimiento en línea diagonal.

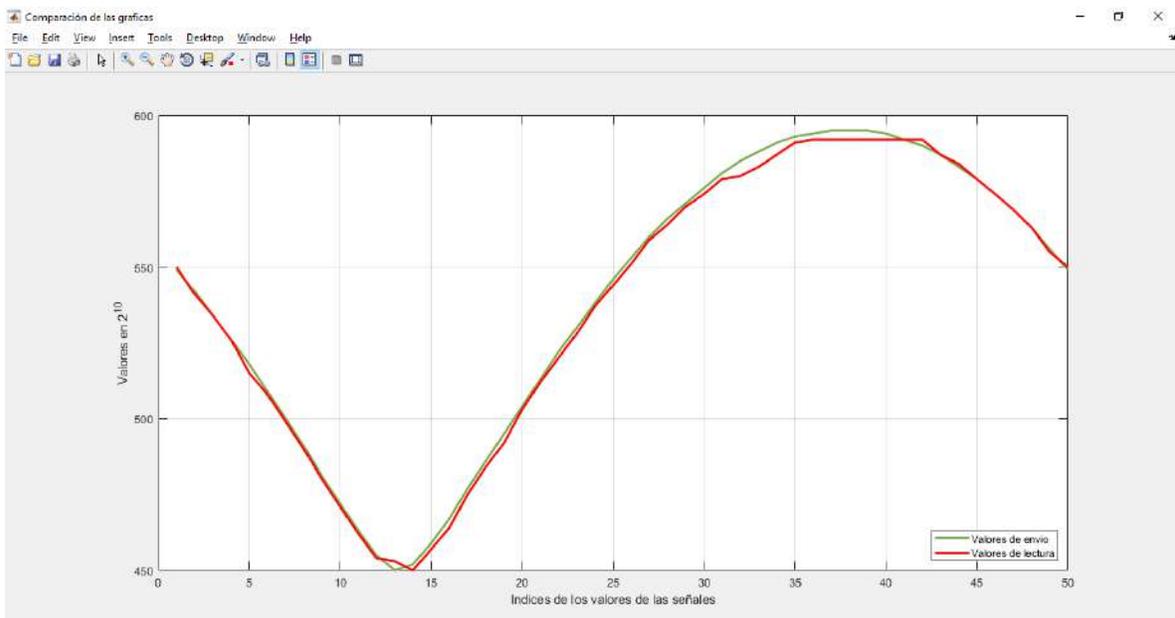


Figura 4.28 Grafica de comparación del servomotor número 2 para un movimiento en línea diagonal.

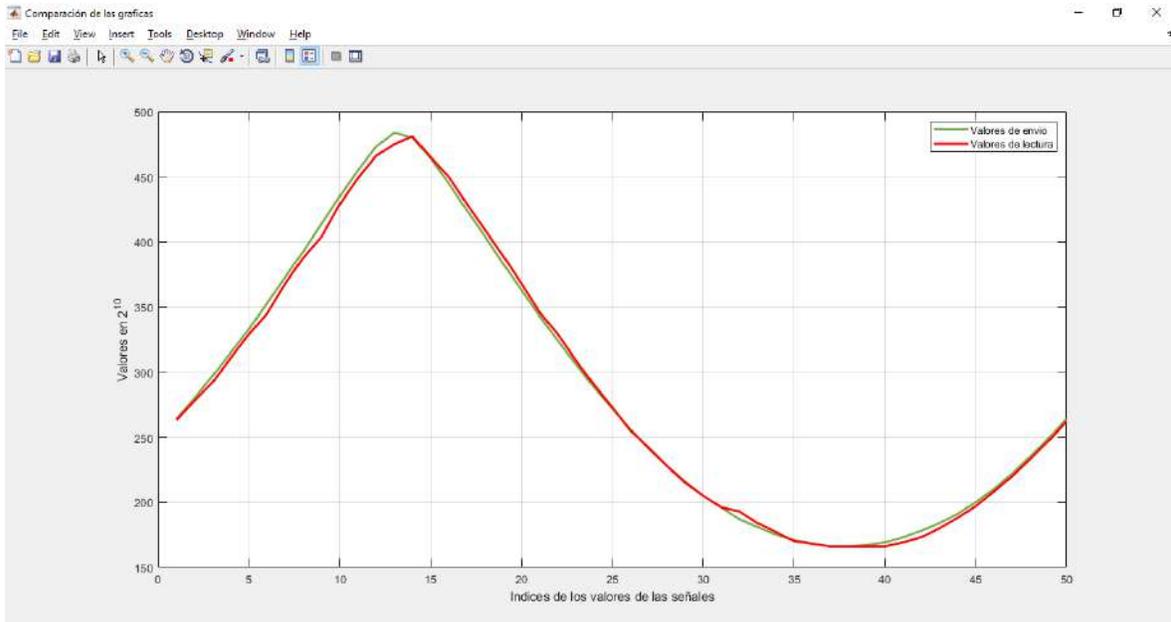


Figura 4.29 Grafica de comparación del servomotor número 3 para un movimiento en línea diagonal.

- **Prueba de un movimiento elíptico**

La prueba para un movimiento en elipse se muestra en la interfaz de la Figura 4.30, este movimiento inicia desde el punto más alto mostrado en la imagen de la derecha de la interfaz. En esta prueba al igual que las anteriores se comparan las gráficas de los valores enviados y los valores de respuesta de los servomotores.

La grafica de los valores con los que responde el servomotor número 1 muestra un comportamiento muy similar al de la gráfica de los valores que se le envían, solamente en unos índices después del 10 y antes del 35 se observa picos que se alejan de los valores que debe tener el servomotor 1. En la gráfica de respuesta del servomotor número 2 se observa que existe una gran separación después del índice 10, pero por los demás valores mantiene cercanos sus valores de respuesta a los valores enviados, por lo que su comportamiento es cercano al que se espera. La grafica de los datos de respuesta del servomotor 3 es muy similar a la de los datos enviados, con la excepción de dos separaciones en los índices después del 10 y antes del 35, con todo los demás mantiene el comportamiento esperado para hacer el movimiento que se le envía.

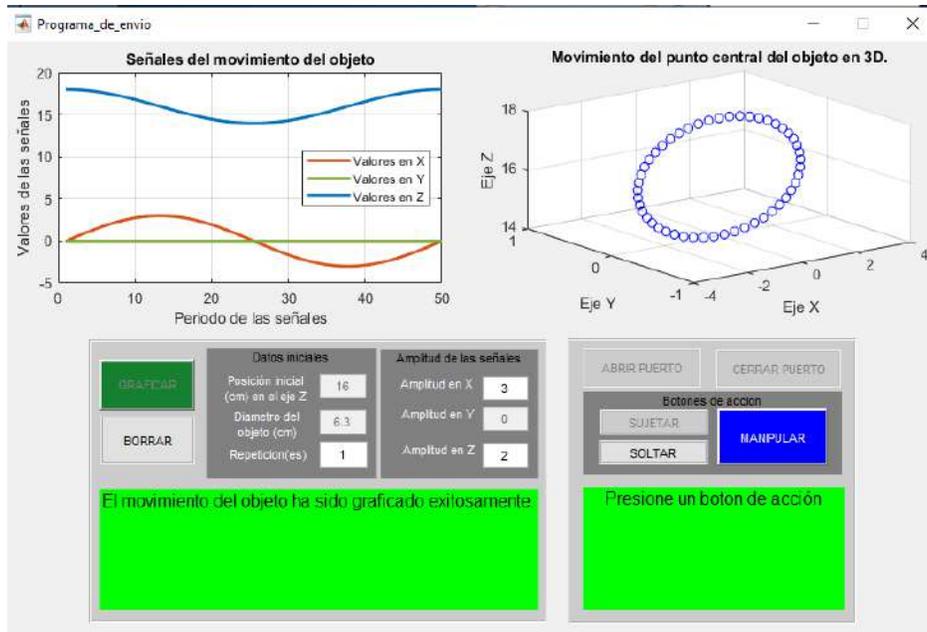


Figura 4.30 Interfaz de control con el movimiento programado en elipse.

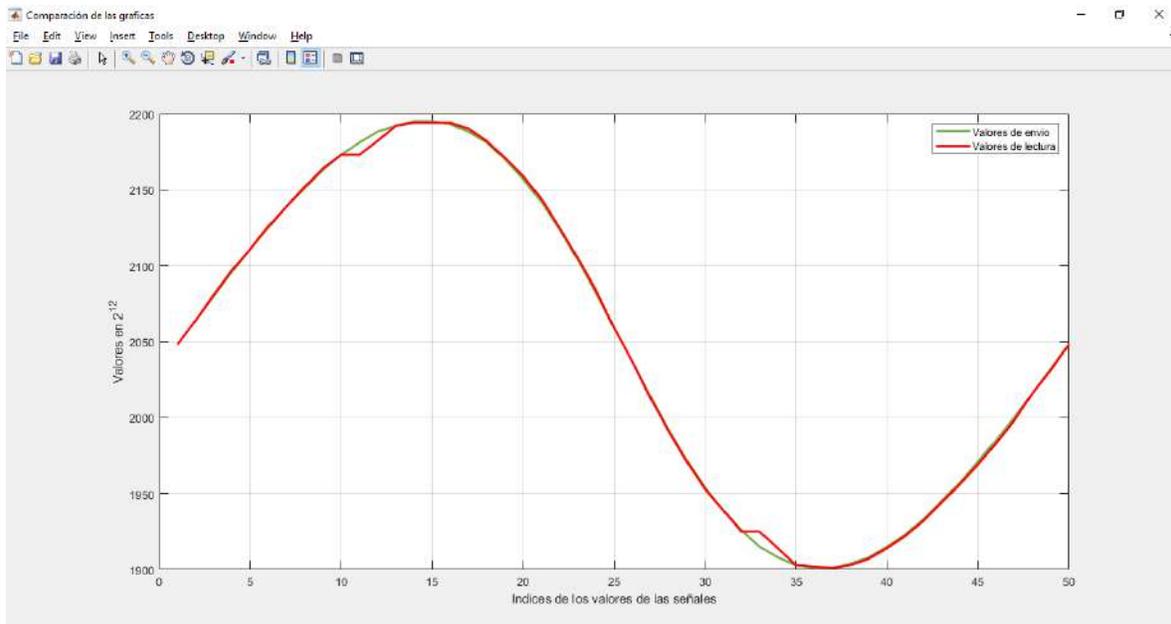


Figura 4.31 Grafica de comparación del servomotor número 1 para un movimiento en elipse.

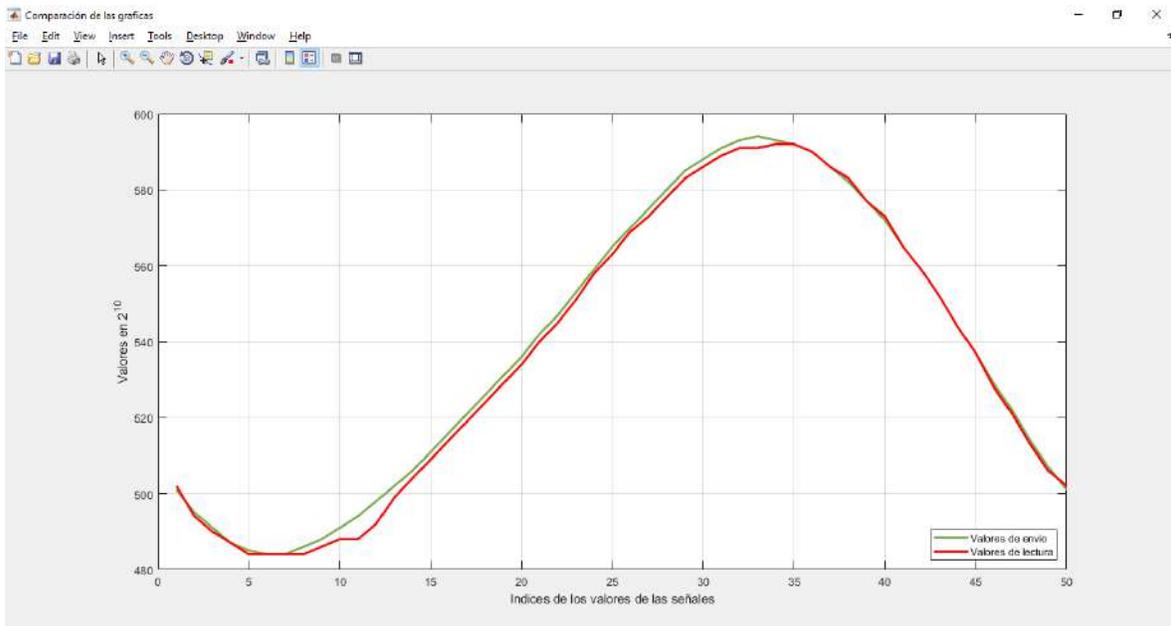


Figura 4.32 Grafica de comparación del servomotor número 2 para un movimiento en elipse.

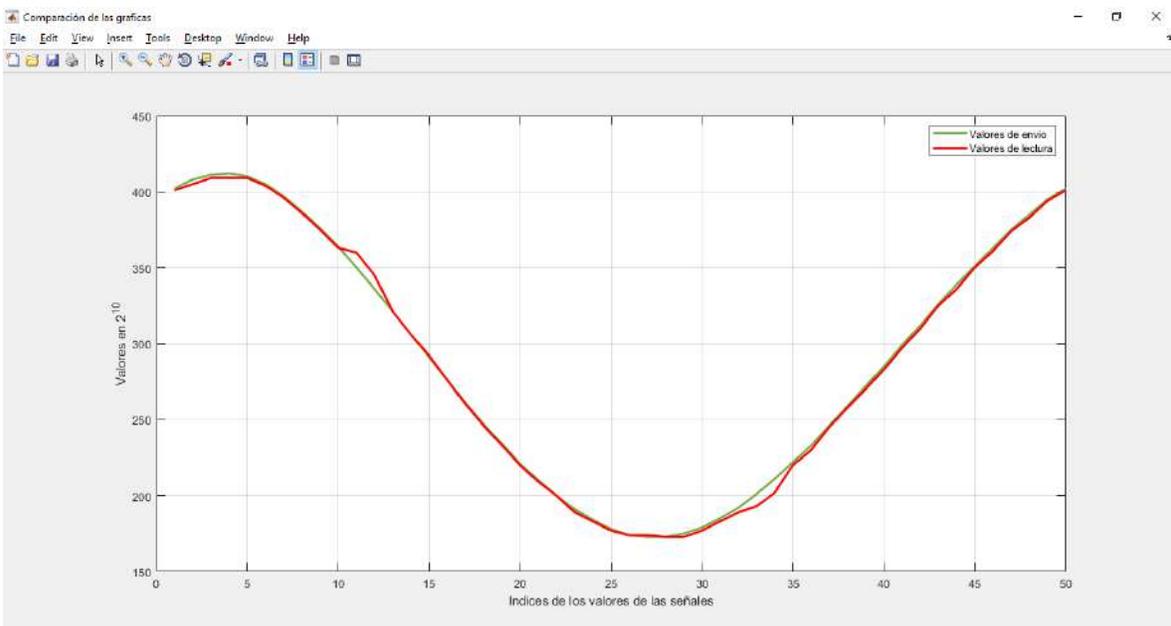


Figura 4.33 Grafica de comparación del servomotor número 3 para un movimiento en elipse.

Las gráficas en estas pruebas presentan algunos errores en el alcance de los servomotores para obtener el valor indicado, pero en general los movimientos de los servomotores son adecuados para que la mano robótica logre una manipulación exitosa. Los errores que se observan en las gráficas de los servomotores pueden que se deban a un mal cambio de posición, motivado por el hecho de que siempre se mantiene una misma velocidad al cambiar de un punto a otro y, por ende, provocando que los servomotores no sean capaces de obtener los valores debidos. Lo anterior también puede provenir del hecho de que la mano robótica solamente

tiene modelado las condenas cinemáticas de sus dedos y no las cadenas dinámicas, pero aun así con el modelo cinemático es suficiente para tener en mente el funcionamiento más básico de este tipo de robots paralelos.

Ventajas y desventajas del proyecto

La mano robótica construida en este trabajo contiene carencias como el hecho de que no posee una gran capacidad de ejecución de fuerza como la vista en uno de los trabajos referenciados; también del hecho de que carece de una retroalimentación cuando los dedos están sujetando el objeto y, la única forma de saber que la mano robótica está realizando dicha acción es mediante la vista del usuario de la interfaz; además se le puede considerar una desventaja el no poseer de un sistema de control, aunque esto esté más ligado al modelo dinámico de la mano robótica que al cinemático, siendo este último el que se utiliza para construir la mano robótica. En cuanto a la interfaz, esta no posee una gran amplitud de movimientos manipulación a realizar sobre el objeto, y la comunicación entre esta y los actuadores es lenta provocando que la manipulación sea lenta.

En cambio, la mano robótica construida cumple con el objetivo de manipular un objeto, con la capacidad de realizar manipulaciones sencillas, pero aun así estos movimientos de manipulación sirven para poner a prueba los servomotores. La mano robótica posee ocho grados de libertad y cada uno de ellos con un actuador, lo cual les permite tener una mayor independencia de movimiento a cada servomotor y además provocando que los dedos puedan llegar a tener una mayor amplitud de posturas en el momento de manipulación. La mano robótica, aunque no ejerza una gran cantidad de fuerza en el momento de la sujeción de un objeto, puede mover el objeto sin necesidad de estar montada en un brazo robótico, además de que esta mano robótica sirve como un aditamento al brazo robótico Yaskawa MH5LF.

4.3 Conclusiones y prospectivas

El robot paralelo de ocho grados de libertad posee una movilidad lo suficientemente capaz para manipular un objeto, sin que el modelo matemático sea muy difícil de obtener o que el control del mismo sea muy complicado. El uso de la notación de Denavit-Hartenberg propicia un análisis sencillo de comprender sobre los mecanismos paralelos con unos principios básicos de seguir, esta notación permite que la obtención de las ecuaciones de la arquitectura de un robot paralelo sea sencilla, siempre y cuando las cadenas de dicho robot paralelo no sean cadenas redundantes o posean muchas incógnitas en cada cadena, porque de lo contrario se complicaría el querer obtener las soluciones de las incógnitas en determinadas cadena cinemáticas, provocando también que puedan surgir más errores al tratar

de obtener las soluciones de este mecanismos paralelo. El robot paralelo por su estructura de cadena cinemática cerrada tiene una mayor precisión en sus movimientos, y el desarrollo de un robot de ocho grados de libertad, encontrándose este entre las manos robóticas diestras y las pinzas robóticas, es más sencillo en el desarrollo de sus ecuaciones y control que las primeras, y poseyendo más capacidades de acción que las segundas, encontrándose en un punto intermedio muy adecuado como una adición a los brazos robóticos, siendo estos últimos robots con cadenas cinemáticas abiertas con una mayor rango de movimiento, pero sin las capacidades de los robots paralelos.

En este trabajo se logró desarrollar la mano robótica de ochos grados de libertad con una interfaz de control que permite a un usuario manipular un objeto esférico. Los actuadores utilizados en este proyecto poseen un gran potencial a la hora de manipular el objeto, usando dos tipos de servomotores en el robot paralelo, uno proporcionando con su torque y resolución de movimiento una gran estabilidad en la base de la mano robótica, mientras que los servomotores que se encuentran en las falanges medial y distal de la mano robótica aun poseen el suficiente torque y resolución de movimiento para poder ajustarse a las posiciones que se requiere para manipular un objeto. Las capacidades de los actuadores utilizados son simplemente de activación y de una velocidad angular constate, dejando de lado la aceleración que estos mismos pueden llegar a tener, aun así, dejando con ellos una cierta base para futuras mejoras y desarrollos en este trabajo.

El trabajo realizado en esta tesis puede ser ampliado con agregados como el análisis dinámico de la mano robótica, permitiendo de esta forma manipular las variables de aceleración que poseen los servomotores y por ende lograr un movimiento de manipulación más fluido; además los métodos de orientación y posicionamiento también puede ser un agregado a la mano robótica que le permita un mejor desempeño en la ejecución de las manipulaciones. También se le puede agregar sensores de presión en las puntas de los dedos para que de esta forma la mano robótica reconozca si la mano robótica está sujetando el objeto, permitiendo saber que fuerza es requerida para mantener sujetado el objeto. El agregar en la interfaz una mayor variedad de movimientos de manipulación sobre un objeto, además de poder manipular objetos de diferentes formas, ya sea un cilindro, un cubo o una figura irregular. Por último, agregar una tarjeta de control específica para la mano robótica, para que de esta forma solamente se le envíe a esta tarjeta la rutina de manipulación y con ello evitar que el proceso de manipulación no sea tarado o lento.

5 BIBLIOGRAFÍA

- 8203 OnRobot Three Finger Gripper 3FG15 - EMI Corp. (11 de 2020). Obtenido de 8203 OnRobot Three Finger Gripper 3FG15 - EMI Corp: <https://www.emicorp.com/item/OR%2B103666/8203-OnRobot-Three-Finger-Gripper-3FG15/>
- Angeles, J. (2014). *Fundamentals of Robotic Mechanical Systems, Theory, Methods, and Algorithms*. Montreal, QC, Canada: Ed. Springer.
- Atlas | Boston Dynamics. (Mayo de 2020). Obtenido de Atlas | Boston Dynamics: <https://www.bostondynamics.com/atlas>
- Challacombe, B. J., Khan, M. S., Murphy, D., & Dasgupta, P. (2006). The history of robotics in urology. *Springer*.
- Correa Martínez, F. (2014). *Desarrollo e implementación de una arquitectura abierta eléctrica y electrónica para un robot de seis grados de libertad*. UAQ, Querétaro.
- DH Robotics 3 Adaptive Gripper, Lifting ... - IndiaMART. (11 de 2020). Obtenido de DH Robotics 3 Adaptive Gripper, Lifting ... - IndiaMART: <https://www.indiamart.com/proddetail/3-finger-adaptive-gripper-22613332473.html>
- Hernández Martínez, E. E. (2010). *Diseño mecatrónico óptimo de simuladores paralelos a través del modelado y simulación*. UAQ, Querétaro, Qro.
- Kim, E.-H., Lim, M.-T., & Lee, Y.-K. (2010). Analysis Grasp Stability for Multi-fingered Robot Hand. *International Conference on Control, Automation and Systems, Gyeonggi-do, Korea*.
- Kuttan K. K., A. (2007). *Robotics*. New Dheli, India: I. K. International Publishing House Pvt. Ltd.
- M.C. González Gutiérrez, C. (2013). *Diseño y construcción de un robot de seis grados de libertad*. UAQ, Centro Universitario Santiago de Querétaro, Qro.
- Merlet, J.-P. (2006). *Parallel Robots*. Netherlands: Ed. Springer.
- Nocks, L. (2007). *The Robot: the life story of a thecnology*. Westport, Connecticut, London: Greenwood Press.
- Norton, R. L. (2009). *Diseño de maquinaria: síntesis y análisis de maquinas y mecanismos*. Ciudad de México, D. F.: Mc Graw Hill.
- Poole, H. (2012). *Fundamentals of Robotics Engineering*. New York, New York: Ed. Springer.
- Robotiq 3-Finger Gripper - Think Solutions LLC. (11 de 2020). Obtenido de Robotiq 3-Finger Gripper - Think Solutions LLC: <https://thinkbotsolutions.com/products/robotiq-3-finger>
- ROBOTIS HAND. (11 de 2021). Obtenido de ROBOTIS: <https://www.robotis.us/robotis-hand/>

- Robots and robotic devices - Vocabulary.* (05 de 2020). Obtenido de ISO 8373:2012(en), Robots and robotic devices - Vocabulary: <https://www.iso.org/obp/ui/#iso:std:iso:8373:ed-2:v1:en>
- Song, W., & Zhang, W. (2018). The CPAM Hand: Coupling-Parallel-Adaption Merged Robot Hand. *IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids) Beijing, China.*
- Three Finger Centric SoftGripper - Developer - Soft Gripping.* (11 de 2020). Obtenido de Three Finger Centric SoftGripper - Developer - Soft Gripping: <https://www.soft-gripping.shop/en/three-fingered-centric-soft-gripper-developer.html>
- Versius Surgical Robotic System - CMR Surgical.* (05 de 2020). Obtenido de Versius Surgical Robotic System - CMR Surgical: <https://cmrsurgical.com/versius/>
- Wang, X., Zhao, Y., Xiao, Y., Zhu, Y., & Wang, Y. (2018). Design of a Underactuated Robotic Hand with Three Articulated Fingers. *IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC).*
- Yamaguchi, K., Hirata, Y., & Ko, K. (2015). Underactuated Robot Hand for Dual-arm Manipulation. *IEEE International Conference on Intelligent Robots and Systems (IROS) Congress Center Hamburg, Germany.*

6 APÉNDICES

6.1 Apéndice 1: programa principal de la interfaz de control

Función del para la ejecución de la interfaz

```
function varargout = Programa_de_envio(varargin)
% PROGRAMA_DE_ENVIO MATLAB code for Programa_de_envio.fig
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @Programa_de_envio_OpeningFcn, ...
                  'gui_OutputFcn',  @Programa_de_envio_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Programa_de_envio is made visible.
function Programa_de_envio_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
handles.Parametro = 0;
guidata(hObject, handles);

function varargout = Programa_de_envio_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

% --- Executes on button press in Boton_Agarrar.
function Boton_Agarrar_Callback(hObject, eventdata, handles)
handles.Click_boton = 2;
guidata(hObject,handles);

set(hObject, 'Enable', 'off');
set(handles.Boton_C, 'Enable', 'off');

% --- Executes on button press in Boton_Manipular.
function Boton_Manipular_Callback(hObject, eventdata, handles)
handles.Click_boton = 3;
guidata(hObject,handles);

set(hObject, 'Enable', 'off');
set(handles.Boton_B, 'Enable', 'off');
```

```

set(handles.Boton_Soltar,'Enable','off');

% --- Executes on button press in Boton_Soltar.
function Boton_Soltar_Callback(hObject, eventdata, handles)
handles.Click_boton = 4;
guidata(hObject,handles);

set(hObject,'Enable','off');
set(handles.Boton_Manipular,'Enable','off');

% --- Executes on button press in Boton_C.
function Boton_C_Callback(hObject, eventdata, handles)

handles.Click_boton = 5;
guidata(hObject,handles);

set(hObject,'Enable','off');
set(handles.Boton_Manipular,'Enable','off');
set(handles.Boton_Agarrar,'Enable','off');
set(handles.Boton_Soltar,'Enable','off');

```

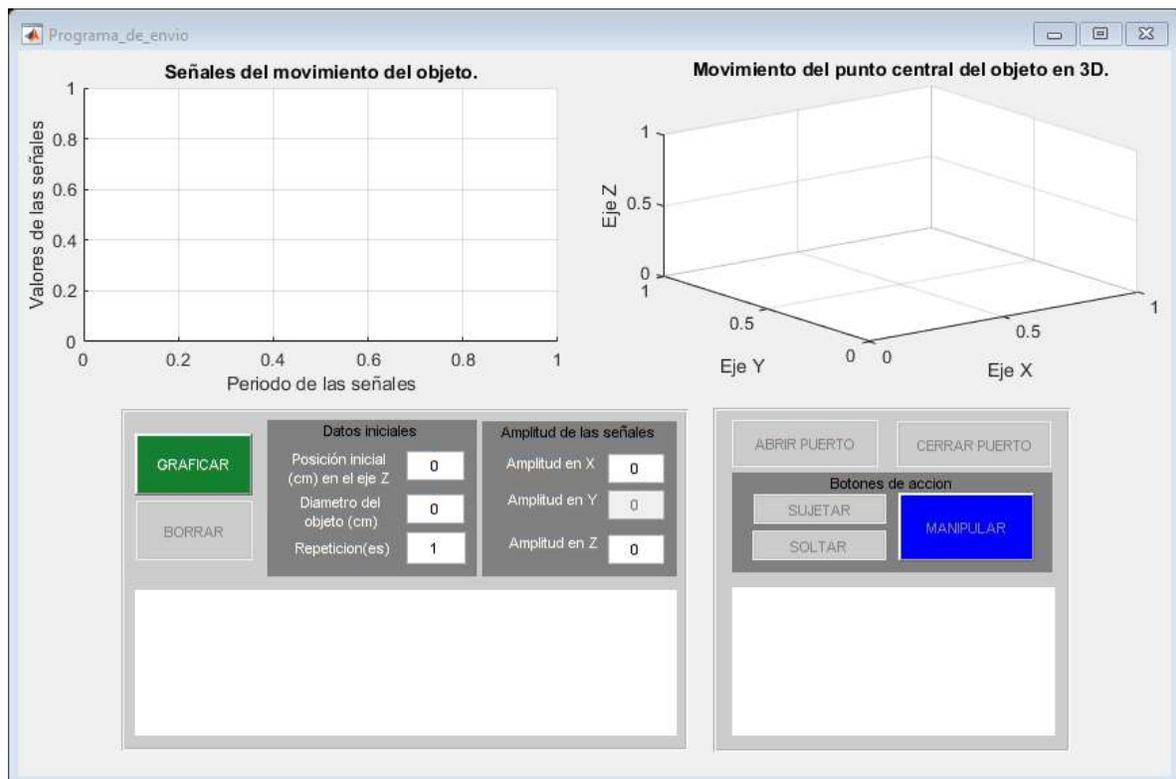


Figura 6.1 Interfaz de control.

Función de ejecución para el botón ABRIR PUERTO

```
function Boton_I_Callback(hObject, eventdata, handles)
clc;
set(hObject,'Enable','off');

% DECLARACIÓN DE LAS LIBRERIAS PARA LOS SERVOMOTORES DYNAMIXEL
lib_name = '';

if strcmp(computer, 'PCWIN')
    lib_name = 'dxl_x86_c';
elseif strcmp(computer, 'PCWIN64')
    lib_name = 'dxl_x64_c';
elseif strcmp(computer, 'GLNX86')
    lib_name = 'libdxl_x86_c';
elseif strcmp(computer, 'GLNXA64')
    lib_name = 'libdxl_x64_c';
elseif strcmp(computer, 'MACI64')
    lib_name = 'libdxl_mac_c';
end

% Load Libraries
if ~libisloaded(lib_name)
    [notfound, warnings] = loadlibrary(lib_name, 'dynamixel_sdk.h', 'addheader',
    'port_handler.h', 'addheader', 'packet_handler.h', 'addheader', 'group_sync_write.h');
end

% ID DE CADA SERVOMOTOR
DXL1_ID    = 1;
DXL2_ID    = 2;
DXL3_ID    = 3;
DXL4_ID    = 4;
DXL5_ID    = 5;
DXL6_ID    = 6;
DXL7_ID    = 7;
DXL8_ID    = 8;

% VERSION DEL PROTOCOLO DE COMUNIACIÓN
PROTOCOL    = 1.0;

% CONFIGURACION PREDETERMINADA
BAUDRATE    = 9600;
DEVICENAME  = 'COM3';

COMM_SUCCESS          = 0;           % Communication Success result value
COMM_TX_FAIL         = -1001;       % Communication Tx Failed

% INICIALIZACION DEL PUERTO DE COMUNIACIÓN
port_num = portHandler(DEVICENAME);
packetHandler();

dxl_comm_result = COMM_TX_FAIL;      % Resultado de la comunicación
dxl_addparam_result = false;        % Resultado del AddParam
```

```

dxl_error = 0; % Error de dynamixel

% APERTURA DEL PUERTO DE COMUNICACIÓN
% Abrir puerto
set(handles.Text_Accionar,'Background','c');
set(handles.Text_Accionar,'Foreground','k');
str = "Abriendo el puerto";
set(handles.Text_Accionar,'String',str);
pause(2);

if (openPort(port_num))
    set(handles.Text_Accionar,'Background','g');
    set(handles.Text_Accionar,'Foreground','k');
    str = "Se logró abrir el puerto.";
    set(handles.Text_Accionar,'String',str);
    pause(2);

else
    unloadlibrary(lib_name);

    set(handles.Text_Accionar,'Background','r');
    set(handles.Text_Accionar,'Foreground','w');
    str = "No se pudo abrir el puerto, verifique que el cable del controlador este
conectado a la computadora.";
    set(handles.Text_Accionar,'String',str);
    pause(2);

    set(handles.Boton_I,'Enable','on');
    return;
end

% Establecer la velocidad en baudios del puerto
if (setBaudRate(port_num, BAUDRATE))
    set(handles.Text_Accionar,'Background','g');
    set(handles.Text_Accionar,'Foreground','k');
    str = "Se logró cambiar la velocidad en baudios.";
    set(handles.Text_Accionar,'String',str);
    pause(2);

else
    unloadlibrary(lib_name);

    set(handles.Text_Accionar,'Background','r');
    set(handles.Text_Accionar,'Foreground','w');
    str = "No se pudo cambiar la velocidad en baudios, desconecte y conecte el cable del
controlador a la computadora.";
    set(handles.Text_Accionar,'String',str);
    pause(2);

    set(handles.Boton_I,'Enable','on');
    return;
end

```

```

% Activación de los servomotores
Activar_servos(dx1_comm_result, dx1_error, COMM_SUCCESS, port_num, PROTOCOL, DXL1_ID,
DXL2_ID, DXL3_ID, DXL4_ID, DXL5_ID, DXL6_ID, DXL7_ID, DXL8_ID)
pause(0.001);

% Apertura de la mano robótica
posicion = 0;
index = 1;
Posicion_asignada(dx1_comm_result, dx1_addparam_result, COMM_SUCCESS, dx1_error,
port_num,PROTOCOL, index, posicion, DXL1_ID, DXL2_ID, DXL3_ID, DXL4_ID, DXL5_ID, DXL6_ID,
DXL7_ID, DXL8_ID)
set(handles.Boton_C, 'Enable', 'on');
set(handles.Boton_Agarrar, 'Enable', 'on');

% Seleccionamiento de la acción a realizar
handles.Click_boton = 1;
guidata(hObject,handles);
handles.Parametro = 1;
guidata(hObject,handles);

% Menú de selección de la acción
while 1
    pause(0.001);
    handles = guidata(hObject);

    switch handles.Click_boton
        case 1 % Caso de espera
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            set(handles.Text_Accionar, 'Background', 'g');
            set(handles.Text_Accionar, 'Foreground', 'k');
            str = "Presione un boton de acción";
            set(handles.Text_Accionar, 'String', str);

        case 2 % La mano robótica SUJETA el objeto a manipular
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            %%%
            posicion = 1;
            index = 1;

            set(handles.P_i_z, 'Enable', 'off');
            set(handles.Diametro_0, 'Enable', 'off');
            set(handles.Text_Accionar, 'Background', 'c');
            set(handles.Text_Accionar, 'Foreground', 'k');
            str = "Sujetando el objeto.";
            set(handles.Text_Accionar, 'String', str);

            pause(0.001);
            Posicion_asignada(dx1_comm_result, dx1_addparam_result, COMM_SUCCESS,
dx1_error, port_num, PROTOCOL, index, posicion, DXL1_ID, DXL2_ID, DXL3_ID, DXL4_ID,
DXL5_ID, DXL6_ID, DXL7_ID, DXL8_ID)

            set(handles.Boton_Manipular, 'Enable', 'on');
            set(handles.Boton_Soltar, 'Enable', 'on');

```

```

handles.Par_Sujetar = 1;
guidata(hObject,handles);
handles.Click_boton = 1;
guidata(hObject,handles);

case 3 % Envió de los datos de posicion del servomotor para MANIPULAR el objeto
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
set(handles.Text_Accionar,'Background','b');
set(handles.Text_Accionar,'Foreground','w');
str = "Manipulando el objeto.";
set(handles.Text_Accionar,'String',str);
set(handles.Ciclo_mov,'Enable','off');

% Repeticiones del movimiento de manipulación
ciclo = str2num(get(handles.Ciclo_mov,'String'));

if (ciclo > 0)
    pause(0.001);
    Enviar_datos(dxl_comm_result, dxl_addparam_result, COMM_SUCCESS, dxl_error,
port_num, PROTOCOL, index, ciclo, DXL1_ID, DXL2_ID, DXL3_ID, DXL4_ID, DXL5_ID, DXL6_ID,
DXL7_ID, DXL8_ID)

else
    set(handles.Text_Accionar,'Background','r');
set(handles.Text_Accionar,'Foreground','w');
str = "El número de repeticiones debe ser al menos uno.";
set(handles.Text_Accionar,'String',str);
pause(3);

end

set(handles.Boton_Manipular,'Enable','on');
set(handles.Boton_Soltar,'Enable','on');
set(handles.Boton_B,'Enable','on');
set(handles.Ciclo_mov,'Enable','on');

handles.Click_boton = 1;
guidata(hObject,handles);

case 4 % La mano robótica SUELTA el objeto
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
set(handles.Text_Accionar,'Background','c');
set(handles.Text_Accionar,'Foreground','k');
str = "Soltando el objeto.";
set(handles.Text_Accionar,'String',str);

posicion = 0;
index = 1;
pause(0.001);
Posicion_asignada(dxl_comm_result, dxl_addparam_result, COMM_SUCCESS,
dxl_error, port_num, PROTOCOL, index, posicion, DXL1_ID, DXL2_ID, DXL3_ID, DXL4_ID,

```



```

function Am_z_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultuicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Am_y_Callback(hObject, eventdata, handles)
function Am_y_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultuicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Am_x_Callback(hObject, eventdata, handles)
function Am_x_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultuicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function P_i_z_Callback(~, eventdata, handles)
function P_i_z_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultuicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Diametro_0_Callback(hObject, eventdata, handles)
function Diametro_0_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultuicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Ciclo_mov_Callback(hObject, eventdata, handles)
function Ciclo_mov_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultuicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

Función de ejecución para el botón GRAFICAR

```
function Boton_G_Callback(hObject, eventdata, handles)
handles.Click_boton = 6;
guidata(hObject,handles);

am_x = str2double(get(handles.Am_x,'String'));
am_y = str2double(get(handles.Am_y,'String'));
am_z = str2double(get(handles.Am_z,'String'));
pi_z = str2double(get(handles.P_i_z,'String'));

diam = str2double(get(handles.Diametro_0,'String')); % Diametro del objeto
ciclo = str2double(get(handles.Ciclo_mov,'String')); % Repicione del movimiento de
manipulacion

if (ciclo > 0)
    if ((diam >= 1) && (diam <= 20))

        % Radio del objeto, se le quita un 5% al diametro del objeto
        r0 = (diam - (diam * 0.1))/2;
        dis = r0 + 1.49;

        % Calculo de los limites de posicionamiento del obejto para los dedos
        Zmax = sqrt( (12.25)^2 - ( dis - 7.9)^2 ) + 6.775;
        Xmax = -sqrt( (12.25)^2 - ( pi_z - 6.775)^2 ) + 7.9;
        Zmin = sqrt( (7.219)^2 - ( dis - 7.9)^2 ) + 6.775;

        Amp1_Xmax = abs(Xmax - dis);
        Amp1_Zmax = abs(Zmax - pi_z);
        Amp1_Zmin = abs(Zmin - pi_z);

        % Valores de posición minimo y maximo en Z asignados por el usuario
        z_max = pi_z + am_z;
        z_min = pi_z - am_z;

        if (am_x <= Amp1_Xmax)

            if (z_max <= (pi_z + Amp1_Zmax)) && (z_min >= (pi_z - Amp1_Zmin))
                fc = ciclo;
                N = ciclo * 49;
                fs = N;

                Sx = zeros([(N+1) 1]);
                Sy = zeros([(N+1) 1]);
                Sz = zeros([(N+1) 1]);

                for k = 0:N
                    Sx(k+1, 1) = am_x*sin(2*pi*fc*k/fs );
                    Sy(k+1, 1) = am_y*sin(2*pi*fc*k/fs);
                    Sz(k+1, 1) = pi_z + am_z*cos(2*pi*fc*k/fs);
                end

                % Genera la tabla de los datos a enviar a los servomotores de la mano
```

robótica

```
Bandera = Generar_tabla(Sx, Sy, Sz, N, r0);

if (Bandera == 0)

    axes(handles.axes1)
    plot(Sx(:,1), 'Linewidth',2, 'Color',[0.8500 0.3250 0.0980]);
    hold on
    plot(Sy(:,1), 'Linewidth',2, 'Color',[0.4660 0.6740 0.1880]);
    hold on
    plot(Sz(:,1), 'Linewidth',2, 'Color',[0 0.4470 0.7410]);
    legend({'Valores en X', 'Valores en Y', 'Valores en
Z'}, 'Location', 'east')
    title('Señales del movimiento del objeto.')
    xlabel('Periodo de las señales')
    ylabel('Valores de las señales')
    zoom on
    grid on

    axes(handles.axes2)
    plot3(Sx(:,1), Sy(:,1), Sz(:,1), 'bo');
    title('Movimiento del punto central del objeto en 3D.')
    xlabel('Eje X')
    ylabel('Eje Y')
    zlabel('Eje Z')
    grid on

    set(handles.Text_Graficar, 'Background', 'g');
    set(handles.Text_Graficar, 'Foreground', 'k');
    str = "El movimiento del objeto ha sido graficado exitosamente.";
    set(handles.Text_Graficar, 'String', str);

    set(hObject, 'Enable', 'off');
    set(handles.Boton_B, 'Enable', 'on');

    handles = guidata(hObject);
    switch handles.Parametro
        case 0
            set(handles.Boton_I, 'Enable', 'on');
        case 1
            handles = guidata(hObject);
            if (handles.Par_Sujetar == 1)
                set(handles.Boton_Manipular, 'Enable', 'on');
            end
        end

elseif (Bandera == 1)
    set(handles.Text_Graficar, 'Background', 'r');
    set(handles.Text_Graficar, 'Foreground', 'w');
    str = "NO SE PUEDE GRAFICAR, porque los servos no pueden alcanzar altura
hacia bajo del punto inicial, por lo que se sugiere cambiar al amplitud en eje Z o subir el
valor del punto inicial.";
    set(handles.Text_Graficar, 'String', str);
end
```

```

end

elseif (z_max >= (pi_z + Ampl_Zmax)) && (z_min >= (pi_z - Ampl_Zmin))
    set(handles.Text_Graficar,'Background','r');
    set(handles.Text_Graficar,'Foreground','w');
    str = ['NO SE PUEDE GRAFICAR, porque la amplitud prouesta de ' num2str(am_z)
' cm en Z, desde posición inicial de ' num2str(pi_z) ' cm supera el limite que se puede
alcanzar, se recomienda bajar la posición inicial o disminiur la amplitud.'];
    set(handles.Text_Graficar,'String',str);
elseif (z_max <= (pi_z + Ampl_Zmax)) && (z_min <= (pi_z - Ampl_Zmin))
    set(handles.Text_Graficar,'Background','r');
    set(handles.Text_Graficar,'Foreground','w');
    str = ['NO SE PUEDE GRAFICAR, porque la amplitud prouesta de ' num2str(am_z)
' cm en Z, desde posición inicial de ' num2str(pi_z) ' cm es menor al limite al que se
puede alcanzar, se recomienda aumentar la posición inicial o disminiur la amplitud.'];
    set(handles.Text_Graficar,'String',str);
elseif (z_max >= (pi_z + Ampl_Zmax)) && (z_min <= (pi_z - Ampl_Zmin))
    set(handles.Text_Graficar,'Background','r');
    set(handles.Text_Graficar,'Foreground','w');
    str = ['NO SE PUEDE GRAFICAR, porque la amplitud prouesta de' num2str(am_z)
' cm en Z, desde posición inicial de ' num2str(pi_z) ' cm es mayor a la amplitud maxima de
' num2str(Ampl_Zmax) ' cm en Z, se recomienda disminiur la amplitud.'];
    set(handles.Text_Graficar,'String',str);
end

elseif (am_x > Ampl_Xmax)
    set(handles.Text_Graficar,'Background','r');
    set(handles.Text_Graficar,'Foreground','w');
    str = ['NO SE PUEDE GRAFICAR, porque la amplitud en el eje x debe ser menor a
' num2str(Ampl_Xmax) ' cm, por lo que se te sugiere disminiur la amplitud en x a un valor
menor.'];
    set(handles.Text_Graficar,'String',str);
end

elseif ((diam < 1) || (diam > 20))
    set(handles.Text_Graficar,'Background','r');
    set(handles.Text_Graficar,'Foreground','w');
    str = "NO SE PUEDE GRAFICAR, porque el diametro del objeto no se encuentra entre 1
o 20 cm, por favor cambia a un objeto con un diametro entre 1 y 20 cm, y presione el boton
GRAFICAR.";
    set(handles.Text_Graficar,'String',str);
end

elseif (ciclo == 0) || (ciclo < 0)
    set(handles.Text_Graficar,'Background','r');
    set(handles.Text_Graficar,'Foreground','w');
    str = "NO SE PUEDE GRAFICAR, porque debe de haber almenos una repetición para realizar
el movimiento del objeto.";
    set(handles.Text_Graficar,'String',str);
end

% --- Executes on button press in Boton_B.
function Boton_B_Callback(hObject, eventdata, handles)
handles.Click_boton = 7;

```

```

guidata(hObject,handles);

set(hObject,'Enable','off');

handles = guidata(hObject);
switch handles.Parametro
    case 0
        set(handles.Boton_I,'Enable','off');
    case 1
        set(handles.Boton_Manipular,'Enable','off');
end

% Limpiar la tabla de datos a enviar
fileID = fopen('Tabla.txt','w');
fclose(fileID);

% Limpiar la grafica de la izquierda
cla(handles.axes1,'reset');
axes(handles.axes1)
title('Señales del movimiento del objeto.')
xlabel('Periodo de las señales')
ylabel('Valores de las señales')
grid on

% Limpiar la grafica de la derecha
cla(handles.axes2,'reset');
axes(handles.axes2)
plot3(0,0,0);
title('Movimiento del punto central del objeto en 3D.')
xlabel('Eje X')
ylabel('Eje Y')
zlabel('Eje Z')
grid on

set(handles.Text_Graficar,'Background','w');
set(handles.Text_Graficar,'Foreground','k');
str = "Los datos han sido borrados.";
set(handles.Text_Graficar,'String',str);
set(handles.Boton_G,'Enable','on');

% --- Executes during object creation, after setting all properties.
function axes1_CreateFcn(hObject, eventdata, handles)
title('Señales del movimiento del objeto.')
xlabel('Periodo de las señales')
ylabel('Valores de las señales')
grid on

% --- Executes during object creation, after setting all properties.
function axes2_CreateFcn(hObject, eventdata, handles)
plot3(0,0,0);
title('Movimiento del punto central del objeto en 3D.')
xlabel('Eje X')
ylabel('Eje Y')

```

```
zlabel('Eje Z')
grid on
```

6.2 Apéndice 2: funciones secundarias del programa principal

Función para la activación general de los servomotores

```
function Activar_servos(dx1_comm_result, dx1_error, COMM_SUCCESS, port_num, PROTOCOL,
DXL1_ID, DXL2_ID, DXL3_ID, DXL4_ID, DXL5_ID, DXL6_ID, DXL7_ID, DXL8_ID)
% Activación
ENABLE      = 1;

% Dirección del servomotor: AX-12A
AX_TORQUE_ENABLE      = 24;
AX_MOVING_SPEED      = 32;

% Dirección del servomotor: XM430-w210
XM_TORQUE_ENABLE      = 64;
XM_PROFILE_VELOCITY    = 112;

% Velocidades de los servomotores
SPEED_AX      = 54;  %(99 = 10.99 rev/min) ó (54 = 5.99 rev/min)
SPEED_XM      = 26;  %(48 = 10.99 rev/min) ó (26 = 5.95 rev/min)

% Activación de los servomotores del dedo número 1
DXL_ON(port_num, PROTOCOL, DXL1_ID, XM_TORQUE_ENABLE, ENABLE, XM_PROFILE_VELOCITY,
SPEED_XM, 1)
Port_Check(dx1_comm_result, dx1_error, COMM_SUCCESS, port_num, PROTOCOL, 1);

DXL_ON(port_num, PROTOCOL, DXL2_ID, AX_TORQUE_ENABLE, ENABLE, AX_MOVING_SPEED, SPEED_AX, 0)
Port_Check(dx1_comm_result, dx1_error, COMM_SUCCESS, port_num, PROTOCOL, 1);

DXL_ON(port_num, PROTOCOL, DXL3_ID, AX_TORQUE_ENABLE, ENABLE, AX_MOVING_SPEED, SPEED_AX, 0)
Port_Check(dx1_comm_result, dx1_error, COMM_SUCCESS, port_num, PROTOCOL, 1);

% Activación de los servomotores del dedo número 2
DXL_ON(port_num, PROTOCOL, DXL4_ID, XM_TORQUE_ENABLE, ENABLE, XM_PROFILE_VELOCITY,
SPEED_XM, 1)
Port_Check(dx1_comm_result, dx1_error, COMM_SUCCESS, port_num, PROTOCOL, 1);

DXL_ON(port_num, PROTOCOL, DXL5_ID, AX_TORQUE_ENABLE, ENABLE, AX_MOVING_SPEED, SPEED_AX, 0)
Port_Check(dx1_comm_result, dx1_error, COMM_SUCCESS, port_num, PROTOCOL, 1);

DXL_ON(port_num, PROTOCOL, DXL6_ID, AX_TORQUE_ENABLE, ENABLE, AX_MOVING_SPEED, SPEED_AX, 0)
Port_Check(dx1_comm_result, dx1_error, COMM_SUCCESS, port_num, PROTOCOL, 1);

% Activación de los servomotores del dedo número 3
DXL_ON(port_num, PROTOCOL, DXL7_ID, AX_TORQUE_ENABLE, ENABLE, AX_MOVING_SPEED, SPEED_AX, 0)
```

```

Port_Check(dx1_comm_result, dx1_error, COMM_SUCCESS, port_num, PROTOCOL, 1);

DXL_ON(port_num, PROTOCOL, DXL8_ID, AX_TORQUE_ENABLE, ENABLE, AX_MOVING_SPEED, SPEED_AX, 0)
Port_Check(dx1_comm_result, dx1_error, COMM_SUCCESS, port_num, PROTOCOL, 1);

```

Función para el cálculo de los ángulos Theta 3, Theta 4 y Theta 5

```

function [TH3, TH4, TH5] = Angulos_theta(Px, Py, Pz, r0, Theta1, Theta6, Theta7, caso)
% Datos de las distancia de un dedo de la mano robótica
r1 = 1.5;                r2 = r0;
a3 = 2.25;               a4 = 6.75;           a5 = 5.5;
a7 = r1 + r2;           b1 = 4.525;           b3 = 7.9;

% Transformación de grados a radianes de los ángulos th1, th6 y th7
th1 = Theta1 * pi/180;   th6 = Theta6 * pi/180;   th7 = Theta7 * pi/180;

% Matriz de orientacion %%%%%%%%%%
Q = Matriz_de_orientacion(th1, th6, th7);

% Punto C %%%%%%%%%%
Cx = Px - a7*Q(1,1);
Cy = Py - a7*Q(2,1);
Cz = Pz - a7*Q(3,1) - b1;

% Theta 3 %%%%%%%%%%
s1 = Cx*sin(th1) - Cy*cos(th1);

if caso == 0
    th3 = 0;
else
    th3 = atan( s1/Cz );
end
TH3 = th3 * 180/pi;

% Theta 5 %%%%%%%%%%
k1 = s1*sin(th3) + Cz*cos(th3) - a3;
k2 = Cx*cos(th1) + Cy*sin(th1) - b3;
k3 = k1^2 + k2^2 - a4^2 - a5^2;
s2 = 2*a4*a5;

th5 = -1*acos(k3/s2);
TH5 = th5 * 180/pi;

% Theta 4 %%%%%%%%%%
x1 = a4 + a5*cos(th5);
x2 = a5*sin(th5);
s4 = sqrt( x1^2 + x2^2 );
s5 = atan(x2/x1);

```

```
th4 = asin(k2/s4) - s5;
TH4 = th4 * 180/pi;
```

Función para el cálculo de la matriz de orientación

```
function MQ = Matriz_de_orientacion(TH1, TH6, TH7)

% Datos iniciales de cada marco de referencia
t1 = TH1;          aph1 = pi/2;    lmd1 = cos(aph1);    mi1 = sin(aph1);
t2 = pi/2;        aph2 = pi/2;    lmd2 = cos(aph2);    mi2 = sin(aph2);
t3 = 0;           aph3 = pi/2;    lmd3 = cos(aph3);    mi3 = sin(aph3);
t4 = 0;           aph4 = 0;       lmd4 = cos(aph4);    mi4 = sin(aph4);
t5 = 0;           aph5 = 0;       lmd5 = cos(aph5);    mi5 = sin(aph5);
t6 = TH6 - pi/2; aph6 = pi/2;    lmd6 = cos(aph6);    mi6 = sin(aph6);
t7 = TH7;         aph7 = 0;       lmd7 = cos(aph7);    mi7 = sin(aph7);

% Matrices de orientación de cada marco de referencia
M1 = [cos(t1) -lmd1*sin(t1)  mi1*sin(t1);
      sin(t1)  lmd1*cos(t1) -mi1*cos(t1); 0 mi1 lmd1];

M2 = [cos(t2) -lmd2*sin(t2)  mi2*sin(t2);
      sin(t2)  lmd2*cos(t2) -mi2*cos(t2); 0 mi2 lmd2];

M3 = [cos(t3) -lmd3*sin(t3)  mi3*sin(t3);
      sin(t3)  lmd3*cos(t3) -mi3*cos(t3); 0 mi3 lmd3];

M4 = [cos(t4) -lmd4*sin(t4)  mi4*sin(t4);
      sin(t4)  lmd4*cos(t4) -mi4*cos(t4); 0 mi4 lmd4];

M5 = [cos(t5) -lmd5*sin(t5)  mi5*sin(t5);
      sin(t5)  lmd5*cos(t5) -mi5*cos(t5); 0 mi5 lmd5];

M6 = [cos(t6) -lmd6*sin(t6)  mi6*sin(t6);
      sin(t6)  lmd6*cos(t6) -mi6*cos(t6); 0 mi6 lmd6];

M7 = [cos(t7) -lmd7*sin(t7)  mi7*sin(t7);
      sin(t7)  lmd7*cos(t7) -mi7*cos(t7); 0 mi7 lmd7];

% Matriz de oreintación
MQ = M1*M2*M3*M4*M5*M6*M7;
```

Función para la desactivación general de los servomotores

```
function Desactivar_servos(port_num, PROTOCOL, DXL1_ID, DXL2_ID, DXL3_ID, DXL4_ID, DXL5_ID,
DXL6_ID, DXL7_ID, DXL8_ID)
% Desactivación
DISABLE      = 0;

% Dirección del servomotor: AX-12A
AX_TORQUE_ENABLE  = 24;

% Dirección del servomotor: XM430-w210
XM_TORQUE_ENABLE  = 64;

% Servomotores del dedo número 1
DXL_OFF(port_num, PROTOCOL, DXL1_ID, XM_TORQUE_ENABLE, DISABLE);
DXL_OFF(port_num, PROTOCOL, DXL2_ID, AX_TORQUE_ENABLE, DISABLE);
DXL_OFF(port_num, PROTOCOL, DXL3_ID, AX_TORQUE_ENABLE, DISABLE);

% Servomotores del dedo número 2
DXL_OFF(port_num, PROTOCOL, DXL4_ID, XM_TORQUE_ENABLE, DISABLE);
DXL_OFF(port_num, PROTOCOL, DXL5_ID, AX_TORQUE_ENABLE, DISABLE);
DXL_OFF(port_num, PROTOCOL, DXL6_ID, AX_TORQUE_ENABLE, DISABLE);

% Servomotores del dedo número 3
DXL_OFF(port_num, PROTOCOL, DXL7_ID, AX_TORQUE_ENABLE, DISABLE);
DXL_OFF(port_num, PROTOCOL, DXL8_ID, AX_TORQUE_ENABLE, DISABLE);
```

Función para la activación individual de los servomotores y la asignación de la velocidad

```
function DXL_ON(port, PROTOCOL, ID, TORQUE_ENABLE, ON, ADDR_SPEED, SPEED_VAL, SERVO)
% Activación del torque
write1ByteTxRx(port, PROTOCOL, ID, TORQUE_ENABLE, ON);

% Declaración de la velocidad
if SERVO == 0
    write2ByteTxRx(port, PROTOCOL, ID, ADDR_SPEED, SPEED_VAL);
else
    write4ByteTxRx(port, PROTOCOL, ID, ADDR_SPEED, typecast(int32(SPEED_VAL), 'uint32'));
end
```

Función para la desactivación individual de los servomotores

```
function DXL_OFF(port, PROTOCOL, ID, TORQUE_ENABLE, OFF)

% Desactivación del torque
writeByteTxRx(port, PROTOCOL, ID, TORQUE_ENABLE, OFF);
```

Función de inspección del puerto de comunicación

```
function Port_Check(dx1_comm_result, dx1_error, COMM_SUCCESS, port_num, PROTOCOL, in)

dx1_comm_result = getLastTxRxResult(port_num, PROTOCOL);
dx1_error = getLastRxPacketError(port_num, PROTOCOL);
if dx1_comm_result ~= COMM_SUCCESS
    fprintf('%s\n', getTxRxResult(PROTOCOL, dx1_comm_result));
elseif dx1_error ~= 0
    fprintf('%s\n', getRxPacketError(PROTOCOL, dx1_error));
elseif in == 1
    fprintf('Dynamixel has been successfully connected \n');
end
```

Función para la generación de la tabla con los valores a enviar a cada servomotor

```
function Bandera = Generar_tabla(Sx, Sy, Sz, N, r0)

% Declaración de la matriz para los valores de los ángulos TH3, TH4, TH5
THs_D1 = zeros([(N+1) 3]); % Matriz del dedo #1
THs_D2 = zeros([(N+1) 3]); % Matriz del dedo #2
THs_D3 = zeros([(N+1) 3]); % Matriz del dedo #3

% Obtención de los ángulos para cada dedo de la mano robótica
for i = 1:(N+1)
    % Dedo #1 THs_D1(TH3, TH4, TH5)
    [THs_D1(i,1), THs_D1(i,2), THs_D1(i,3)] = Angulos_theta( Sx(i,1), Sy(i,1), Sz(i,1), r0,
120, 0, 0, 1);

    % Dedo #2 THs_D2(TH3, TH4, TH5)
    [THs_D2(i,1), THs_D2(i,2), THs_D2(i,3)] = Angulos_theta( Sx(i,1), Sy(i,1), Sz(i,1), r0,
-120, 0, 0, 1);

    % Dedo #3 THs_D3(TH3, TH4, TH5)
    [THs_D3(i,1), THs_D3(i,2), THs_D3(i,3)] = Angulos_theta( Sx(i,1), Sy(i,1), Sz(i,1), r0,
0, 0, 0, 0);
end
```

```

% Limites de movimiento angular de los servomotores
Servos_Lim = zeros([(N+1) 1]); % Vector de los limites de cada falange
Bandera = 0;

for i = 1:(N+1)
    % Casos de la falange Proximal
    if (( THs_D1(i,1) < -90) || ( THs_D1(i,1) > 90)) || (( THs_D2(i,1) < -90) || (
THs_D2(i,1) > 90)) || (( THs_D3(i,1) < -90) || ( THs_D3(i,1) > 90))
        Servos_Lim(i,1) = 1;
        Bandera = 1;
    end

    % Casos de la falange Medial
    if (( THs_D1(i,2) < -70) || ( THs_D1(i,2) > 90)) || (( THs_D2(i,2) < -70) || (
THs_D2(i,2) > 90)) || (( THs_D3(i,2) < -70) || ( THs_D3(i,2) > 90))
        Servos_Lim(i,1) = 1;
        Bandera = 1;
    end

    % Casos de la falange Distal
    if (( THs_D1(i,3) < -109) || ( THs_D1(i,3) > 90)) || (( THs_D2(i,3) < -109) || (
THs_D2(i,3) > 90)) || (( THs_D3(i,3) < -109) || ( THs_D3(i,3) > 90))
        Servos_Lim(i,1) = 1;
        Bandera = 1;
    end
end

% Matriz de valores de los angulos TH3, Th4, Th5 tranformados
% en un rango de 0 a 4095 para TH3, y un rango de 0 a 1023 para TH4 y TH5
THs_T = zeros([(N+1) 8]); % Matriz contenedora de los datos a enviar de cada dedo

% Tranformacion de los ángulos en rangos de 10 bits y 12 bits
for i = 1:(N+1)
    THs_T(i,1) = Transformar(0, THs_D1(i,1));
    THs_T(i,2) = Transformar(1, THs_D1(i,2));
    THs_T(i,3) = Transformar(1, THs_D1(i,3));

    THs_T(i,4) = Transformar(0, THs_D2(i,1));
    THs_T(i,5) = Transformar(1, THs_D2(i,2));
    THs_T(i,6) = Transformar(1, THs_D2(i,3));

    THs_T(i,7) = Transformar(1, THs_D3(i,2));
    THs_T(i,8) = Transformar(1, THs_D3(i,3));
end

% Generación de la tabla de datos a enviar
fileID = fopen('Tabla.txt','w');
fprintf(fileID, '%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\n', THs_T');
fclose(fileID);

```

Función de transformación de los datos en ángulos a valores de 10 bits o 12 bits

```
function TH = Transformar(caso, th)

if caso == 1
    TH = round((th + 150)/( 300/1024 ));
else
    TH = round((th + 180)/( 360/4096 ));
end
```

Función de asignación de posiciones específicas para los servomotores

```
function Posicion_asignada(dx1_comm_result, dx1_addparam_result, COMM_SUCCESS, dx1_error,
port_num, PROTOCOL, index, posicion, DXL1_ID, DXL2_ID, DXL3_ID, DXL4_ID, DXL5_ID, DXL6_ID,
DXL7_ID, DXL8_ID)

% Direcciones del servomotor: XM430-w210
XM_GOAL_POSITION      = 116;
XM_PRESENT_POSITION  = 132;

% Direcciones del servomotor: AX-12A
AX_GOAL_POSITION      = 30;
AX_PRESENT_POSITION   = 36;

% Longitud de los datos en bytes
AX_LEN  = 2;
XM_LEN  = 4;

% Umbral del estado de movimiento del servomotor
DXL_MOVING_STATUS_THRESHOLD = 6;

% Grupo de escritura para cada servomotor
group_num_AX = groupSyncwrite(port_num, PROTOCOL, AX_GOAL_POSITION, AX_LEN); % Servomotor
AX-12A
group_num_XM = groupSyncwrite(port_num, PROTOCOL, XM_GOAL_POSITION, XM_LEN); % Servomotor
XM430-w210

% Asignación del valor a tomar por la mano robótica
if posicion == 0

    % valores de la posición de activación y
    % desactivación de los servomotores

    %Dedo #1
    dx1_goal_position_1 = [2048 2048];
    dx1_goal_position_2 = [819 512];
    dx1_goal_position_3 = [205 512];

    %Dedo #2
```

```

dxl_goal_position_4 = [2048 2048];
dxl_goal_position_5 = [819 512];
dxl_goal_position_6 = [205 512];

%Dedo #3
dxl_goal_position_7 = [819 512];
dxl_goal_position_8 = [205 512];
else

% valores de la posición de sujeción del objeto

fileID = fopen('Tabla.txt','r');
foratSpec = '%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\n';
val = fscanf(fileID,foratSpec,[8 1]);
fclose(fileID);

%Dedo #1
dxl_goal_position_1 = [val(1,1) val(1,1)];
dxl_goal_position_2 = [val(2,1) val(2,1)];
dxl_goal_position_3 = [val(3,1) val(3,1)];

%Dedo #2
dxl_goal_position_4 = [val(4,1) val(4,1)];
dxl_goal_position_5 = [val(5,1) val(5,1)];
dxl_goal_position_6 = [val(6,1) val(6,1)];

%Dedo #3
dxl_goal_position_7 = [val(7,1) val(7,1)];
dxl_goal_position_8 = [val(8,1) val(8,1)];
end

% Dynamixel#1 asignacion de valor objetivo
dxl_addparam_result = groupSyncwriteAddParam(group_num_XM, DXL1_ID,
typecast(int32(dxl_goal_position_1(index)), 'uint32'), XM_LEN);
if dxl_addparam_result ~= true
    fprintf(['ID:%03d] groupSyncwrite addparam failed', DXL1_ID);
    return;
end

% Dynamixel#2 asignación de valor objetivo
dxl_addparam_result = groupSyncwriteAddParam(group_num_AX, DXL2_ID,
dxl_goal_position_2(index), AX_LEN);
if dxl_addparam_result ~= true
    fprintf(['ID:%03d] groupSyncwrite addparam failed', DXL2_ID);
    return;
end

% Dynamixel#3 asignación de valor objetivo
dxl_addparam_result = groupSyncwriteAddParam(group_num_AX, DXL3_ID,
dxl_goal_position_3(index), AX_LEN);
if dxl_addparam_result ~= true
    fprintf(['ID:%03d] groupSyncwrite addparam failed', DXL3_ID);
    return;
end

```

```

% Dynamixel#4 asignación de valor objetivo
dxl_addparam_result = groupSyncWriteAddParam(group_num_XM, DXL4_ID,
typecast(int32(dxl_goal_position_4(index)), 'uint32'), XM_LEN);
if dxl_addparam_result ~= true
    fprintf('ID:%03d groupSyncwrite addparam failed', DXL4_ID);
    return;
end

% Dynamixel#5 asignación de valor objetivo
dxl_addparam_result = groupSyncWriteAddParam(group_num_AX, DXL5_ID,
dxl_goal_position_5(index), AX_LEN);
if dxl_addparam_result ~= true
    fprintf('ID:%03d groupSyncwrite addparam failed', DXL5_ID);
    return;
end

% Dynamixel#6 asignación de valor objetivo
dxl_addparam_result = groupSyncWriteAddParam(group_num_AX, DXL6_ID,
dxl_goal_position_6(index), AX_LEN);
if dxl_addparam_result ~= true
    fprintf('ID:%03d groupSyncwrite addparam failed', DXL6_ID);
    return;
end

% Dynamixel#7 asignación de valor objetivo
dxl_addparam_result = groupSyncWriteAddParam(group_num_AX, DXL7_ID,
dxl_goal_position_7(index), AX_LEN);
if dxl_addparam_result ~= true
    fprintf('ID:%03d groupSyncwrite addparam failed', DXL7_ID);
    return;
end

% Dynamixel#8 asignación de valor objetivo
dxl_addparam_result = groupSyncWriteAddParam(group_num_AX, DXL8_ID,
dxl_goal_position_8(index), AX_LEN);
if dxl_addparam_result ~= true
    fprintf('ID:%03d groupSyncwrite addparam failed', DXL8_ID);
    return;
end

% Escritura sincronizada del valor objetivo
groupSyncWriteTxPacket(group_num_XM);
groupSyncWriteTxPacket(group_num_AX);

dxl_comm_result = getLastTxRxResult(port_num, PROTOCOL);
if dxl_comm_result ~= COMM_SUCCESS
    fprintf('%s\n', getTxRxResult(PROTOCOL, dxl_comm_result));
end

% Borrar almacenamiento de parámetros de la escritura sincronizada
groupSyncWriteClearParam(group_num_XM);
groupSyncWriteClearParam(group_num_AX);

```

```

while 1

    % Dynamixel#1 lectura del valor actual
    dxl1_present_position = read4ByteTxRx(port_num, PROTOCOL, DXL1_ID,
XM_PRESENT_POSITION);
    Port_Check(dxl_comm_result, dxl_error, COMM_SUCCESS, port_num, PROTOCOL, 0);

    % Dynamixel#2 lectura del valor actual
    dxl2_present_position = read2ByteTxRx(port_num, PROTOCOL, DXL2_ID,
AX_PRESENT_POSITION);
    Port_Check(dxl_comm_result, dxl_error, COMM_SUCCESS, port_num, PROTOCOL, 0);

    % Dynamixel#3 lectura del valor actual
    dxl3_present_position = read2ByteTxRx(port_num, PROTOCOL, DXL3_ID,
AX_PRESENT_POSITION);
    Port_Check(dxl_comm_result, dxl_error, COMM_SUCCESS, port_num, PROTOCOL, 0);

    % Dynamixel#4 lectura del valor actual
    dxl4_present_position = read4ByteTxRx(port_num, PROTOCOL, DXL4_ID,
XM_PRESENT_POSITION);
    Port_Check(dxl_comm_result, dxl_error, COMM_SUCCESS, port_num, PROTOCOL, 0);

    % Dynamixel#5 lectura del valor actual
    dxl5_present_position = read2ByteTxRx(port_num, PROTOCOL, DXL5_ID,
AX_PRESENT_POSITION);
    Port_Check(dxl_comm_result, dxl_error, COMM_SUCCESS, port_num, PROTOCOL, 0);

    % Dynamixel#6 lectura del valor actual
    dxl6_present_position = read2ByteTxRx(port_num, PROTOCOL, DXL6_ID,
AX_PRESENT_POSITION);
    Port_Check(dxl_comm_result, dxl_error, COMM_SUCCESS, port_num, PROTOCOL, 0);

    % Dynamixel#7 lectura del valor actual
    dxl7_present_position = read2ByteTxRx(port_num, PROTOCOL, DXL7_ID,
AX_PRESENT_POSITION);
    Port_Check(dxl_comm_result, dxl_error, COMM_SUCCESS, port_num, PROTOCOL, 0);

    % Dynamixel#8 lectura del valor actual
    dxl8_present_position = read2ByteTxRx(port_num, PROTOCOL, DXL8_ID,
AX_PRESENT_POSITION);
    Port_Check(dxl_comm_result, dxl_error, COMM_SUCCESS, port_num, PROTOCOL, 0);

    if ~( (abs(dxl_goal_position_1(index) - typecast(uint32(dxl1_present_position),
'int32')) > DXL_MOVING_STATUS_THRESHOLD) || (abs(dxl_goal_position_2(index) -
dxl2_present_position) > DXL_MOVING_STATUS_THRESHOLD) || (abs(dxl_goal_position_3(index) -
dxl3_present_position) > DXL_MOVING_STATUS_THRESHOLD) || (abs(dxl_goal_position_4(index) -
typecast(uint32(dxl4_present_position), 'int32')) > DXL_MOVING_STATUS_THRESHOLD) ||
(abs(dxl_goal_position_5(index) - dxl5_present_position) > DXL_MOVING_STATUS_THRESHOLD) ||
(abs(dxl_goal_position_6(index) - dxl6_present_position) > DXL_MOVING_STATUS_THRESHOLD) ||
(abs(dxl_goal_position_7(index) - dxl7_present_position) > DXL_MOVING_STATUS_THRESHOLD) ||
(abs(dxl_goal_position_8(index) - dxl8_present_position) > DXL_MOVING_STATUS_THRESHOLD) )

```

```

        break;
    end

end

end

```

Función de envío de los valores para los servomotores

```

function Enviar_datos(dx1_comm_result, dx1_addparam_result, COMM_SUCCESS, dx1_error,
port_num, PROTOCOL, index, ciclo, DXL1_ID, DXL2_ID, DXL3_ID, DXL4_ID, DXL5_ID, DXL6_ID,
DXL7_ID, DXL8_ID)

% Direcciones del servomotor: XM430-w210
XM_GOAL_POSITION      = 116;
XM_PRESENT_POSITION   = 132;

% Direcciones del servomotor: AX-12A
AX_GOAL_POSITION      = 30;
AX_PRESENT_POSITION   = 36;

% Longitud de los datos en bytes
AX_LEN  = 2;
XM_LEN  = 4;

% Umbral del estado de movimiento para cada servomotor
DXL_MOVING_STATUS_THRESHOLD = 10;

% Grupo de escritura para cada servomotor
group_num_AX = groupSyncwrite(port_num, PROTOCOL, AX_GOAL_POSITION, AX_LEN); % Servomotor
AX-A12A
group_num_XM = groupSyncwrite(port_num, PROTOCOL, XM_GOAL_POSITION, XM_LEN); % Servomotor
XM430-w210

% Datos de envío
tabla_c = (ciclo * 49) + 1;
fileID = fopen('Tabla.txt','r');
foratSpec = '%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\n';
val = fscanf(fileID,foratSpec,[8 tabla_c]);
fclose(fileID);

% Enviando datos
[~,c] = size(val);

for k = 1 : c

    % valores objetivo para el dedo #1
    dxl_goal_position_1 = [val(1, k) val(1, c+1-k)];
    dxl_goal_position_2 = [val(2, k) val(2, c+1-k)];
    dxl_goal_position_3 = [val(3, k) val(3, c+1-k)];

    % valores objetivo para el dedo #2
    dxl_goal_position_4 = [val(4, k) val(4, c+1-k)];

```

```

dxl_goal_position_5 = [Val(5, k) Val(5, c+1-k)];
dxl_goal_position_6 = [Val(6, k) Val(6, c+1-k)];

% Valores objetivo para el dedo #3
dxl_goal_position_7 = [Val(7, k) Val(7, c+1-k)];
dxl_goal_position_8 = [Val(8, k) Val(8, c+1-k)];

% Dynamixel#1 asignacion de valor objetivo
dxl_addparam_result = groupSyncWriteAddParam(group_num_XM, DXL1_ID,
typecast(int32(dxl_goal_position_1(index)), 'uint32'), XM_LEN);
if dxl_addparam_result ~= true
    fprintf('[ID:%03d] groupSyncwrite addparam failed', DXL1_ID);
    return;
end

% Dynamixel#2 asignación de valor objetivo
dxl_addparam_result = groupSyncWriteAddParam(group_num_AX, DXL2_ID,
dxl_goal_position_2(index), AX_LEN);
if dxl_addparam_result ~= true
    fprintf('[ID:%03d] groupSyncwrite addparam failed', DXL2_ID);
    return;
end

% Dynamixel#3 asignación de valor objetivo
dxl_addparam_result = groupSyncWriteAddParam(group_num_AX, DXL3_ID,
dxl_goal_position_3(index), AX_LEN);
if dxl_addparam_result ~= true
    fprintf('[ID:%03d] groupSyncwrite addparam failed', DXL3_ID);
    return;
end

% Dynamixel#4 asignación de valor objetivo
dxl_addparam_result = groupSyncWriteAddParam(group_num_XM, DXL4_ID,
typecast(int32(dxl_goal_position_4(index)), 'uint32'), XM_LEN);
if dxl_addparam_result ~= true
    fprintf('[ID:%03d] groupSyncwrite addparam failed', DXL4_ID);
    return;
end

% Dynamixel#5 asignación de valor objetivo
dxl_addparam_result = groupSyncWriteAddParam(group_num_AX, DXL5_ID,
dxl_goal_position_5(index), AX_LEN);
if dxl_addparam_result ~= true
    fprintf('[ID:%03d] groupSyncwrite addparam failed', DXL5_ID);
    return;
end

% Dynamixel#6 asignación de valor objetivo
dxl_addparam_result = groupSyncWriteAddParam(group_num_AX, DXL6_ID,
dxl_goal_position_6(index), AX_LEN);
if dxl_addparam_result ~= true
    fprintf('[ID:%03d] groupSyncwrite addparam failed', DXL6_ID);
    return;
end

```

```

end

% Dynamixel#7 asignación de valor objetivo
dxl_addparam_result = groupSyncWriteAddParam(group_num_AX, DXL7_ID,
dxl_goal_position_7(index), AX_LEN);
if dxl_addparam_result ~= true
    fprintf('[ID:%03d] groupSyncwrite addparam failed', DXL7_ID);
    return;
end

% Dynamixel#8 asignación de valor objetivo
dxl_addparam_result = groupSyncWriteAddParam(group_num_AX, DXL8_ID,
dxl_goal_position_8(index), AX_LEN);
if dxl_addparam_result ~= true
    fprintf('[ID:%03d] groupSyncwrite addparam failed', DXL8_ID);
    return;
end

% Escritura sincronizada del valor objetivo
groupSyncWriteTxPacket(group_num_XM);
groupSyncWriteTxPacket(group_num_AX);

dxl_comm_result = getLastTxRxResult(port_num, PROTOCOL);
if dxl_comm_result ~= COMM_SUCCESS
    fprintf('%s\n', getTxRxResult(PROTOCOL, dxl_comm_result));
end

% Borrar almacenamiento de parámetros de la escritura sincronizada
groupSyncWriteClearParam(group_num_XM);
groupSyncWriteClearParam(group_num_AX);

while 1

    % Dynamixel#1 lectura del valor actual
    dxl1_present_position = read4ByteTxRx(port_num, PROTOCOL, DXL1_ID,
XM_PRESENT_POSITION);
    Port_Check(dxl_comm_result, dxl_error, COMM_SUCCESS, port_num, PROTOCOL, 0);

    % Dynamixel#2 lectura del valor actual
    dxl2_present_position = read2ByteTxRx(port_num, PROTOCOL, DXL2_ID,
AX_PRESENT_POSITION);
    Port_Check(dxl_comm_result, dxl_error, COMM_SUCCESS, port_num, PROTOCOL, 0);

    % Dynamixel#3 lectura del valor actual
    dxl3_present_position = read2ByteTxRx(port_num, PROTOCOL, DXL3_ID,
AX_PRESENT_POSITION);
    Port_Check(dxl_comm_result, dxl_error, COMM_SUCCESS, port_num, PROTOCOL, 0);

    % Dynamixel#4 lectura del valor actual
    dxl4_present_position = read4ByteTxRx(port_num, PROTOCOL, DXL4_ID,
XM_PRESENT_POSITION);
    Port_Check(dxl_comm_result, dxl_error, COMM_SUCCESS, port_num, PROTOCOL, 0);

```

```

    % Dynamixel#5 lectura del valor actual
    dxl5_present_position = read2ByteTxRx(port_num, PROTOCOL, DXL5_ID,
AX_PRESENT_POSITION);
    Port_Check(dxl_comm_result, dxl_error, COMM_SUCCESS, port_num, PROTOCOL, 0);

    % Dynamixel#6 lectura del valor actual
    dxl6_present_position = read2ByteTxRx(port_num, PROTOCOL, DXL6_ID,
AX_PRESENT_POSITION);
    Port_Check(dxl_comm_result, dxl_error, COMM_SUCCESS, port_num, PROTOCOL, 0);

    % Dynamixel#7 lectura del valor actual
    dxl7_present_position = read2ByteTxRx(port_num, PROTOCOL, DXL7_ID,
AX_PRESENT_POSITION);
    Port_Check(dxl_comm_result, dxl_error, COMM_SUCCESS, port_num, PROTOCOL, 0);

    % Dynamixel#8 lectura del valor actual
    dxl8_present_position = read2ByteTxRx(port_num, PROTOCOL, DXL8_ID,
AX_PRESENT_POSITION);
    Port_Check(dxl_comm_result, dxl_error, COMM_SUCCESS, port_num, PROTOCOL, 0);

    if ~( (abs(dxl_goal_position_1(index) - typecast(uint32(dxl1_present_position),
'int32')) > DXL_MOVING_STATUS_THRESHOLD) || (abs(dxl_goal_position_2(index) -
dxl2_present_position) > DXL_MOVING_STATUS_THRESHOLD) || (abs(dxl_goal_position_3(index) -
dxl3_present_position) > DXL_MOVING_STATUS_THRESHOLD) || (abs(dxl_goal_position_4(index) -
typecast(uint32(dxl4_present_position), 'int32')) > DXL_MOVING_STATUS_THRESHOLD) ||
(abs(dxl_goal_position_5(index) - dxl5_present_position) > DXL_MOVING_STATUS_THRESHOLD) ||
(abs(dxl_goal_position_6(index) - dxl6_present_position) > DXL_MOVING_STATUS_THRESHOLD) ||
(abs(dxl_goal_position_7(index) - dxl7_present_position) > DXL_MOVING_STATUS_THRESHOLD) ||
(abs(dxl_goal_position_8(index) - dxl8_present_position) > DXL_MOVING_STATUS_THRESHOLD) )

        break;
    end

end
end

```