



UNIVERSIDAD POLITÉCNICA DE VALENCIA

**Departamento de Ingeniería de Sistemas y Automática
(DISA)**

MÁSTER UNIVERSITARIO EN AUTOMÁTICA E INFORMÁTICA INDUSTRIAL

**Diseño, implementación y control de una pinza servo-controlada
para un exoesqueleto**

TRABAJO DE FIN DE MÁSTER

AUTOR:

Camacho Muñoz, Jefferson Fernando

TUTOR:

Zotovic, Ranko

VALENCIA - ESPAÑA

2016

Índice general

Índice de figuras	IV
Índice de tablas	VII
1. ALCANCE DE LA INVESTIGACIÓN	2
1.1. Objetivos	2
1.1.1. Objetivo general	2
1.1.2. Objetivos específicos	2
1.2. Justificación	2
1.3. Metodología	3
2. ESTADO DEL ARTE	6
2.1. Introducción	6
2.2. Robótica	7
2.3. Sistema robótico multimodal	9
2.4. Exoesqueletos	13
2.4.1. Clasificación de los exoesqueletos	15
2.4.1.1. Exoesqueletos No portátiles	15
2.4.1.2. Exoesqueletos Portátiles	16
2.4.2. Aplicaciones de los exoesqueletos robóticos	16
2.4.2.1. Amplificadores de potencia	17
2.4.2.2. Dispositivo para realimentación háptica y de fuerzas	20
2.4.2.3. Rehabilitación y entrenamiento motor	21
2.4.2.4. Estudio e investigación	22
2.4.3. Power Assist suit Kawassaki	23
2.5. Efecto Final	25
2.5.1. Herramienta	25
2.5.1.1. Criterios de selección de manipuladores	26
2.5.2. Pinza Robótica	27
2.6. Pinza robótica como prótesis	29

3. MATERIALES Y MÉTODOS	32
3.1. Hardware	32
3.2. Fundamentos para el diseño de la pinza	32
3.2.1. Puntos de contacto	32
3.2.2. Número de dedos a utilizar	33
3.2.3. Fuerza de acuerdo a los puntos de presión	34
3.2.4. Desplazamiento para el agarre	37
3.2.4.1. Área a ocupar	37
3.2.4.2. Máximo de Apertura	37
3.2.4.3. Desgaste Mecánico	38
3.2.4.4. Fuerza Máxima	38
3.2.4.5. Objetos a sujetar	39
3.2.5. Tamaño de los dedos	40
3.3. Diseño de la pinza	40
3.3.1. Cara inferior de la pinza	41
3.3.2. Cara superior de la pinza	42
3.3.3. Componente izquierdo y derecho compuesto por dos dedos cada uno	43
3.3.4. Ejes de rotación	45
3.4. Sensores de Fuerza	46
3.5. Diseño de circuitos electrónicos	49
3.6. Software	50
3.6.1. Arquitectura de programación	51
3.6.1.1. Interfaz USB	53
3.6.1.2. SDRAM	53
3.6.2. Identificación del sistema	54
3.6.2.1. Filtro del sensor de retroalimentación	55
3.6.2.2. Obtener datos	57
3.6.2.3. Calculo de la función de transferencia	58
3.6.2.4. Simulación y validación de las ecuaciones obtenidas	61
3.6.2.5. Diseño de controlador	63
3.6.3. Control	66
3.6.3.1. Control por Velocidad	66
3.6.3.2. Control por Posición	70
3.6.3.3. Control por Fuerza	75

4. FUNCIONAMIENTO Y PRUEBAS	82
4.1. Conexión Tarjeta stm32f4xx	82
4.1.1. GPIO a utilizar	82
4.2. Librerías a utilizar	84
5. CONCLUSIONES Y RECOMENDACIONES	88
5.1. Conclusiones	88
5.2. Recomendaciones	90
Bibliografía	91
ANEXOS	94
A. Cotas de la pinza	95
B. Librerías Usadas	99
B.1. USB	99
B.2. ADC	100
B.3. PWM	100
B.4. SDRAM	100
B.5. TIMER	100
B.6. Librerías generales	101
B.7. Librerías creadas para el presente proyecto	101
C. Esquemas para circuito de sensores	102
D. Código desarrollado	104
D.1. main.c	104
D.2. acciones.c	111
D.3. acciones.h	117

Índice de figuras

1.1. Etapas de elaboración del proyecto	4
2.1. Crecimiento de la automatización robotizada[3]	7
2.2. Diagrama de bloques del sistema neuromuscular-esquelético humano[30]	10
2.3. Esquema en lazo cerrado del sistema de control[31]	11
2.4. Configuración del exoesqueleto robótico para estudios del movimiento del miembro superior. Izq: Esquema del movimiento; Der: Dispositivo adaptado sobre el miembro superior[30]	12
2.5. Tarjeta Raspberry Pi 3	13
2.6. Encoder con sensor óptico	13
2.7. Baterías montadas en exoesqueleto[9]	14
2.8. Marcibionics Exoesqueleto mecanismo de movilidad[10]	14
2.9. Marco de exoesqueleto	15
2.10. Clasificación en base a la portabilidad (a) No portátiles, (b) Portátiles	16
2.11. Exoesqueleto HULC[14]	18
2.12. Sistema teleoperado master-esclavo[15]	20
2.13. Interfaz Háptica Cybergrasp[16]	21
2.14. Exoesqueleto para rehabilitación[17]	21
2.15. Diseño biomecánico de una prótesis robótica[20]	22
2.16. Brazo accionado de forma neumática Kawasaki[29]	24
2.17. Esquema de traje Kawasaki[29]	24
2.18. Exoesqueleto desarrollado por la empresa Kawasaki	25
2.19. Efectores finales utilizados frecuentemente	26
2.20. Grados de libertad en una mano humana	28
2.21. Doble pinza basculante[24]	29
2.22. Prótesis de manos completas	29
3.1. Formas de objetos posibles[32]	33
3.2. Puntos de aplicación de fuerza	35
3.3. Desplazamientos Angular y Rectilíneo	38
3.4. Dimensiones de piñones[34]	41

3.5. Cara inferior de la pinza	42
3.6. Cara superior de la pinza	43
3.7. Componentes Izquierdo o derecho con sus respectivas partes	44
3.8. Unión entre los dedos y los ejes de la cara inferior	45
3.9. Unión entre los dedos y los ejes de la cara superior	45
3.10. Sensor de Fuerza	46
3.11. Divisor de Voltaje [35]	46
3.12. Conjunto de pesas para pruebas de sensor de fuerza	47
3.13. Salida - Peso aplicado Resistor 10 k Ω	48
3.14. Salida - Peso aplicado Resistor 1 k Ω	48
3.15. Salida - Peso aplicado Resistor 2 k Ω	49
3.16. Diseño de placa con entradas y salidas de sensores de fuerza	49
3.17. Esquema inversor de voltaje	50
3.18. Esquema de conexión puente h	50
3.19. Arquitectura de control general	51
3.20. Maquina de estados para control general	52
3.21. Maquina de estados para obtención de datos	52
3.22. Datos Obtenidos sin filtro	54
3.23. Datos Obtenidos con filtrado mediante capacitor electrolítico	55
3.24. Filtro promediador con 30 muestras	56
3.25. Filtro promediador con 50 muestras	57
3.26. Sistema de experimentación con ganancia 3 en el proporcional	58
3.27. Diseño en simulink para el esquema en bucle cerrado	62
3.28. Esquema real y esquema simulado con la función de transferencia obtenida	62
3.29. Esquema en bucle abierto y en bucle cerrado	63
3.30. Comparación en bucle abierto y en bucle cerrado	63
3.31. Diseño del controlador PD Sisotool	64
3.32. Respuesta al impulso mediante Sisotool	64
3.33. Respuestas ante referencia con controlador PD y P	65
3.34. Esquema de control para Velocidad	66
3.35. Esquema de control por estados	67
3.36. Esquema de funcionamiento para Velocidad	67
3.37. Ruido en el sensor del motor para determinar la velocidad del mismo	68
3.38. Desplazamiento constante para determinar la velocidad	69
3.39. Esquema de control para Posición	71
3.40. Esquema de funcionamiento para Posición	72
3.41. Esquema de control PID[38]	73
3.42. Algoritmo de control PID	74

3.43. Apertura de la pinza dentro del rango que va de 115 a 145 grados	75
3.44. Esquema de funcionamiento para Fuerza	76
3.45. Esquema de control para Fuerza	77
3.46. Objeto sujetado con una fuerza equivalente al 80 % de la capacidad total de la pinza	78
3.47. Objeto sujetado con una fuerza equivalente al 90 % de la capacidad total de la pinza	79
3.48. Objeto sujetado con una fuerza equivalente al 50 % de la capacidad total de la pinza	80
4.1. Definiciones en Target	84
4.2. Algunas de las librerías utilizadas para el proyecto	84
4.3. Path Discovery	85
4.4. Puerto a usar por la tarjeta Discovery	85
4.5. Posibles acciones a realizar	85
4.6. Forma de ejecutar las diferentes acciones	86
4.7. Ingresando mas de un valor para determinar posición	86
A.1. Medidas Cara inferior de la pinza	96
A.2. Medidas Cara superior de la pinza	97
A.3. Medidas ambos dedos	98
C.1. Placa acondicionada para sensores de fuerza	102
C.2. Top silk para nombrar los diferentes puntos de la placa	103

Índice de tablas

2.1. CAUSAS COMUNES DE DEBILIDAD MUSCULAR	19
2.2. HERRAMIENTAS TERMINALES PARA ROBOTS [22]	25
3.1. Acciones a realizar, necesidad y posibilidad de realizarla	39
3.2. Velocidad en diferentes instantes	70
4.1. GPIO Disponibles para funcionar como ADC's	83

ALCANCE DE LA INVESTIGACIÓN

1

ALCANCE DE LA INVESTIGACIÓN

1.1. Objetivos

1.1.1. Objetivo general

Diseñar, implementar y controlar una pinza servo-controlada para un exoesqueleto.

1.1.2. Objetivos específicos

- Diseñar una pinza para que pueda ser controlada mediante un servo-motor .
- Implementar la pinza en un exoesqueleto el cual ya se ha venido trabajando en diferentes proyectos.
- Realizar un control de velocidad y posición de acuerdo a los sensores de fuerza que el modelo contiene en su estructura.

1.2. Justificación

El incremento de la población con disfuncionalidad en sus extremidades superiores o inferiores, en las últimas décadas a ido en aumento, teniendo su origen en la mayoría de casos en accidentes causados por vehículos a motor, la facilidad para adquirir uno y la imprudencia o la impericia del personal humano, y en menor número enfermedades de tipo congénitas.

Con el fin de disminuir la dificultad para realizar tareas de forma cotidiana a personas que han sufrido algún tipo de amputación, o, que padecen de movilidad en sus extremidades, se ha dado paso a que se dedique especial atención a la investigación y desarrollo de prototipos exo-esqueléticos, que sean capaces de facilitar a quien los usa, el desempeño de sus actividades cotidianas, para de esta forma proporcionar al individuo, seguridad, confianza, reincorporarse e independizarse en las actividades diarias de la sociedad y de su entorno.

Es por tales motivos que se desarrolla el presente proyecto, demostrando de una manera eficiente, que es posible diseñar implementar y controlar un sistema que permita a quien lo porta; sostener objetos,

manipularlos, y hacer uso de los mismos, sin atrofiar su estructura física con precisión y exactitud.

1.3. Metodología

Para partir con el proyecto se analiza la metodología que se prevé seguir para lograr el correcto funcionamiento del dispositivo, la misma que se ha compuesto de tres etapas:

1. La primera etapa comprende el análisis para la creación de prototipo del proyecto a desarrollar, donde se realiza una recolección de información para la ejecución del mismo, recopilando datos tales como: tareas a realizar, peso de los objetos que se pretende manipular, peso máximo que el prototipo podría presentar, dimensiones, etc. Los prototipos serán basados en documentos Técnicos sobre como deberían estar estructurados para evitar fisuras y duración del componente, con estos datos de carácter informativo se definirá los posibles diseños a realizar y las ventajas de utilizar uno u otro en conformidad con los objetivos del proyecto.
2. La segunda etapa se enfoca en el desarrollo del diseño definitivo, para lo cuál se utilizaría la impresión 3D¹ como objeto de validación de los prototipos diseñados en la etapa anterior, si el objeto diseñado no cumple de forma eficiente con los objetivos para los cuales ha sido creado, se re-diseñara y volverá a imprimir hasta tener un objeto el cual cumpla con los objetivos planteados para el mismo.

Durante esta etapa también se realizará un estudio de los sensores a utilizar, obteniendo los resultados necesarios para su acondicionamiento, diseño de esquemas electrónicos y fabricación de la placa electrónica que se encargará de la manipulación de los mismos.

Con el diseño definitivo de la pinza y la adecuación de los sensores para su control, se procede a su montaje dentro de un entorno enfocado únicamente al funcionamiento de la pinza sin considerar aún el resto del exo-esqueleto.

3. La tercera etapa está enfocada al control y pruebas de funcionamiento las cuales han sido preparadas en la etapa anterior, documentar conclusiones y recomendaciones como resultado de varias pruebas efectuadas, con los sistemas en perfecto funcionamiento, cubriendo de esta forma cada uno de los objetivos propuestos.

La metodología a seguir en el proyecto se resume en la figura 1.1.

¹https://es.wikipedia.org/wiki/Impresión_3D

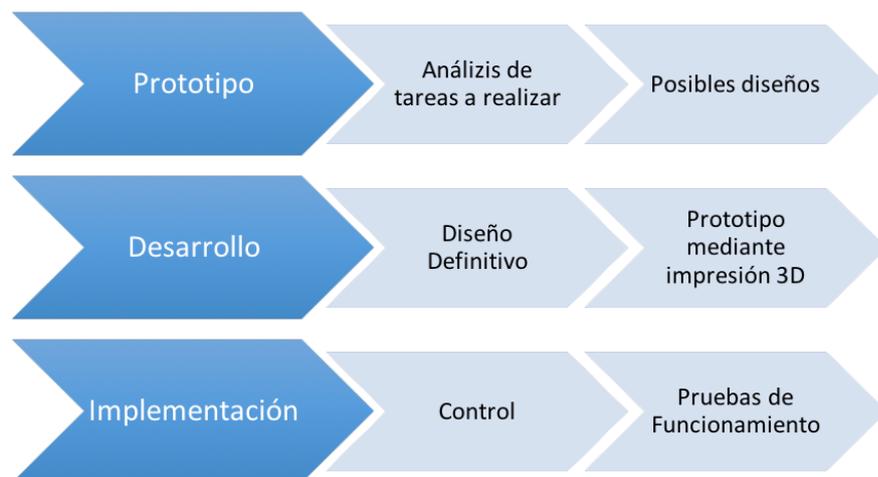


Figura 1.1: Etapas de elaboración del proyecto

ESTADO DEL ARTE

2

ESTADO DEL ARTE

2.1. Introducción

Son varios siglos que han transcurrido desde el diseño, planificación y puesta en marcha del primer aparato mecánico, Arquitas de Tarento, en cuyo honor se nombro al cráter Arquitas en la Luna, construyo un ave mecánica que funcionaba a vapor a quien llamo "La paloma", considerado el padre de la ingeniería mecánica y uno de los maestros clásicos de la robótica occidental o "Herón de Alejandría¹" quien creó varios dispositivos automáticos (Autómatas²) los cuales estaban dispuestos para que los usuarios pudiesen modificarlos, Hsieh-Fec, Al-Jazari, Roger Bacon, Juanelo Turriano, Leonardo da Vinci, Vaucanson o von Kempelen construyeron robots en la edad media, el renacimiento y el clasicismo, En el año de 1942 Asimov publica las tres leyes de la robótica con lo que da inicio a la robótica moderna, El desarrollo de robots en los campos bélicos, industrial y aeroespacial durante el siglo XX permite la aparición de robots con gran precisión útiles en cirugía como el robot quirúrgico "da Vinci³"[1].

El término robot fue utilizado por primera vez por Karel Capek en su obra de teatro denominada "Los robots universales de Rossum", robot viene del vocablo checo robota "que significa "trabajo", en el sentido de la obligatoriedad entendido para la servidumbre como trabajo forzado o para los esclavos, en este mismo concepto amo-esclavo, se basan los diseños actuales de robots, Años mas tarde la obra se adapto para la película denomida "Metropolis⁴", quedando acentuado el término que definiría a este tipo de máquinas.

Sin quedarse estancado el avance de la tecnología durante el siglo XVIII se dieron avances en el terreno industrial, como la máquina de vapor de Thomas Newcomen, perfeccionada después por Humphrey Potter que introdujo un novedoso concepto, la retroalimentación y por James Watt. En 1801 Joseph Marie Jacquard, un empresario textil, realiza una aportación fundamental a la robótica al diseñar un sistema de funcionamiento automático de los telares, programando sus movimientos. Se trata de un cartón mul-

¹<http://scielo.isciii.es/pdf/ae/v31n2/original1.pdf>

²<http://historiaautomatas.blogspot.com.es/2010/06/grecia-arquitas-de-tarento.html>

³<http://www.palexmedical.com/es/specialty.cfm?id=robotica-da-vinci>

⁴<http://www.alohacriticon.com/cine/criticas-peliculas/metropolis-fritz-lang/>

tipificado que permite tipificar algunas tareas y repetirlas de forma idéntica[1].

Tras constantes e importantes avances en la robótica se emprendió también en el diseño, control e implementación de exo-esqueletos de apoyo a las tareas realizadas por los humanos, dando un nuevo y amplio campo de investigación, apoyo y uso a la robótica.

2.2. Robótica

Robot formalmente se les denomina a las máquinas automáticas programables capaces de realizar determinadas operaciones de manera autónoma y reemplazar a los humanos de manera preferente en tareas pesadas, peligrosas, repetitivas o de acuerdo como lo define RIA (Robot Industries Asociation) "Manipulador funcional reprogramable, capaz de mover material, piezas, herramientas o dispositivos especializados mediante movimientos variables programados, con el fin de realizar tareas diversas[2]".

Los Robots han tenido en los últimos años un considerable incremento de intervención en las actividades humanas, ya sea por su eficiencia, precisión o reducción de costos en torno a la mano de obra, su mayor intervención se encuentra en la industria ya que podría trabajar las 24 horas durante todo el año y sin lugar a duda aumentaría la producción a mas bajo coste, lo cual ha hecho que la gran mayoría de industrias opte por estas tecnologías en los últimos años, tal como lo muestra la encuesta desarrollada por Rankia en su publicación denominada <El futuro de la automatización robotizada>, dándonos a conocer el suministro anual de robots industriales por región entre los años 2009 y 2013 tal como se muestra en la figura 2.1.

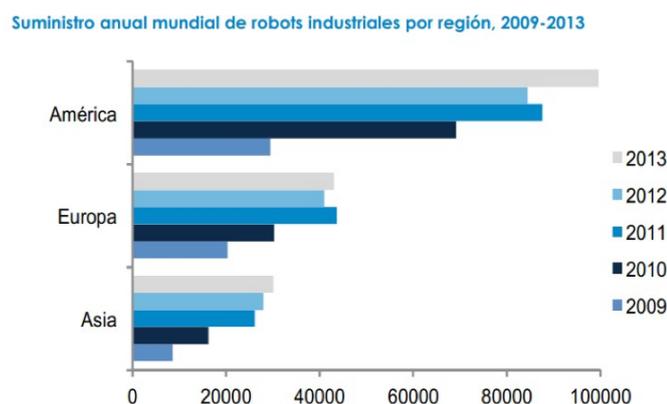


Figura 2.1: Crecimiento de la automatización robotizada[3]

Cada día es más evidente la inmersión de los mecanismos autónomos en trabajo conjunto con los seres humanos, lo cual lleva a una mayor atención en la inversión, investigaciones y exigencias de los mecanismos desarrollados y a desarrollarse, al mismo tiempo que ha ido creciendo la industria y la sociedad han aumentado los campos de investigación sobre ellos llegando a diferentes ramas como son:

- Aérea y espacial

- **Trasporte y automoción**
- **Medioambiental**
- **Misiones de rescate**
- **Vigilancia y seguridad**
- **Militar**
- **Tareas domésticas**
- **Entretenimiento y ocio**
- **Agricultura y ganadería**
- **Educación e información**
- **Asistencia y cuidado ancianos**
- **Construcción**
- **Quirúrgica**
- **Rehabilitación**
- **Exoesqueletos y prótesis**
- **Arte y patrimonio**
- **Submarina**

Existen aspectos de los mencionados anteriormente en los cuales ya se han desarrollado importantes avances en referencia a la tecnología que los precede como en el campo de la agricultura y ganadería cuyo principio fundamental de funcionamiento se basa en el reconocimiento del cultivo mediante imágenes que son procesadas y a su vez envían órdenes precisas en cuanto a las tareas a realizar, con alguna planta del cultivo, En la actualidad, existen pequeños robots que realizan tareas de búsqueda y transmisión de información sobre un determinado terreno de una forma totalmente nueva en el sector agrícola, aunque a futuro se prevé el uso de drones para realizar un recorrido más extendido y más rápido de un determinado entorno[4], estos equipos correspondientes al área aérea y espacial han ido evolucionando en cuanto a alcance, tiempo de vuelo, transmisión de datos y manipulación de los mismos se refiere.

También existen importantes avances en la construcción ya que la aplicación de un sistema de automatización en la construcción incrementa la productividad y la calidad de una determinada obra, de concretarse los proyectos en marcha se pretende reducir el tiempo de construcción en un 70 %, De esta manera, se garantiza un trabajo más eficiente, así como un incremento de calidad y con prestaciones de las edificaciones acorde con las necesidades de la sociedad actual, Santiago Martínez de la Casa de la

Universidad Carlos III de Madrid dice: que el estudio se centra en la introducción de robots industriales comerciales en el entorno de la construcción, favoreciendo el ensamblaje de múltiples productos y que pueden ser transportado de obra en obra[5].

Se podría también hablar del campo submarinos cuyos vehículos son aquellos no tripulados sumergibles que pueden ser controlados desde la superficie, pueden ser altamente maniobrables y al menos, llevan una cámara de video y luces. Se selecciona el tipo de robot a usar dependiendo del entorno u objetivo, pueden llevar diferentes accesorios tales como: sonares, brazos manipuladores y sensores de variados tipos. Actualmente hay varias empresas que se dedican a la investigación y desarrollo de este tipo de robots ya que muchos científicos u empresas los utilizan para llevar a cabo sus investigaciones marítimas[6].

Sin duda existen varios avances en todas las líneas de investigación mencionadas en el presente documento pudiendo ser una de las ramas que cuenta con mayor inversión y a la que se le atribuyen grandes avances y la línea militar, los robots militares se remontan a la Segunda Guerra Mundial y la Guerra Fría en la forma de Alemania minas Goliat y la Unión Soviética teletanks, El MQ-1 Predator drone fue cuando agentes de la CIA vieron los beneficios prácticos por primera vez en la fantasía de lo que en ese entonces era, la utilización de robots aéreos para recabar información de inteligencia, no obstante existe quienes creen que el futuro de la guerra moderna se peleará por los sistemas de armas automáticos¹, Es por tal motivo que potencias mundiales invierten grandes cifras en su investigación y desarrollo[7].

Parte de estas investigaciones no solo abarcan el desarrollo de armamento bélico, también se investiga la posibilidad de usando estructuras mecánicas controladas, se pueda proveer al ser humano de capacidades de fuerza, precisión, etc, que de otra manera sería imposible adquirirlas, ha estos aparatos mecánicos se les denomina exo-esqueletos, misma denominación que forma parte de una de las líneas de investigación mencionadas anteriormente.

Conscientes del potencial que pudiesen aportar los exo-esqueletos, se ha extendido su línea de investigación para dotar a personas que han sido despojadas de movilidad en sus extremidades o por ausencia de ellas, ha pernoctar en sus capacidades para investigar sobre la forma en como sustituirlas, para que el individuo que las posee se reintegre a sus actividades diarias de forma fiable y con la seguridad que solo sus extremidades naturales le fueron capaces de dar un día.

2.3. Sistema robótico multimodal

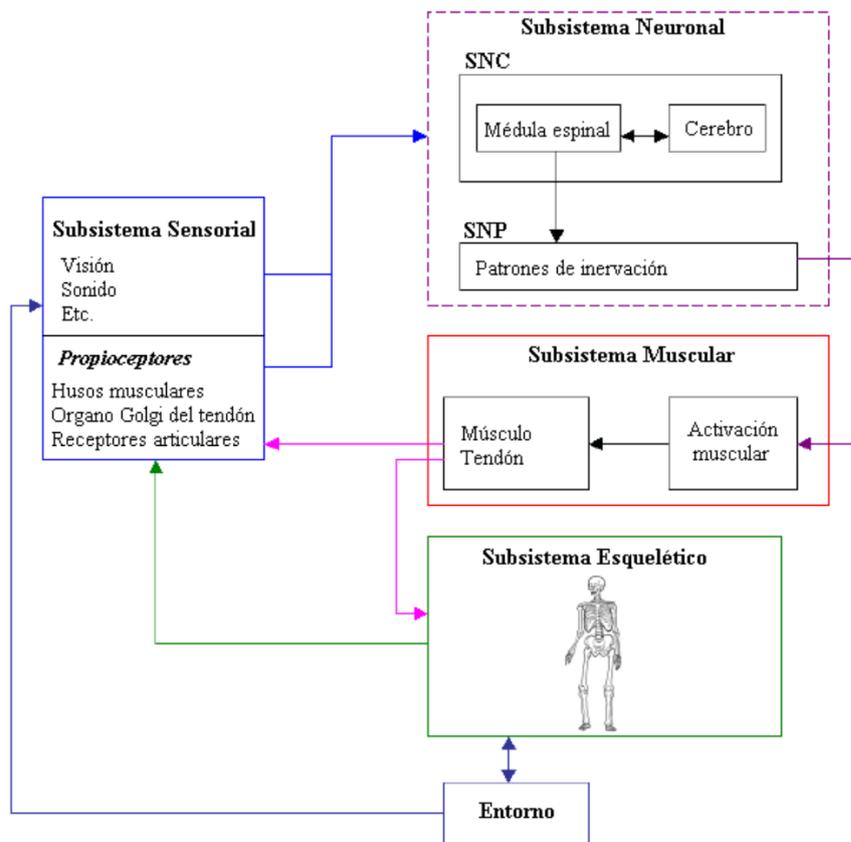
El trabajo de investigación denominado "Sistema Robótico Multimodal para Análisis y Estudios en Biomecánica, Movimiento Humano y Control Neuromotor" desarrollado por <Andrés Felipe Ruíz Olaya>, en la Universidad Carlos III de Madrid², es orientada a: diseñar, implementar y evaluar un sistema,

¹https://en.wikipedia.org/wiki/Military_robot

²<http://www.uc3m.es/Inicio>

compuesto por un exoesqueleto robótico de miembro superior y módulos independientes, que al actuar en conjunto dan una solución multimodal dentro del estudio neuromotor humano, desarrollando tanto el exoesqueleto como diversas herramientas, han sido integradas en un sistema común en esquema multimodal, siendo objeto de estudio la articulación del codo se ha determinado la viabilidad de estimar la impedancia mecánica a partir de la activación de los músculos asociados al movimiento de dicha articulación, específicamente bíceps y tríceps braquial.

Desde este punto de vista y en referencia al movimiento humano, existen diversos subsistemas que componen el sistema principal el cual enlaza una señal generada en el sistema nervioso central a un evento de movimiento humano. Teniendo como base que uno de los desafíos que afrontan los investigadores en neurociencia, involucrados en el análisis del movimiento humano, es entender que eventos o secuencia de los mismo, conecta la actividad neuronal a la contracción muscular, producto de lo cual se genera el movimiento humano y la interrelación entre los distintos subsistemas que componen el sistema motor humano para lo cual se plantea el esquema mostrado en la figura 2.2, con el fin de comprender su funcionamiento[30].



SNC: Sistema Nervioso Central
 SNP: Sistema Nervioso Periférico

Figura 2.2: Diagrama de bloques del sistema neuromuscular-esquelético humano[30]

De forma sencilla se puede simplificar el esquema en una entrada producto de un estímulo neurológico produciendo una salida traducida como una activación muscular, desde el punto de vista técnico al cual se acogen otros autores se vería como el esquema de la figura 2.3, en el cual T_{FF} corresponde a la estimación del bloque predictivo o feed forward y el T_{FB} corresponde al componente de realimentación feedback, resultando las propiedades elásticas del músculo[31].

Este esquema de control compuesto por una serie de bloques anticipativos o predictivos y diversos lazos de realimentación o feedback. Desde la perspectiva de la ingeniería de control, a un sistema de control se le aplica una perturbación externa y se lo cataloga al mismo mediante la observación de los cambios en su dinámica. De esta forma, se podría decir que para las investigaciones del movimiento humano y control neuromotor existen escenarios en los cuales es necesario adquirir y registrar la actividad eléctrica del sujeto, en sincronía con las mediciones cinemáticas y dinámicas producidas por el movimiento humano para modelar diversos aspectos del sistema de control[31].

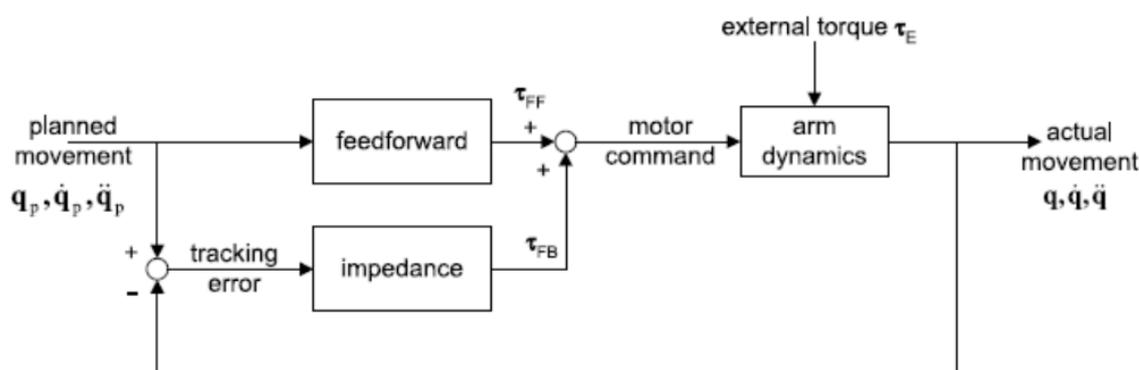


Figura 2.3: Esquema en lazo cerrado del sistema de control[31]

Con base en los puntos anteriores se desarrollo en primera instancia, un dispositivo robótico el cual fuese capaz de aplicar diferentes pares definidos sobre el movimiento voluntario del sujeto, usando soportes para el exoesqueleto los cuales proporcionen una transmisión efectiva de la carga sobre el miembro superior, para lo cual se ha usado materiales termo-plásticos como elementos de sujeción esta etapa concluida se muestra en la figura 2.4.

Este dispositivo ha sido diseñado tomando en consideración que se debe mantener un rango de movimientos en un espacio seguro para quien lo porta, ajustando su cinemática al miembro superior y restringiendo el rango de movimientos, a uno menor que el perteneciente al mismo miembro en cada una de sus articulaciones, se ha considerado también que el uso del dispositivo no altere a gran escala los patrones fisiológicos de la persona, ya que los movimientos del dispositivo dependen del umbral de presión de estos, limitando de esta manera los pares máximos a ser aplicados por el dispositivo robótico en las articulaciones y sus zonas de aplicación.

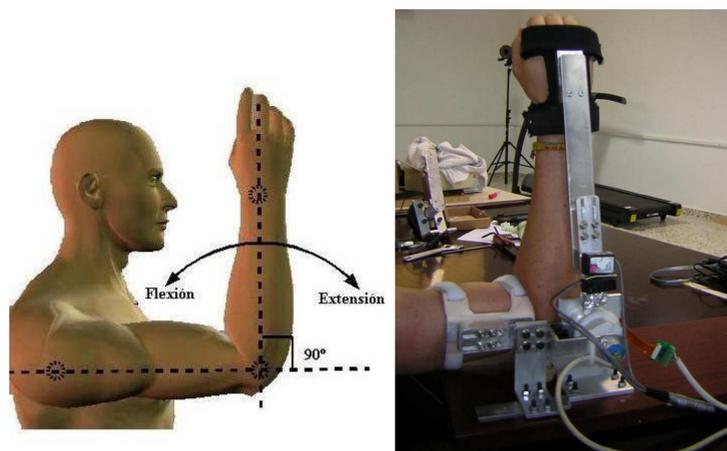


Figura 2.4: Configuración del exoesqueleto robótico para estudios del movimiento del miembro superior. Izq: Esquema del movimiento; Der: Dispositivo adaptado sobre el miembro superior[30]

El dispositivo robótico diseñado anteriormente sirve como dispositivo de medida y actuación otros elementos como la herramienta de adquisición de EMG¹, permiten registrar otras variables de interés asociadas a la actividad bioeléctrica, ya que cuenta con una arquitectura de control abierta, permite ejecutar en tiempo real algoritmos de control desarrollados en MATLAB², siendo posible la validación del hardware y adquisición de señales del mismo.

La experimentación del dispositivo en un entorno controlado a permitido comparar y validar pares generados y aplicados por el exoesqueleto, siendo posible la configuración de una serie de condiciones de operatividad sobre seguimiento de perfiles de fuerza, reproduciendo varios entornos dinámicos y generando perturbaciones mecánicas.

De este modo y mediante las aportaciones descritas anteriormente investigación llevada a cabo por Felipe Ruíz, deja constancia de tres componentes detalladas como lo son: científica, tecnológica y experimental en sujetos reales, cuya componente tecnológica se mostró en la figura 2.4, la componente científica implicada en el modelado de la dinámica del codo conjuntamente con esta información y la información EMG de los músculos asociados a dicha articulación. desarrollando de esta forma un sistema robótico para realizar estudios experimentales del sistema neuromotor humano.

Este proyecto ha dejado abiertas nuevas líneas de investigación tales como: Estudio del movimiento humano (al nivel del miembro superior) bajo cargas externas, ampliando su utilización a otras articulaciones como la muñeca. Configuración de experimentos que involucren no solo control postural sino también control del movimiento, etc. Estas líneas propuestas entre otras son de las cuales ha partido el presente proyecto.

¹<http://www.aboutkidshealth.ca/En/HealthAZ/Multilingual/ES/Pages/electromyography-emg.aspx>

²<http://es.mathworks.com/products/matlab/>

2.4. Exoesqueletos

El exoesqueleto no es mas que un aparato mecánico el cual sostiene al cuerpo desde afuera, a diferencia del esqueleto que lo conforman los huesos, el cual sostiene al cuerpo desde adentro; son usualmente diseñados para aumentar las capacidades de las personas como su fuerza, movilidad, resistencia o para ejercer actividades físicas que sin ellos les fuese imposibles.

Los exoesqueletos pueden estar compuestos de varios componentes como:

- **Controlador:** Parte del equipo de la electrónica del dispositivo, actuará como el cerebro del mismo tomando lecturas proporcionadas por los sensores y tomando como referencia las mismas decidir que acciones deben ejecutar los actuadores, si la lógica y la medición de las mismas es correctas permitirá al usuario pararse, caminar, levantar objetos u otras actividades para las cuales halla sido programado, un posible controlador podría ser la tarjeta Raspberry que se muestra en la figura 2.5



Figura 2.5: Tarjeta Raspberry Pi 3

- **Sensores:** Ya sean de fuerza, visión, distancia u otras variables que se deseen, obtendrán la información del entorno en que se ejecuta su lectura, para posteriormente ser transmitido al controlador, uno de estos sensores podría ser el encoder el cual conjuntamente con un sensor óptico, figura 2.6, entregan la información correspondiente al número de vueltas de un motor eléctrico.



Figura 2.6: Encoder con sensor óptico

- **Baterías:** Teniendo como función principal el movimiento del exoesqueleto, deben ser capaces de portar la energía suficiente para su movimiento y energizar los componentes eléctricos o electrónicos que pueda contener, estas deben recargarse rápidamente y no pesar demasiado para que el exoesqueleto no sea voluptuoso o demasiado pesado como es el caso de la figura 2.7.



Figura 2.7: Baterías montadas en exoesqueleto[9]

- **Actuadores:** Son los que se encargarán de los movimientos del exoesqueleto siendo usualmente de tipo eléctrico o hidráulico, estos ejecutan sus movimientos usando la energía enviada por las baterías y la información transmitida por el controlador, llevando el exoesqueleto a posiciones precisas y sin demasiado esfuerzo para la persona que los usa este tipo de dispositivos podrían lucir como el de Marcibionics¹ que se muestra en la figura 2.8.

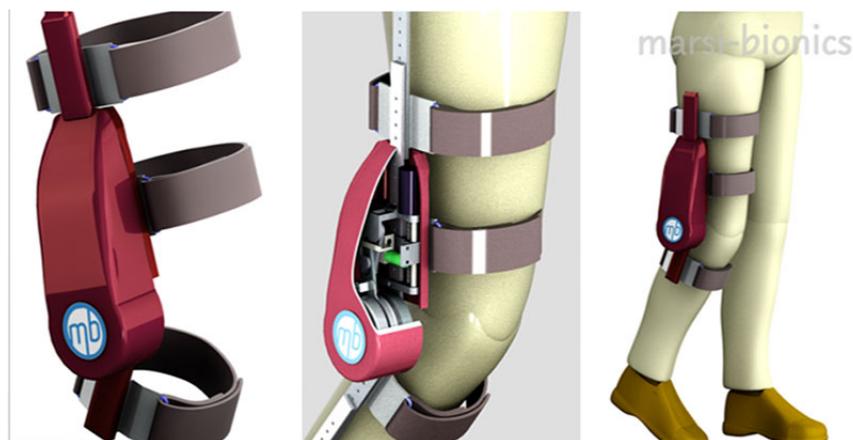


Figura 2.8: Marcibionics Exoesqueleto mecanismo de movilidad[10]

- **Marco:** El marco debe de ser de un material ligero y resistente como se muestra en la figura 2.9, ya que va a soportar no solo el peso de los objetos que vaya a levantar sino también el peso del sujeto que lo utilice, no debe suponer mayor esfuerzo y debe ser seguro para quien lo porte a fin de que no se lastime o sufra algún accidente relacionado con el uso del exoesqueleto[8].

¹<http://www.smartherapy.org/crowdfunding-marsi-bionics/>



Figura 2.9: Marco de exoesqueleto

A diferencia de las estructuras que comprende el diseño de un exoesqueleto completo, existen también exoesqueletos diseñados para partes específicas del cuerpo ya sea un brazo una pierna o la mano, siendo estas usadas por quienes carecen de la extremidad o porque han perdido la movilidad del mismo, estos diseños aparte de las exigencias de portabilidad, también es necesario que sean precisos, ya que interactuarán de manera directa con su portador y con el entorno que los rodea, cumpliendo tareas sencillas para el cuerpo humano pero de mayor complejidad para una extremidad robótica; como es el caso de levantar un recipiente que contenga líquido y verterlo en otro, su complejidad es mayor ya que quien lo manipula se limita a su necesidad y la extremidad robótica mediante sus diferentes sensores, debe interactuar no solo con el entorno sino también con las decisiones de la persona que lo porta, este ámbito es el que se encuentra actualmente en fase de estudio aunque ya se han tenido grandes avances.

2.4.1. Clasificación de los exoesqueletos

Los exoesqueletos se pueden clasificar en función de los actuadores utilizados (neumático, eléctrico o hidráulico), cadena cinemática, tipo de control, o aplicación, los diseños actuales en su mayoría se basan en interfaces sujetas a una base externa, pudiendo ser clasificados como portátiles o no portátiles, esta última clasificación se la puede apreciar en la figura 2.10.

2.4.1.1. Exoesqueletos No portátiles

A diferencia de los portátiles estos se encuentran anclados a una base fija, pudiendo ser esta el suelo, la pared o el techo, ya que el peso no recae en el usuario, este tipo de exoesqueletos suele ser mucho más pesado y a su vez más complejas, ya que pueden contar con mayor número de actuadores o instrumentos capaces de generar un torque mayor debido a la robustez de su estructura, siendo uno de los puntos más considerados la seguridad que puedan ofrecer los mismos (véase figura 2.10 a).

Ya que no están restringidos en peso o volumen se puede utilizar cualquier dispositivo sobre los mismos, inclusive transmisiones mecánicas como cables o poleas, siendo posible el uso de herramientas tradicionales para las diferentes transmisiones de fuerza[11].

2.4.1.2. Exoesqueletos Portátiles

Es un dispositivo mediante el cual el usuario tiene que ser quien soporte el peso del exoesqueleto o un dispositivo móvil como una silla de ruedas que se mueva conjuntamente con el sujeto, de esta forma los puntos de presión o fuerza son ejercidos sobre el usuario en los puntos de sujeción (véase figura 2.10 b).

Ya que existen restricciones de peso, se inhibe el uso de un gran número de actuadores, pero también se reduce la cantidad de grados de libertad para poder ejercer fuerzas de forma apropiada sobre los miembros del sujeto, es por tal motivo que los exoesqueletos portátiles requieren especial atención en el número de grados de libertad[11].

Este tipo de características en particular obliga a quien lo diseñe a crearlo con menos peso y volumen, evitando de esta manera la fatiga muscular, por ende sus actuadores deben corresponderse en potencia y peso o potencia y volumen para que puedan ser usados en este tipo de interfaces, siendo clasificados también en dos grupos exoesqueletos para el brazo o para la mano.

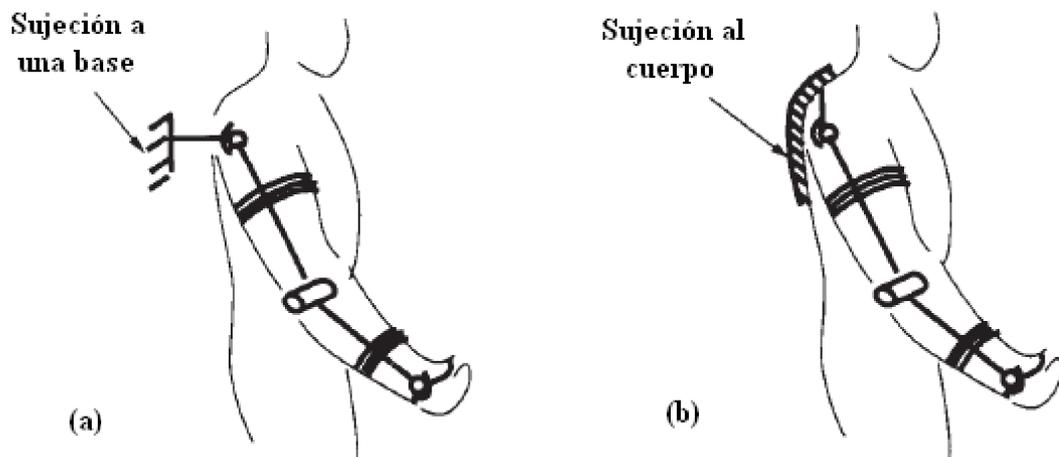


Figura 2.10: Clasificación en base a la portabilidad (a) No portátiles, (b) Portátiles

2.4.2. Aplicaciones de los exoesqueletos robóticos

Aunque los exoesqueletos han sido utilizados en diferentes aplicaciones. En forma de concepto estos sistemas se pueden utilizar en las siguientes aplicaciones.

- Amplificadores de potencia.
 - Asistencia y compensación funcional.

- Dispositivo para re-alimentación háptica y de fuerzas
 - Dispositivos de un sistema de teleoperación maestro/esclavo.
 - Dispositivos hápticos.
- Rehabilitación y entrenamiento motor.
- Estudio e investigación.
 - Biomecánica.
 - Neurofisiología y control motor.
 - Trastornos de movimiento.

2.4.2.1. Amplificadores de potencia

Como su nombre lo indica, son capaces de amplificar la capacidad física del sujeto que lo porta, aunque también puede ser usado por personas que hayan sufrido algún tipo de accidente el cual limite sus capacidades físicas o motoras[13].

Un estudio llevado a cabo por la Universidad de Pamplona¹ acerca de las Órtesis² activas, establece que los exoesqueletos son de gran ayuda durante la terapia ya que permiten interactuar directamente con el humano emulando sus movimientos, esto se debe a que lo conforma un mecanismo cuya estructura está acoplada externamente y sus juntas y eslabones corresponden a las partes del cuerpo humano que simula, el contacto entre estos sistemas y el cuerpo humano permite la transferencia de potencia mecánica y señales de información[12].

Cuando funcionan como amplificadores de potencia, el usuario transmite señales al exoesqueleto, ya sea mediante estímulos a sensores de diferente tipo o por tareas que fuesen pre-definidas teniendo como fin una determinada tarea, el usuario que utiliza este sistema puede desarrollar pesadas tareas de índole físico con el mínimo esfuerzo.

Un ejemplo notable en el desarrollo de este tipo de exoesqueletos es el desarrollado por los EE:UU para sus soldados el denominado HULC.

Exoesqueleto HULC: El Human Universal Load Carrier (HULC) o portador de carga universal humano es como la empresa "Lokheed Martin" denominó al exoesqueleto desarrollado para las fuerzas militares de los EE:UU, en una serie de pruebas en conjunto con Natick Soldier Research Development and Engineering Center(NSRDEC) el potencial de los exoesqueletos para aliviar el esfuerzo y la fatiga de

¹<http://www.unipamplona.edu.co/>

²<http://ortopedia1.com/ortesis>

los soldados al desplazarse con cargas demasiado pesadas que debían ser trasladadas a grandes distancias.

El personal militar a menudo lleva cargas pesadas de combate que aumenta el estrés en el cuerpo lo cual conduce a inevitables lesiones, con el exoesqueleto denominado HULC se pretende transferir el peso de estas cargas al suelo a través del exoesqueleto con piernas de titanio sin pérdida de movilidad[14].

Es un exoesqueleto que tiene forma humana y su accionamiento es de tipo hidráulico, proporcionando al usuario la posibilidad de llevar cargas de hasta 200 libras por períodos prolongados de tiempo sin importar por el terreno que se desplace, es muy flexible permitiendo al usuario realizar distintas posiciones corporales sin verse afectado por el uso del mismo como se aprecia en la figura 2.11.



Figura 2.11: Exoesqueleto HULC[14]

Cuenta con micro-ordenador el cual garantiza que el exo-esqueleto se mueva en concordancia con quien lo porta, sus componentes pueden ser reemplazados fuera de entornos controlados y su batería de larga duración permite el uso del dispositivo durante varias horas sin verse afectado su funcionamiento, si el sistema llegase a quedar sin energía suministrada por la batería es capaz de soportar una carga máxima prescindiendo de la misma.

El personal de investigación del Natick Soldier Center se encuentran evaluando también como el funcionamiento de HULC afecta el desempeño de los soldados. Además, las pruebas biomecánicas serán capaces de medir la energía gastada por un soldado cuando utiliza el exoesqueleto HULC y las pruebas desarrolladas evaluarán la facilidad en que sus usuarios aprenden a utilizar este sistema, realizando varias cargas y moviéndose a diferentes velocidades por diferentes terrenos.

Aunque la empresa Lockheed Martin se ha dedicado sus investigaciones al desarrollo de exoesqueletos para el campo militar, esta también dedicando parte de las mismas al uso de exoesqueletos en el campo

industrial[14].

Asistencia y compensación funcional

Quienes padecen de problemas neuromusculares pueden perder con el tiempo la capacidad de realizar actividades cotidianas con sus extremidades superiores, esto se debe a la pérdida de masa muscular o debilidad de la misma, este tipo de padecimientos pueden ser causados por varias enfermedades como distrofia muscular, atrofia muscular espinal, lesiones en la médula espinal, y ciertos niveles de atrofas producidas a edades avanzadas del paciente, en la tabla 2.1 se resumen algunas de las enfermedades que causan afectaciones en el sistema muscular de los seres humanos produciendoles perdida parcial o total de la movilidad de las extremidades [11].

Enfermedades
Apoplejía
Lesión de la médula espinal
Lesión nerviosa periférica
Esclerosis lateral amiotrófica
Distrofia muscular
Osteoartritis, artritis reumatoidea
Terapia prolongada con corticosteroides
Diabetes
Miopatía
Neuropatía del enfermo crítico
Poliomielitis
Intoxicación por arsénico
Miastenia gravis
Miotonía congénita
Nervio periférico y raíces nerviosas
Inmovilización prolongada
Síndrome de Guillain-Barre

Tabla 2.1: CAUSAS COMUNES DE DEBILIDAD MUSCULAR

Los robots para la asistencia a humanos ha sido considerado por muchas empresas como ente importante de investigación sobre todo en aquellos países donde la tasa de envejecimiento de la población es alto y es necesaria la atención de la misma y también la asistencia adecuada y precisa en personas discapacitadas.

Uno de los puntos mas importantes para poder el sistema asistir en las funciones motoras es la información o señales de control, ya que la misma debe reflejar las intenciones o acciones voluntarias del usuario, siendo algunas de las formas de entregar esta información las: mediciones de movimientos, fuerzas de interacción con el dispositivo, activación muscular o actividad cerebral[30].

2.4.2.2. Dispositivo para realimentación háptica y de fuerzas

Dispositivos de un sistema de teleoperación maestro/esclavo

Comprendida como el conjunto de tecnologías, la teleoperación permite al operador realizar remotamente diferentes tareas, ya sea porque estas conllevan riesgo para el operador o el acceso para el mismo no es posible [15].

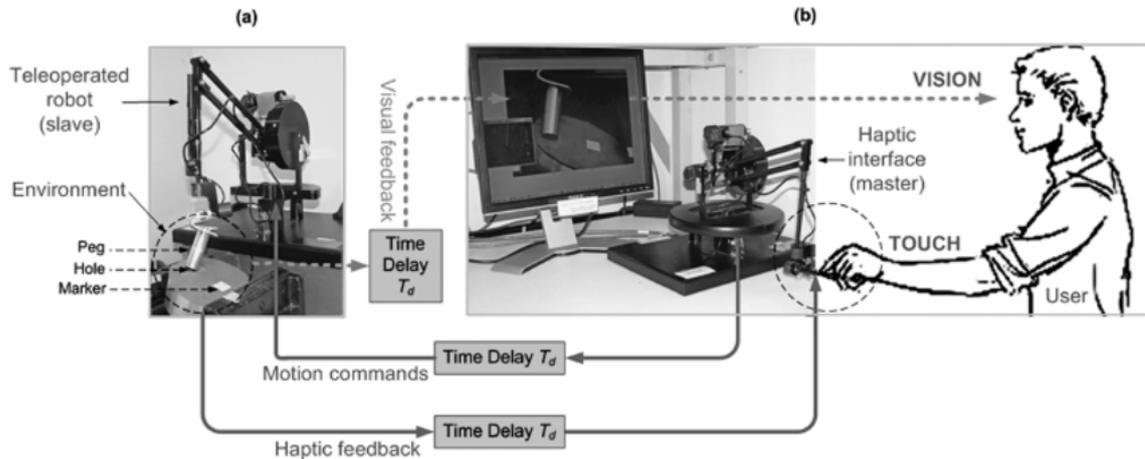


Figura 2.12: Sistema teleoperado master-esclavo[15]

La interacción de fuerzas en el dispositivo que hace de maestro (véase figura 2.12 b), se transmiten a la parte esclava (véase figura 2.12 a), a través de un canal de comunicación retardada, la posición y la fuerza ejercidas por el maestro y el esclavo son iguales, Una vídeo dirigido desde el entorno del esclavo proporciona una percepción visual de las interacciones entre el esclavo y el medio ambiente para el operador, y está deliberadamente diseñado para sufrir el mismo retardo de los canales de comunicación como la regeneración de la fuerza.

También se utiliza exoesqueletos como sistemas teleoperados donde el operador utiliza el exoesqueleto y los movimientos realizados por el mismo son reproducidos en un esclavo, pudiendo ser este únicamente un brazo, Las fuerzas aplicadas sobre un brazo robótico por el entorno son reflejadas al maestro y aplicadas por los actuadores del exoesqueleto en el brazo del operador.

Dispositivos hápticos

Una interfaz háptica es aquella que permite manipular objetos en interfaces virtuales o simuladas, en este contexto el termino háptico se refiere a los exoesqueletos capaces de proporcionar re-alimentación de fuerza o tacto al operador, de esta forma, un objeto en un entorno simulado puede ser explorado o palpado por el operador, mientras la estructura exoesquelética y sus actuadores proporcionan fuerza de realimentación, emulando objetos reales tanto en sus propiedades mecánicas como de textura (véase figura 2.13).

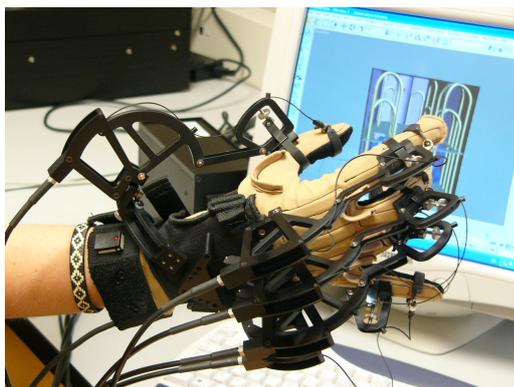


Figura 2.13: Interfaz Háptica Cybergrasp[16]

2.4.2.3. Rehabilitación y entrenamiento motor

En neuro-rehabilitación existen diversas terapias para los pacientes que tienen las extremidades superiores paralizadas a causa de lesiones del sistema nervioso, central o periférico, el objetivo de estas terapias es recuperar la función motora de los pacientes, mejorar la coordinación de los movimientos, o prevenir lesiones secundarias.

Siendo las técnicas de rehabilitación en su mayoría mecánicas y repetitivas se pretende reiniciar el proceso de re-aprendizaje interno del sistema nervioso, uno de los mecanismos exoesqueleticos para ayudar con este proceso es el "Lokomat ¹", el cual se ha establecido como una intervención efectiva para mejorar la función de caminar sobre el suelo en pacientes con fallos neurológicos en todo el mundo mostrando resultados positivos y mejoras funcionales.[17] (vease figura 2.14)".



Figura 2.14: Exoesqueleto para rehabilitación[17]

¹<https://www.hocoma.com/world/en/products/lokomat/>

Si se considera que el proceso de rehabilitación podría ser una rutina repetitiva de carácter mecánico, las aplicaciones robóticas podrían ayudar en este tipo de procesos. Diversas investigaciones han demostrado diferentes resultados y beneficios en terapias de neuro-rehabilitación usando plataformas robóticas, evaluando la rehabilitación en pacientes principalmente con apoplejía mediante el uso de dispositivos robóticos como exoesqueletos[18] [19].

2.4.2.4. Estudio e investigación

Biomecánica

Ya que los exoesqueletos en sí, son diseñados para emular los movimientos humanos o ayudar a los mismos a desarrollar actividades de mayor exigencia física, debe ser antropomórfico y ergonómico, tanto en forma como en función, por tales motivos es necesario conocer la biomecánica del cuerpo y sus cambios ante las diferentes actividades que se valla a realizar ya sea caminar, correr levantar carga o cualquier actividad que involucre esfuerzo para el ser humano.

Esto puede ser observado al momento de colocar un actuador, ya que el mismo se debe colocar en la misma posición que corresponde a la de los humanos, siendo su objetivo principal simular el movimiento de los músculos durante su actuación, esto se ejemplifica en la figura 2.15 donde se trata de compensar mediante actuadores los esfuerzos en los músculos de la pierna.



Figura 2.15: Diseño biomecánico de una prótesis robótica[20]

Neurofisiología y control motor

Las diferentes investigaciones llevadas a cabo para recuperar capacidades neuro-motoras de un ser humano se están empezando a realizar mediante exoesqueletos los cuales pueden aplicar fuerza, o provocar estímulos al cuerpo humano.

Este es el caso de «Ekso Bionics¹», el cual cuenta con un motor eléctrico, batería recargable, y varios sensores que permiten a un discapacitado ponerse en pie y andar, la diferencia entre este exoesqueleto y los anteriormente mostrados es que no está diseñado para portar cosas o realizar recorridos su programación y diseño están orientados únicamente a la rehabilitación del individuo.

Un terapeuta aplica dos modos principales de rehabilitación: ejercicios pasivos y con asistencia activa. En el primero el tratamiento mueve el brazo mediante movimientos específicos evitando que el paciente realice fuerza alguna, en el segundo modo de rehabilitación el paciente realiza diferentes movimientos hasta que los complete en caso de no poder completar el mismo el paciente realizará el movimiento hasta donde no pueda aplicar más fuerza, en este instante el tratante ayudara a completar el movimiento [21].

Ya que las actividades mencionadas anteriormente son de carácter repetitivo y mecánico, se pueden utilizar exoesqueletos para de esta manera ayudar al sistema nervioso durante el proceso de re-aprendizaje. también realizando movimientos visualmente guiados se puede realizar el análisis de varias áreas sensoriales y motoras del cerebro permitiendo al sistema exoesquelético cuantificar y manipular características del movimiento.

Transtornos de movimiento

Los trastornos de movimiento como la discinesia, son alteraciones presentadas por el cuerpo, provocando movimientos involuntarios, aunque su tratamiento varía de persona a persona y no siempre la causa es la misma, en este aspecto se puede utilizar exoesqueletos durante la rehabilitación, los cuales proporcionarán información de los diferentes movimientos como; frecuencia con la cual se producen, sitio en el cual se genera la señal etc, dando al tratante la posibilidad de una rehabilitación más acertado.

De esta manera el exoesqueleto también puede ser usado en un entorno experimental el comportamiento o reacción que tiene un número determinado de pacientes ante diferentes medicamentos suministrados o dosis de los mismos.

2.4.3. Power Asist suit Kawassaki

La empresa Kawassaki ha venido desde años atrás desarrollando tecnologías para facilitar el transporte o carga de objetos pesados es así que en el año 1991 desarrollan un brazo robótico y en 1994 una pierna robótica siendo su objetivo final un traje que sea capaz de ayudar al usuario en el transporte de objetos.

El brazo robótico figura 2.16, estaba compuesto por actuadores neumáticos utilizando tubos de goma para rotar el codo y de un mango del mismo material para detectar el movimiento del brazo, luego de varias pruebas y basados en los mismos principios de funcionamiento en el año de 1994 desarrollaron la pierna[29].

¹<http://eksobionics.com/healthcare/healthcare-professionals/>



Figura 2.16: Brazo accionado de forma neumática Kawasaki[29]

En el año de 1994 la misma empresa desarrollo un traje de potencia compuesto por ambos brazos piernas y cintura basados en el mismo principio usado anteriormente mediante actuadores de goma, los brazos, cintura y las piernas estaban conectadas entre si y de la misma manera estaban enlazados al usuario, dándole el aspecto de traje cuyo esquema se muestra en la figura 2.17.

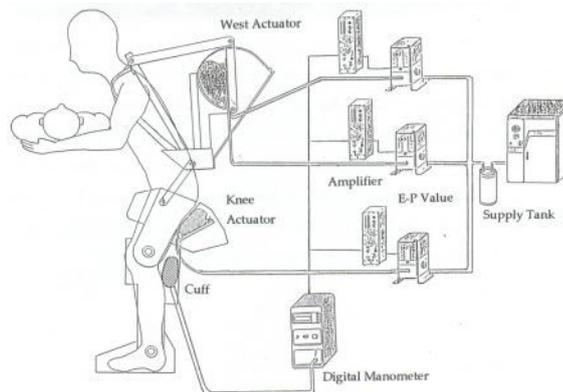


Figura 2.17: Esquema de traje Kawasaki[29]

En los diseños primarios el traje había estado sujeto a cables para su alimentación y mangueras para el suministro de aire, hasta que en el año 2002 realizaron una disminución considerable del gasto de energía y de los sistemas de suministro de aire logrando que estos sean portables, Los brazos, las piernas y la cintura tienen actuadores rotativos neumáticos, accionados directamente con bombas de aire micro suministrados por baterías portátiles de níquel, las fuerzas musculares se detectan con un nuevo sensor de dureza del músculo que utiliza una punta sensora montada en un dispositivo de película gruesa de polímero, el microordenador incorporado calcula el par necesario para mantener una posición de acuerdo con las ecuaciones derivadas de la mecánica del cuerpo; utilizando los ángulos de la articulación y el par necesario, se combina con las señales de salida para crear las órdenes de control.

En los años próximos iría evolucionando a tal punto que en el año 2005 la empresa contaba con un traje mucho mas robusto, de esta forma, a avanzado continuamente logrando en los últimos años, diseñar un exoesqueleto acorde a los objetivos planteados, quedando tal como se observa en la figura 2.18.



Figura 2.18: Exoesqueleto desarrollado por la empresa Kawassaki

2.5. Efector Final

En las aplicaciones industriales para completar el trabajo de un robot, se debe utilizar dispositivos adicionales, los cuales se podría denominar como periféricos del robot, El termino para denominar al dispositivo que esta unido a la muñeca dentro de robótica es actuador final, esta herramienta esta diseñada especialmente para hacer que el robot de propósito general cumpla con una tarea en específico[22].

Los efectores finales pueden dividirse en dos grandes grupos:

- Herramienta
- Gripper(Pinza)

2.5.1. Herramienta

Las herramientas utilizadas como efectores finales, están diseñadas para cumplir con una tarea determinada en la cual deba aplicarse una operación sobre las piezas de trabajo algunas de estas herramientas y su función se indican en la tabla 2.2. En cada uno de estos casos la pieza trabaja unida a la muñeca para llevar a cabo el trabajo asignado.

Herramienta	Función que desarrolla
Soldadura por punto	Dos electrodos que se cierran sobre la pieza a soldar
Soplete soldadura de arco	Aportan el flujo de electrodo el cual se funde
Cucharón para colada	Se utiliza en trabajos de fundición
Atornillador	Suelen incluir la alimentación de tornillos
Fresa-lija	Para perfilar, eliminar rebabas, pulir, etc
Pistola de pintura	Para pulverización de la pintura
Cañón láser	Para corte de materiales, soldadura o inspección
Cañón de agua a presión	Para corte de materiales

Tabla 2.2: HERRAMIENTAS TERMINALES PARA ROBOTS [22]

Algunos de los manipuladores se pueden apreciar en las imágenes de la figura 2.19.

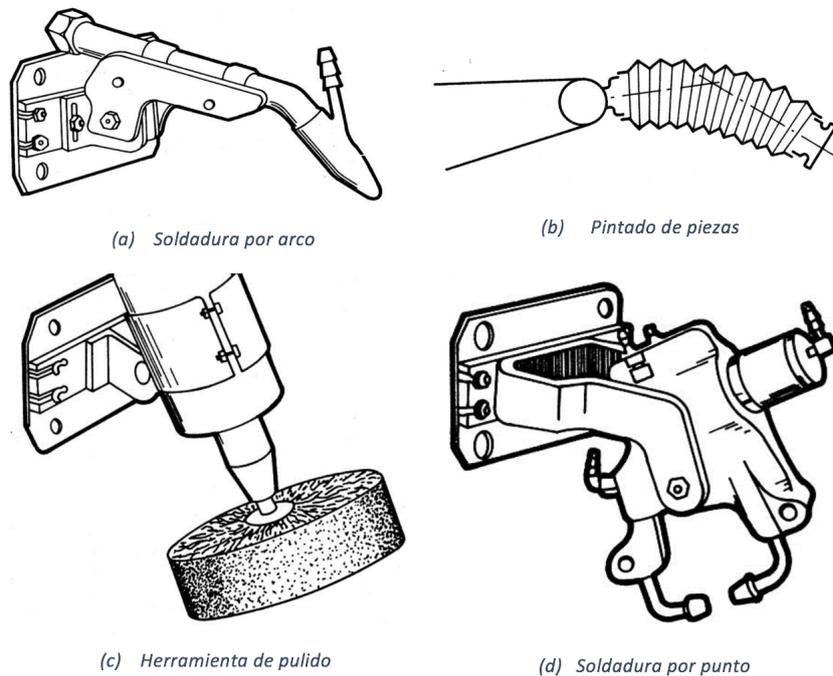


Figura 2.19: Efectores finales utilizados frecuentemente

2.5.1.1. Criterios de selección de manipuladores

Existen varios criterios de selección para los manipuladores entre los cuales los más usados son:

- **Trabajo de la pieza al ser manejada.-** En función del robot en el cual se utilizará la pieza se definen las características de la misma tales como:
 - Dimensiones
 - Masa
 - Potencia
 - Geometría de robot
- **Actuadores** Los actuadores pueden ser de diferentes características dependientes de las piezas con las que se va a trabajar pudiendo ser de tipo:
 - Mecánico
 - Al Vacío
 - Magnético
- **Materiales** En función del entorno y piezas en las cuales se va a trabajar se puede definir el material del efector final

- Fuerte, rígido, durable
 - Esfuerzo continuo
 - Costo y factibilidad en producción
 - Coeficiente de fricción
 - Entorno de trabajo
- **Posicionamiento** En aplicaciones industriales es importante que exista repetibilidad en la cadena de producción por lo cual el efecto final se elige dependiendo de:
- Longitud del manipulador
 - Longitud del robot y repetibilidad
 - Tolerancias

De forma tradicional se puede decir según [23] que los criterios de selección de una herramienta son:

1. Un Peso bajo para permitir tener un manejo de carga más útil, aumento de las aceleraciones, disminución en tiempo de ciclo.
2. Dimensiones mínimas dispuestas por el tamaño de la pieza de trabajo.
3. Rango más ancho de las partes de la pieza usando inserciones, y movimientos ajustables.
4. Rigidez para mantener la exactitud del robot y reducir vibraciones.
5. Fuerza máxima solicitada, seguridad y prevención de daño en los productos.
6. La fuente de poder debe estar prontamente disponible para el robot.
7. El mantenimiento debe ser fácil y rápido.
8. Formas de seguridad para que el material no se deje caer cuando falte la fuente de poder.

Mediante estos criterios de selección en nuestros efectores finales podemos adaptar nuestro mecanismo a las funciones que deseemos, es por tal motivo considerar estos criterios entre otros que se puedan presentar al momento de seleccionar el efecto final con el cual se va a trabajar.

2.5.2. Pinza Robótica

También conocidas como efecto final en un brazo robótico; es un sistema cuya principal función es sujetar objetos para obtener un determinado efecto, es importante que las pinzas que se desarrollen puedan moverse en tres dimensiones a fin de que se imite la movilidad de una mano real.

Considerando los grados de libertad, una pinza robótica puede tener un máximo de seis grados de libertad, expresados en tres movimientos lineales en las tres dimensiones de los ejes x , y , z y tres movimientos giratorios α_1 , α_2 y α_3 , sobre los ejes x , y , z . Se puede considerar que existen máquinas que

cuentan con más de 6 grados de libertad, a las cuales se les conoce como grados de libertad mecánica o de libertad de movimientos[24], Según Bejcy¹ una mano humana puede ejecutar movimientos con seis grados de libertad tal como se muestra en la figura 2.20.

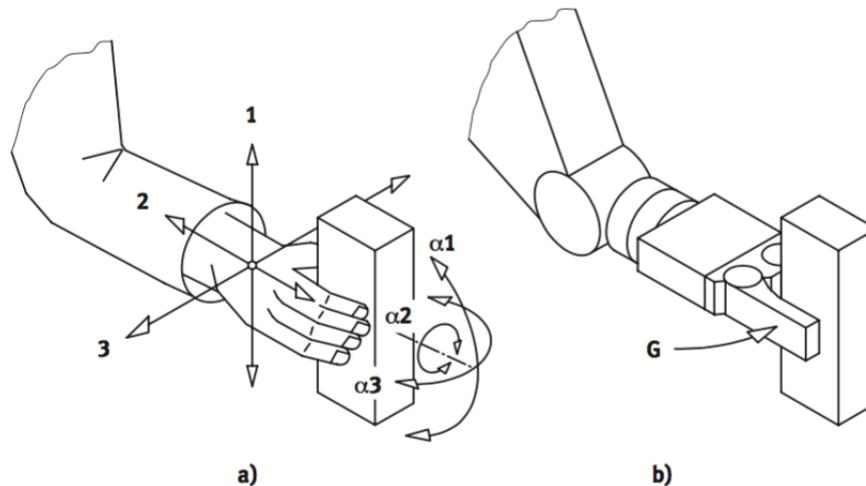


Figura 2.20: Grados de libertad en una mano humana

Los movimientos de empuje se describen como:

1. Vertical, arriba/abajo
2. Frontal, avance/retroceso
3. Lateral, izquierdo/derecho

De otra manera siguiendo la terminología de aviación según [25] se describen como:

1. $\alpha 1$ Cabeceo, inclinación
2. $\alpha 2$ Balanceo, torsión
3. $\alpha 3$ Guiñada, giro

Es importante tener en cuenta que los dedos de la pinza no son considerados como un grado de libertad debido a que su movimiento no interviene en la trayectoria de la pinza.

Los sistemas de pinzado pueden ser de diferentes formas dependiendo del elemento u objeto que se desee sostener o con el cual se vaya a laborar; un ejemplo de esto son las diferentes pinzas industriales que se elaboran para un determinado uso, tal es el caso de la figura 2.21, en el cual se ha colocado dos pinzas estándar sobre un placa basculante en la cual un cilindro neumático es el que se encarga del movimiento, el cambio de pinza se hace de forma rápida y se amortigua en el final del recorrido. Este tipo de soluciones suele ser utilizada en equipos de verificación.

¹<https://www-robotics.jpl.nasa.gov/news/newsStory.cfm?NewsID=117>

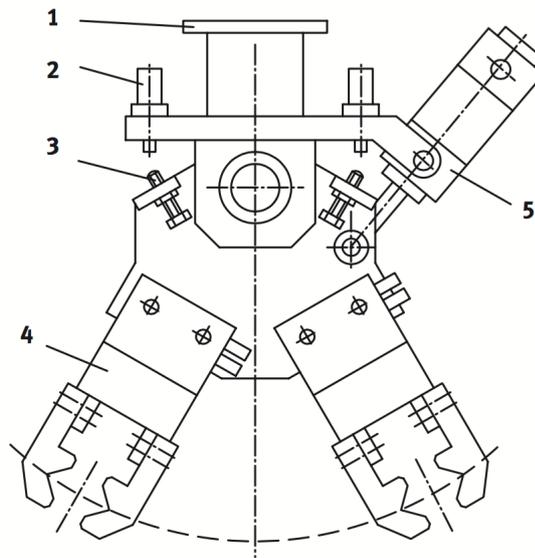


Figura 2.21: Doble pinza basculante[24]

2.6. Pinza robótica como prótesis

Se ha llevado a cabo diferentes estudios y diseños de prótesis mecánicas como [26] [27] para sustituir dedos o la mano de un ser humano, tomando en consideración que en los mismos no se considera el peso de la misma ya que se considera que parte de la mano o en su defecto el resto del miembro superior(brazo) se encuentra completo.

Basados en el principio de reconstrucción total de la mano se ha desarrollado prótesis como las que se muestran en la figura 2.22, la figura (a) ha sido desarrollada por ottobock¹, la cual puede sujetar objetos de diferente tamaño ya sean frágiles o robustos, esto se debe a que está combinada con prótesis de brazo con control mioeléctrico.



Figura 2.22: Protésis de manos completas

Su característica especial reside en que las manos pueden abrirse para reaccionar rápidamente en tiempo real, de forma similar a como lo hace una mano natural. Esto le brinda espontaneidad y la capacidad

¹<http://www.ottobock.es/>

de usar su mano protésica de forma casi natural.

Característicamente las prótesis mioeléctricas desarrolladas por ottobock reaccionan normalmente a las respectivas señales musculares del usuario. Las manos SensorHand Speed y VariPlus Speed de la misma empresa disponen de hasta seis programas de control que permiten adaptar la prótesis de mano a las necesidades particulares del usuario, dependiendo de si el usuario puede controlar una o dos señales musculares. Su capacidad de control permite ajustar la fuerza y la velocidad de agarre [28].

En la Figura 2.22 (b), se muestra la mano desarrollada por la Universidad Manuela Beltrán, el prototipo cuenta con 5 grados de libertad que al igual que la mano de la figura 2.22 (a), se controla a través de señales mioeléctricas, las cuales se emiten al momento que el musculo del antebrazo se contrae, logrando de esta forma movimientos sencillos y básicos; para lograr el control y manipulación de la misma se necesita de un proceso de adaptación mediante terapias.

El desarrollo de estos prototipos esta orientado a la existencia del brazo y ante brazo en la persona, lo cual hace posible la lectura de señales mioeléctricas para el control de movimientos.

MATERIALES Y MÉTODOS

3

MATERIALES Y MÉTODOS

3.1. Hardware

Para el diseño e implementación de la pinza tanto en la parte mecánica como electrónica se ha utilizado las siguientes herramientas:

- Solid Works¹
- Meshmixer²
- Labcenter Electronics(Proteus y Ares)³

El software Solid Works, se utilizó para crear un diseño en 3D de la pinza el cual posteriormente se lo tendría en físico mediante la impresión 3D, este procedimiento se lo lleva a cabo con el objetivo de variar el diseño he ir mejorando cuantas veces sea necesario hasta obtener un prototipo lo más óptimo posible, en tanto que el software de Labcenter, se utilizó para diseñar los circuitos eléctricos a los cuales irán conectados, tanto los sensores de fuerza como el servomotor.

3.2. Fundamentos para el diseño de la pinza

El diseño de la pinza se basó en las formas y pesos de objetos que se quería ha llegar a sujetar con la misma, ya sea desde sujetar una cuchara, botella, vaso u otro tipo de objetos habituales.

3.2.1. Puntos de contacto

Cada objeto tiene su forma pero el diseño de la pinza no se centra en un objeto en particular a diferencia de ello se basará en varias formas, pudiendo ser: planas, redondas, esféricas, cilíndricas, cúbicas, etc(vease figura 3.1). Es decir la pinza debe tener una forma general que englobe las diferentes presentaciones de los objetos a sujetar.

¹<http://www.solidworks.es>

²<http://www.meshmixer.com>

³<https://www.labcenter.com>

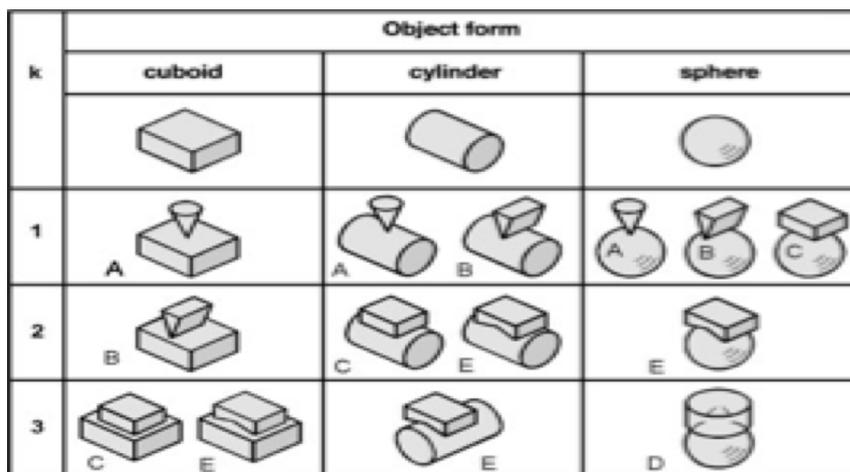


Figura 3.1: Formas de objetos posibles[32]

Otro de los factores a tener en cuenta es la fuerza que se va a aplicar, cantidad de sensores con los cuales se va a contar y región donde van a estar ubicados los mismos, teniendo en cuenta las diferentes formas que puede haber, en la figura 3.1, se puede apreciar los diferentes puntos de contacto que puede proporcionar una pinza teniendo en cuenta su diseño.

A efectos de puntos de presión considerados, se podría decir que la mejor forma de ajustar un objeto sería la forma <E> tanto para el objeto cubico, cilíndrico y esférico sin embargo este tipo de agarre sería perjudicial al momento de medir la fuerza que se aplique a un objeto, ya que si se toma de ejemplo un bolígrafo este quedaría perfectamente ajustado al contorno de las puntas de la pinza con la forma <E> de la figura 3.1, pero si se tomase un objeto de un diámetro mayor o el área de contacto sea mayor que el espacio entre los puntos de presión, se vería afectado el contacto del objeto con el sensor de fuerza, lo cual podría provocar error en las mediciones del sensor o inclusive la inexistencia de variación en las mediciones que se tome del mismo.

Es por tal motivo que la mejor opción que se podría tomar en torno a la forma de la punta de los dedos y lugares donde fuese a existir sensores de fuerza, es la <C> de la figura 3.1, la ventaja de esta forma es que al poseer una cara plana puede tomar cualquier tipo de objeto aplicando la misma presión en cada uno de ellos, sin importar su área de contacto debido a que este diseño se implementa con un único punto de contacto.

3.2.2. Número de dedos a utilizar

1. Si se llegase a utilizar un único dedo (véase figura 3.2 A), sería necesario un objeto inmóvil al cual se fuese a transmitir la fuerza aplicada, dando como resultado una limitación al tamaño y la forma de los objetos que la pinza pudiese sostener, a la fuerza que se pudiese aplicar sobre los mismos y las dimensiones que el objeto o dedo inmóvil debiese tener para asegurar la estabilidad del objeto.

2. Con dos dedos (véase figura 3.2 B) se eliminarían los problemas planteados anteriormente sin embargo si el objeto es sometido a una fuerza externa el objeto pudiese desestabilizarse, resbalar y caer, si los dedos no están perfectamente alineados pudiesen producirse fracturas en la pinza al aplicar demasiada fuerza, también resultaría complicado sujetar objetos que no tuviesen una cara plana o que fuesen redondos, sería necesario un mayor control para poder sostener perfectamente el objeto con el cual se trabaja.
3. La posibilidad de trabajar con tres dedos (véase figura 3.2 C) parecería resolver los problemas de los apartados uno y dos sin embargo se sigue manteniendo el posible error mencionado en el apartado dos que si no se sitúa exactamente la pinza donde se encuentra el objeto este se deslizaría dificultando la sujeción del mismo otro inconveniente sería en el caso de sostener un objeto en forma cónica; tomando en consideración que se va a trabajar con un único motor y tres dedos nos da como resultado que uno de los dedos estaría fijo o dos de ellos se moverían uniformemente, no podría estar fijo ya se incidiría nuevamente e los problemas del apartado uno, y si se mueven uniformemente los sensores en la punta de los dedos presentarían variaciones en sus datos pudiesen ser estas diferentes en cada uno de ellos.
4. Como solución a los problemas mencionados en los diferentes apartados se optó por el uso de cuatro dedos (véase figura 3.2 D) basándose en las siguientes ventajas
 - **Control** El objeto al entrar en contacto con uno de los dedos y estando el mismo dentro del área de los demás en caso de ser arrastrado; se desplazará, hasta posicionarse en el centro entre los cuatro dedos y quedando presto para la manipulación del mismo.
 - **Agarre** En caso de usar la misma figura cónica indicada en el apartado 3 los sensores que primero entren en contacto con el objeto serán quienes transfieran los datos para el control de la pinza tomando como consideración que estos sensores deben pertenecer uno al dedo derecho y uno al dedo izquierdo.
 - **Estabilidad** Al contar con una área de contacto mayor se asegurará que exista más estabilidad.
 - **Datos de procesamiento** Al existir mayor número de muestras se puede optimizar el control en base a las mismas y las regiones de las cuales fueron proporcionadas ayudarán a dar una idea del tipo de objeto que se quiere sujetar y el control preciso para su efecto.

3.2.3. Fuerza de acuerdo a los puntos de presión

La fuerza que se va a aplicar sobre el objeto es de esencial importancia, ya que de no aplicarse de una manera controlada el objeto podría:

- Caer
- Sufrir daños en su estructura
- No se pueda tomar el objeto

- Dañar la estructura de la pinza

A manera de demostración sobre la fuerza que se puede aplicar a cada objeto se analizará la figura 3.2, y la superficie de contacto que cada dedo tuviese sobre el objeto. Ya que en el apartado anterior se determino que existiría un solo punto de contacto por cada dedo y el número de dedos sería igual a cuatro, en esta imagen se una perspectiva de como cada punto de contacto influiría sobre el objeto dependiendo de su forma y del número de dedos que apliquen fuerza sobre la misma.

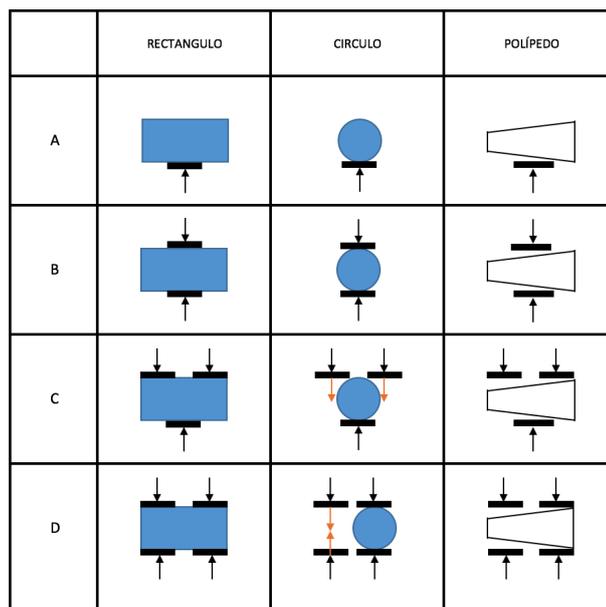


Figura 3.2: Puntos de aplicación de fuerza

La superficie de presión que cada dedo va a aplicar sobre el objeto viene determinado por la ecuación 3.1 tomada de [32], donde como fuerza máxima se aplicará un Newton y suponiendo que el punto de contacto del dedo es cuadrado y su lado mide 22 mm:

- p = Superficie de presión
- F_K = Fuerza aplicada
- a = Ancho del dedo
- b = Alto del dedo

$$p = \frac{F_K}{a * b} \quad (3.1)$$

$$p = 20,66 \quad (3.2)$$

Esta ecuación es únicamente en el caso de que el dedo entre en contacto con toda la superficie del objeto en caso de que solo sea un punto la superficie de presión viene determinada por la ecuación 3.3 donde:

- m = Masa del objeto
- E_r = Coeficiente de elasticidad
- r = radio del Objeto

$$p = m * \sqrt[3]{\frac{F_K * E_r^2}{r^2}} \quad (3.3)$$

En caso de que la superficie de contacto sea una línea la ecuación para determinar la superficie de presión esta determinada por 3.4.

- L = longitud de contacto

$$p = 0,418 * \sqrt{\frac{2 * F_K * E_r}{L * d}} \quad (3.4)$$

Al momento que se realiza el contacto entre la pinza con el material también va a existir una deformación o aplanamiento en la superficie de la pinza la cual puede ser determinada por la ecuación 3.5 tanto para objetos esféricos o cilindros.

- σ = Total de aplanamiento
- E = Módulo de Young
- r = radio de curvatura de la esfera

$$\sigma = 1,23 * \sqrt[3]{\frac{F_K^2}{E^2 * r}} \quad (3.5)$$

En caso de que los módulos de Young sean diferentes se determina E con la ecuación 3.6

$$E = \frac{2 * E_1 * E_2}{E_1 + E_2} \quad (3.6)$$

De la misma forma si existe curvatura entre la pinza y el objeto se determina el radio de curvatura a través de la ecuación 3.7

$$\frac{1}{r} = \frac{1}{r_1} + \frac{1}{r_2} \quad (3.7)$$

De esta manera se puede determinar la superficie de presión en diferentes objetos y de acuerdo a como estos sean sujetados por la pinza; de tal forma que si aplicamos una determinada fuerza al motor se pueda asimilar la fuerza transmitida al objeto a través de la superficie de presión.

Ya que no todos los objetos se los puede tomar con tan solo la punta de los dedos se ha decidido agregar un sensor más por cada dedo en el centro del mismo, logrando de esta forma facilitar el agarre de objetos de mayor tamaño como: botellas, vasos, o cualquier otro objeto con diámetros parecidos o mayores.

3.2.4. Desplazamiento para el agarre

Como desplazamiento para el agarre se entenderá a la forma en la cuál se va a abrir y cerrar la pinza tomando en cuenta las aportaciones en física hechas por [33] y los siguientes aspectos:

- Área a ocupar
- Máximo de apertura
- Desgaste Mecánico
- Fuerza máxima
- Objetos a sujetar

Teniendo en consideración que se va a trabajar con un único servomotor el desplazamiento tiene que ser capaz de completar sus movimientos con el mismo, en este sentido se analizarán tanto el desplazamiento angular y desplazamiento rectilíneo, ventajas y desventajas de ocupar cada uno de estos desplazamientos.

3.2.4.1. Área a ocupar

El área a ocupar por parte del dispositivo debe ser proporcional al resto del brazo, para evitar sobreesfuerzos por parte de los motores que pertenecen a las articulaciones del brazo, antebrazo y muñeca.

Desplazamiento Angular En el desplazamiento angular; el área a ocupar por parte del dispositivo será la necesaria para ubicar el servomotor o el valor correspondiente a la suma de los diámetros de los piñones que se utilice.

Desplazamiento rectilíneo En este tipo de desplazamiento el área depende de la apertura máxima que se desee tener de la pinza en caso de usar un desplazamiento por tornillo, si el desplazamiento se hiciese a través de una circunferencia sujeta al piñón del motor el área sería la correspondiente al desplazamiento lateral producido al girar el motor.

3.2.4.2. Máximo de Apertura

La apertura máxima que se pretende lograr en la pinza es la necesaria para sostener por ejemplo una botella de dos litros de capacidad o para sostener un vaso de agua.

Desplazamiento Angular La apertura máxima podría estar determinada por las dimensiones de los dedos al punto máximo donde aún sea posible el agarre o por topes mecánicos o digitales con los cuales cuente la pinza.

Desplazamiento Rectilíneo La máxima apertura en este desplazamiento está determinada por el ancho de la pinza.

3.2.4.3. Desgaste Mecánico

El desgaste mecánico que se pueda producir depende de la cantidad de veces que se abra y cierre la pinza ya que es una pinza de asistencia se asume que el desgaste va a ser el mismo tanto para un desplazamiento angular como para un desplazamiento rectilíneo sin embargo la diferencia radica en el número de componentes que sufran este desgaste, pudiendo ser en el primer caso de los ejes de giro y de los piñones de giro que estuviesen adheridos a los dedos para producir el movimiento de los mismos, en tanto que en el desplazamiento rectilíneo será las guías por las cuales se desplaza el eje produciéndose un desgaste similar también de este.

3.2.4.4. Fuerza Máxima

La fuerza máxima que puede proporcionar la pinza, estará determinada por el sitio en el cual se aplique la fuerza o también podría ser determinada por el torque máximo que pudiese proporcionar el motor.

Desplazamiento Angular Para este desplazamiento la fuerza es transmitida de forma radial, de modo que la resistencia que aplique el objeto sobre la pinza deforme la pinza de acuerdo a la elasticidad de la misma (véase figura 3.3).

Desplazamiento Rectilíneo El desplazamiento rectilíneo al aplicar fuerza para efectuar el cierre de la pinza deforma más el material de la pinza que en un desplazamiento angular, esto se debe a la transmisión de fuerzas entre el punto que se aplica la misma y el punto de aplicación de la pinza tal como se muestra en la figura 3.3

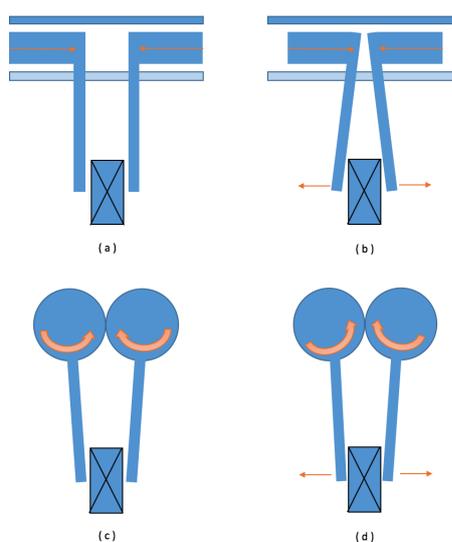


Figura 3.3: Desplazamientos Angular y Rectilíneo

3.2.4.5. Objetos a sujetar

Los objetos a utilizar como modelo de prueba han sido determinados por Amir¹, las diferentes necesidades han sido sometidas a análisis pudiendo ser algunas de ellos factibles, otras a ingresar en modelo de prueba y otras para trabajos futuros, estas actividades se las puede observar en la tabla 4.1.

Acción	Necesidad	Factibilidad
Capacidad de pinzar un objeto y meter/sacar en/de un/a bolso/a	alto	No
Recorrido de ascensión y descenso 300 mm con carga	alto	Posible
Capacidad de verter como Leche en una taza	alto	Si
Recorrido de giro para verter y retorno 95 grados	alto	Si
Capacidad de usar calzador	alto	Posible
Capacidad de abrochar y desabrochar botones	bajo	No
Capacidad de subir y bajar cremallera de 1 pantalón	alto	Posible
Capacidad de poner / quitar las gafas	alto	Si
Capacidad de abrir y aguantar una paraguas	bajo	Posible
Capacidad de meter y sacar el plato de la comida en/de la nevera o microondas	alto	Si
Capacidad de meter y sacar una taza de café en/de la nevera o microondas	alto	Si
Capacidad de conectar/desconectar un enchufe en una base de enchufe	alto	Posible
Capacidad de poder limpiarse después de defecar en un bidet	alto	No
Capacidad de teclear el teclado	alto	Si
Capacidad de manejar el volante y los mandos de un coche	alto	No
Capacidad de usar un lápiz para un teclado táctil en ascensores, móviles etc.	alto	Si
Capacidad de tomar medicamentos, píldoras, pomadas, gotas	alto	Posible
Capacidad de rascarse	alto	No
Capacidad de poder introducir la llave en su ranura y abrir/cerrar la puerta	alto	Posible

Tabla 3.1: Acciones a realizar, necesidad y posibilidad de realizarla

El paciente a dado un grado de necesidad alto o bajo en función del número de veces que realiza una determinada actividad o la importancia que tiene la misma en su vida cotidiana, la factibilidad se ha esquematizado en tres factores siendo estos; posible, si y no, considerando a la circunstancia posible que se podría realizar pero sin embargo no en la entrega del primer prototipo y a la factibilidad (si) que se espera entregar en el primer prototipo, la factibilidad utilizada como (no) se refiere a que en los primeros proyectos o en la presente entrega no se la va a considerar sin embargo en trabajos futuros pudiese ser analizada y desarrollar ya prototipos incluyendo estas actividades.

Para la decisión de si una tarea es factible o no se ha basado en:

- Control Que un ser humano normal requiere para desarrollar la actividad
- Esfuerzo que realizar la actividad supondría para el exoesqueleto

¹Amir Paciente con parálisis en las extremidades superiores, este paciente proporciono las necesidades básicas que precisa

- Peso que soportaría la pinza
- Fuerza necesaria para realizar la actividad
- No generación de singularidades por parte del brazo
- Contacto directo con el cuerpo u otro ser humano

Considerando que se han analizado todos los puntos hasta ahora expuestos, ventajas y desventajas de cada desplazamiento se ha decidido por utilizar un desplazamiento angular, el cual fijara su movimiento utilizando el servo motor como eje de rotación, proporcionando un movimiento como se muestra en las figuras 3.3 C y D.

3.2.5. Tamaño de los dedos

El tamaño de los dedos esta en función de los objetos que se pretenda tomar con el mismo o de las actividades que vaya a realizar; en el capítulo anterior se mostró las actividades que va a realizar la pinza lo cual dará una perspectiva de el tipo de objetos que va a sujetar y de la proporción y peso de estos.

Ya que entre todos los objetos el más grande pudiese ser una botella de 2 Litros, la pinza tiene que ser capaz de abrirse lo necesario para tomar este objeto y que al momento de cerrarse tenga la suficiente fuerza para sostener el objeto, y pudiese ser manipulado.

3.3. Diseño de la pinza

El diseño e impresión del prototipo permitirá evaluar los planteamientos anteriormente hechos, dando la posibilidad de realizar varias pruebas o re-diseño en caso de que este sea herrado en un principio.

Para el diseño en tres dimensiones de la pinza se utilizó el software Solid works en el cual se han construido las piezas por separado y mediante la impresión 3D se ha unido todo para obtener el prototipo.

El diseño esta compuesto de las siguientes partes:

- Cara inferior de la pinza
- Cara superior de la pinza
- Componente Izquierdo y derecho compuestos por dos dedos cada uno
- Ejes de rotación

3.3.1. Cara inferior de la pinza

La cara inferior de la pinza es donde irán asentados tanto el circuito electrónico para los sensores y los ejes de rotación.

Este objeto que forma parte de la pinza esta dispuesto con cinco agujeros, de los cuales; tres serán utilizados por los pernos que sujetarán la cara superior con la inferior, y, los dos centrales que servirán como eje de rotación, la distancia entre un eje de rotación y el otro esta determinada por el tipo de piñones que se va a utilizar y la profundidad de la parte dentada de estos piñones.

Ya que uno de los objetivos de hacer este tipo de pinza era hacerla ligera y el área ocupada por la estructura para su movimiento fuese lo más pequeña posible; se ha utilizado piñones de 25,6 mm de diámetro cuyas dimensiones se muestran en la figura 3.4

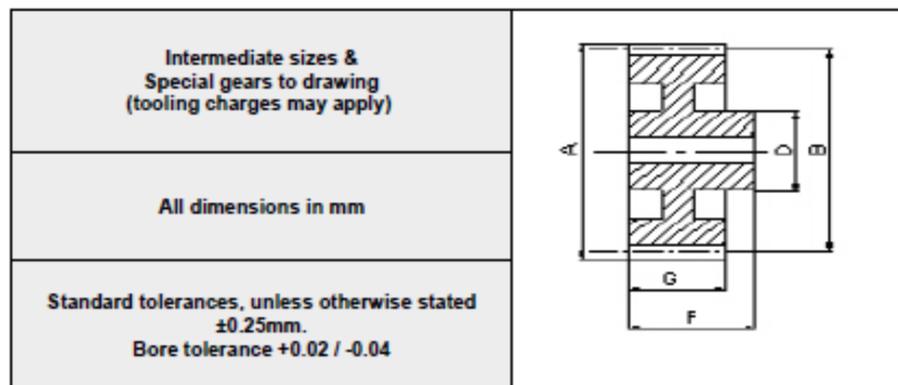


Figura 3.4: Dimensiones de piñones[34]

Las dimensiones y composición del componente se corresponden de la siguiente manera:

- Número de dientes = 30
- A = 25,6 mm
- B = 24 mm
- D = 12 mm
- F = 9 mm
- G = 4 mm

Ya que la distancia de B corresponde al punto medio de un diente de un extremo al medio de su contrapuesto, por lo tanto matemáticamente se obtiene que la distancia entre el punto medio del eje de un piñón y el otro es igual a 24 mm, quedando unidos de tal forma que no se produzca un rozamiento descontrolado al momento de ejecutar el movimiento de los dedos o se pueda ver afectado la estructura

del piñón con el uso cotidiano.

Conocidos los sitios donde estarán ubicados los ejes del movimiento se determina las dimensiones de la cara inferior sumando las áreas de los piñones más las áreas de los pernos y una perforación la cual servirá de guía para que no desvíen su trayectoria los dedos cuando se estén moviendo, quedando la pieza como se muestran en la figura 3.5



Figura 3.5: Cara inferior de la pinza

3.3.2. Cara superior de la pinza

Sobre la cara superior de la pinza se ha posicionado el servo motor que se encarga del movimiento de los dedos, también está ubicado uno de los ejes de movimiento, el segundo eje corresponde al motor, de la misma forma que en la cara inferior cuenta con perforaciones para los pernos que aseguran la estabilidad de la pinza.

Para evitar que exista incremento del espesor en la estructura de movimiento de la pinza, se ha resaltado los puntos de agarre del servomotor, y se ha disminuido el espesor en la superficie del objeto, quedando el espacio necesario para el servomotor y los agujeros para que atravesase el piñón de movimiento correspondiente al mismo.

Las dimensiones de la cara superior en cuanto a su espesor, ancho y alto son las mismas de la cara inferior y los agujeros centrales son céntricos con la misma, también contará con los agujeros que servirán de guías para el movimiento de los dedos.

Ya que va a realizarse impresión 3D para el modelado físico del prototipo la estructura debe tener en todo momento soportes para evitar deformaciones , es por este motivo que en los sitios donde va a estar ubicado el servomotor existen una especie de ramificaciones, estos son soportes creados por el programa meshmixer para realizar una correcta impresión del prototipo, de no utilizar los mismos la impresión no sería posible.

Tomando en cuenta los aspectos mencionados se diseño la cara superior (véase figura 3.6) de la pinza.



Figura 3.6: Cara superior de la pinza

3.3.3. Componente izquierdo y derecho compuesto por dos dedos cada uno

El componente tanto izquierdo como derecho serán relativamente iguales, la diferencia radica en que el movimiento generado en el componente derecho esta determinado por el motor, y el movimiento en el izquierdo esta determinado por el derecho.

Los piñones que se utilizo para el cálculo del área de las caras inferior y superior, serán ubicados en cada uno de los dedos ambos piñones están sujetos mediante pernos a la estructura de cada componente.

En la estructura del componente se ha hecho dos agujeros por los cuales pasará tanto los ejes de rotación como los componentes que servirán de guía para los cuales se dejo los agujeros en la cara inferior y superior.

En estos dispositivos al igual que en la cara superior se precisaba de soportes por las elevaciones que se presentan en ambos extremos de los componentes y los sitios donde van a ir ubicados los sensores.

Se ha diseñado una estructura para cada dedo en forma de trapecioide, con el fin de proporcionar un mejor agarre a componentes redondos o de dimensiones grandes, mejorar la fuerza que la pinza pueda aplicar sobre los objetos a sostener, y aplicar mayores puntos de presión en determinados objetos tratando de que esto no influya en el control de la pinza.

Existen agujeros hechos también, para ubicar los sensores de fuerza, la dimensión de estos corresponde a las medidas de los sensores a utilizar, es conveniente dejar un offset de 1 mm por lado a las dimensiones del sensor, esto se realiza debido a que en el momento de la impresión el material se expande y provoca que disminuya el ancho real interno y externo de la pieza.

Una de las características que posee la mano humana es que la piel tiene un coeficiente de rozamiento óptimo para cualquier material lo que disminuye la fuerza que se deba aplicar sobre el mismo para sostenerlo, en este mismo contexto se ha buscado un material que sea económico y preste las mejores características con vidrio, papel o plástico, siendo el caucho uno de los mejores materiales para este trabajo, por lo tanto se dejarán las ondulaciones necesarias en la mano para colocar este material.

Con los aspectos analizados se ha diseñado los componentes izquierdo y derecho con sus respectivos dedos, ondulaciones para el material anti- corrosivo y los sensores de fuerza (véase figura 3.7)

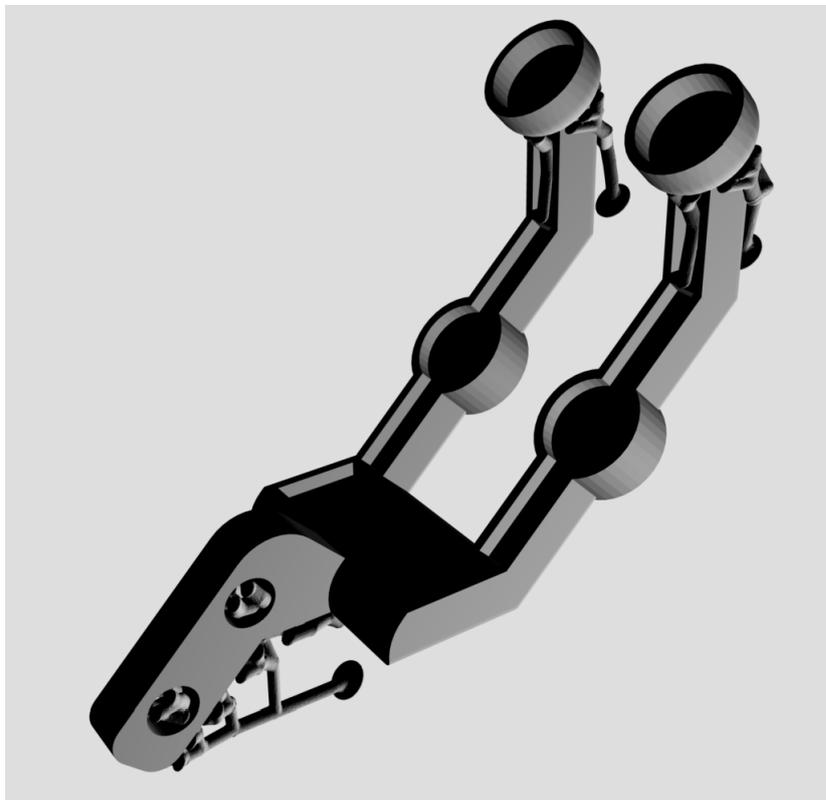


Figura 3.7: Componentes Izquierdo o derecho con sus respectivas partes

3.3.4. Ejes de rotación

Los ejes de rotación diseñados para el presente proyecto tienen tres diámetros diferentes el primero para ajustarse al agujero que se dejó en el componente de los dedos, el segundo para servir de tope mecánico entre la cara inferior y el eje que corresponde al tercer diámetro.

En la parte superior del eje que se mencionó en el párrafo anterior se ha realizado un agujero en el centro del objeto con el propósito de que en el componente izquierdo se introduzca una unión cuya función sea llegar al punto de rotación en la cara superior, y en el componente derecho sirva únicamente para fortalecer la unión entre el piñón y el componente derecho.

Las dimensiones de cada estructura pueden ser determinadas de acuerdo a los requerimientos del usuario o por factibilidad en su montaje, en este caso se ha utilizado las dimensiones que se muestran en el Anexo A, el presente proyecto ha utilizado un piñón de 6 mm de diámetro interno por lo tanto el diámetro de la unión dos será de 5 mm (1 mm se espera en expansión del material de impresión).

Por lo tanto el resultado del diseño de las uniones será el que se muestra en las figuras 3.8 y 3.9

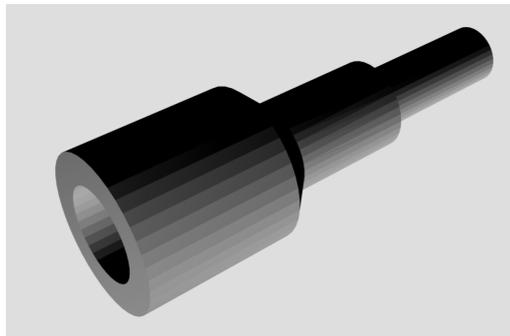


Figura 3.8: Unión entre los dedos y los ejes de la cara inferior

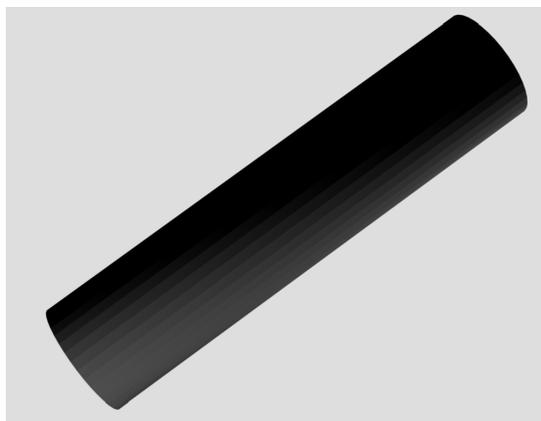


Figura 3.9: Unión entre los dedos y los ejes de la cara superior

3.4. Sensores de Fuerza

Para la calibración y adecuación de los sensores de fuerza, se ha utilizado diferentes masas las cuales representan las variaciones de fuerza que se pudiesen aplicar al sensor; el peso de las masas varía entre 10 g hasta 1000 g, se ajustara el resultado de las mismas con el fin de obtener a la salida una variación entre 0 y 3.3 Voltios, el mismo que permitirá la lectura de datos por parte de la tarjeta de programación.

Los sensores de Fuerza a utilizar son resistivos (véase figura 3.10), tienen una sensibilidad de hasta 1 kilogramo y su resistencia varía de entre 0 hasta 10 K Ω .



Figura 3.10: Sensor de Fuerza

Ya que su resistencia varía de entre 0 a 10 K Ω se puede utilizar un divisor de voltaje, de esta manera se determinaría la resistencia teniendo como salida el voltaje máximo soportado por el Pin de entrada de la tarjeta de adquisición de datos, también con el objetivo de que haya la menor perdida de datos posibles se ha añadido un amplificador operacional que servirá como acople de impedancias, para tener la mayor transparencia posible en la transmisión de datos.

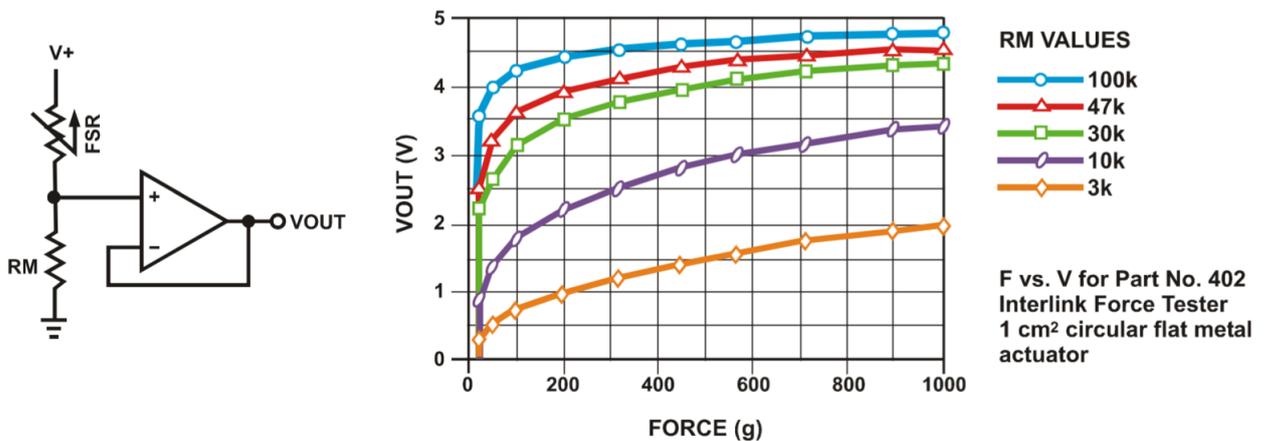


Figura 3.11: Divisor de Voltaje [35]

Para determinar la resistencia R_m del circuito que se muestra en la figura 3.11, se utiliza como referencia el esquema emitido por el fabricante, cuyos valores de resistencia vienen determinados por la ecuación 3.8.

$$V_{out} = \frac{R_2}{1 + \frac{RFSR}{R_m}} \quad (3.8)$$

En la fase de experimentación se pudo corroborar el mejor comportamiento del FRSR, siendo el mismo con una resistencia con un valor igual a $2\text{ K}\Omega$, este valor proporciona una salida de hasta 2.8 V con carga máxima.

También se tomo en consideración los diferentes puntos donde se haría presión dando como resultado que: si se genera presión en las esquinas el voltaje es diferente al que se genera en el centro del componente con una relación de 50 por ciento, es decir que la sensibilidad del mismo es mayor a medida que el punto de presión se aproxime al centro del sensor, en la figura 3.12 se puede apreciar tanto las diferentes masas que se uso para la calibración del sensor así como también el punto de presión que se utilizo de referencia para todas las medidas.



Figura 3.12: Conjunto de pesas para pruebas de sensor de fuerza

De igual manera se ha analizado el comportamiento del componente en diferentes tiempos de haber sido colocada la carga esto es relativamente importante al momento de realizar el estudio de los datos que emite este componente, este tipo de experimento no presento variaciones en la salida de voltaje del amplificador operacional.

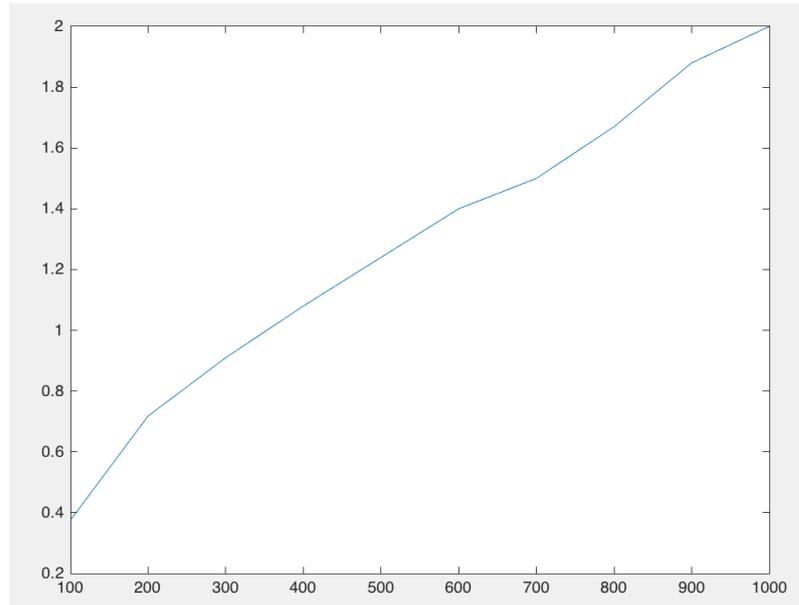


Figura 3.13: Salida - Peso aplicado Resistor 10 k Ω

Como se puede apreciar en la figura 3.13 el voltaje máximo de salida nos da un valor de 2 V ya que la entrada de el ADC a utilizar, permite una entrada de hasta 3.3 V se estaría perdiendo resolución por lo tanto este valor no es adecuado.

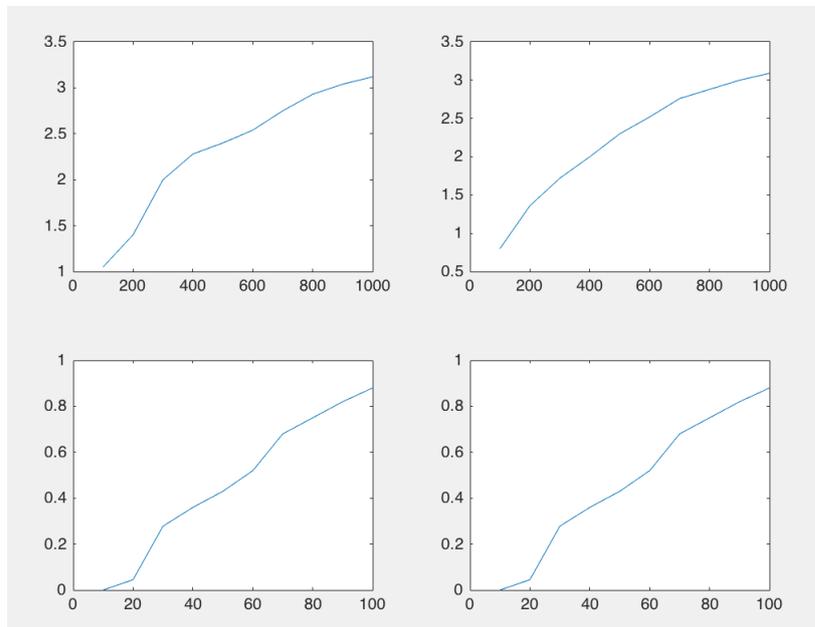


Figura 3.14: Salida - Peso aplicado Resistor 1 k Ω

Como se puede apreciar en las diferentes figuras el mas estable es el resistor de 2 K Ω , con el resistor de 1 K Ω se desestabilizaba el sistema frecuentemente y su salida es menos constante en referencia al resistor de 2 K Ω , se ha utilizado otros valores de resistencia como 3 K Ω , 4.7 K Ω , etc, que son valores comerciales pero la resolución de la salida en determinados casos sobrepasaba los 3.3 V y en otros casos

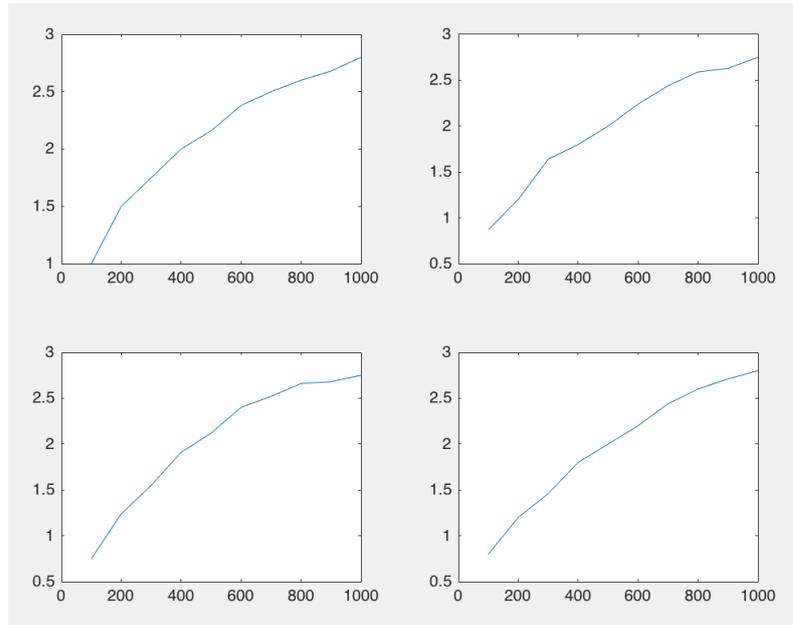


Figura 3.15: Salida - Peso aplicado Resistor 2 kΩ

tenía el mismo comportamiento que con el resistor de 1 KΩ, por lo tanto se trabajará con un R_m de 2 KΩ.

El amplificador que se utilizó para la obtención de estos datos es el LM358P especificado por el fabricante [35] como uno de los más indicados para trabajar con las FRSR.

3.5. Diseño de circuitos electrónicos

Una vez desarrolladas las pruebas con diferentes sensores de fuerza se procedió al diseño de las diferentes placas; la placa para los sensores de fuerza se diseñó con el objetivo de que sea lo más compacta posible, para que pueda ser contenida dentro de la pinza, en la figura 3.16 se puede apreciar el diseño final.

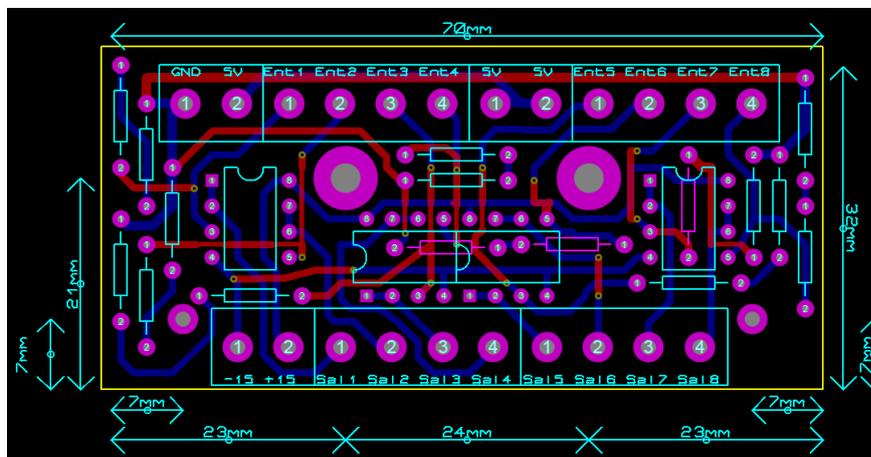


Figura 3.16: Diseño de placa con entradas y salidas de sensores de fuerza

Tanto para la alimentación del motor como de los amplificadores operacionales y tomando en consideración que estos funcionan con voltaje positivo y negativo se diseñó un inversor de voltaje con el integrado TL 7660 el cual permite trabajar con voltajes de entre 1 y 10 voltios, el esquemático de este sensor se muestra en la figura 3.17, el mismo ha sido tomado de la hoja de datos del fabricante [36].

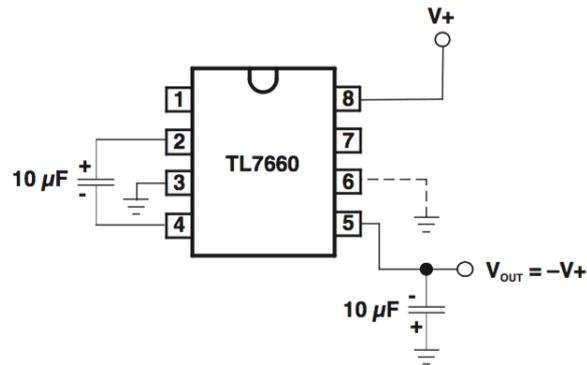


Figura 3.17: Esquema inversor de voltaje

La última placa que se diseñó contiene un puente h el cual permitirá modificar el sentido de giro del motor cuyo esquema de conexión típico se lo muestra en la figura 3.18, este esquema se tomo de la hoja de datos del fabricante [37]

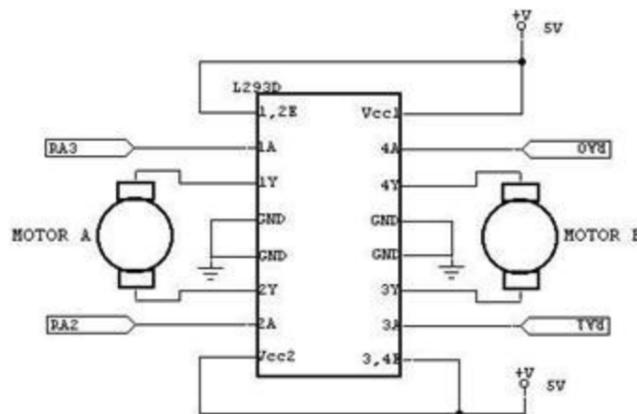


Figura 3.18: Esquema de conexión puente h

3.6. Software

Para el diseño del software se ha basado el proyecto en la arquitectura que se muestra en la figura 3.19 donde el sistema en general a una escala superior estará controlando por la tarjeta Raspberry Pi o Rose, esta a su vez enviará comandos o acciones de control a la tarjeta Discovery que a su vez enviará y solicitará información de una segunda tarjeta Discovery, esta ultima será quien controle la pinza.

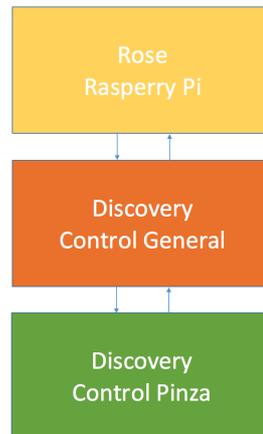


Figura 3.19: Arquitectura de control general

Este esquema de funcionamiento es un planteamiento inicial del proyecto, posteriormente se tiene pensado implementar una única tarjeta Discovery que se comunique directamente con el controlador principal que será el sistema Rose o la tarjeta Raspberry Pi.

El objetivo de esta arquitectura a partir de la tarjeta Discovery de control general es; que mediante cuatro bits de comunicación entre esta tarjeta y la de control de la pinza, se definan diferentes acciones a realizar, en el presente proyecto para realizar las diferentes pruebas de movimiento, velocidad, posicionamiento y fuerza de la pinza se ha implementado comunicación USB en la cual se ha desarrollado una interfaz para controlar las diferentes acciones de la pinza, pudiendo ser estas:

- Apertura y cierre de pinza a velocidad predeterminada
- Apertura y cierre de pinza a velocidad específica
- Control de Posicionamiento para apertura o cierre de la pinza
- Control de Fuerza para sostener objetos
- Sensores o dedos a utilizar
- Obtención de datos usando SDRAM

Si se utiliza 4 bits en lo posterior tendremos 16 diferentes acciones a realizar; se pudiese implementar otros tipos de comunicación entre las tarjetas Discovery como SPI, bluetooth, USART, los datos que sean transmitidos por la tarjeta Discovery principal serán recibidos por la tarjeta secundaria que manipulará la pinza y mediante un selector decidirá que acción realizar.

3.6.1. Arquitectura de programación

Con arquitectura de programación referiremos el presente proyecto al diseño de la maquina de estados que se encargara de referenciar las diferentes acciones realizadas o a realizar y el cumplimiento o fallo

de las mismas, esta arquitectura estará basada entre la interfaz de comunicación USB y el hardware que acompaña a la pinza.

Tomando en consideración que se debe realizar la identificación de los diferentes sistemas, antes de ingresar a la etapa de control, se presentará una maquina de estados diferente para realizar la captación de datos para el diseño de los controladores.

De esta manera se presentan las maquinas de estados contenidas en las figuras 3.20 3.21

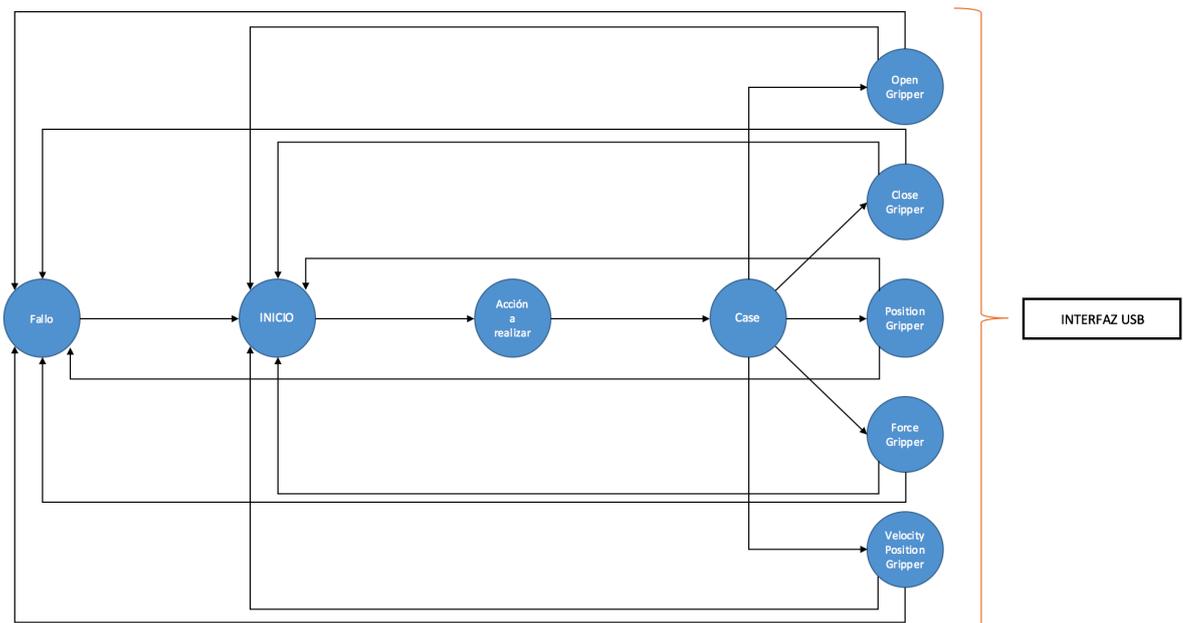


Figura 3.20: Maquina de estados para control general

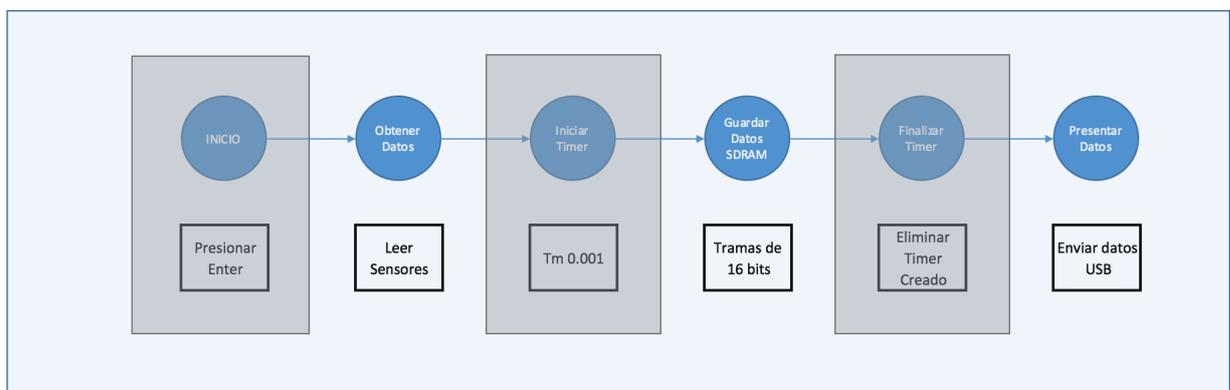


Figura 3.21: Maquina de estados para obtención de datos

Debido a que en ambos casos en lo que se entiende sobre las maquinas de estados se utilizará la interfaz USB y en todo el proyecto en si es importante considerar la configuración del mismo.

3.6.1.1. Interfaz USB

Para la implementación de la interfaz USB se ha utilizado las librerías proporcionados por [39], teniendo en consideración que para utilizar las mismas se utilizará los diferentes paquetes que se muestran en el Anexo B.

Teniendo en consideración que lo que se envía y recibe son caracteres se puede trabajar tanto con los mismos como con el referente decimal de cada uno de ellos, si el objetivo fuese trabajar con los caracteres que se reciben, mediante una tabla ascii se puede definir el caracter a utilizar en este caso se trabajara con el caracter 13 que corresponde a la tecla enter.

Siendo el caso de que se necesiten otros valores para posicionar el motor, definir la fuerza, o la velocidad se trabaja con la función `atoi(constant char)`, de esta manera cada vez que se ingrese un valor ya sea letra o número este se guardara en una variable tipo array y luego se convertirá en una constante mediante esta función, si el caso fuese de que se ingresare una letra se ha parametrizado el programa para que las rechace.

Al momento de implementar USB se inhabilitan los puntos PA9 y PA10 de la tarjeta Discovery estos puntos sirven para transmisión y recepción de datos del USB, por lo tanto los mismos no se podrán utilizar.

La comunicación USB ha pesar de que nos facilita la obtención de datos, mediante la practica se ha determinado que no es lo suficientemente rápida como para recibir los datos en un tiempo de muestreo de 1ms provocando excesiva perdida de los mismos, interrupciones en la operatividad del programa entre otros.

Es por esta razón que se ha implementado la lectura y escritura de datos mediante la SDRAM interna de la tarjeta Discovery, de esta manera los datos pueden ser leídos en lo posterior y transmitidos por USB evitando eficientemente todos los inconvenientes antes mencionados.

3.6.1.2. SDRAM

La memoria dinámica de acceso aleatorio con la que cuenta la tarjeta Discovery es de 2 Mb lo cual es suficiente para almacenar los datos de proporcionados durante la identificación y la posterior lectura de los mismos.

Ya que la memoria esta distribuida por diferentes direcciones de almacenamiento pudiendo ser de 8, 16 o 32 bits, se debe tener en consideración que tanto la escritura como la lectura se debe hacer en el mismo punto de memoria es decir si un dato fuese almacenado en la dirección (0x01AB) se tendrá que leer de (0x01AB).

Si la escritura esta definida para hacer cada 8 bits todas las escrituras posteriores se debe realizar con el mismo valor, si el caso fuese con 16 o 32 se debe realizar el mismo procedimiento.

En el presente proyecto se ha estimado 16 bits de escritura para cada dato de muestreo, empezando en la dirección 0 hasta el final del muestreo.

Las diferentes librerías que se utiliza para la implementación de la SDRAM se muestran en el Anexo B.

Al momento de implementar el uso de la SDRAM queda inhabilitado el Timer 1 de la tarjeta discovery y otros puertos GPIO pudiendo ser estos directamente determinados o los mismos pudiesen ser alternados con otros, estas especificaciones se encuentran dentro de las librerías.

3.6.2. Identificación del sistema

Para la identificación se va a tomar datos que han sido guardados anteriormente en la SDRAM por las razones anteriormente explicadas, es importante tener en consideración que para este proceso se han definido diferentes posiciones del motor para poder tomar estos datos.

Al momento de empezar a captar los datos se pudo observar que existe demasiado ruido proporcionado por parte del potenciómetro que acompaña al movimiento del motor, dando como resultado la gráfica que se muestra en la figura 3.22

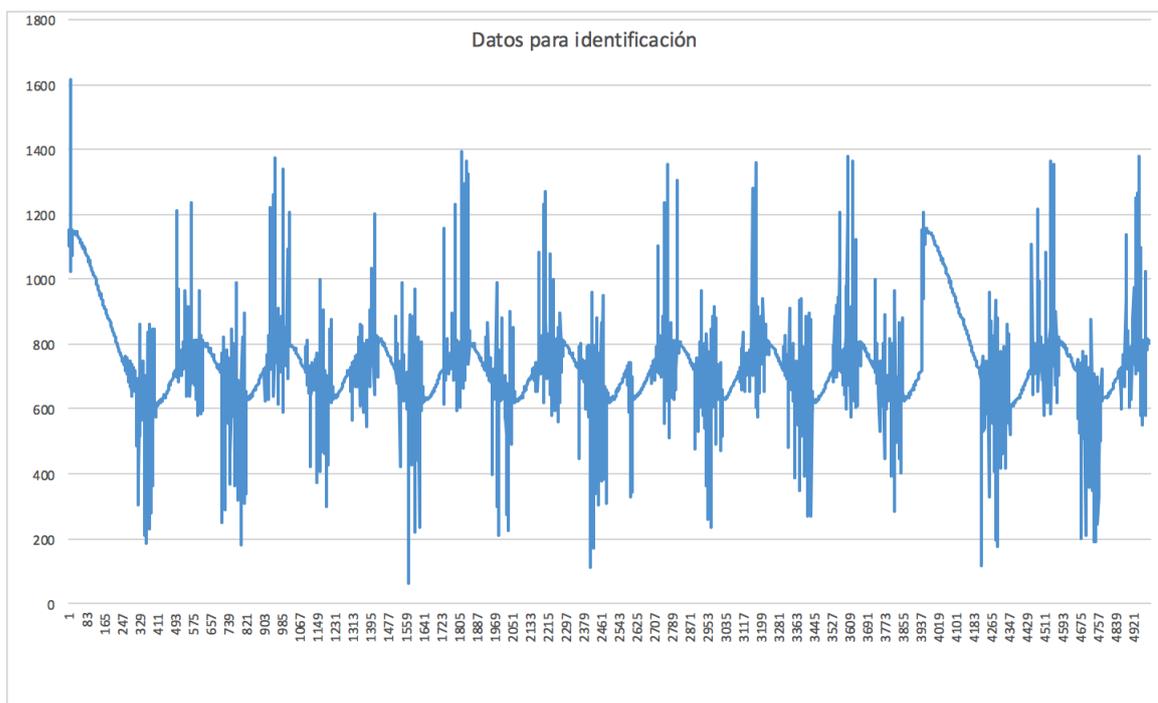


Figura 3.22: Datos Obtenidos sin filtro

3.6.2.1. Filtro del sensor de retroalimentación

Se ha tratado de realizar dos tipos de filtrado de estos datos el primero es mediante un capacitor y el segundo mediante un filtrado por software.

El resultado del filtrado mediante un capacitor se muestra en la figura 3.23 lo cual nos demuestra que no es un filtrado esencialmente bueno ya que el ruido que elimina es bajo.

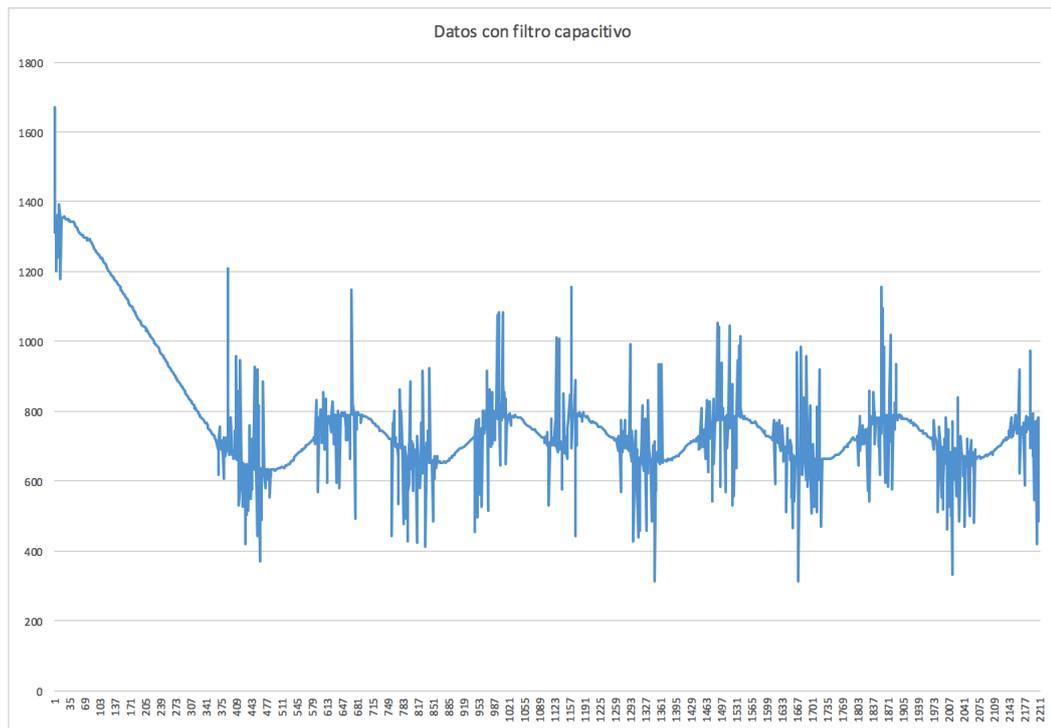


Figura 3.23: Datos Obtenidos con filtrado mediante capacitor electrolítico

La segunda opción es realizar un filtrado usando un filtro pasa bajas o filtro promediador ¹, tal como su nombre lo dice se trata de tomar varios datos, y luego promediar la cantidad de datos tomados.

En software se ha definido de la siguiente manera:

Se crea un vector con el número de muestras que se vaya a tomar `float muestras[n]`, se rellena de ceros todas los espacios del vector mediante un ciclo for y posteriormente se crea un nuevo bucle con la siguiente estructura

1. Inicia el bucle
2. Se guarda el valor total con la lectura inmediata anterior

$$\text{Total} = \text{Total} - \text{muestras}[i]$$

¹http://www.academia.edu/5301304/FILTROS_DIGITALES_14_FILTROS_DIGITALES

3. Se almacena un nuevo dato de entrada
 $\text{muestras}[i] = \text{ADC}()$
4. Se almacena en el total la nueva muestra sumada la muestra anterior
 $\text{Total} = \text{Total} + \text{muestras}[i]$
5. Se incrementa el número de muestras obtenidas en uno $i++$
6. Se repite el bucle

Este proceso se lo realiza la cantidad de veces que han sido definidas para la muestra y posteriormente se promedia el Total, la eficiencia de este filtro depende del número de muestras que se adquieran del sensor.

En este caso se han definido dos valores máximos para el numero de muestras siendo el primero 30 y en el segundo caso 50.

Para el primer caso se ha obtenido la figura 3.24

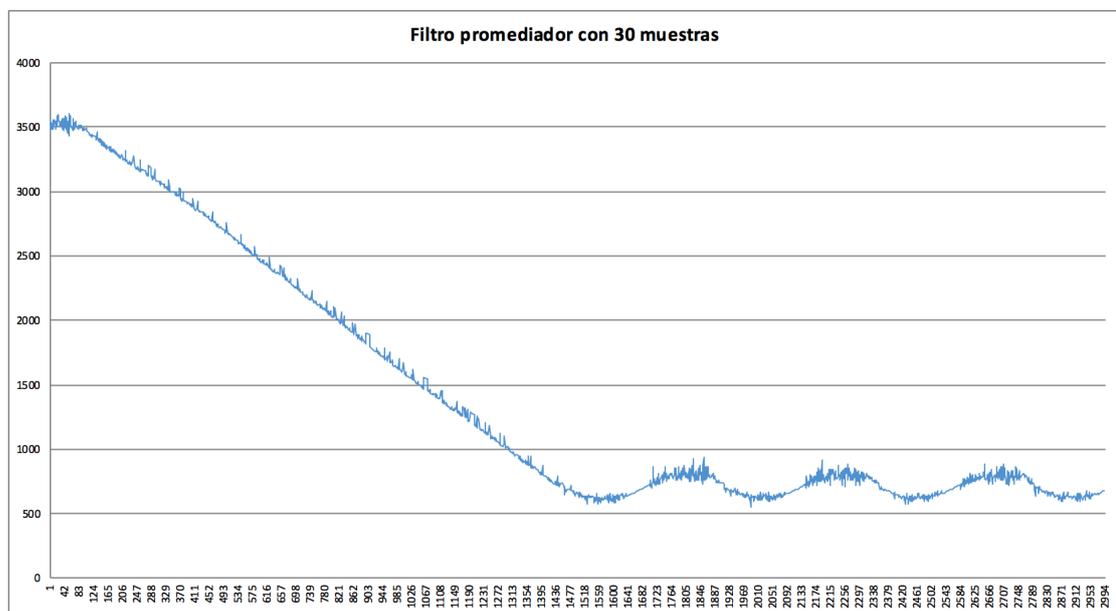


Figura 3.24: Filtro promediador con 30 muestras

En esta figura podemos observar que a pesar de que el filtrado a resultado ser eficiente aún siguen existiendo varios picos de ruido que pueden ocasionar variaciones en la etapa de control por lo tanto se procede a aumentar el número de muestras al filtro promediador a 50.

Se debe tener en cuenta que se puede aumentar el numero de muestras tanto como el procesador que se este usando lo permita y dependiendo del tiempo de muestreo que se le de al sistema.

En este caso en particular el tiempo de muestreo es de 1 ms, pero cada dato será analizado en ns ya que esa velocidad de adquisición nos permite la tarjeta Discovery.

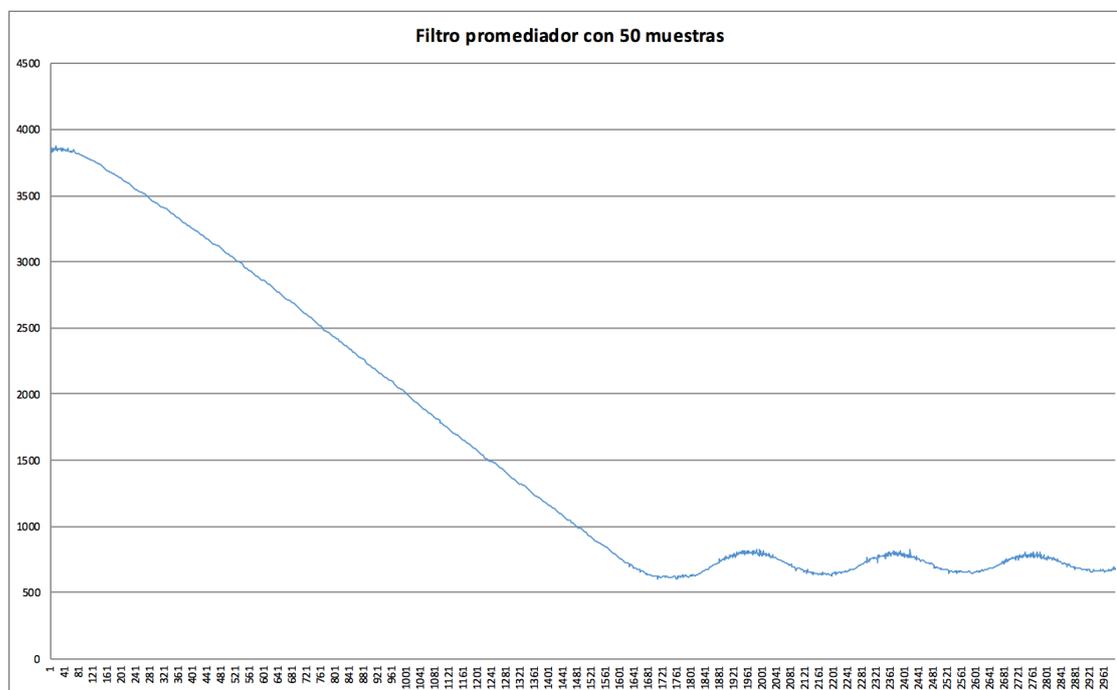


Figura 3.25: Filtro promediador con 50 muestras

Aumentando el número de muestras se puede evidenciar que se ha disminuido en gran proporción el ruido, no se ha aumentando el número de muestras porque no hay cambios representativos en la salida y se carga innecesariamente el procesador de la tarjeta Discovery, por lo tanto para deducir el ruido en el modelo de experimentación se trabajará con un filtro promediador con 50 muestras.

3.6.2.2. Obtener datos

Para obtener los datos en primer lugar se ha definido un controlador P tal que haga subamortiguado el sistema pero sin que la acción de control sature el mismo o el sistema se vuelva inestable, luego de hacer varios experimentos se ha determinado que el valor de 3 para la ganancia en el proporcional cumple estas especificaciones.

En este punto el sistema se vuelve subamortiguado en el régimen transitorio y se estabiliza en aproximadamente de 1 a 2 segundos para el régimen permanente (véase 3.26).

Los diferentes escalones simbolizan la posición del motor la cual va de 0 a 180 grados, como se cuenta con un potenciómetro para definir la ubicación el mismo alimentado con 3 V dará una variación de 0 a 10 k Ω , ya que se trabaja con 12 Bits en el ADC se obtendrá 4096 valores así que la definición de posición corresponderá al rango de 0 a 4096 sin embargo mediante la practica se constata que los valores van de

100 a 4096.

Ya que existe repetibilidad en esta medida se ha agregado un offset el cual arregle este desfase quedando como resultado la figura 3.26 donde se muestra los valores obtenidos y la referencia ingresada.

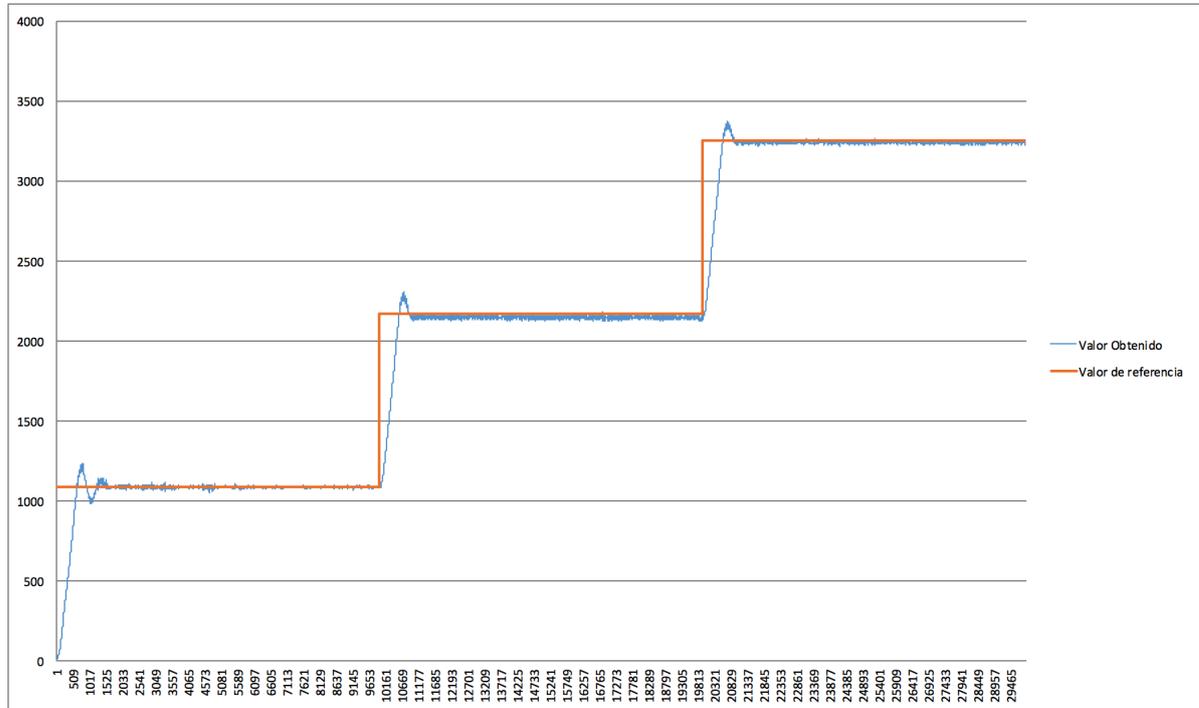


Figura 3.26: Sistema de experimentación con ganancia 3 en el proporcional

Las posiciones que se han ingresado corresponden a 45, 90 y 135 grados respectivamente.

Aunque en principio parecería alcanzar el sistema la referencia sin problemas esto no es real pero la variación es demasiado pequeña como para ser notada en la escala en la cual se presenta la figura.

3.6.2.3. Cálculo de la función de transferencia

Una vez Obtenidos los datos se procederá a obtener la función de transferencia del sistema mediante la ecuación 3.9.

$$G_{\Theta_{bc}}(s) = \frac{k * w_n^2}{s^2 + 2 * \xi * w_n * s + w_n^2} \quad (3.9)$$

Donde:

- k Es la ganancia
- w_n Es la frecuencia natural
- ξ Es el coeficiente de amortiguamiento

En primer lugar se determinará ξ mediante la ecuación 3.10

$$\xi = \frac{-\ln\delta}{\sqrt{(\ln\delta)^2 + \pi^2}} \quad (3.10)$$

El valor de sobre-oscilación δ esta determinado por 3.11

$$\delta = \frac{M_p - \Theta_{bc}(\infty)}{\Theta_{bc}(\infty)} \quad (3.11)$$

La frecuencia natural esta determinada por 3.12

$$w_n = \frac{w_p}{\sqrt{1 - \xi^2}} \quad (3.12)$$

Donde w_p es la frecuencia propia determinada por el tiempo de pico t_p 3.13

$$t_p = \frac{\pi}{w_p} \quad (3.13)$$

Primer escalón

$$\Theta_{bc}(\infty) = AK$$

$$\Theta_{bc}(\infty) = 1083$$

$$A = 1084$$

$$K = 0,99$$

M_p corresponde al valor pico y $\Theta_{bc}(\infty)$ corresponde al valor alcanzado por el sistema, sustituyendo en 3.11

$$\delta = \frac{1209 - 1083}{1083}$$

$$\delta = 0,1163$$

Sustituyendo en 3.10

$$\xi = \frac{-\ln(0,1163)}{\sqrt{(\ln(0,1163))^2 + \pi^2}}$$

$$\xi = 0,5649$$

Despejando y sustituyendo en 3.13

$$w_p = \frac{\pi}{t_p}$$

$$w_p = \frac{3,14}{0,767}$$

$$w_p = 4,08$$

Sustituyendo en 3.12

$$w_n = \frac{4,08}{\sqrt{1 - (0,5649)^2}}$$

$$w_n = 4,945$$

Segundo escalón

$$\Theta_{bc}(\infty) = AK$$

$$\Theta_{bc}(\infty) = 1058$$

$$A = 1084$$

$$K = 0,976$$

Sustituyendo en 3.11

$$\delta = \frac{1188 - 1058}{1058}$$

$$\delta = 0,1228$$

Sustituyendo en 3.10

$$\xi = \frac{-\ln(0,1228)}{\sqrt{(\ln(0,1228))^2 + \pi^2}}$$

$$\xi = 0,555$$

Despejando y sustituyendo en 3.13

$$w_p = \frac{\pi}{t_p}$$

$$w_p = \frac{3,14}{0,75}$$

$$w_p = 4,1888$$

Sustituyendo en 3.12

$$w_n = \frac{4,1888}{\sqrt{1 - (0,555)^2}}$$

$$w_n = 5,0352$$

Tercer escalón

$$\Theta_{bc}(\infty) = AK$$

$$\Theta_{bc}(\infty) = 1096$$

$$A = 1084$$

$$K = 1,01$$

Sustituyendo en 3.11

$$\delta = \frac{1204 - 1096}{1096}$$

$$\delta = 0,0985$$

Sustituyendo en 3.10

$$\xi = \frac{-\ln(0,0985)}{\sqrt{(\ln(0,0985))^2 + \pi^2}}$$

$$\xi = 0,5936$$

Despejando y sustituyendo en 3.13

$$w_p = \frac{\pi}{t_p}$$

$$w_p = \frac{3,14}{0,77}$$

$$w_p = 4,08$$

Sustituyendo en 3.12

$$w_n = \frac{4,08}{\sqrt{1 - (0,5936)^2}}$$

$$w_n = 5,07$$

Mediante el promedio de los diferentes valores obtenidos, se tiene que:

$$k = 0,992$$

$$w_n = 5,0167$$

$$\xi = 0,5711$$

Sustituyendo en 3.9

$$G_{\Theta_{bc}}(s) = \frac{24,8987}{s^2 + 5,7281s + 25,15}$$

3.6.2.4. Simulación y validación de las ecuaciones obtenidas

Para verificar que esta ecuación ha sido determinada correctamente se ha utilizado el software matlab y mediante simulink se ha ingresado la ecuación de transferencia con los diferentes escalones que se

determinaron en un principio y en un tiempo de muestreo igual al utilizado para tomar los datos, el esquema de simulink se muestra en la figura 3.27

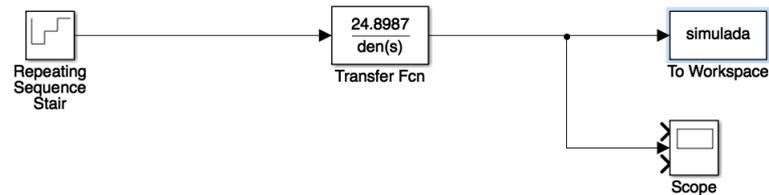


Figura 3.27: Diseño en simulink para el esquema en bucle cerrado

Los datos de esta simulación han sido comparados con el sistema real, dando como resultado la figura 3.28

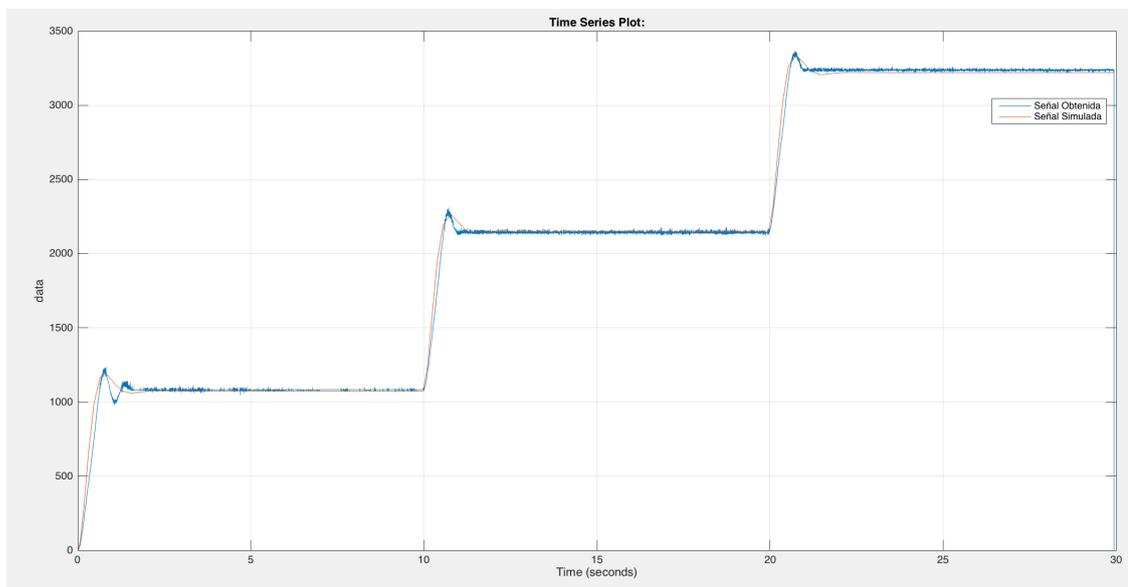


Figura 3.28: Esquema real y esquema simulado con la función de transferencia obtenida

En esta figura podemos observar que tanto la señal obtenida como la simulada son muy parecidas por lo cual se puede definir que esta correctamente implementada la función de transferencia.

Esta ecuación se ha deducido en bucle cerrado, para determinar la ecuación en bucle abierto se utilizará 3.14

$$G_{\Theta} = \frac{G_{\Theta_{bc}}(s)}{K_p(1 - G_{\Theta_{bc}}(s))} \quad (3.14)$$

Dando como resultado

$$G_{\Theta} = \frac{8,3}{s^2 + 5,728s + 0,2513}$$

Se procede a ingresar los resultados en el simulador y se comparará con la señal en bucle cerrado teniendo que ser exactamente iguales las dos dando a la señal en bucle abierto un k_p de 3 que es valor que se ha dado a la ganancia del proporcional para deducir el sistema, quedando el esquema de simulación como se muestra en la figura 3.29

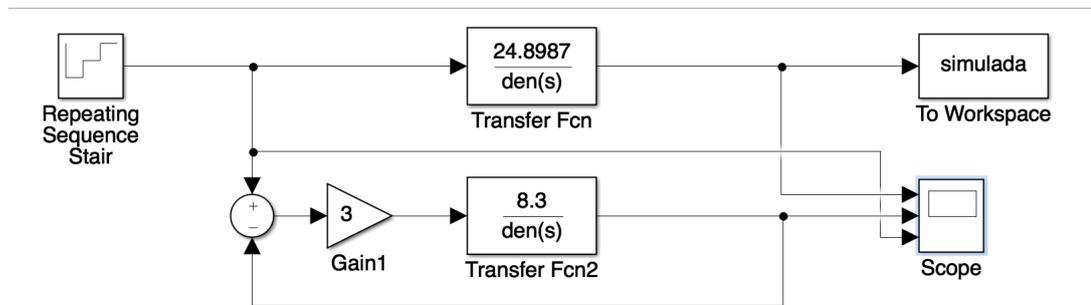


Figura 3.29: Esquema en bucle abierto y en bucle cerrado

Este sistema implementado nos da como resultado

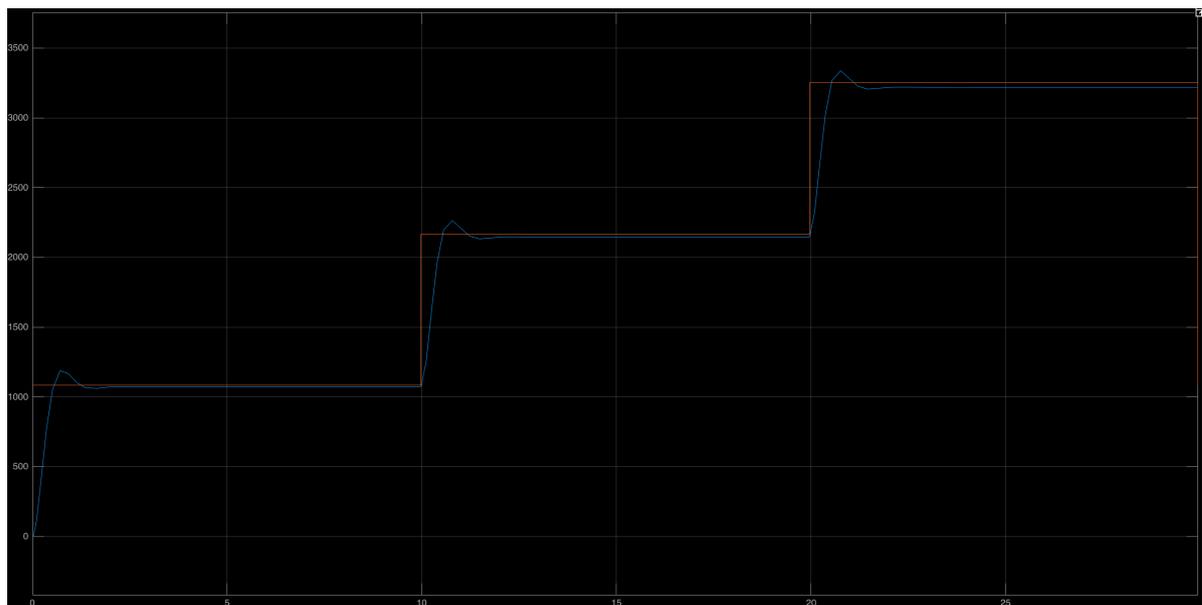


Figura 3.30: Comparación en bucle abierto y en bucle cerrado

Como se puede observar en la figura 3.29 ambos sistemas se corresponden por lo tanto la deducción de bucle abierto está bien desarrollada.

3.6.2.5. Diseño de controlador

Con el objetivo de eliminar el amortiguamiento y mejorar el tiempo de establecimiento que existe en el sistema se ha utilizado la herramienta sisotool de matlab, mediante la cual se ha implementado un cero para mejorar la respuesta en el transitorio y su ganancia para eliminar el amortiguamiento que existe en

el sistema.

La configuración que se ha realizado en sisotool se muestra en la figura 3.31

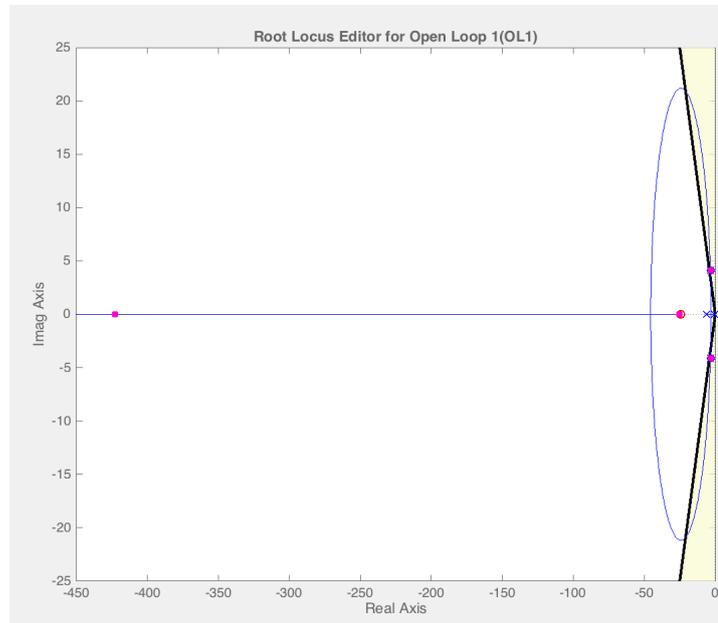


Figura 3.31: Diseño del controlador PD Sisotool

En esta figura también se puede apreciar que el cero está ubicado dentro de especificaciones así como también los polos.

La respuesta al impulso que da esta configuración se muestra en la figura 3.32, esta misma respuesta se ha utilizado para adaptar el sistema variando la ubicación de los polos, los ceros y la ganancia.

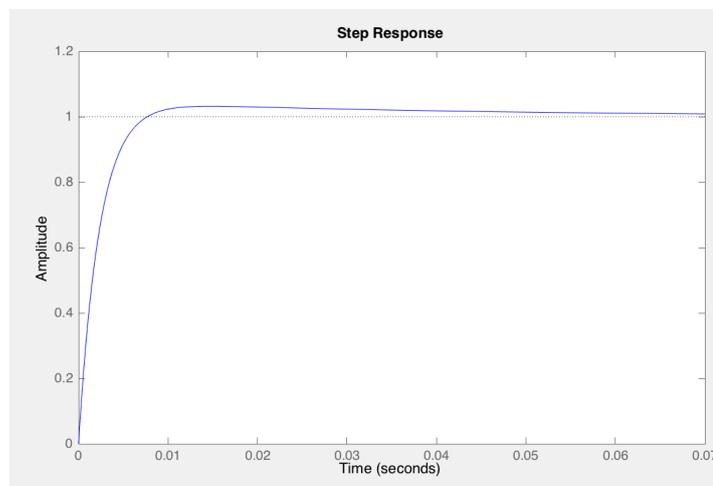


Figura 3.32: Respuesta al impulso mediante Sisotool

Una vez diseñado este sistema se ha realizado los cálculos pertinentes para el sistema, utilizando la ecuación 3.15

$$G_{RPID}(s) = \frac{k_c(T_d s^2 + s + \frac{1}{T_i})}{s} \quad (3.15)$$

Donde por deducción y para un PD se determina que

$$T_d = 1/b \quad (3.16)$$

$$k_c = k_p(b) \quad (3.17)$$

El valor de k_p corresponde a la ganancia que en este caso es de 53.312 y el valor de b es el cero que se ha agregado que en este caso corresponde a 24.2.

Por lo tanto mediante 3.16 se obtiene el tiempo para el derivativo el cual es igual a:

$$T_d = 0,04132$$

Y el valor de la ganancia mediante 3.17 para el derivativo será igual a:

$$k_c = 1290,15$$

Se ha implementado estos valores en el entorno de simulación simulink obteniendo los resultados que se muestran en la figura 3.33

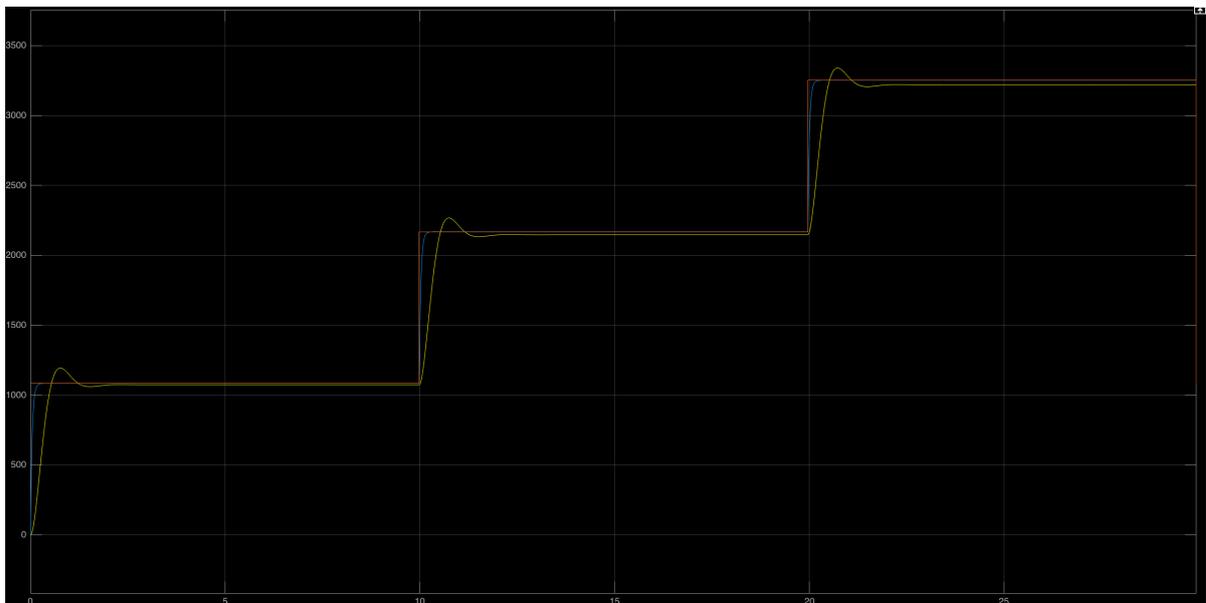


Figura 3.33: Respuestas ante referencia con controlador PD y P

Como se puede apreciar el sistema deja de estar sobreamortiguado, mejora su tiempo de estabilización y alcanza la referencia en el régimen transitorio y en el permanente.

3.6.3. Control

Para el esquema de manipulación de la pinza se ha implementado tres diferentes tipos diferentes de control:

- Control por Velocidad
- Control por Fuerza
- Control por posición

Para cada uno de estos controles se cuenta con un feedback de retro-alimentación estimado por los diferentes sensores contenidos en el prototipo.

Para los diferentes esquemas de control se hará uso de los modelos obtenidos en el proceso de identificación y se adaptará los mismos a los diferentes requerimientos, no es necesario realizar otro calculo de función de transferencia, debido a que los diferentes factores que determinan el comportamiento del sistema tienen como fin proporcionar una acción de control para el mismo motor a diferencia del control por velocidad.

El control por velocidad aunque es determinada para el mismo motor se analiza su modificación partiendo desde los valores más bajos hasta los más elevados.

3.6.3.1. Control por Velocidad

Para el control por velocidad se ha implementado el esquema de control que se muestra en la figura 3.34 donde los puntos a ajustar del sistema son las constantes del controlador, este tipo de control es importante para producir movimientos suaves o realizar mayor esfuerzo de-acuerdo al objetivo que se quiera implementar.

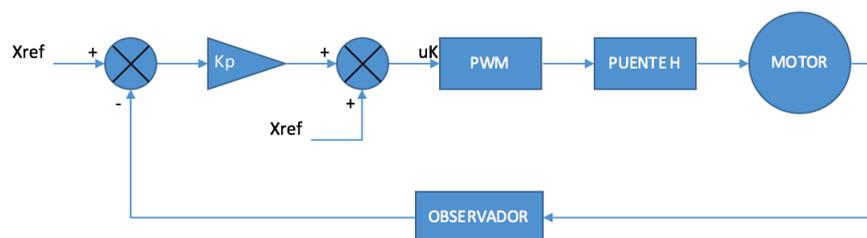


Figura 3.34: Esquema de control para Velocidad

El objetivo de este control es ajustar el duty generado por la tarjeta discovery en un rango de 0 a 100 por ciento, al variar el duty generado por el controlador se varía la velocidad pudiendo ser o mas rápida al llegar al punto 100 en términos de porcentaje o más lenta al llegar al punto 0.

La generación de un pulso PWM cuenta con un ciclo determinado el cual se basa en estados en alto y en bajo que determinará su velocidad (véase figura 3.35)

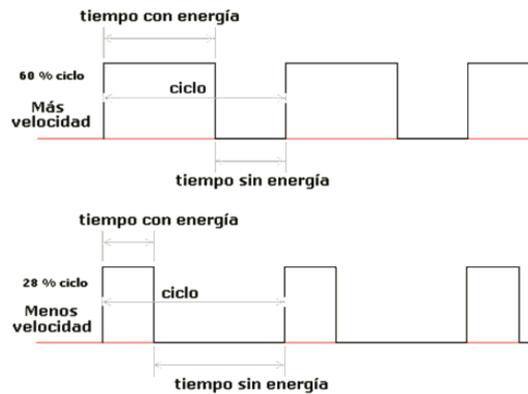


Figura 3.35: Esquema de control por estados

La velocidad del motor radica en el tiempo que se mantenga en alto la señal producida por la salida PWM y la variación de la velocidad se consigue con la variación del duty en un ciclo de trabajo el cual haya sido previamente definido.

El programa que se ha diseñado para tal efecto sigue el esquema de funcionamiento que se muestra en la figura 3.36.

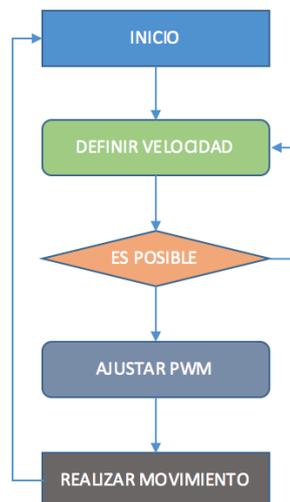


Figura 3.36: Esquema de funcionamiento para Velocidad

El control de velocidad sin embargo no determina hacia que lado gira el motor dando la imposibilidad de retornar a una posición si ya ha pasado la misma, la forma para que un motor se mueva hacia un lado u otro es invirtiendo el sentido de la corriente que pasa por el mismo y esto no se lo puede lograr con la variación del duty.

Es por tal motivo que se ha utilizado un puente H para el control de la corriente que circula por el motor y dos salidas PWM de la tarjeta Discovery, de tal modo que si se realiza un giro con la mitad de potencia del motor se hará con el 50% de la capacidad total en una de las salidas GPIO de la tarjeta Discovery y la otra con la frecuencia más alta que se le pueda dar, de esta manera girará hacia uno u otro lado dependiendo de estas características.

En el caso de que se implemente más motores y sea necesario ocupar una cantidad menor de salidas PWM por cada motor, se podrían aplicar compuertas activando su giro de uno u otro lado con un GPIO y una salida PWM.

El reconocimiento de la velocidad si esta era constante o no, se trato de realizar obteniendo datos de muestreo por cada unidad de tiempo y comparandolos con su inmediato anterior, sin embargo esto no fue posible debido a la cantidad de ruido que existe en el potenciómetro del motor.

Este ruido no filtrado anteriormente no produjo problemas porque simboliza la cantidad de 1 a 2 por cada mil lo cual en las variaciones de posición no representa un valor significativo, pero como en este modelo se compara entre dos valores continuos y se saca la diferencia entre ellos, el valor de ruido es muy significativo tal como se muestra ne la figura 3.37

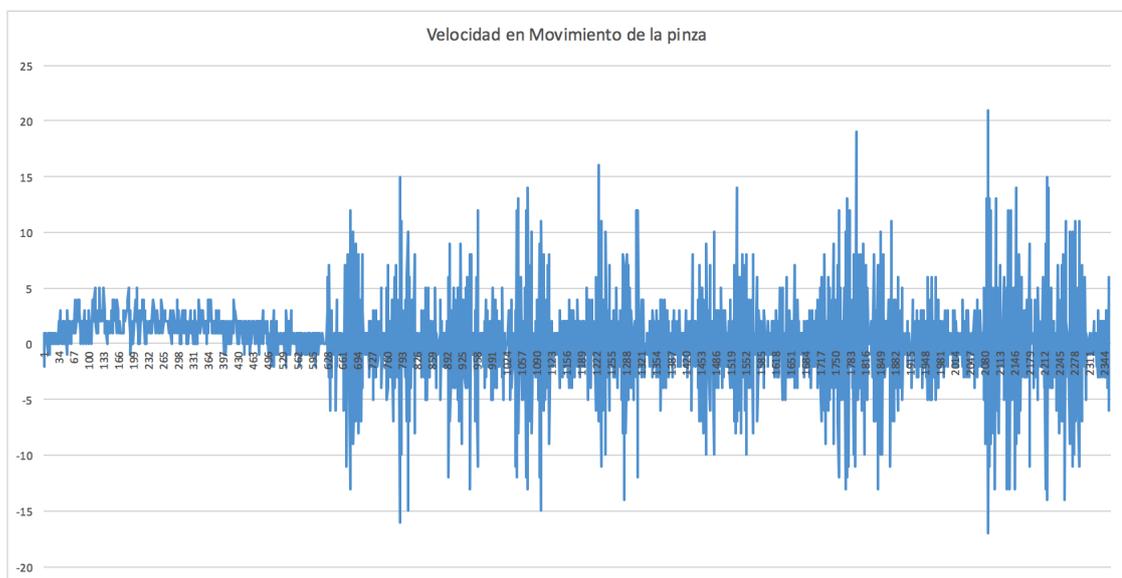


Figura 3.37: Ruido en el sensor del motor para determinar la velocidad del mismo

Como se puede apreciar en la imagen 3.37 existe demasiado ruido en el sensor de retroalimentación el cual nos sirve de guía para determinar la velocidad con la cual esta girando el motor, por lo tanto se ha limitado la constancia a los datos obtenidos por un desplazamiento determinado, es decir tomando como

referencia la pendiente de la recta en un espacio de tiempo.

Tomando varias muestras en diferentes espacios en tiempo progresivo, se ha determinado la velocidad con la cual se mueve el motor, para decir que existe una velocidad constante se entendería que el desplazamiento también debería ser constante, tal como se muestra en la figura 3.38



Figura 3.38: Desplazamiento constante para determinar la velocidad

Para determinar la velocidad con la cual se esta moviendo el motor se aplicará la ecuación 3.18

$$v_d = \frac{\Delta(e)}{\Delta(t)} \quad (3.18)$$

Determinando la velocidad en diferentes instantes de tiempo en ms

$$v_d = \frac{3644 - 3339}{258 - 403}$$

$$v_d = 2,103$$

Transformando a grados por mili-segundo

$$v_d = \frac{2,103 * 180}{4096}$$

$$v_d = -0,092 \frac{\text{grados}}{\text{ms}}$$

Hay que tomar en cuenta que el signo simboliza la dirección de desplazamiento es decir si se desplaza de la posición cero a 180 grados o viceversa.

El valor que va de 0 a 4096 correspondiente al desplazamiento corresponde a la lectura de 12 bits del ADC.

Se ha realizado el mismo cálculo aplicado a diferentes instantes de tiempo en el mismo desplazamiento obteniendo los siguientes resultados (véase tabla 3.2).

Posición final	Posición Inicial	Tiempo final	Tiempo inicial	velocidad en $\frac{\text{grados}}{\text{ms}}$
3122	2901	504	602	-0.098
2896	2818	604	639	-0.097
2624	2504	726	778	-0.10
2352	2113	847	947	-0.10
1889	1564	1045	1192	-0.097

Tabla 3.2: Velocidad en diferentes instantes

Mediante los datos obtenidos en la tabla 3.2 se puede inferir que el dispositivo se está desplazando a una velocidad constante.

Para su implementación dentro del software se ha creado una función denominada:

```
int openCloseGripper(uint16_t valVel, uint16_t valRef);
```

Donde se enviará la velocidad a la cual deseamos que se desplace y la posición a la cual deseamos que vaya, y el valor de retorno nos indicará si ha ejecutado la acción satisfactoriamente o no.

En su caso también se podrían retornar otros valores como la velocidad a la cual se ha desplazado, la posición a la cual ha llegado o desde cual se ha desplazado y el tramo que ha recorrido, esto depende de los datos que se deseen analizar.

3.6.3.2. Control por Posición

Para el control por posición se ha utilizado el servo-motor Tower Pro MG968 DIGI con el cual se ha realizado el modelo de identificación es importante mencionar que se ha eliminado la electrónica interna que realiza el posicionamiento del mismo para que el usuario pueda controlar su comportamiento, y se ha dejado únicamente el motor DC la parte mecánica que compone el servo-motor y el potenciómetro que servirá para determinar la posición del servomotor, con lo cual se tendría un esquema de control tal como se muestra en la figura 3.39.

El potenciómetro que contiene el servomotor, tiene un rango de movilidad de 0 a 180 grados, el voltaje de salida que el mismo genere dependerá de la entrada de alimentación que le demos, tomando en cuenta que la tarjeta Discovery toma los datos de 0 a 3.3 V se tendrá que adaptar el mismo a este valor, sin embargo para no aumentar la electrónica se tomará la salida de 3 V que está disponible en la tarjeta controladora.

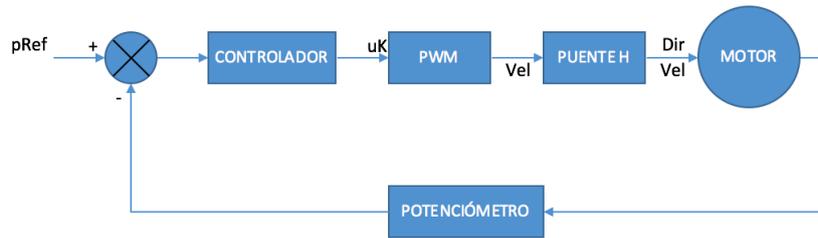


Figura 3.39: Esquema de control para Posición

Existe una diferencia de 300 mV a la entrada del sensor ADC por lo cual se ha ajustado el software a esta variación mediante la ecuación 3.19, donde se considera que el voltaje máximo es de 3.3 V y a este valor se tendrá 12 bits en lectura es decir 4096 variaciones, lo cual es solucionable mediante la regla de tres aplicada, obteniendo la variación máxima de datos que se tendrá a la salida.

$$D_{out} = \frac{3 * 4096}{3,3} \quad (3.19)$$

$$D_{out} = 3723$$

Conocida esta variación de datos que obtendremos a la salida, se puede calcular la posición del servomotor en un rango de 0 a 180 grados, utilizando la ecuación 3.20, dentro del programa.

$$D_{out} = \frac{Pos * 3723}{180} \quad (3.20)$$

De este modo se podrá llegar a la posición en grados que se determine en un rango de 0 a 180, por ejemplo, si se quisiera llegar a la posición de 100 grados, aplicando la ecuación 3.20 nos dará un valor de 2068 aproximadamente como dato de entrada en el ADC; este será el valor de referencia al cual se debe llegar con el controlador implementado, siguiendo este criterio se ha seguido el siguiente esquema de funcionamiento 3.40.

Sin embargo mediante la practica se ha podido constatar que realmente la salida de la tarjeta controladora da un valor de 3.3 V por lo tanto no se aplicará el calculo realizado anteriormente, pero si el caso fuese que se aplique en una tarjeta que su salida de voltaje de efectivamente este valor se deberá considerar los cálculos anteriormente dados.

Para el software que se ha diseñado se debe definir a donde se quiere posicionar el motor mediante la siguiente línea de código(se posicionará en 100 grados).

```
gripperPosControl(100);
```

Lo que hace este comando es llamar a la librería acción que se ha creado para el presente proyecto donde se ajustará el valor de grados al de datos que pudiese generar el ADC, siendo este 2409.

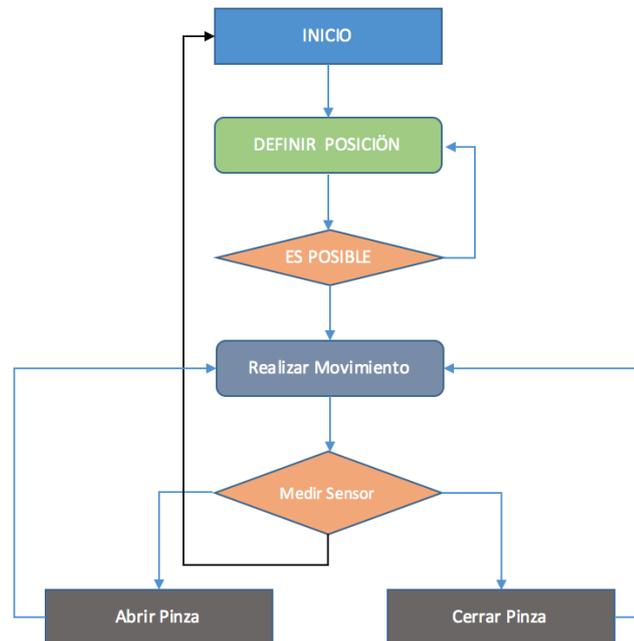


Figura 3.40: Esquema de funcionamiento para Posición

Se leerá la posición actual del servo mediante la línea de código.

```
readPos();
```

La función de este comando es leer el PIN análogo del GPIO correspondiente al potenciómetro (PE6), el valor generado por esta función será almacenado en una variable global de modo que se pueda ocupar en otras funciones.

Este valor conjuntamente con el de referencia al cual se desea llegar serán transmitidos para que proporcionen la acción de control mediante el comando.

```
PIDControl(valPos, valSensor);
```

Controlador.- El software que se ha implementado para el control del motor es un PD en paralelo cuyos cálculos se realizaron en la sección de identificación.

Se ha desarrollado a modo de prueba también un controlador PID sin embargo existía acumulación del error en el integral, lo cual volvía inestable el sistema, generando sobreoscilaciones al momento de llegar a la referencia, para tal efecto se siguió el esquema indicado en la figura 3.41, para el PD se elimina

la parte integral del controlador en paralelo la ganancia y el cero agregado para cumplir especificaciones se han analizado en la etapa de identificación.

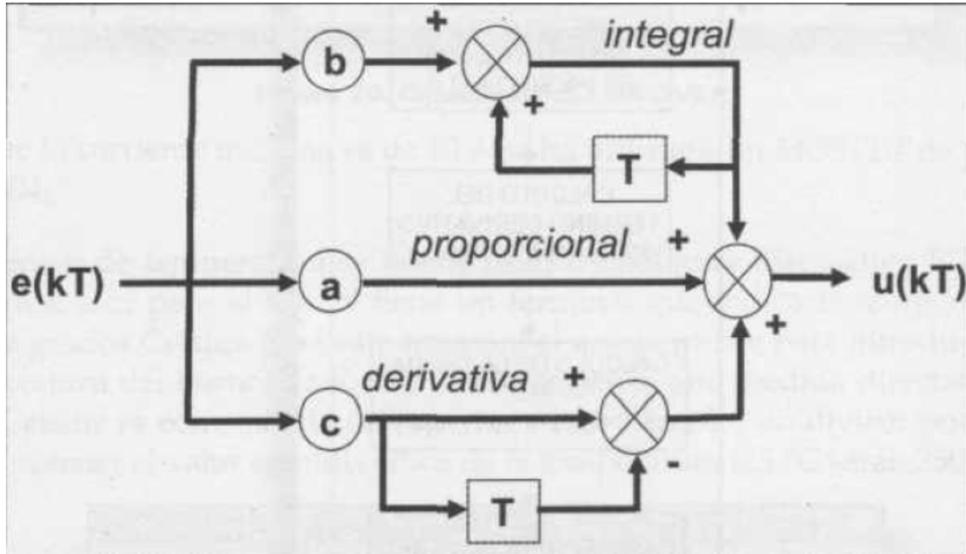


Figura 3.41: Esquema de control PID[38]

En la actualidad muchos dispositivos de carácter industrial utilizan controladores PID o derivados de los mismos, pudiendo ser estos de tipo mecánico, neumático, electrónico, etc, mediante el uso de microprocesadores se puede implementar este tipo de controladores de forma digital, ya que en un controlador PID intervienen tres parámetros diferentes como lo son; de tipo proporcional, integral y derivativo, respondiendo a la ecuación 3.21.

$$u(t) = k_p * e(t) + \frac{k_p}{T_i} * \int_0^t e(t) \partial(t) + k_p * T_d * \frac{\partial(e(t))}{\partial(t)} \quad (3.21)$$

Para la lectura de sensores se ha establecido un tiempo de muestreo de 1 ms, este tiempo de muestreo se ha determinado mediante un timer el cual activará cada mili-segundo una interrupción la cual llamará a la función que realizará la lectura de los datos analógicos pudiendo ser estos datos por parte de los sensores de presión o por parte del potenciómetro.

Mediante la siguiente línea de código se crea el timer en el cual se establece el tiempo de activación de la interrupción denominada CustomTimer1_Task.

```
CustomTimer1 = TM_DELAY_TimerCreate ( 1, 1, 1, CustomTIMER1_Task, NULL);
```

En cada instante que se toman las muestras serán almacenadas en una variable global para ser usadas en la etapa de control.

Para aplicar el controlador se enviará dos parámetros tanto la referencia como la lectura del sensor, los mismos se enviarán a la siguiente función.

```
PIDControl(valRef, valSensor);
```

Dentro de esta función se ha implementado el controlador PD si el caso se diese de que se necesitare un controlador PID su algoritmo de control es el siguiente:

```

/* Inicialización de variables */
ek=0; ek_1=0; ek_2=0;
uk=0; uk_1=0;
yk=0;

while(true)
{
    yk=Conversión_AD();
    ek=referencia-yk;
    /* Cálculo de la acción de control a aplicar en este instante*/
    uk=uk_1+Kp*(ek-ek_1+(T/Ti)*ek_1+(Td/T)*(ek-2*ek_1+ek_2));
    /* Aplicación de la acción de control */
    Conversión_DA(uk);
    /* Actualización de variables para la siguiente iteración */
    uk_1=uk;
    ek_2=ek_1;
    ek_1=ek;
    esperar_periodo_muestreo();
}

```

Figura 3.42: Algoritmo de control PID

Ya que es un controlador en paralelo tipo PD únicamente se elimina la parte integral del mismo.

Este algoritmo nos dará como resultado la acción de control que modificará el duty del PWM para de esta forma llegar a la posición indicada.

Para el ajuste de las constantes del controlador se han desarrollado varias pruebas presentando en las que no se han ajustado al sistema sobre-oscilaciones ya sea en el punto de mayor apertura de la pinza u en otros puntos de posición, inestabilidad, sobre esfuerzos, movimientos bruscos, etc, son algunos de los problemas que durante el ajuste de las constantes del PD se pudieron apreciar, hasta llegar a los valores que no producían ninguno de estos efectos.

El posicionamiento o apertura de la pinza se lo realizara en grados pudiendo ser la apertura máxima de 115 y su cierre máximo de 145 grados.

Basando las pruebas del controlador implementado se han realizado diferentes pruebas en este rango dando como resultado el desplazamiento esperado tal como se muestra en la figura 3.43

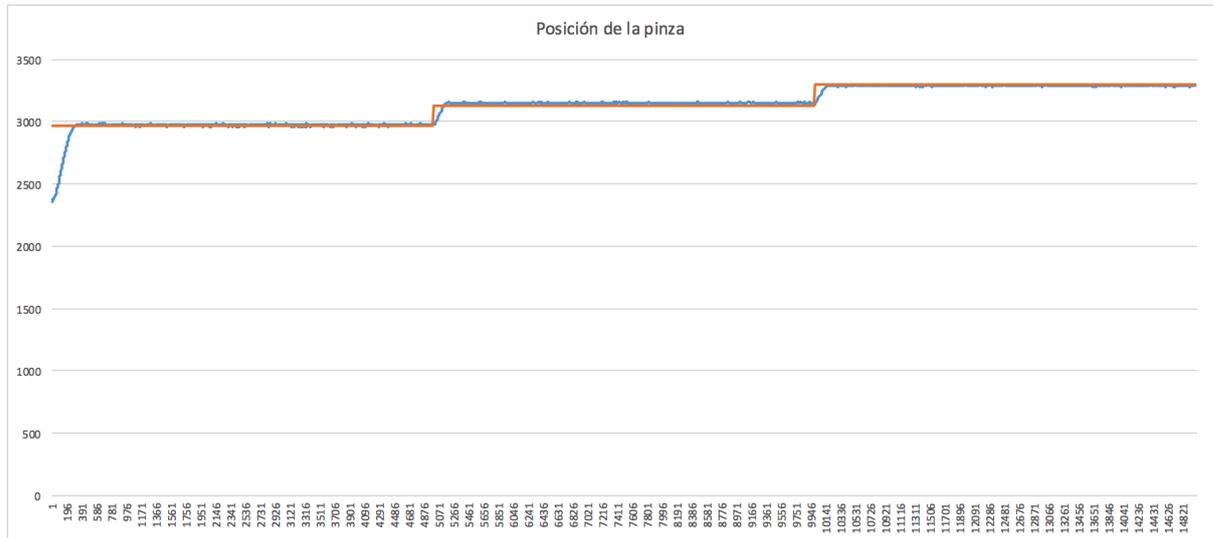


Figura 3.43: Apertura de la pinza dentro del rango que va de 115 a 145 grados

Se puede apreciar como el sistema llega a la posición esperada sin sobre-oscilaciones en el régimen transitorio y además cumple con la referencia, el análisis se baso en tres escalones de 123, 130 y 135 grados correspondientes a cada escalón, quedando verificado la eficiencia del controlador PD diseñado.

3.6.3.3. Control por Fuerza

Para el control por fuerza se ha seguido el mismo procedimiento que el implementado para control por posición la diferencia entre estos dos procesos serán los datos que serán utilizados para realizar la acción de control.

En este caso los datos provienen de la lectura de los sensores de presión ubicados tanto en las puntas de los dedos como en los bordes internos de cada uno de ellos.

Es importante tener en consideración que para desplazarse el motor necesita al menos el 40 % de su capacidad total por lo tanto en tramos pequeños la acción de control que se pueda generar necesita la ganancia suficiente para lograr este movimiento.

Es por este motivo que a partir de una distancia determinada se variarán las constantes del controlador con el fin de lograr desplazarse en tramos pequeños para que el motor pueda producir mayor fuerza al momento de presionar.

Se han variado las constante del controlador de modo que se pueda producir este efecto, sin embargo los datos no se han podido mostrar debido a que el uso de la SDRAM ocupa GPIO's estimados para la lectura del ADC.

Para la aplicación del esquema de fuerza se analizará la estructura de funcionamiento a ejecutar (véase figura 3.44)

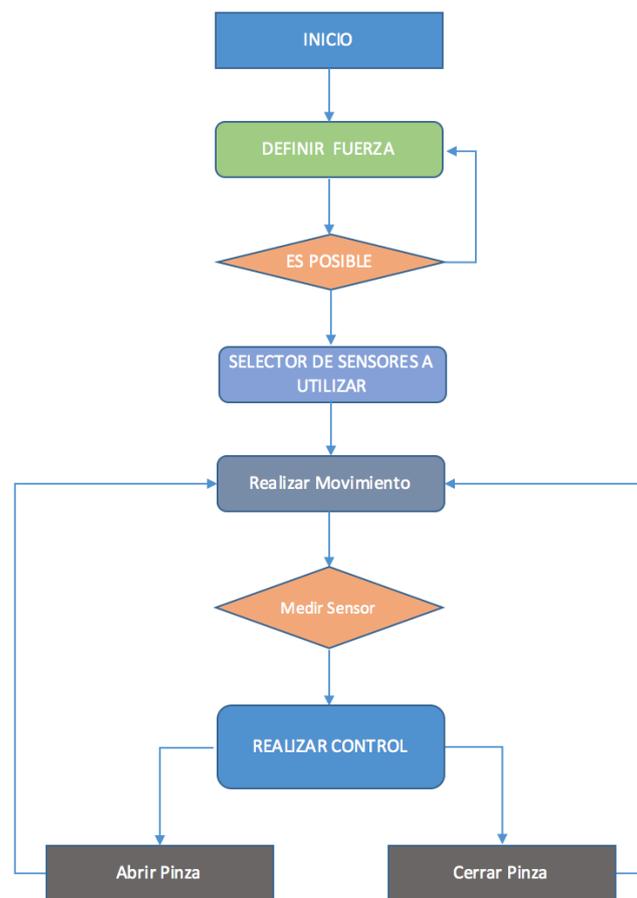


Figura 3.44: Esquema de funcionamiento para Fuerza

Cada uno de estos sensores transmitirán información la cual puede ser analizada en conjunto o independientemente dependiendo de la acción que se desea realizar.

Tal es el caso de que si se desea tomar una botella dependiendo del tamaño se pudiesen utilizar los sensores ubicados en las puntas o los internos, pero si el caso fuese tomar un bolígrafo u objetos más pequeños se podría utilizar únicamente los sensores de las puntas.

El análisis de estos sensores se podría hacer de varias formas y dependiendo de las conclusiones u acciones a las cuales queramos llegar, para el presente proyecto se ha optado por realizar la captura de

datos de todos los sensores y sacar el promedio de todos ellos de esta manera se estimara la fuerza que se esta aplicando, y comparandola con la referencia mediante el controlador se determinará la acción de control.

El esquema de control para Fuerza se muestra en la figura 3.45

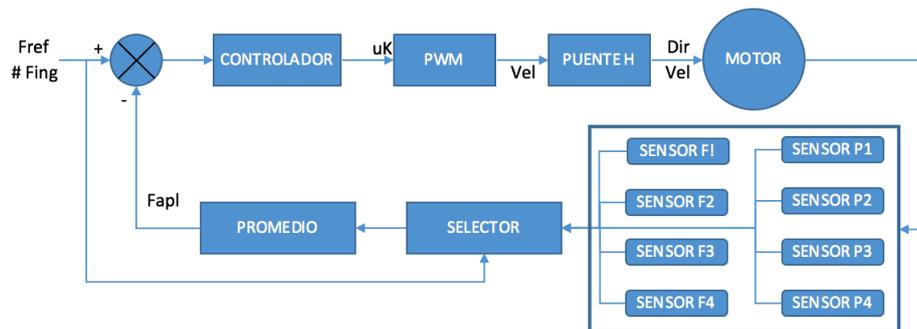


Figura 3.45: Esquema de control para Fuerza

El selector que se muestra en el esquema realizará la selección de cuantos y cuales dedos se va a utilizar para el análisis de fuerza aplicada, por lo tanto la llamada a la interrupción se hará de la misma forma que en el caso de posición con la diferencia que contendrá el parámetro fingers que determina que dedos se va a utilizar, de la siguiente manera.

```
CustomTimer1 = TM_DELAY_TimerCreate ( 1, 1, 1, CustomTIMER1_Task, fingers);
```

Aplicando esta línea de código se lograra realizar la lectura de los ADC necesarios.

El resultado de la lectura de los sensores y la fuerza de referencia serán transmitidos al controlador, donde también han sido ajustados los parámetros de las constantes para no producir sobre-oscilaciones, movimientos bruscos, o golpes a los objetos que se puedan producir por la inercia con la que se mueve el motor.

Para tener constancia de la fuerza que esta aplicando la pinza sobre el objeto se han realizado diferentes experimentos mediante los cuales; con poca fuerza no pueda levantar o sujetar determinados objetos, y con el incremento de la misma se los pueda sujetar y manipular sin problemas(vease figuras 3.46, 3.47, 3.48)

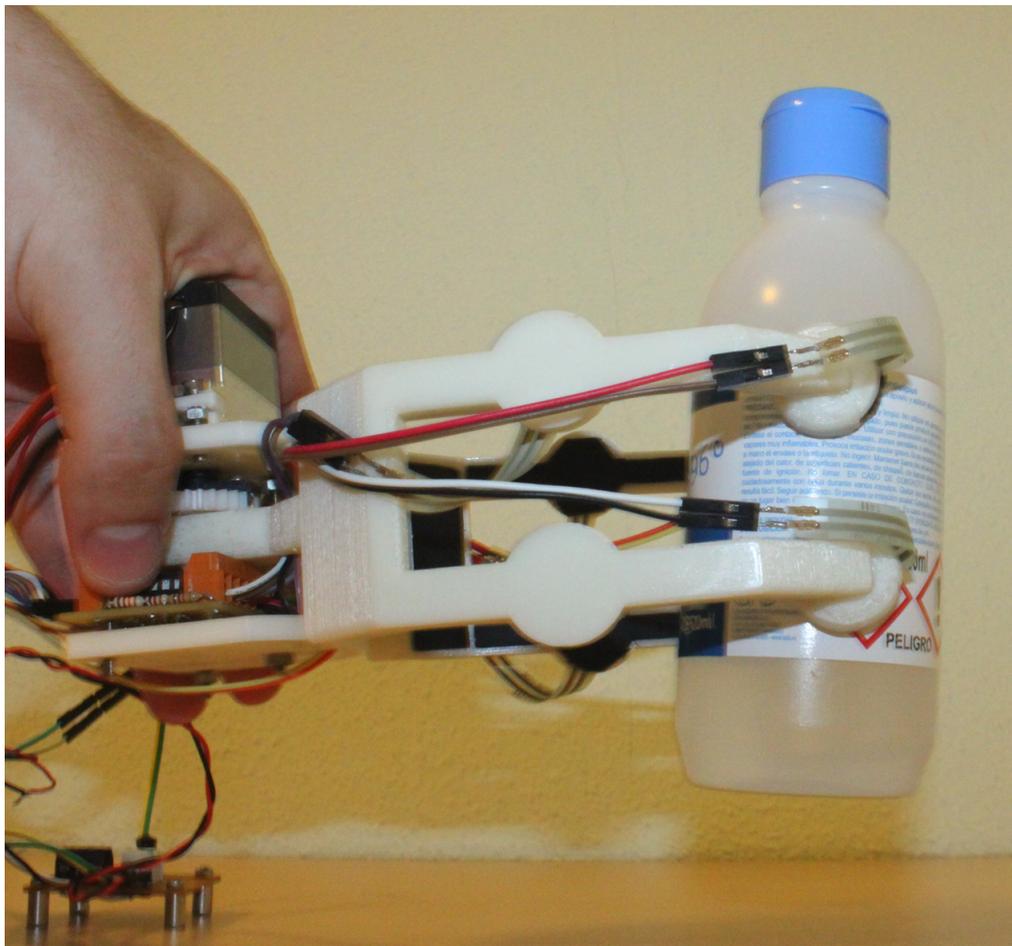


Figura 3.46: Objeto sujetado con una fuerza equivalente al 80 % de la capacidad total de la pinza

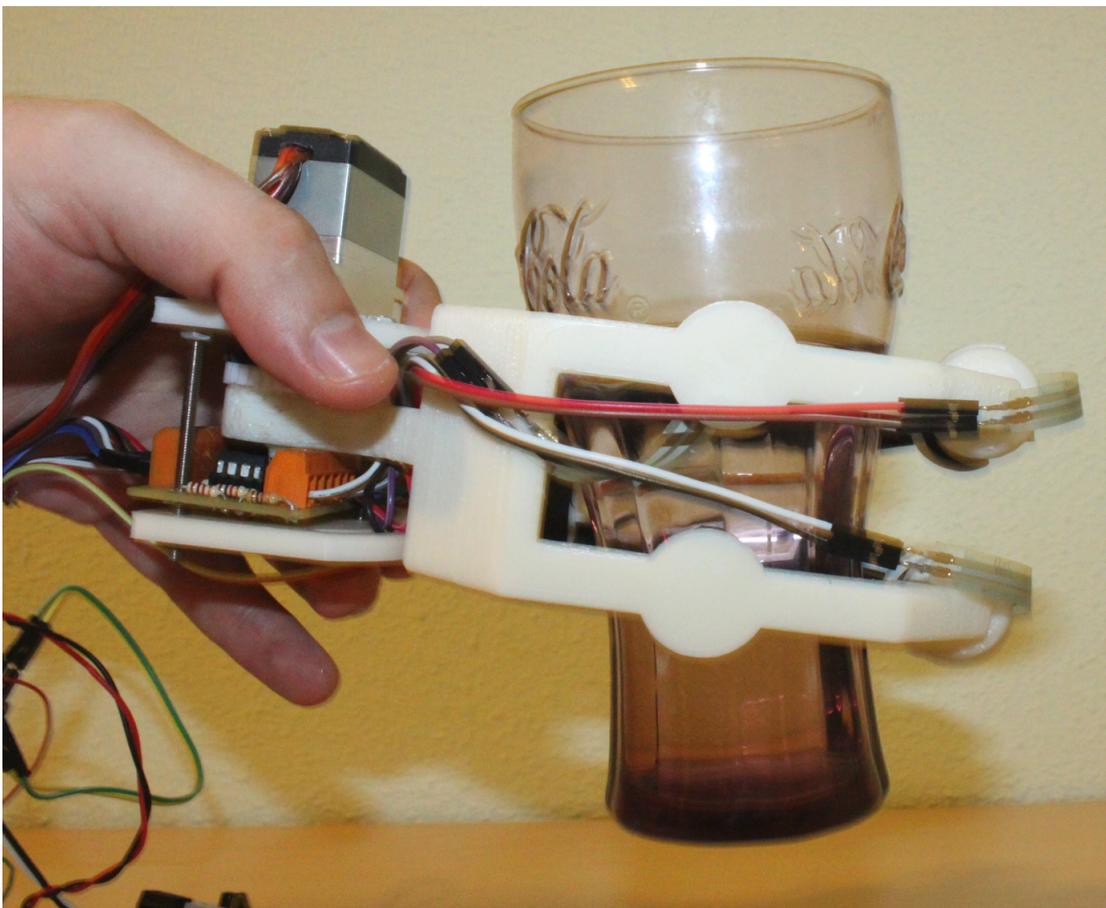


Figura 3.47: Objeto sujetado con una fuerza equivalente al 90 % de la capacidad total de la pinza

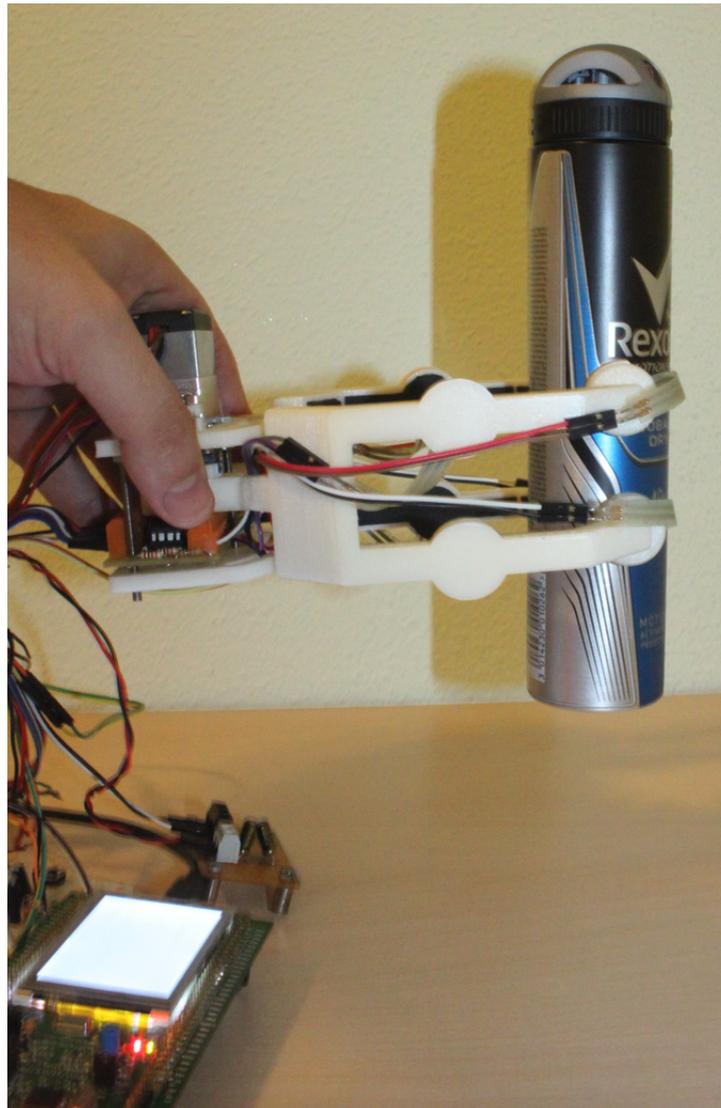


Figura 3.48: Objeto sujetado con una fuerza equivalente al 50 % de la capacidad total de la pinza

FUNCIONAMIENTO Y PRUEBAS

4

FUNCIONAMIENTO Y PRUEBAS

4.1. Conexión Tarjeta stm32f4xx

Para la conexión de la tarjeta con la pinza debemos tener en cuenta los siguientes puntos

- Puertos Análogos a utilizar
- Salidas PWM
- Interrupciones Externas
- GPIO's ocupados por el USB
- GPIO's ocupados por el SDRAM

Teniendo en cuenta los puntos anteriores, se ha realizado una evaluación de los posibles puertos a utilizar para la generación y obtención de las diferentes señales ya que la tarjeta Discovery dispone de varios GPIO, pero no todos son capaces de actuar de forma similar ya sea como ADC's, DAC's, generando señales PWM, activando el lcd, o que no se encuentren en funcionamiento por otros componentes como el USB o la SDRAM, suponiendo que se utilizará estos componentes al mismo tiempo.

4.1.1. GPIO a utilizar

Para la lectura de los datos análogos se ha utilizado la librería `tm_stm32f4xx_adc.h` proporcionada por [39], mediante la cual se dispone de 16 posibles puertos a utilizar en tres diferentes combinaciones, sin embargo como se ha dicho antes no todos están disponibles por lo que se ha hecho un análisis de todos estos puertos obteniendo los siguientes resultados.

De los cuales se ha determinado que en el presente proyecto los puertos funcionales a utilizar son.

- GPIOA 0,1,2,3,4,7
- GPIOB 0,1
- GPIOC 0,1,2,3,4,5

CHANNEL	ADC1	ADC2	ADC3
0	PA0	PA0	PA0
1	PA1	PA1	PA1
2	PA2	PA2	PA2
3	PA3	PA3	PA3
4	PA4	PA4	PF6
5	PA5	PA5	PF7
6	PA6	PA6	PF8
7	PA7	PA7	PF9
8	PB0	PB0	PF10
9	PB1	PB1	PF3
10	PC0	PC0	PC0
11	PC1	PC1	PC1
12	PC2	PC2	PC2
13	PC3	PC3	PC3
14	PC4	PC4	PF4
15	PC5	PC5	PF5

Tabla 4.1: GPIO Disponibles para funcionar como ADC's

Para la generación del PWM se ha utilizado los puertos GPIOA PA7 y GPIOA PA5 con diferentes timers para poder variar el duty.

Para los sensores externos que determinarán mecánicamente que la pinza a llegado a sus puntos donde no se produzcan singularidades se ha determinado los GPIOD PD2 y PD3, los cuales estarán activos al modo de interrupción.

Para la conexión del puerto USB se utiliza los GPIO PA9 y PA10 si el caso fuese necesario que se cambie los mismos esto se puede definir en el archivo denominado defines.h, en este mismo archivo se puede definir.

- Oscilador a utilizar
- GPIO's a ocupar por la SDRAM(referirse a la librería para el cambio de los mismos)
- Timer's a utilizar
- Cambiar puertos de conexión USB
- Puertos a usar con USART

Otra forma de definir estos modos es mediante la ventana Target correspondiente al programa keil tal como se muestra en la figura

En el presente proyecto se ha realizado las definiciones en la ventana de target siendo las cuales se listan a continuación.

1. STM32F429_439xx.- Tarjeta con la cual se va a trabajar definiciones dentro de la carpeta a la cual hace referencia

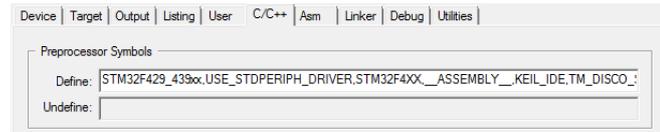


Figura 4.1: Definiciones en Target

2. USE_STDPERIPH_DRIVER.- Drivers a usar como USB, SDRAM, ADC's, etc.
3. STM32F4XX.- STM Disponibles
4. __ASSEMBLY__.- Ensamblador del programa keil
5. KEIL_IDE
6. TM_DISCO_STM32F429_DISCOVERY.- Librería Disco para la tarjeta usada
7. USE_USB_OTG_HS.- Definiciones para usar la SDRAM
8. HSE_VALUE=8000000,PLL_M=8,PLL_N=360,PLL_P=2,PLL_Q=7.- Definiciones para el oscilador externo

4.2. Librerías a utilizar

Durante el proyecto se han utilizado diferentes librerías (véase figura 4.2) las cuales controlarán interrupciones, adc, gpio, timers, pwm, etc, el listado completo se ubica en el Anexo B.

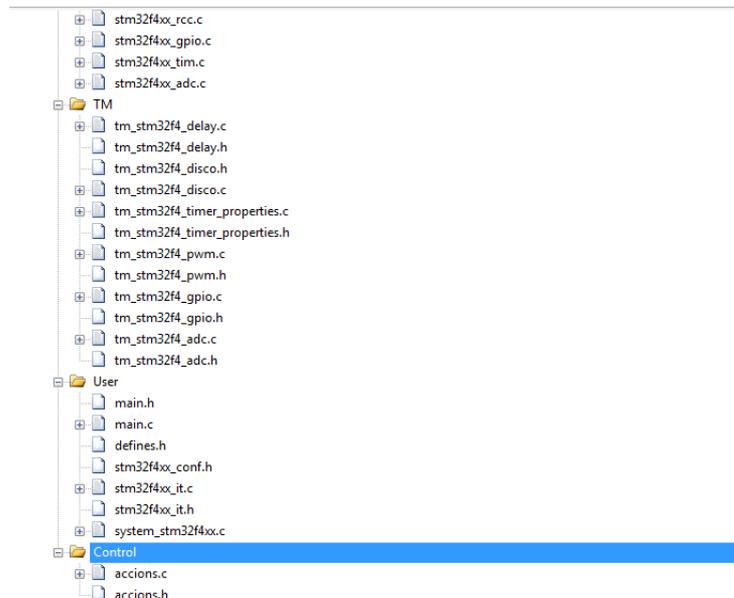


Figura 4.2: Algunas de las librerías utilizadas para el proyecto

También es importante tomar en cuenta las configuraciones que se deben realizar para el correcto funcionamiento de la tarjeta las cuales se realizarán, en el caso de utilizar keil¹, en el apartado target

¹<http://www2.keil.com/mdk5/uvision/>

donde se añadirán los paths de las librerías que se va a utilizar en el apartado C/C++(véase figura 4.3)

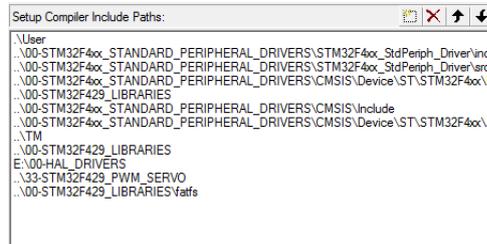


Figura 4.3: Path Discovery

El software keil también permite hacer configuraciones correspondientes a memorias externas, configuración para utilizar la ventana propia del simulador en el cual mediante la opción printf se pueden mostrar los resultados, en el presente proyecto esto no ha sido necesario ya que los datos son obtenidos mediante comunicación USB.

Para la comunicación entre la tarjeta y el ordenador se utiliza el programa SerialTools¹ en el cuál se seleccionará el puerto ocupado por la tarjeta tal como se muestra en la figura 4.4

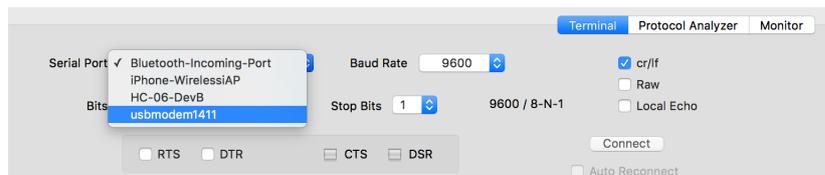


Figura 4.4: Puerto a usar por la tarjeta Discovery

La comunicación se ha configurado por default a 9600 baudios, 8 bits de datos, un bit de parada y sin paridad, esta configuración también puede ser modificada en el archivo defines.h

Una vez realizada la conexión con la Tarjeta utilizando las teclas a(mayúscula o minúscula) y la tecla enter se presentarán las diferentes acciones que puede realizar la pinza, como se muestra en la figura 4.5

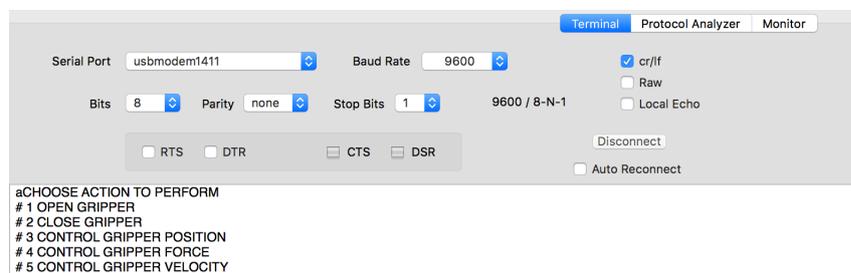


Figura 4.5: Posibles acciones a realizar

Mediante el teclado numérico del ordenador seleccionamos una opción(ejem. # 1 Open Gripper) y presionamos enter la tarjeta realizará la acción determinada y posteriormente nos dará un mensaje correspondiente a la acción realizada(véase figura 4.6).

¹<https://itunes.apple.com/es/app/serialtools/id611021963?mt=12>

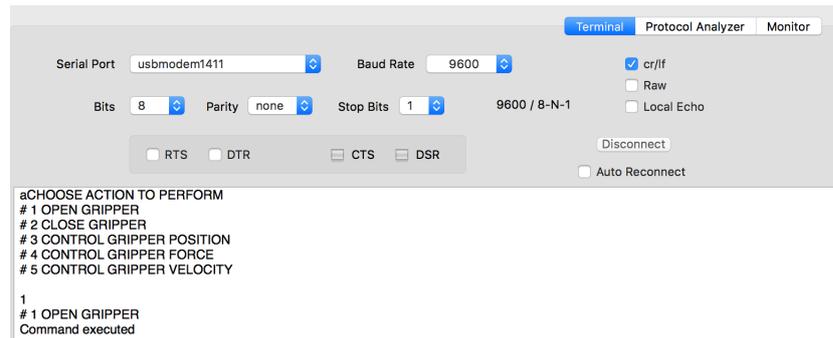


Figura 4.6: Forma de ejecutar las diferentes acciones

En el caso de que se desee abrir o cerrar la pinza en una determinada posición seleccionaremos la opción 3 y determinaremos la posición a la cual deseamos que se mueva (véase figura 4.7), el mismo procedimiento se realizará tanto para fuerza como para velocidad.

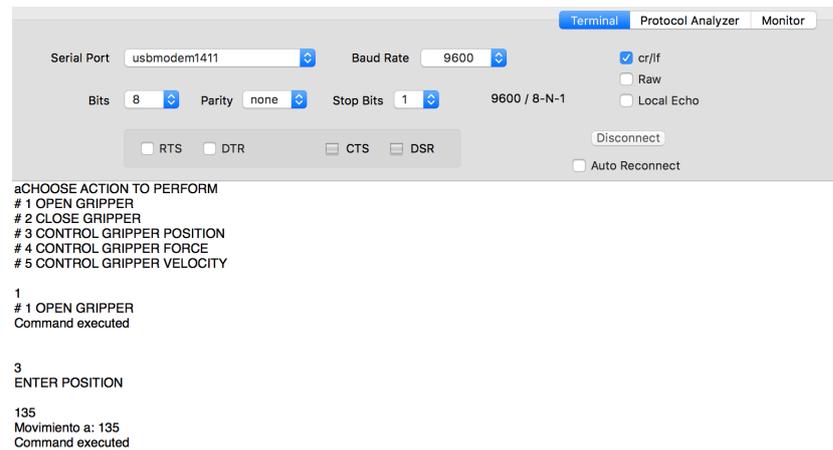


Figura 4.7: Ingresando mas de un valor para determinar posición

Los esquemas para conexión de las diferentes placas se muestran en el Anexo C.

El software implementado para el presente proyecto se muestra en el Anexo D.

**CONCLUSIONES Y
RECOMENDACIONES**

5

CONCLUSIONES Y RECOMENDACIONES

Cumplidos finalmente los objetivos propuestos y finalizada la investigación y ejecución del presente TFM, se exponen las siguientes conclusiones y recomendaciones.

5.1. Conclusiones

1. Queda demostrado mediante la práctica y experimentación, que es posible la implementación y control de un pinza con un único servo-motor de bajas capacidades.
2. Se ha diseñado una pinza la cual cuenta con cuatro dedos funcionales los cuales mediante la practica han determinado su funcionamiento, mejorando el agarre, para determinados objetos.
3. La calibración de los sensores de fuerza es básica ya que de esta depende el control que posteriormente tendrá la pinza en el sentido de la fuerza con la que ejecute las acciones de agarre.
4. El potenciómetro con el cual cuenta el servo motor genera demasiado ruido a la salida del mismo, mediante la experimentación y con un número determinado en 50 muestras se ha podido disminuir eficientemente la cantidad de ruido que este genera.
5. La identificación de la función de transferencia que se ha realizado para el presente proyecto a permitido desarrollar controladores que funcionen de manera optima tanto para posición como para fuerza
6. La comunicación USB mediante la experimentación se ha definido que no es lo suficientemente robusta como para trasmitir datos cada mili segundo produciéndose: perdida de datos, colapso en el sistema, lectura de datos errónea por parte de la tarjeta al ser sobrecargado el procesador de la misma.
7. El rango máximo de movilidad de la pinza esta determinado a 30 grados, dependiendo del diseño este se pudiese ampliar o reducir.
8. El uso de la SDRAM es necesario para poder almacenar los datos que posteriormente serán presentados por el canal USB, para lo cual se ha definido una dirección de memoria de 16 bits por

muestra, se ha llegado a almacenar en el presente proyecto hasta 50000 muestras en la fase de experimentación e identificación sin presentar error alguno

9. Si los datos presentados por la SDRAM son totalmente incoherentes al análisis que se está realizando, es probable que no se hayan almacenado los datos o no esté leyendo correctamente la dirección donde se ha almacenado los mismos.
10. El motor para iniciar su movimiento necesita al menos del 40 % de su capacidad total, esto se debe al rozamiento que existe entre los ejes a los cuales se encuentra sujeto el motor y los diferentes componentes que acompañan al movimiento, como los dedos, sensores, piñones, etc.
11. El controlador que se ha desarrollado para el control de posición cumple especificaciones, evitando el sobre-amortiguamiento del sistema en el régimen transitorio y la estabilidad del mismo en el régimen permanente, también alcanza sin problemas la referencia.
12. Se ha probado el motor para diferentes posiciones y no ha dado problemas para alcanzar las mismas salvo en aquellos casos que se quiera trasladar de un grado a su inmediato superior en la escala de uno debido al rozamiento, ya que este parámetro ha sido ajustado mediante la práctica el sistema ejecuta el movimiento para la mayoría de los casos.
13. La constante que se ha determinado para medir la velocidad se lo ha hecho llevando el motor desde su posición más baja a la más alta y evaluando la pendiente que este tipo de movimientos ha producido, constatando de que el valor generado es constante y uniforme, las pequeñas variaciones que en el mismo se presentan se deben al ruido del potenciómetro.
14. Se ha probado la estimación de fuerza con diferentes objetos siendo necesario aplicar mayor fuerza a los que tienen mayor peso para poder sostenerlos, a los mismos si se les aplica menor fuerza la pinza no es capaz de sostenerlos, validando el funcionamiento de este tipo de control.
15. No ha sido posible constatar visualmente los datos proporcionados por cada uno de los sensores de fuerza ya que al momento de utilizar los mismos, no se puede trabajar con la SDRAM, lo cual evita que se guarden los datos generados por los sensores e imposibilita la lectura de los mismos a través de la pantalla, sin embargo para determinar que los sensores están llegando a ejercer la fuerza indicada, se ha hecho uso de un condicional el cual devolvería un determinado estado cuando la fuerza aplicada sea igual a la referencia, obteniendo resultados positivos.

5.2. Recomendaciones

1. Es recomendable trabajar con un motor de mejores prestaciones que el usado en el presente proyecto de esta manera se podría aplicar mayor presión a los objetos.
2. Tener un buen diseño de la pinza es un punto indispensable en la partida del proyecto, en el cual se deben considerar las acciones que se va a realizar, el peso que tiene que soportar, la fuerza que va a ejercer, etc.
3. No sobre-esforzar el modelo de prueba si el mismo esta desarrollado en ABS, este material es demasiado frágil para aplicar cantidades altas de torque en el motor pudiendo producir fisuras o la in-habilitación del modelo experimental.
4. Se debe tener especial consideración en el manejo y programación de la interfaz USB ya que esta puede servir de gran ayuda en la obtención de datos e interacciones repetitivas con el dispositivo pudiendo variar los datos repetidamente a modo de experimentación.
5. Es necesario tener una idea clara sobre lo que se va ha hacer y que dispositivos se va a usar si los dispositivos que se van a usar son los mencionados en el presente proyecto es recomendable empezar por montar el USB, SDRAM, ADC, PWM, GPIO, en ese orden respectivamente, esto evitará que se tenga que modificar código cada vez que se adhiere un nuevo dispositivo.
6. Es importante tomar varias muestras para poder definir un modelado eficiente para la pinza diseñada, sobre todo es recomendable experimentar en le rango que se va a trabajar y con todos los dispositivos montados, así no se tendrá que modificar en lo posterior constantes como las ganancias del controlador o la constante de rozamiento.
7. Al momento de compilar el programa es preferible tener deshabilitada la opción de búsqueda dentro del Target esto aumentará la velocidad con la cuál el software compila.

Bibliografía

- [1] SÁNCHEZ MARTÍN FM, MILLÁN RODRÍGUEZ F, SALVADOR BAYARRI J, PALOU REDORTA J, RODRÍGUEZ ESCOVAR F, ESQUENA FERNÁNDEZ S, VILLAVICENCIO MAVRICH H *Historia de la robótica: de Arquitas de Tarento al robot Da Vinci*, pp. 69-75 ,(2006) 6, 7
- [2] RIA *Robotic Industries Association* 12 Diciembre. (2015),[en línea],disponible en:<http://www.robotics.org/> 7
- [3] ALARCON, MARIA ISABEL *Fidelity: el futuro de la automatización robotizada* 12 Diciembre. (año),[en línea],disponible en:<http://www.rankia.com/blog/fondos-inversion/2417082-fidelity-futuro-automatizacion-robotizada> IV, 7
- [4] PÉREZ, OMAR *Agricultura* 15 Diciembre. (2015),[en línea],disponible en:<http://perezrodriguez1.blogspot.com.es/2014/07/robotica-en-la-agricultura-y-la.html> 8
- [5] UNIVERSIDAD CARLOS III MADRID *Robots que reducen tiempo y dinero en la construcción* 15 Diciembre. (2015),[en línea],disponible en:http://portal.uc3m.es/portal/page/portal/actualidad_cientifica/noticias/robots_construccion 9
- [6] VALPARAISO *Robótica Submarina* 17 Diciembre. (2015),[en línea],disponible en:<http://www.skyringmarine.cl/index.htm> 9
- [7] RT *Video: marines de EE.UU practican con el asombroso robot militar de Google* 18 Diciembre. (2015),[en línea],disponible en:<https://actualidad.rt.com/actualidad/186615-video-marines-eeuu-practican-asombroso> 9
- [8] DR. GENE EMMER *Exoesqueleto* 05 Enero. (2016),[en línea],disponible en:<http://www.exoesqueleto.com.es/iquestcoacutemo-funciona-un-exoesqueleto.html> 14
- [9] SAMUEL FRANCO DOMÍNGUEZ *Blog de rehabilitación que mira el futuro* 17 Enero. (2016),[en línea],disponible en:<http://www.rehabilitacionblog.com/2012/02/exoesqueleto-robotico-de-kawasaki.html> IV, 14
- [10] MARSİ-BIONICS *Smart Teraphy* 17 Enero. (2016),[en línea],disponible en:<http://www.smartherapy.org/crowdfunding-marsi-bionics/> IV, 14
- [11] FRANCISCO PÉREZ MARCIAL *Diseño, Análisis y Modelado Cinemático de un Exoesqueleto Pasivo de Extremidad Inferior con Propósito de Rehabilitación*, pp. 17-100 ,(2011) 16, 19
- [12] ING. ANDRÉS E. GÓMEZ H., ING. MAURICIO CARRILLO, MSc. JOHN FABER ARCHILA DÍAZ *ESTUDIO DE AYUDAS TECNICAS TIPO EXOESQUELETO PARA APOYO A PACIENTES CON DESORDENES EN EL MOVIMIENTO*, pp. 52-59 ,(2009) 17
- [13] J. ROSEN, M. BRAND, M. FUCHS, AND M. ARCAN *A myosignal-based powered exoskeleton system. IEEE Transaction On Systems, Man and Cybernetics - Part A: Systems and Humans*, pp. 210-222 ,(2001) 17
- [14] LOKHEED MARTIN *Lokheed Martin* 17 Enero. (2016),[en línea],disponible en:<http://www.lockheedmartin.com/> IV, 18, 19

-
- [15] MICHAEL C. YIPA, MAHDI TAVAKOLIBAND, ROBERT D. HOWE *Performance Analysis of a Haptic Telemanipulation Task under Time Delay*, pp. 651-673 ,(2011) IV, 20
- [16] CYBER GRASP *Cyber Glove System* 20 Enero. (2016),[en línea],disponible en:<http://www.cyberglvesystems.com/cybergrasp/> IV, 21
- [17] LOKOMAT *TERAPIA FUNCIONAL DE MARCHA ROBOTIZADA* 22 Enero. (2016),[en línea],disponible en:<https://www.hocomat.com/world/en/products/lokomat/> IV, 21
- [18] J. A. COZENS. *Robotic assistance of an active upper limb exercise in neuro- logically impaired patient. IEEE Transaction On Rehabilitation Engineering*, pp. 254-256 ,(1999) 22
- [19] H. I. KREBS, N. HOGAN, M. L. AISEN, AND B. T. VOLPE. *Robot-aided neurore- habilitation. IEEE Transaction On Rehabilitation Engineering*, pp. 75-88 ,(1998) 22
- [20] EQUIPO DE AMAZINGS *Prótesis robótica con músculos artificiales para rehabilitación de pie y tobillo* 08 Febrero. (2016),[en línea],disponible en:<http://noticiasdela ciencia.com/> IV, 22
- [21] CHRISTIAN ORTEGA LOUBON¹, JULIO CÉSAR FRANCO². *Neurofisiología del aprendizaje y la memoria. Plasticidad Neuronal*, pp. 1-7 ,(2010) 23
- [22] VICTOR R. GONZALEZ *Elementos terminales o actuadores finales*, 10 Febrero. (2016),[en línea],disponible en:<http://platea.pntic.mec.es/vgonzale> VII, 25
- [23] FIB *Elementos Finales* 15 Febrero. (2016),[en línea],disponible en:<http://www-pagines.fib.upc.es> 27
- [24] SAULOMA *Pinzas y ejes de mano* 18 Febrero. (2016),[en línea],disponible en:<http://www.profesaulosuna.com> IV, 28, 29
- [25] SIEGERT, H.-J.; BOCIONEK, S.: ROBOTIK *Programmierung intelligenter Roboter*, pp. 446-523 ,(1996) 28
- [26] CÉSAR AUGUSTO QUINAYÁS-BURGOS, MARIELA MUÑOZ-AÑASCO, ÓSCAR ANDRÉS VIVAS-ALBÁN, CARLOS ALBERTO GAVIRIA-LÓPEZ *Diseño y construcción de la prótesis robótica de mano*, pp. 224-236 ,(2010) 29
- [27] CHRISTIAN AUGUSTO SILVA CASTELLANOS, JHON EDISON MUÑOZ RIAÑOS, DIEGO ALEXANDER GARZÓN ALVARADO, NANCY STELLA LANDÍNEZ PARRA, OCTAVIO SILVA CAICEDO *Revista Cubana de Investigaciones Biomédicas*, pp. 15-41 ,(2011) 29
- [28] OTTOBOCK *Prótesis de mano SensorHand Speed y VariPlus Speed* 01 Marzo. (2016),[en línea],disponible en:<http://www.ottobock.es/> 30
- [29] KAWASSAKI *departamento de mecatrónica robótica* 22 Enero. (2016),[en línea],disponible en:<http://www.rm.kanagawa-it.ac.jp/> IV, IV, 23, 24
- [30] ANDRÉS FELIPE RUÍZ OLAYA *Sistema Robótico Multimodal para Análisis y Estudios en Biomecánica, Movimiento Humano y Control Neuromotor*, pp. 12-256 ,(2008) IV, IV, 10, 12, 19
- [31] K.P. TEE, E. BURDET, C.M. CHEW, AND T.E. MILNER *A model of force and impedance in human arm movements*, pp. 368-375 ,(2004) IV, 11
- [32] G. J. MONKMAN, S. HESSE, R. STEINMANN, H. SCHUNK *Robot Grippers*, pp. 19-72 ,(2007) IV, 33, 35
- [33] UN ESPACIO PARA LA FÍSICA!!! *Conociendo la física* 04 Abril. (2016),[en línea],disponible en:<https://conociendolafisica.files.wordpress.com/2010/04/8mcu.pdf> 37
- [34] RS *RS Components* 07 Abril. (2016),[en línea],disponible en:<http://docs-europe.electrocomponents.com/webdocs/09ad/0900766b809ad3ef.pdf> IV, 41

- [35] INTERLINK ELECTRONICS *FSRR Integration Guide y Evaluation Parts Catalog*, pp. 1-24 ,(2009) v, 46, 49
- [36] TEXAS INSTRUMENTS *TL7660CMOS VOLTAJE CONVERTER*, pp. 1-23 ,(2006) 50
- [37] TEXAS INSTRUMENTS *L293x Quadruple Half-H Drivers*, pp. 1-21 ,(2015) 50
- [38] ILBER ADONAYT RUGE RUGE *METODO BASICO PARA IMPLEMENTAR UN CONTROLADOR DIGITAL PID EN UN MICROCONTROLADOR PIC PARA DESARROLLO DE APLICACIONES A BAJO COSTO*, pp. 1-10 ,(2014) v, 73
- [39] TILEN MAJERLE *stm32f4 Discovery* Mayo-julio. (2016),[en línea],disponible en:<http://stm32f4-discovery.net> 53, 82

ANEXOS

ANEXO A

Cotas de la pinza

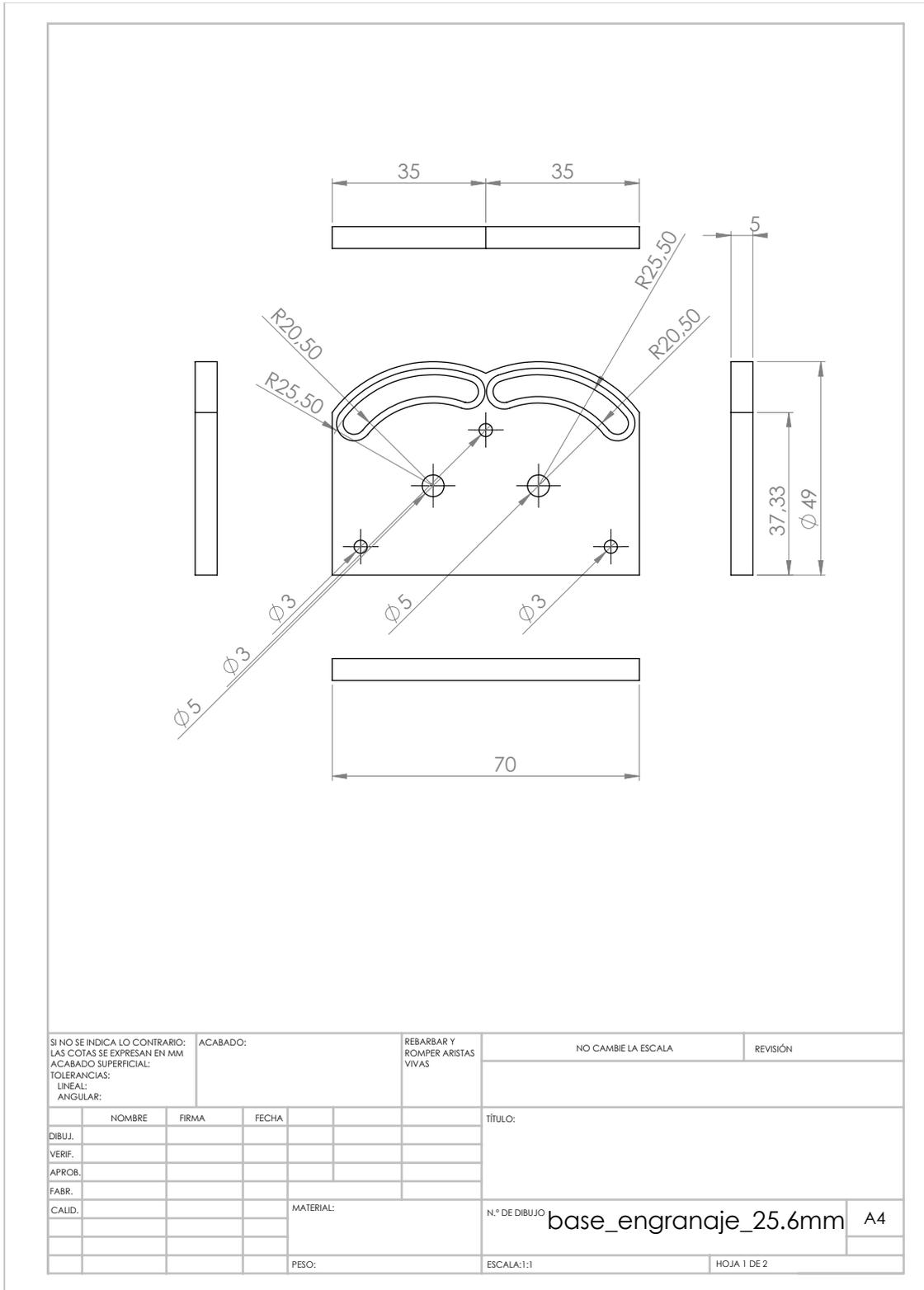


Figura A.1: Medidas Cara inferior de la pinza

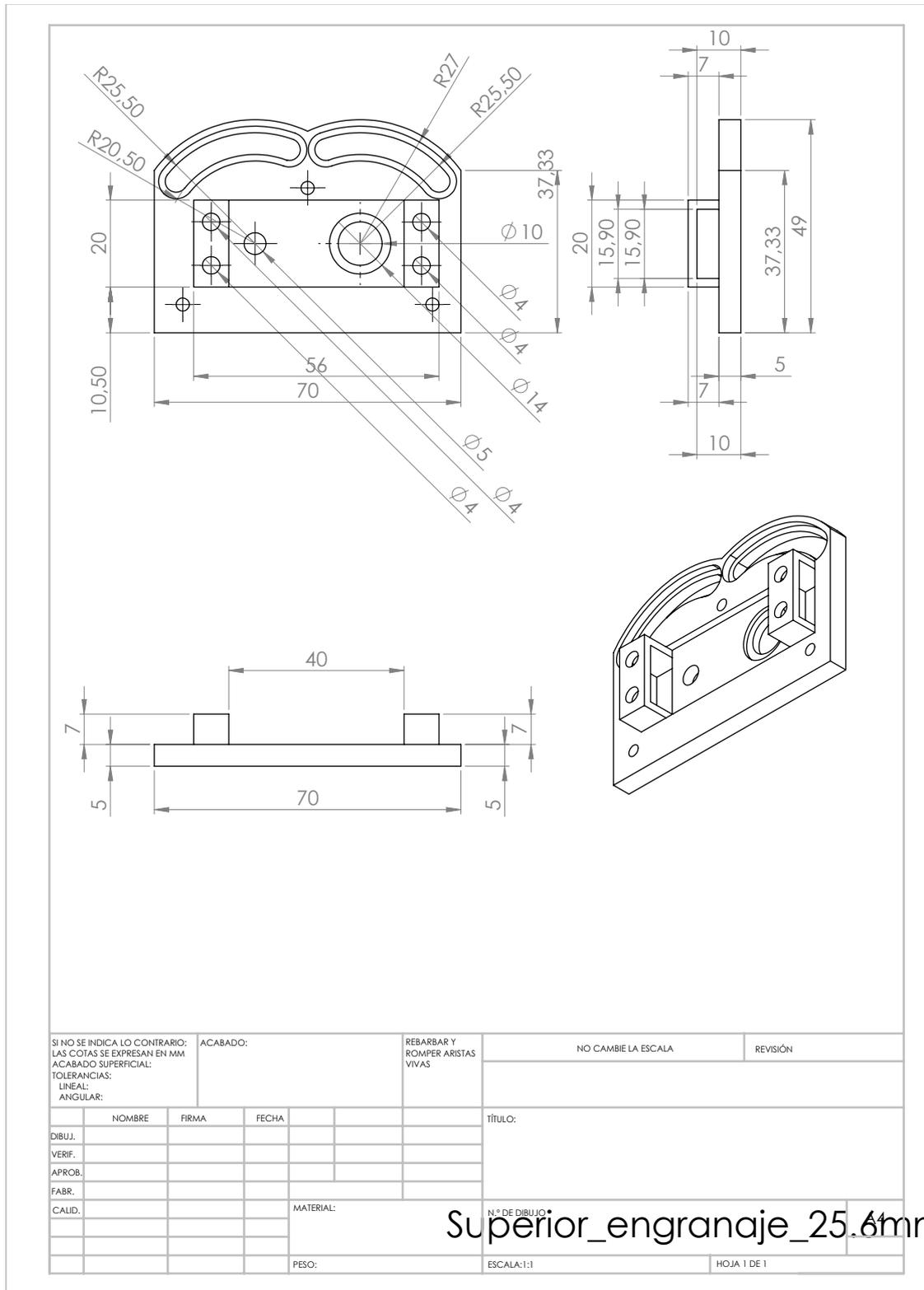


Figura A.2: Medidas Cara superior de la pinza

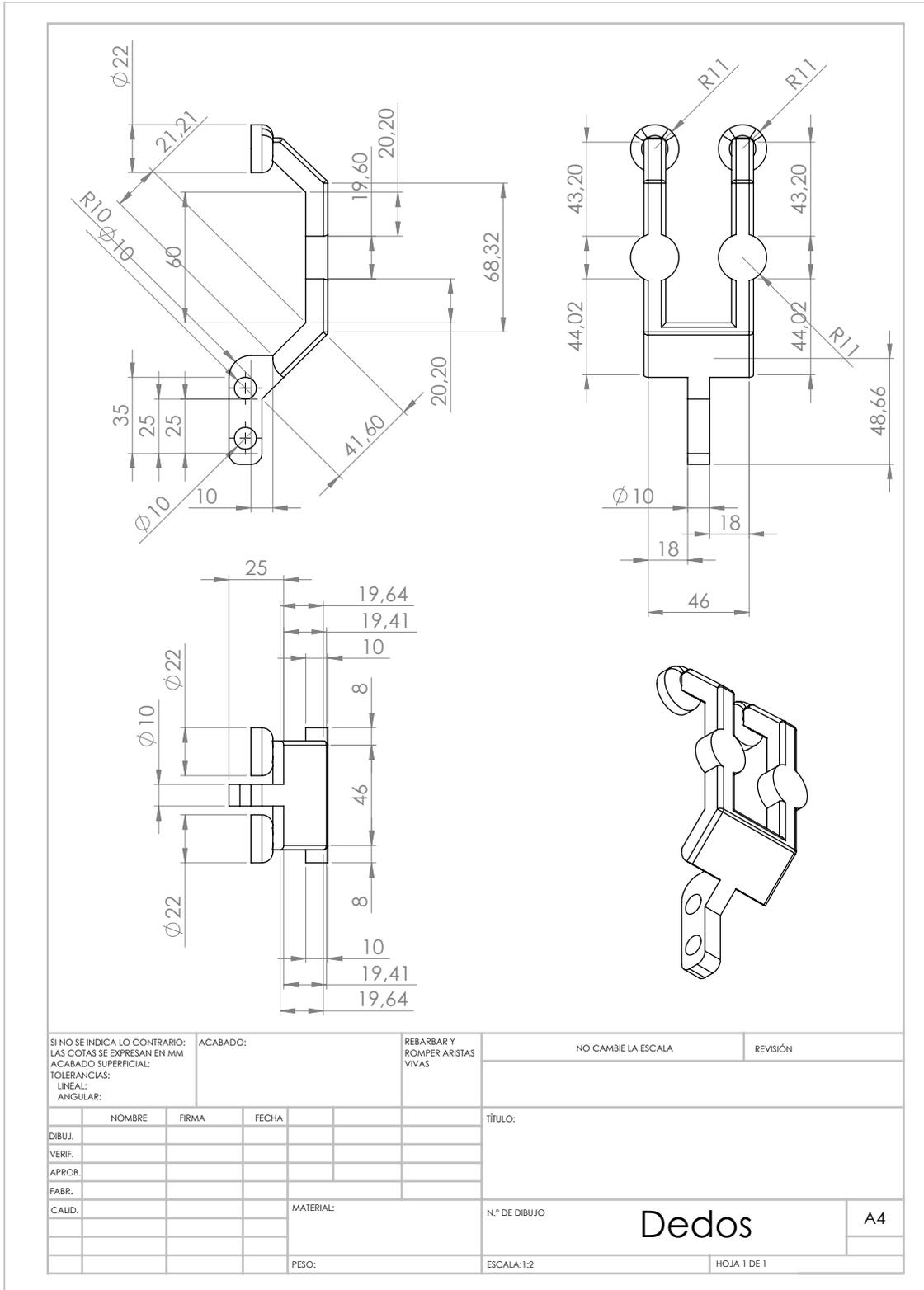


Figura A.3: Medidas ambos dedos

ANEXO B

Librerías Usadas

B.1. USB

1. tm_stm32f4_usb_vcp.c
2. tm_stm32f4_usb_vcp.h
3. usb_bsp.c
4. usb_bsp.h
5. usb_conf.h
6. usb_core.c
7. usb_core.h
8. usb_dcd_int.c
9. usb_dcd_int.h
10. usb_dcd.c
11. usb_dcd.h
12. usb_defines.h
13. usb_regs.h
14. usb_cdc_core.c
15. usb_cdc_core.h
16. usb_cdc_vcp.c
17. usb_cdc_vcp.h
18. usb_def.h

19. usb_desc.c
20. usb_desc.h
21. usb_ioreq.c
22. usb_ioreq.h
23. usb_req.c
24. usb_req.h
25. usb_usr.c
26. usb_usr.h

B.2. ADC

1. tm_stm32f4_adc.c
2. tm_stm32f4_adc.h
3. stm32f4xx_adc.c

B.3. PWM

1. tm_stm32f4_pwm.c
2. tm_stm32f4_pwm.h

B.4. SDRAM

1. tm_stm32f4_sdram.c
2. tm_stm32f4_sdram.h

B.5. TIMER

1. tm_stm32f4_timer_properties.c
2. tm_stm32f4_timer_properties.h
3. stm32f4xx_tim.c

B.6. Librerías generales

1. tm_stm32f4_delay.c
2. tm_stm32f4_delay.h
3. tm_stm32f4_disco.c
4. tm_stm32f4_disco.h
5. tm_stm32f4_gpio.c
6. tm_stm32f4_gpio.h
7. stm32f4xx_rcc.c
8. stm32f4xx_exti.c
9. stm32f4xx_gpio.c
10. stm32f4xx_fmc.c

B.7. Librerías creadas para el presente proyecto

1. acciones.c
2. acciones.h

ANEXO C

Esquemas para circuito de sensores

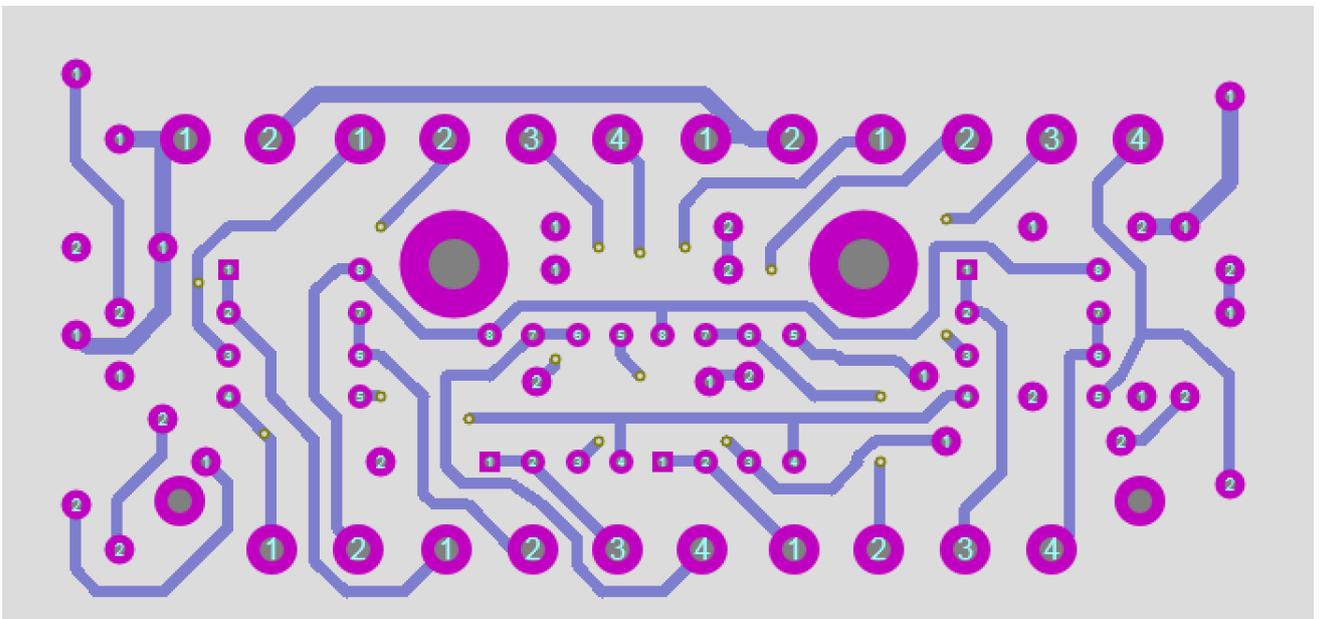


Figura C.1: Placa acondicionada para sensores de fuerza

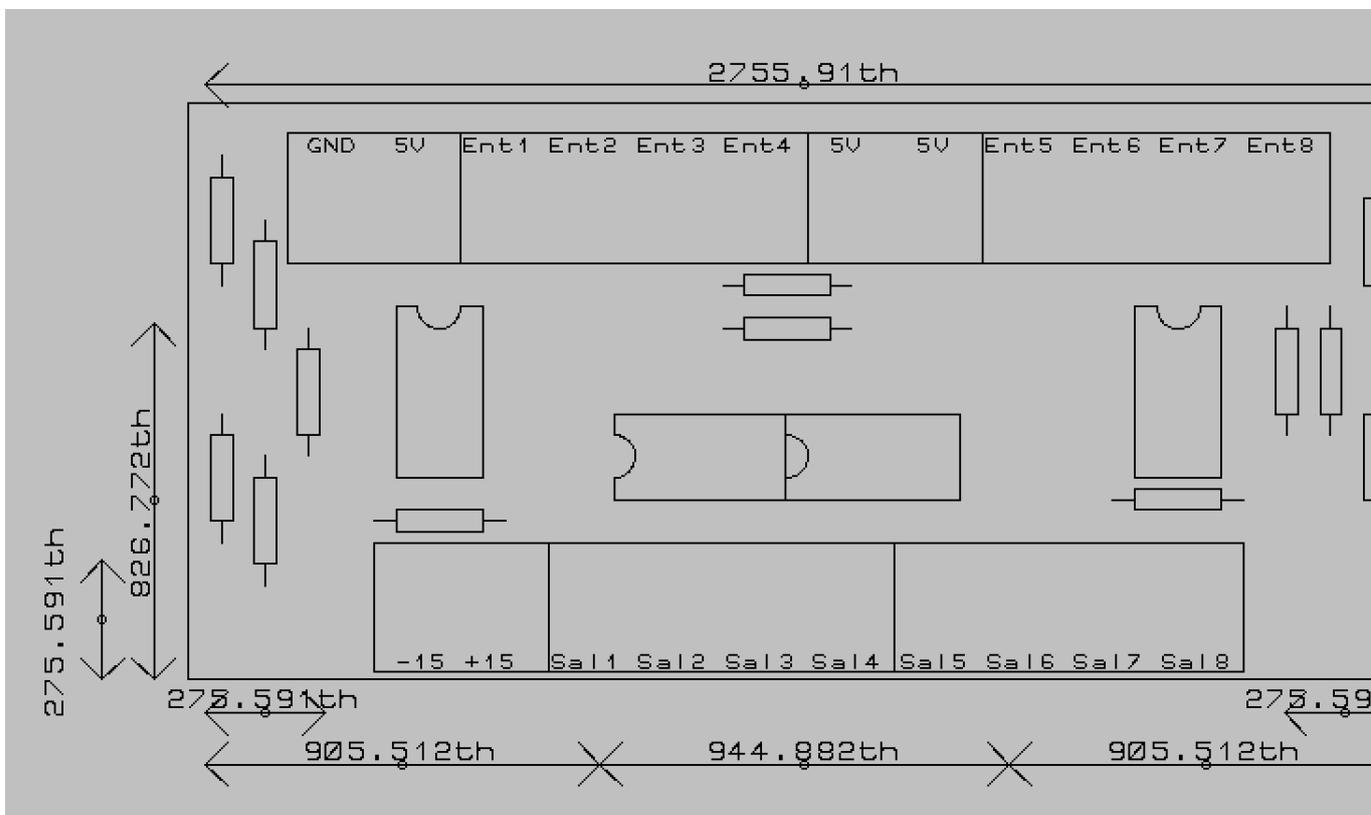


Figura C.2: Top silk para nombrar los diferentes puntos de la placa

ANEXO D

Código desarrollado

D.1. main.c

```
/**
 * Keil project for gripper
 *
 *
 * @author Jefferson Camacho
 * @email jefferson-camacho@hotmail.com
 * @website -----
 * @ide Keil uVision 5
 * @packs STM32F4xx Keil packs version 2.2.0 or greater required
 * @stdperiph STM32F4xx Standard peripheral drivers version 1.4.0 or greater required
 */
#include "stm32f4xx.h"
#include "defines.h"
#include "tm_stm32f4_pwm.h"
#include "tm_stm32f4_disco.h"
#include "tm_stm32f4_delay.h"
#include "tm_stm32f4_adc.h"
#include <stdio.h>
#include "acciones.h"

#include "tm_stm32f4_usb_vcp.h"
#include "defines.h"
#include <string.h>
#include <stdlib.h>
#include "tm_stm32f4_sdram.h"

#define LED GPIO_PIN_2

char txt[10],tyt[21];
int state=1;

uint8_t c,i;

void setBuffer(void)
{
int total = strlen(txt);
for (int x=0; x <= total; x++)
{
txt[x]= '\0';
}
}
```

```
total = strlen(tyt);
for (int x=0; x <= total; x++)
{
tyt[x]= '\0';
}
i=0;
}

void options(void)
{
Delayms(100);
TM_USB_VCP_Puts("CHOOSE ACTION TO PERFORM\n");
TM_USB_VCP_Puts("# 1 OPEN GRIPPER\n");
TM_USB_VCP_Puts("# 2 CLOSE GRIPPER\n");
TM_USB_VCP_Puts("# 3 CONTROL GRIPPER POSITION\n");
TM_USB_VCP_Puts("# 4 CONTROL GRIPPER FORCE\n");
TM_USB_VCP_Puts("# 5 CONTROL GRIPPER VELOCITY\n\n");
}

void acciones(int a)
{
int b;
switch(a)
{
case 1:
TM_USB_VCP_Puts("# 1 OPEN GRIPPER\n");
state = 1;
while(state == 1)
{
state = gripperPosControl(120);
}
setBuffer();
break;

case 2:
TM_USB_VCP_Puts("# 2 CLOSE GRIPPER\n");
state = 1;
while(state == 1)
{
state = gripperPosControl(143);
}
setBuffer();
break;

case 3:
setBuffer();
state = 5;
TM_USB_VCP_Puts("ENTER POSITION\n");
while(state == 5)
{
if (TM_USB_VCP_Getc(&c) == TM_USB_VCP_DATA_OK)
{
TM_USB_VCP_Putc(c);
if (c == 13)
{
a = atoi(txt);
if (a <= 109 || a >= 146)
{
TM_USB_VCP_Puts("THE POSITION IS NOT POSSIBLE RANGE 115 TO 143\n");
TM_USB_VCP_Puts("ENTER ANOTHER VALUE\n");

```



```
else
{
state = 1;
while(state == 1)
{
state = gripperForceControl(b,a);
}
setBuffer();
}
// state = gripperForceControl(2000,0);
// // if (state == 2)
// // TM_USB_VCP_Puts("Sharp Object\n"), state = 1;
// // if (TM_USB_VCP_Getc(&c) == TM_USB_VCP_DATA_OK)
// // {
// // TM_USB_VCP_Putc(c);
// // if (c == 13)
// // {
// // TM_USB_VCP_Puts("Loose Object\n"), state = 2;
// // state = gripperPosControl(130);
// // }
// // }
// }
// setBuffer();
// }
}
else
{
txt[i] = (char) c;
i++;
}
}
}
}
}
else
{
txt[i] = (char) c;
i++;
}
}
}
break;

case 5:
break;
}
Delaysms(100);
}

int main(void) {
//int16_t read;
int a;
//
// /* Initialize system */
SystemInit();
//
// /* Initialize delay functions */
TM_DELAY_Init();

startPWM();
```

```

startADC();

    TM_USB_VCP_Init();
// TM_ADC_Init(ADC1, ADC_Channel_9);
// TM_ADC_Init(ADC1, ADC_Channel_11);

/* Initialize onboard leds */
TM_DISCO_LedInit();
//startOpenCloseControl();

// /* Set PWM to 1KHz frequency on timer TIM2 */
// /* 50Hz = 20ms = 20000us */
// TM_PWM_InitTimer(TIM1, &TIM1_Data, 1000);
// TM_PWM_InitTimer(TIM2, &TIM2_Data, 1000);
//
// /* Initialize PWM on TIM2, Channel 1 and PinsPack 2 = PA5 Conectado al Pin 2 del Puente H*/
// /* Initialize PWM on TIM1, Channel 2 and PinsPack 1 = PA7 Conectado al Pin 7 del Puente H*/
// BOTONES de fin de carrera conectados a PD2 y PD3 verificaci n en estado alto*/
// Valores ADC Para control del Motor Conectados al PIN PF6*/
//
// /* Set channel 1 value, 1500us = servo at center position */
// //TM_PWM_SetChannelMicros(&TIM2_Data, TM_PWM_Channel_1, 00);
//

/*Puertos de Conexi n ADC
ADC_Channel0 0 PA0
ADC_Channel3 0 PA3
ADC_Channel7 0 PA7
ADC_Channel8 0 PBO
ADC_Channel9 0 PB1
ADC_Channel10 0 PC0
ADC_Channel11 0 PC1
ADC_Channel12 0 PC2*/
//
TM_GPIO_Init(GPIOD, LED, TM_GPIO_Mode_OUT, TM_GPIO_OType_PP, TM_GPIO_PuPd_NOPULL, TM_GPIO_Speed_Hi

if (TM_SDRAM_Init()) {
TM_DISCO_LedOn(LED_GREEN);
} else {
TM_DISCO_LedOn(LED_RED);
}

while (1) {
if (TM_USB_VCP_GetStatus() == TM_USB_VCP_CONNECTED)
{
TM_DISCO_LedOn(LED_RED);
/* If something arrived at VCP */
if (TM_USB_VCP_Getc(&c) == TM_USB_VCP_DATA_OK)
{
TM_USB_VCP_Putc(c); //Acciones(c);

if (c == 65 || c == 97) //if c = a or A
{
options();
setBuffer();
}
else if (c == 13) //key Carrer
{
a = atoi(txt); //Take all data of txt

```



```

//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%EVALUACION ADC's%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
}
}

```

D.2. acciones.c

```

/**
 * @author   Jefferson Camacho
 * @email    jefferson-camacho@hotmail.com
 * @version  v1.0
 * @ide      Keil uVision
 * @license  GNU GPL v5
 * @brief    Accions Gripper library for STM32F4xx
 *
 * @verbatim
-----
Copyright (C) Jefferson Camacho, 2016

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program. If not, see <http://www.gnu.org/licenses/>.
-----
@endverbatim
 */

#include "stm32f4xx.h"
#include "defines.h"
#include "tm_stm32f4_pwm.h"
#include "tm_stm32f4_disco.h"
#include "tm_stm32f4_delay.h"
#include "tm_stm32f4_adc.h"
#include <stdio.h>
#include "acciones.h"

#include <time.h>//incluida para temporizar la funci n PD_trayectoria
#include <string.h>
#include <stdint.h>//clock
//#include "tm_stm32f4_fatfs.h"

#include "tm_stm32f4_sdram.h"
#include "tm_stm32f4_usb_vcp.h"

#define ButtonA GPIO_PIN_2
#define ButtonB GPIO_PIN_3

TM_PWM_TIM_t TIM2_Data;
TM_PWM_TIM_t TIM1_Data;

```

```

int8_t State;

int16_t valSensorPos;
int16_t valSensorForce;

int timeCont=0;

void CustomTIMER1_Task(void* UserParameters);
int PIDControl(uint16_t valRef, uint16_t valSensor, int8_t ctrl);
TM_DELAY_Timer_t* CustomTimer1;
//int8_t count;
int cont;
char tyt1[21];

void setBuffer1(void)
{
int total = strlen(tyt1);
for (int x=0; x <= total; x++)
{
tyt1[x]= '\0';
}
}

void startOpenCloseControl(void)
{
TM_GPIO_Init(GPIOD, ButtonA, TM_GPIO_Mode_IN, TM_GPIO_OType_PP, TM_GPIO_PuPd_DOWN, TM_GPIO_Speed_Low);
TM_GPIO_Init(GPIOD, ButtonB, TM_GPIO_Mode_IN, TM_GPIO_OType_PP, TM_GPIO_PuPd_DOWN, TM_GPIO_Speed_Low);

TM_ADC_Init(ADC3, ADC_Channel_4);
}

void startPWM(void)
{
//PWM righth Turn

TM_PWM_InitTimer(TIM2, &TIM2_Data, 1000);

TM_PWM_InitChannel(&TIM2_Data, TM_PWM_Channel_1, TM_PWM_PinsPack_2);

TM_PWM_SetChannelMicros(&TIM2_Data, TM_PWM_Channel_1, 1000);

//PWM left Turn

TM_PWM_InitTimer(TIM3, &TIM1_Data, 1000);

TM_PWM_InitChannel(&TIM1_Data, TM_PWM_Channel_2, TM_PWM_PinsPack_1);

TM_PWM_SetChannelMicros(&TIM1_Data, TM_PWM_Channel_2, 1000);
}

//static int buttonCareer(void)
//{
// int8_t val = 0;
// if (TM_GPIO_GetInputPinValue(GPIOD, ButtonA) == 1 || TM_GPIO_GetInputPinValue(GPIOD, ButtonB) == 1)
// val = 1;
// else

```

```

// val = 0;
//
// return val;
//}

int openCloseGripper(uint16_t valVel, uint16_t valRef)
{
float uk, valPos;
valPos = (4096*valRef)/170;
State = 1;
CustomTimer1 = TM_DELAY_TimerCreate(1, 1, 1, CustomTIMER1_Task, NULL);
while (State == 1)
{
uk = PIDControl(valPos, valSensorPos, 0);
if (uk > 0)
{
if (uk > valVel){
uk = valVel;}
TM_PWM_SetChannelMicros(&TIM2_Data, TM_PWM_Channel_1, (1000-uk));
TM_PWM_SetChannelMicros(&TIM1_Data, TM_PWM_Channel_2, 1000);
}
else
{
if (uk < -valVel){
uk = -valVel;}
TM_PWM_SetChannelMicros(&TIM1_Data, TM_PWM_Channel_2, (1000+uk));
TM_PWM_SetChannelMicros(&TIM2_Data, TM_PWM_Channel_1, 1000);
}
}
return 2;
}

/* Para que se mueva a la velocidad deseada se tiene que restar la accion a 1000 es decir (1000 -
void startADC(void)
{
TM_ADC_Init(ADC1, ADC_Channel_0);//Finger Sensor
TM_ADC_Init(ADC1, ADC_Channel_3);//Finger Sensor
//TM_ADC_Init(ADC1, ADC_Channel_7);//Finger Sensor
TM_ADC_Init(ADC1, ADC_Channel_8);//Finger Sensor
TM_ADC_Init(ADC1, ADC_Channel_9);//Finger Sensor
TM_ADC_Init(ADC1, ADC_Channel_10);//Finger Sensor
TM_ADC_Init(ADC1, ADC_Channel_11);//Finger Sensor
TM_ADC_Init(ADC1, ADC_Channel_12);//Finger Sensor

TM_ADC_Init(ADC3, ADC_Channel_4);//Position Sensor
}

void readADC(int8_t val)
{
float result1 = 0;
float result2 = 0;
float result3 = 0;
float result4 = 0;
float result = 0;
switch(val)
{
case 0: //Two internal fingers
result1 = TM_ADC_Read(ADC1, TM_ADC_Channel_9);

```

```

result2 = TM_ADC_Read(ADC1, TM_ADC_Channel_11);
result = (result1+result2)/2;
break;

case 1: //Cuatro Dedos Externos
result1 = TM_ADC_Read(ADC1, TM_ADC_Channel_9);
result2 = TM_ADC_Read(ADC1, TM_ADC_Channel_8);
result3 = TM_ADC_Read(ADC1, TM_ADC_Channel_10);
result4 = TM_ADC_Read(ADC1, TM_ADC_Channel_11);
result = (result1+result2+result3+result4)/4;
break;

case 2: //Dos dedos Internos
result1 = TM_ADC_Read(ADC1, TM_ADC_Channel_12);
result2 = TM_ADC_Read(ADC1, TM_ADC_Channel_7);
result = (result1+result2)/2;
break;

case 3: //Cuatro Dedos Internos
result1 = TM_ADC_Read(ADC1, TM_ADC_Channel_9);
result2 = TM_ADC_Read(ADC1, TM_ADC_Channel_8);
result3 = TM_ADC_Read(ADC1, TM_ADC_Channel_10);
result4 = TM_ADC_Read(ADC1, TM_ADC_Channel_11);
result = (result1+result2+result3+result4)/4;
break;
}
valSensorForce = result;
}

int PIDControl(uint16_t valRef, uint16_t valSensor, int8_t ctrl)
{
int Kp = 3,Ki = 0;
float Kd = 1287.44, T = 0.001;
static float upk,uik,udk,uk;
static float ek,ek_1 = 0,uik_1 = 0;

// if (ctrl == 0)
// Kp = 10,Ki = 2000,Kd = 10;
// else if (ctrl == 1)
// Kp = 20,Ki = 2000,Kd = 10;

ek = valRef - valSensor;

upk = Kp * ek;

uik = uik_1 + Ki * T * ek_1;

udk = Kd/T * (ek - ek_1);

uk = upk + uik + udk;

uik_1=uik;

ek_1=ek;

return uk;
}

int gripperForceControl(uint16_t valRef, int8_t fingers)
{

```

```

timeCont = 0;
CustomTimer1 = TM_DELAY_TimerCreate(1, 1, 1, CustomTIMER1_Task, (void *)fingers);
State = 1;
float uk;
while(State == 1)
{
uk = PIDControl(valRef, valSensorForce, 1);
if (timeCont >= 30000) //Time 5s
{
TM_DELAY_TimerDelete(CustomTimer1);//Stop the timer
State = 0; //Finish bucle
timeCont=0;
// scont = 0; //Restart Ram position to save
}
if (uk > 0)
{
if (uk > 1000){
uk = 1000;}
TM_PWM_SetChannelMicros(&TIM2_Data, TM_PWM_Channel_1, (1000-uk));
TM_PWM_SetChannelMicros(&TIM1_Data, TM_PWM_Channel_2, 1000);
}
else
{
if (uk < -1000){
uk = -1000;}
TM_PWM_SetChannelMicros(&TIM1_Data, TM_PWM_Channel_2, (1000+uk));
TM_PWM_SetChannelMicros(&TIM2_Data, TM_PWM_Channel_1, 1000);
}
// if (valRef <= valSensorForce+10 && valRef >= valSensorForce-10 )
// {
// cont++;
// if (cont >= 10000)
// {
// State = 0;
// cont = 0;
// TM_PWM_SetChannelMicros(&TIM1_Data, TM_PWM_Channel_2, 1000);
// TM_PWM_SetChannelMicros(&TIM2_Data, TM_PWM_Channel_1, 1000);
// }
// }
}
return 2;
}

int gripperPosControl(uint16_t valRef)
{
timeCont = 0;
float uk, valPos;
valPos = (4096*valRef)/170;
State = 1;
CustomTimer1 = TM_DELAY_TimerCreate(1, 1, 1, CustomTIMER1_Task, NULL);
while (State == 1)
{
uk = PIDControl(valPos, valSensorPos, 0);
if (timeCont >= 5000) //Time 5s
{
TM_DELAY_TimerDelete(CustomTimer1);//Stop the timer
State = 0; //Finish bucle
timeCont=0;
// scont = 0; //Restart Ram position to save
}
}

```

```

if (uk > 0)
{
if (uk > 1000){
uk = 1000;}
TM_PWM_SetChannelMicros(&TIM2_Data, TM_PWM_Channel_1, (1000-uk));
TM_PWM_SetChannelMicros(&TIM1_Data, TM_PWM_Channel_2, 1000);
}
else
{
if (uk < -1000){
uk = -1000;}
TM_PWM_SetChannelMicros(&TIM1_Data, TM_PWM_Channel_2, (1000+uk));
TM_PWM_SetChannelMicros(&TIM2_Data, TM_PWM_Channel_1, 1000);
}
//if (valPos <= valSensorPos+2 && valPos >= valSensorPos-2 )
// if(valPos == valSensorPos)
// {
// cont++;
// if (cont >= 30000)
// {
// cont = 0;
// TM_PWM_SetChannelMicros(&TIM1_Data, TM_PWM_Channel_2, 1000);
// TM_PWM_SetChannelMicros(&TIM2_Data, TM_PWM_Channel_1, 1000);
// }
// }
}
return 2;
}

void CustomTIMER1_Task(void* fingers) {

int Lecturas[30], Total = 0, i = 0;
//static uint32_t scont = 0;

// Variables para velocidad
// int vel;
// static uint32_t lastpost = 0;

int8_t val = (int8_t)fingers;
readADC(0);

while (i < 30)
{
Lecturas[i] = 0;
i++;
}

i=0;

while (i < 30) //Potentiometer Filter
{
if (i <= 29)
{
Total = Total - Lecturas[i];
Lecturas[i] = TM_ADC_Read(ADC3, TM_ADC_Channel_4);
Total = Total + Lecturas[i]-100;
i++;
}
}
}

```

```

valSensorPos = Total/30; //Average filter
Total = 0;
i = 0;
// vel = valSensorPos - lastpost;

//TM_SDRAM_Write16(scont, valSensorPos); //save in the ram
//scont += 16; //Add 16 bits for next value
timeCont++;
//lastpost = valSensorPos;
}

```

D.3. acciones.h

```

/**
 * @author Jefferson Camacho
 * @email jefferson-camacho@hotmail.com
 * @version v1.0
 * @ide Keil uVision
 * @license GNU GPL v5
 * @brief Accions Gripper library for STM32F4xx
 *
@verbatim
-----
Copyright (C) Jefferson Camacho, 2016

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program. If not, see <http://www.gnu.org/licenses/>.
-----
@endverbatim
*/

#ifndef acciones_H
#define acciones_H

void startPWM(void);

void startOpenCloseControl(void);

int openCloseGripper(uint16_t valVel, uint16_t valRef);

void closeGripper(void);

int gripperForceControl(uint16_t valRef, int8_t fingers);

void startADC(void);

```

```
void readADC(int8_t val);  
int gripperPosControl(uint16_t valRef);  
#endif
```