



**UNIVERSIDAD PEDAGOGICA
NACIONAL**

Educadora de educadores

FACULTAD DE CIENCIA Y TECNOLOGIA

DISEÑO DE UN BRAZO ROBOTICO CLASIFICADOR DE OBJETOS

TRABAJO DE GRADO

PARA OBTENER EL TITULO DE PREGRADO DE

LICENCIADO EN ELECTRONICA

PRESENTA

ROYER SANABRIA ARIZA

STEPHANNY ORTIZ HERRAN

DIRECTOR

PROFESOR ALBERTO MORALES

BOGOTA, 2015.

NOTA DE ACEPTACIÓN

FIRMA DEL DIRECTOR

JURADO

JURADO

RESUMEN

Este proyecto de grado presenta un trabajo de desarrollo el cual se centra en el diseño mecánico, modelamiento cinemático y control de un brazo robótico de tres grados de libertad clasificador de tuercas, arandelas y tornillos. Para su implementación se hizo uso de técnicas matemáticas y herramientas software como SolidWorks y en Matlab.

En una primera etapa se realizó una breve reseña sobre los brazos robóticos y su aplicación; así como la descripción básica de los elementos técnicos y analíticos que conforman un sistema robótico de este tipo.

Seguidamente, se procedió a desarrollar el modelo cinemático, y la simulación en Matlab para su construcción y diseño mecánico. Junto al diseño mecánico se integra el sistema de iluminación, el efector final o electroimán y la cámara. Por último se desarrolló el controlador Fuzzy y el procesamiento digital de imágenes para las pruebas de funcionamientos.

Finalmente, se expone la implementación completa del brazo robótico junto con sus conclusiones y posibles mejoras.

Palabras clave: brazo robot, diseño mecánico, cinemática, visión artificial.

ABSTRACT

This project is presented to show the development of the work centered in the mechanic design, cinematic modeling and control of a screw classifier robotic arm with 3 grades of freedom. For this purpose, mathematical techniques and software tools like Solid Work and Matlab were used.

As a first level a review of robotic arms and their applications were done, as well as a basic description of the technical and analytic elements that conform this type of robotic system. The next procedure was the development of the cinematic model and the Matlab simulation for the mechanic building and design.

Along with the mechanic design the illumination design, the end Effector or Electromagnet and the camera are integrated. And lastly, a Fuzzy control was developed as well as the digital processing of images for the performance testing.

Finally, the complete implementation of the robotic arm is presented along with conclusions and possible improvements.

Key words: Robot arm, Mechanic design, cinematic and artificial vision.

AGRADECIMIENTOS

A mi familia, por ser mi guía, mi apoyo y su amor incondicional. Gracias a ustedes soy lo que soy.

A mi compañero y amigo de trabajo porque a pesar que hubo momentos difíciles hoy logramos esta meta propuesta.

A mi familia, a mi estrella y a María.

RESUMEN ANALITICO EN EDUCACION-RAE

| 1. Información General | |
|-------------------------------|---|
| Tipo de documento | Trabajo de grado |
| Acceso al documento | Universidad Pedagógica Nacional. Biblioteca Central |
| Título del documento | Diseño de un brazo robótico clasificador de objetos |
| Autor(es) | Ortiz Herrán, Derly Stephanny; Sanabria Ariza, Royer Yesid |
| Director | Morales Alberto |
| Publicación | Bogotá. Universidad Pedagógica Nacional, 2015. 135p. |
| Unidad Patrocinante | Universidad Pedagógica Nacional. |
| Palabras Claves | BRAZO ROBÓTICO, DISEÑO MECÁNICO, CINEMÁTICA, VISIÓN ARTIFICIAL. |

| 2. Descripción |
|--|
| <p>Trabajo de grado presentado para obtener el título de Licenciado en Electrónica, en el cual se propone el diseño de un brazo robótico de tres grados de libertad para clasificar tres clases de objetos como son: tuercas, tornillos y arandelas. Utiliza técnicas de control, de programación y de tratamientos de imágenes para cumplir su objetivo. Sus aplicaciones a mayor escala se observan en la industria; por las temáticas abordadas para su desarrollo es muy útil para la formación académica cursada.</p> |

| 3. Fuentes |
|--|
| <p>PONTIFICIA UNIVERSIDAD JAVERIANA Facultad de ingeniería Electrónica Proyecto de grado para la obtención de del título en ingeniería electrónica TITULO: "CONTROL PARA UN BRAZO ROBOT COLOCADO SOBRE LA PLATAFORMA MÓVIL ÚRSULA". AUTORES: Marcela Aparicio Gonzales y Johana Carolina Orjuela BOGOTA-COLOMBIA.</p> |

INSTITUTO POLITECNICO NACIONAL DE MEXICO

Faculta de Ingeniería

Proyecto de grado para la obtención de del título maestro en manufactura

TÍTULO: “Manufactura de un brazo robótico Scara”

AUTOR: Ingeniero Víctor Hernández Hernández

CIUDAD DE MEXICO-MEXICO

4. Contenidos

El presenta trabajo desarrolla las temáticas de la siguiente manera:

En el capítulo 1 se describen las generalidades de esta clase de brazos robóticos, su morfología, la estructura y ciertas aplicaciones.

En el capítulo 2 se realiza el diseño mecánico y el ensamblaje de la estructura (selección de motores y torques).

En el capítulo 3 se desarrolla la cinemática de un robot (cinemática directa e inversa) y la técnica de control utilizada.

El capítulo 4, integra las técnicas de procesamiento de imagen y el uso de tarjetas electrónicas y actuadores.

Por último, el capítulo 5 incluye las pruebas de funcionamiento, posibles mejoras y los resultados esperados. Posteriormente se resumen las conclusiones a las que se llegaron.

5. Metodología

En una primera etapa se realizó una breve reseña sobre los brazos robóticos y su aplicación; así como la descripción básica de los elementos técnicos y analíticos que conforman un sistema robótico de este tipo.

Seguidamente, se procedió a desarrollar el modelo cinemático, y la simulación en Matlab para su construcción y diseño mecánico. Junto al diseño mecánico se integra el sistema de iluminación, el efector final o electroimán y la cámara. Por último se desarrolló el controlador Fuzzy y el procesamiento digital de imágenes para las pruebas de funcionamientos.

Finalmente, se expone la implementación completa del brazo robótico junto con sus conclusiones y posibles mejoras.

6. Conclusiones

1. La metodología de trabajo permite obtener resultados satisfactorios en el desarrollo de un proyecto. En este caso apporto orden y argumentación en cada avance de los objetivos planteados.
2. Los conceptos de cinemática inversa y directa y los cálculos matemáticos permitieron obtener parámetros importantes como la posición y el torque de cada una de las articulaciones del brazo robótico.
3. Este trabajo apporto a nuestra formación como Docentes en Electrónica habilidades de análisis en la solución de problemas, herramientas técnicas como el uso de Software de diseño y habilidades matemáticas que se unen a los conocimientos adquiridos en las áreas cursadas en la carrera.
4. Los resultados obtenidos en la simulación de la planta con la obtención de cinemática inversa fueron validados con los resultados obtenidos en la planta real. Lo que quiere decir que la obtención de cinemática inversa si resuelve el problema de posicionamiento del electroimán.
5. El controlador Fuzzy careció de precisión en algunas zonas del área de trabajo. Esto se debe a la falta de membresías y variables de entrada. Se puede mejorar el controlador integrando las entradas x , y ; y obteniendo los resultado de los ángulos de los motores requeridos para cada

punto del área de trabajo. Esto trae como consecuencia más costo computacional pero mejora la precisión del controlador.

6. Los momentos de HU son parámetros decisivos del reconocimiento de objetos porque aportan, de forma numérica, información única e invariantes a traslación, rotación o cambio de escala, la forma de los objetos.

| | |
|-----------------------|--|
| Elaborado por: | Derly Stephanny Ortiz Herrán y Royer Yesid Sanabria Ariza. |
| Revisado por: | Alberto Morales |

| | | | |
|--|---|-----|------|
| Fecha de elaboración del Resumen: | 4 | dic | 2015 |
|--|---|-----|------|

Índice general

Preliminares

| | |
|---------------|----|
| Introducción | 18 |
| Justificación | 20 |
| Objetivos | 21 |
| Antecedentes | 22 |

CAPITULO 1. Marco Teórico

| | |
|---|----|
| 1. Generalidades | 24 |
| 1.1. – Robot Industrial | 25 |
| 1.1.1-Clasificacion de robots industrial. | 26 |
| 1.1.2 – Morfología del Robot. | 28 |
| 1.1.2.1 – Estructura mecánica. | 28 |
| 1.1.2.2 – Sensores. | 29 |
| 1.1.2.3 – Actuadores | 32 |
| 1.1.2.3.1-Motores de corriente continua | 33 |
| 1.1.2.3.2-Motores de corriente alterna | 34 |
| 1.1.2.3.3-Motores paso a paso | 34 |
| 1.1.2.3.4-Servomotores | 35 |
| 1.1.3 – Sistema de control | 35 |

CAPITULO 2.-Diseño mecánico

| | |
|----------------------------------|----|
| 2.1 –Selección de articulaciones | 38 |
|----------------------------------|----|

| | |
|---|----|
| 2.2 – Calculo matriz de torque | 39 |
| 2.3 –Selección de motores | 41 |
| 2.3.1-Comunicación | 44 |
| 2.4 –Diseño y construcción del brazo robótico | 47 |
| 2.5-Diseño y construcción de la cabina | 51 |
| 2.5.1–Iluminación | 53 |

CAPITULO 3.- Cinemática y control

| | |
|---|----|
| 3.1 -Método Denavit-Hartenberg | 54 |
| 3.2 -Cinemática Directa | 56 |
| 3.3 – Cinemática Inversa | 59 |
| 3.4 –Simulación en matlab | 61 |
| 3.5-Controlador Fuzzy | 65 |
| 3.5.1- Estructura de un controlador Fuzzy | 65 |
| 3.6- Implementación del controlador | 65 |
| 3.7- Pruebas del funcionamiento del brazo robótico con el controlador Fuzzy | 72 |

72

CAPITULO 4.-Vision artificial

| | |
|----------------------------------|----|
| 4.1-Ajuste de cámara | 75 |
| 4.2- Análisis de piezas | 77 |
| 4.2.1-Area | 77 |
| 4.2.2-Momentos invariantes de HU | 81 |

| | |
|--|------------|
| 4.3-Algoritmo de reconocimiento de objetos | 89 |
| 4.4 Pruebas de tratamiento de imágenes | 90 |
| | |
| CAPITULO 5.-Implementacion final | |
| 5.1-Pruebas de funcionamiento | 93 |
| 5.2-Inconvenientes y recomendaciones | 96 |
| | |
| Conclusiones | 97 |
| Bibliografía | 99 |
| Anexos | 100 |

INDICE DE FIGURAS

| | |
|---|----|
| Figura 1.1.- Tipos de articulaciones | 28 |
| Figura 1.2.-Encoder | 31 |
| Figura 1.3.-Sensor CMOS | 32 |
| Figura 1.4 a.-Planta SISO | 36 |
| Figura 1.4 b.-Planta MIMO | 36 |
| Figura 1.4c.-Planta SIMO | 36 |
| Figura 1.4d.-Planta MIMO | 36 |
| Figura 1.5.-Controlador PID | 37 |
| Figura 2.1a.-Robot tipo RPP | 39 |
| Figura 2.1b.-Robot tipo RRR | 39 |
| Figura 2.2.-Modelo de robot RRR | 41 |
| Figura 2.3 a.-Conexión motores Dynamixel | 44 |
| Figura 2.3b-Configuración de comunicación | 44 |
| Figura 2.4.-Orden de comunicación | 45 |
| Figura 2.5.-Comunicación Matlab y Dynamixel | 46 |
| Figura 2.6.-Comunicación completa | 47 |
| Figura 2.7.- Modelado del brazo robótico | 47 |
| Figura 2.8.-Implementación del electroimán | 48 |
| Figura 2.9.-Circuito electroimán | 49 |
| Figura 2.10.-Ensamble brazo robótico | 50 |
| Figura 2.11.-Area de trabajo | 51 |

| | |
|--|----|
| Figura 2.12.-Modelado de cabina | 52 |
| Figura 2.13.-Cabina y brazo robótico | 52 |
| Figura 3.1.-Sistema de coordenadas | 57 |
| Figura 3.2.-Cinematica inversa | 62 |
| Figura 3.3.-Articulacion 1 | 62 |
| Figura 3.4.- Articulación 2 | 63 |
| Figura 3.5.-Articulacion 3 | 63 |
| Figura 3.6.-Robot en Simulink | 63 |
| Figura 3.7.-Bloque cinemática inversa | 64 |
| Figura 3.8.-Fuzzy control | 65 |
| Figura 3.9 a.-Teorema y razón trigonométrica | 66 |
| Figura 3.9b.-Esquema del controlador | 66 |
| Figura 3.10 a.-Estado inicial simulado | 67 |
| Figura 3.10b.-Estado inicial real | 67 |
| Figura 3.11.-Fuzzy-control | 69 |
| Figura 3.12.-Membresia H | 70 |
| Figura 3.13.-Membresia Tetha 2 | 70 |
| Figura 3.14.-Membresia Tetha 3 | 71 |
| Figura 3.15.-Comportamiento del controlador | 71 |
| Figura 3.16.-Aplicación de ecuación | 72 |
| Figura 3.17.-Mapa | 73 |
| Figura 3.18.-Mapa con zonas de color | 73 |
| Figura 4.1.-Obtencion de información | 75 |

| | |
|---|----|
| Figura 4.2.-Ajuste de cámara. | 76 |
| Figura 4.3 a.-Imagen binaria | 77 |
| Figura 4.3b.-Imagen por Matlab | 77 |
| Figura 4.4 a.-Piezas reales | 78 |
| Figura 4.4b.-Piezas formato binario | 78 |
| Figura 4.5.-Tornillo en formato binario | 79 |
| Figura 4.6.-Algoritmo de reconocimiento de imágenes | 83 |
| Figura 4.7.-Imagen de tratamiento de imagen | 90 |
| Figura 4.8 - Prueba de tratamiento de imagen | 91 |
| Figura 5.1.-Esquema implementación final | 92 |
| Figura 5.2.- Esquema del algoritmo completo | 93 |
| Figura 5.3.-Piezas en el área de trabajo | 95 |

INDICE DE TABLAS

| | |
|---|----|
| Tabla 1.1.-Clasificacion de robots | 26 |
| Tabla 1.2.-Generaciones de robots | 27 |
| Tabla 1.3.-Clases de sensores | 30 |
| Tabla 2.1.-Caracteristicas de motores | 42 |
| Tabla 2.2.-Datos de los elementos del brazo | 49 |
| Tabla 2.3.-Datos piezas | 49 |
| Tabla 3.1.-Calculo de articulaciones | 57 |
| Tabla 3.2.-Angulos de cada motor | 68 |
| Tabla 3.3.-Cinematica inversa | 68 |
| Tabla 3.4.-Comparacion | 69 |
| Tabla 3.5.-Membresia H (in-out) | 71 |
| Tabla 4.1.-Area y posición | 79 |
| Tabla 4.2.-Area arandela | 79 |
| Tabla 4.3.-Area tornillo | 80 |
| Tabla 4.4.-Area tuerca | 80 |
| Tabla 4.5.-Momento HU | 83 |
| Tabla 4.6.-HU1 arandela | 83 |
| Tabla 4.7.-HU2 arandela | 84 |
| Tabla 4.8.- HU3 arandela | 84 |
| Tabla 4.9.- HU4 arandela | 84 |

| | |
|--|----|
| Tabla 4.10.- HU5 arandela | 84 |
| Tabla 4.11.- HU6 arandela | 84 |
| Tabla 4.12.- HU7 arandela | 85 |
| Tabla 4.13.-HU1 tornillo | 85 |
| Tabla 4.14.- HU2 tornillo | 85 |
| Tabla 4.15.- HU3 tornillo | 86 |
| Tabla 4.16.- HU4 tornillo | 86 |
| Tabla 4.17.- HU5 tornillo | 86 |
| Tabla 4.18.- HU6 tornillo | 86 |
| Tabla 4.19.- HU7 tornillo | 86 |
| Tabla 4.20.-HU1 tuerca | 87 |
| Tabla 4.21.- HU2 tuerca | 87 |
| Tabla 4.22.- HU3 tuerca | 87 |
| Tabla 4.23.- HU4 tuerca | 87 |
| Tabla 4.24.- HU5 tuerca | 88 |
| Tabla 4.25.- HU6 tuerca | 88 |
| Tabla 4.26.- HU7 tuerca | 88 |
| Tabla 4.27.-Análisis de momentos de HU | 89 |
| Tabla 4.28.-Estadística de tratamiento de imagen | 91 |
| Tabla 5.1.-Pruebas finales y estadísticas de error | 94 |

INTRODUCCION

La incorporación de los brazos robóticos en la industria ha permitido la reducción de gastos, la mejora de los productos y la realización de tareas de manera más eficaz y rápida.

Un brazo robótico es un manipulador articulado con múltiples funciones, que ha sido diseñado para mover elementos, materiales o herramientas; gracias a que sus movimientos han sido programados mediante técnicas de control y procesamiento de la información del entorno.

Nuestra formación profesional no puede dejar a un lado el auge que presenta hoy día la robótica donde participan disciplinas como la teoría de control, álgebra, informática, mecánica y electrónica. Por tanto, este trabajo beneficia nuestra formación como futuros Licenciados en Electrónica aportando de manera significativa conocimientos propios de la carrera y permitiendo profundizar en otros. Estas características lo convierten en una alternativa de práctica para demás compañeros en formación en el área de control.

El trabajo está enfocado en el diseño de un brazo robótico de tres grados de libertad que clasifica tuercas, tornillos y arandelas empleando técnicas de diseño, modelamiento cinemático, control y visión artificial. Como metodología se usó primeramente el análisis y cálculo de variables; luego el diseño mecánico y la simulación de sólidos para su posterior implementación. Por

último la experimentación consistió en pruebas de funcionamiento donde el brazo robótico detecto los objetos y los organizo según pertenecía.

El texto fue organizado en orden y en relación a los objetivos propuestos que se describen en la secuencia de desarrollo de cada uno de los capítulos de esta manera: En el capítulo 1 se describen las generalidades de esta clase de brazos robóticos, su morfología, la estructura y ciertas aplicaciones. En el capítulo 2 se realiza el diseño mecánico y el ensamblaje de la estructura (selección de motores y torques). En el capítulo 3 se desarrolla la cinemática de un robot (cinemática directa e inversa) y la técnica de control utilizada. El capítulo 4, integra las técnicas de procesamiento de imagen y el uso de tarjetas electrónicas y actuadores. Por último, el capítulo 5 incluye las pruebas de funcionamiento, posibles mejoras y los resultados esperados. Posteriormente se resumen las conclusiones a las que se llegaron.

JUSTIFICACION

Un brazo robótico es utilizado principalmente en la industria para realizar tareas durante más tiempo de manera eficiente y rápida. Sus beneficios no solo inciden en el campo económico (pues son usados en la industria) si no en el campo académico como fue nuestro caso. La construcción de este brazo robótico nos aportó conocimientos matemáticos, habilidades técnicas y analíticas; algunas de estas vistas en el transcurso de la carrera, otras fueron nuevas.

Los elementos mecánicos, de simulación, de procesamiento y de cálculo son pertinentes al área de estudio de la robótica y al campo laboral en que nos desempeñamos como futuros Licenciados en electrónica. Por ejemplo, para el año en curso en su página Web: www.tic.bogota.gov.co/robótica La Alta Consejería Distrital de TIC convocó a docentes, semilleros, grupos de investigación entre otros a participar en la feria anual de Robótica para proponer y exponer proyectos tecnológicos e innovadores. Entes como el anterior impulsan estas ideas y proporcionan a colegios y universidades elementos para el desarrollo de este campo. Este movimiento actual convierte este trabajo de grado en una oportunidad para profundizar y a la vez extender los conocimientos adquiridos para un futuro desempeño laboral.

Eventos como este inciden en universidades con carreras afines, como la nuestra. Por eso proponemos nuestro brazo robótico como herramienta de uso para nuestros compañeros en formación, para sus prácticas y para sus posibles trabajos de grado.

OBJETIVOS

Objetivo general

- Desarrollar un brazo robótico de 3 grados de libertad para clasificar y organizar 3 clases de objetos.

Objetivos específicos.

- Obtener el modelo cinemático del sistema para simularlo en matlab.
- Diseñar el sistema mecánico en SolidWorks para su implementación.
- Generar el algoritmo de adquisición y procesamiento digital de imágenes.
- Implementar el controlador Fuzzy.

ANTECEDENTES

“Control para un brazo robot colocado para la plataforma móvil URSULA”,

Marcela Aparicio Gonzales y Johanna Carolina Orjuela, Pontificia Universidad Javeriana, Colombia, 2005.

Este trabajo de grado describe todo el proceso de diseño de un prototipo que ayude a la labor de detección de minas por medio de un sistema de control que manipula un brazo robótico de tres grados de libertad y con instrucciones que proporcionan un mayor alcance en la búsqueda de minas antipersonas. Sus autores mencionan que la validación de los métodos de control y movimiento se realizó con una plataforma de demostración.

“Manufactura de un brazo robótico SCARA”, Ingeniero Víctor Hernández

Hernández, Instituto Politécnico Nacional, México D.F, 2011.

Esta tesis presenta una investigación en el diseño mecánico, modelado y control de un brazo robótico tipo SCARA de tres grados de libertad para ser utilizado en soldadura TIG sobre componentes metálicos y considerar su proceso de fabricación.

“Diseño de un brazo robótico para aplicaciones fitosanitarias”, Daniel Aristizabal Torres, Universidad Politécnica de Valencia, Valencia, 2009.

Esta tesis presenta un trabajo de investigación en el diseño de un brazo robótico para la portabilidad de un sistema de desinsectación por microondas. Tiene como objetivo dar seguridad al operario durante el tratamiento de piezas y accesibilidad a las esquinas, alturas y puntos ocultos de estructuras arquitectónicas.

CAPITULO 1.-MARCO TEORICO

1. Generalidades

Comprender la complejidad de un brazo robótico, su concepto y algunas de sus aplicaciones, conlleva a recordar una serie de acontecimientos de la historia humana llevados gradualmente con mucho ingenio y fascinación por las máquinas y los dispositivos.

Por ejemplo, en la década de los 40s el termino robot como maquina se desprende del concepto de robot apareciendo el de tele manipulador aproximándose así a lo que es un brazo robótico. En 1948 el ingeniero R.C. Goertz del Argonne National Laboratory desarrollo un tele manipulador capaz de manipular elementos radiactivos protegiendo así al operador y luego incorporándole servo controladores. Para esta misma época Ralph Mosher ingeniero de General Electric desarrollo 2 brazos mecánicos llamado Handy-Man.

Estos inicios hicieron que la industria de carros, marina, aeroespacial y militar se interesaran e invirtieran en su desarrollo. Así que en 1954 el ingeniero George C. Devol estableció los parámetros para un robot industrial y concibe una idea de definición de esta clase de robot como un dispositivo de transferencia de artículos programado. Este junto al director de ingeniería de la división aeroespacial de la empresa Manning Maxwell fundan la Consolidated Controls Corporation e instalan su primera máquina Ultimate (1960) en la empresa General Motors aplicándola en la fundición por inyección. En 1982 el profesor de la Universidad Yamanashi de

Japón desarrolla el concepto de robot SCARA, dispositivo de bajo costo aplicado para el ensamblaje de piezas.

Para esta época el concepto de robot se basaba en películas de ciencia ficción y estos dispositivos distaban mucho de lo que se conocía en el momento; pero aun así en poco se firmaron muchos acuerdos y diversas fábricas emprendieron la competencia para desarrollarlos y así mejorar su productividad. Este auge apunta a que el campo de la robótica busque que los dispositivos, incluidos los brazos robóticos, mejoren su movilidad y autonomía. A este punto la mayoría de conceptos que rodean a lo que es un robot se aplican directamente al robot industrial.

1.1 Robot Industrial

Un robot industrial se puede definir, según la Robot Institute of America (RIA), como: “un manipulador multifuncional reprogramable diseñado para mover materiales, partes, herramientas o dispositivos especializados a través de movimientos programados para la ejecución de una variedad de tareas”. (SAHA, 2010)

Esta definición tiene una modificación por parte de la Federación Internacional de Robótica (FIR) donde distingue un robot manipulador de otros robots:

“Por robot industrial se entiende a una máquina de manipulación automática, reprogramable y multifuncional con tres o más ejes que puede posicionar y orientar materias, piezas, herramientas o dispositivos especiales para ejecución de trabajos diversos en las diferentes etapas de la producción industrial, ya sea en una posición fija o en movimiento”. (SAHA, 2010)

El brazo robótico por aplicación, mecanismos, diseño y complejidad clasifica dentro de la generación de los robots industriales.

1.1.1 Clasificación de robots industriales

Según la Federación Internacional de Robótica (FIR) clasifica así estos dispositivos (BARRIENTOS, Peñin, BALAGUER, & Aracil):

Tabla 1.1 Clases de robots

| | |
|--------|---|
| Tipo A | Manipulador con control manual o telemando |
| Tipo B | Manipulador automático con ciclos reajustables; regulación mediante fines de carrera o topes; control por PLC; accionamiento neumático, hidráulico o eléctrico. |
| Tipo C | Robot programable con trayectoria continua o punto a punto. Carece de conocimientos sobre su entorno. |

Tipo D Robot capaz de adquirir datos de su entorno, readaptando su tarea en función de estos.

Fundamentos de Robótica (1996).

O, según su reprogramación y funcionalidad sin modificaciones físicas; este mismo organismo los clasifica así (BARRIENTOS, Peñin, BALAGUER, & Aracil):

Tabla 1. 2 Generaciones de robots

| | |
|----------------|---|
| 1ª. Generación | <p>Repite la tarea programada secuencialmente.</p> <p>No toma en cuenta las posibles alteraciones de su Entorno.</p> |
| 2ª. Generación | <p>Adquiere información limitada de su entorno y actúa en consecuencia. Puede localizar, clasificar y detectar esfuerzos y adaptar sus movimientos en consecuencia.</p> |
| 3ª Generación | <p>Su programación se realiza mediante el empleo de un lenguaje natural. Posee capacidad para la planificación automática de tareas.</p> |

Fundamentos de Robótica (1996).

Tanto las diversas definiciones como su clasificación colocan a los brazos robóticos como un robot industrial capaz de manipular elementos y que posee un controlador complejo.

1.1.2 Morfología del robot

Los robots, sin importar su clasificación, están formados por los siguientes elementos: estructura mecánica, sistema sensorial, actuadores, sistema de control, y elementos terminales (BARRIENTOS, Peñin, BALAGUER, & Aracil) . A continuación vamos a ver de forma general las características principales de los elementos nombrados.

1.1.2.1 Estructura mecánica:

Los robots están constituidos por una serie de piezas solidas denominadas eslabones conectados entre sí que permiten un movimiento definido en cada articulación. En 1876 Franz Reuleaux identifico los posibles movimientos entre dos eslabones y clasificó los tipos de articulaciones que pueden llevar los robots (CASTILLO & CABERTA):

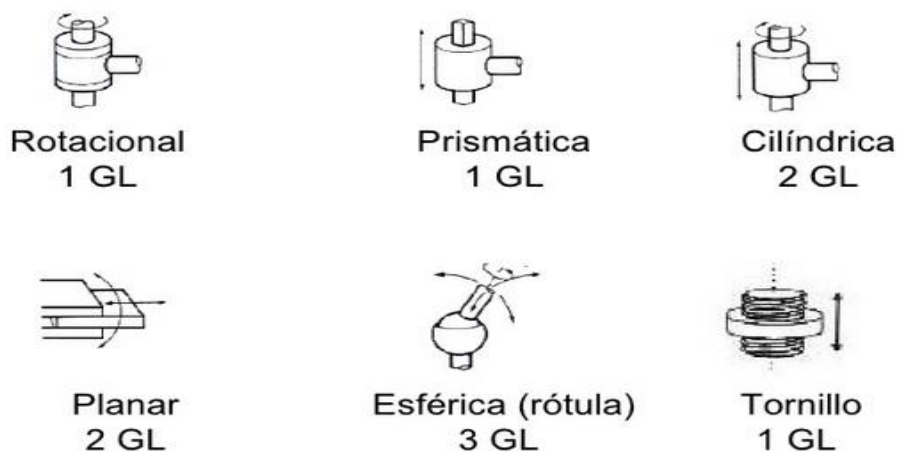


Figura 1.1 Clases de articulaciones. Caracterización de un robot (1996).

Se denomina grado de libertad, Degrees of Freedom (DOF), a cada uno de los movimientos independientes que puede realizar una articulación (BARRIENTOS, Peñin, BALAGUER, & Aracil). En la figura 1.1 observamos los 6 tipos de articulaciones que pueden llevar los robots como también los grados de libertad que proporciona cada articulación. El número de grados de libertad total del robot viene dado por la suma de los grados de libertad de cada articulación.

1.1.2.2 Sensores

Una de las características esenciales que todo robot debe tener es la precisión de sus movimientos y la velocidad adecuada. Para ello es necesario que el robot cuente con sensores que retroalimenten el estado en el que se encuentra y la información necesaria del entorno. Las variables a tener en cuenta para la retroalimentación llegan a ser internas y externas, y dependen de la tarea que ejecute el robot. Las variables internas a medir están sujetas a los mecanismos que componen el robot, algunos ejemplos de variables internas son: la posición y velocidad de cada actuador (motor), la posición de cada eslabón, temperaturas internas de los mecanismos y posición final del actuador, entre otros. Los motores seleccionados cuentan con sensores internos que miden posición y temperatura. Por otro lado, las variables externas vienen dadas por el entorno de trabajo tales como: obstáculos, iluminación del área de trabajo y detección de objetos, entre otros.

En la tabla 1.3 se da información de los sensores internos más utilizados para adquirir información de la posición del robot, la velocidad y presencia de objetos (BARRIENTOS, Peñin, BALAGUER, & Aracil):

Tabla 1.3 Clases de sensores.

| | | |
|-----------|-----------------|------------------------|
| Presencia | Inductivo | |
| | Capacitivo | |
| | Efecto Hall | |
| | Célula Reed | |
| | Ultrasonido | |
| | Óptico | |
| | Contacto | |
| Posición | Analógicos | |
| | | Potenciómetros |
| | | Resolver |
| | | Sincro |
| | | Inductosyn |
| | | LVDT |
| | Digitales | |
| | | Encoders absolutos |
| | | Encoders incrementales |
| | | Regla óptica |
| | | |
| | | |
| Velocidad | Taco generatriz | |

Tabla 1.3. Fundamentos de Robótica (1996).

Actualmente en el mercado existen **sensores de posición** digitales y analógicos. El brazo robótico desarrollado trabaja con un sensor de posición digital o codificador angular de posición (Encoders), ver figura 1.2 , que es un sensor interno que determina la posición de un motor a través de un disco codificado unido al eje de rotación, un foto receptor y un led emisor. Cada vez que gira las ranuras del disco generan un tren de pulsos que junto a una franja de marcas permite indicar el sentido de giro.

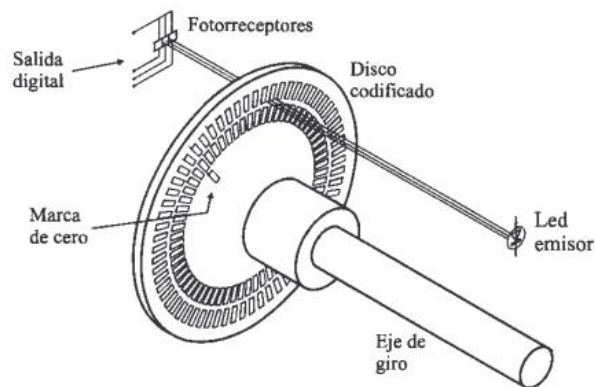


Figura 1.2 Encoders. *Fundamentos de Robótica (1996)*.

Los **sensores de presencia** detectan objetos a un radio de distancia determinado con o sin contacto con éste. En robótica los detectores de presencia informan el estado de entorno a los sensores de posición para que el robot tome una decisión o se utilizan también como sensores externos.

Un ejemplo de **sensor externo** es el Complementary metal-oxide semiconductor (CMOS) mostrado en la figura 1.3. Este sensor captura y procesa la cantidad de luz que hay en su rango de

alcance en el espacio. Muchas de las cámaras que hoy en día están en el mercado trae incorporado un sensor CMOS.

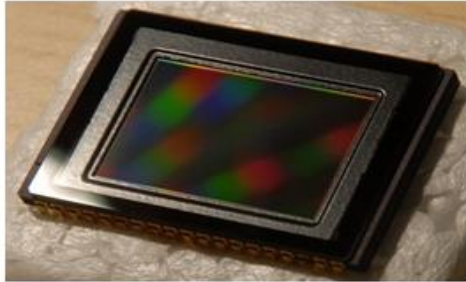


Figura 1.3. Sensor CMOS. Sensores CMOS. Recuperado de:
<https://www.google.com.co/search?q=sensor+CMOS&espv>

1.1.2.3 Actuadores

Los actuadores son los elementos del robot encargados de generar movimientos. Según el objetivo y funcionalidad del robot a construir se debe considerar la potencia, la fuerza, volumen, peso y costo del actuador. Según la energía que utilicen se pueden clasificar en actuadores eléctricos, neumáticos e hidráulicos.

Los **actuadores neumáticos** se pueden en:

- Cilindros neumáticos: por la presión de aire mueve unos émbolos internos, solo consiguen un desplazamiento más no un movimiento continuo.
- Motores neumáticos (de aletas rotativas o de pistones axiales): utilizan eje de rotación y aletas para el movimiento continuo por presión.

Los **actuadores hidráulicos** utilizan aceites minerales que gracias a su grado de compresibilidad permiten sujetarlos a un control continuo y a elevarla su fuerza.

Existen cilindros hidráulicos y motores hidráulicos con el mismo principio de funcionamiento de los actuadores neumáticos.

Esta sección enfatizara en los actuadores servomotores, debido a que fueron los seleccionados para el desarrollo del brazo robótico. Dentro de los actuadores eléctricos pueden distinguirse tres tipos diferentes (M., 1989):

- Motores de corriente continua (DC)
- Motores de corriente alterna (AC)
- Motores paso a paso y servomotores.

1.1.2.3.1 Motores de corriente continúa

Dado que los actuadores transforman la energía eléctrica (en este caso corriente continua c.c) en energía mecánica, es necesario revisarlos mecánicamente. Estos se componen principalmente de:

- Rotor: Es una pieza móvil giratoria que contiene partes laterales cubiertas de embobinado.

- Estator: Pieza fija con bobinados eléctricos que al circular corriente eléctrica genera campo magnético y a su vez el movimiento del rotor.

1.1.2.3.2 Motores de corriente alterna

Esta clase de motores se caracterizan por la interacción ejercida entre el rotor y estator con sus bobinados junto a los campos electromagnéticos generados por la corriente alterna. A diferencia del anterior donde la corriente circula por un conductor en una sola dirección y su magnitud es la misma; la corriente eléctrica presenta valores y direcciones de corriente alternadas.

1.1.2.3.3 Motores paso a paso

Las características de un motor **paso a paso** son ideales para mecanismos que necesiten movimientos precisos; pues el rotor, estator, el embobinado se excitan conmutándose y dando pasos exactos. Este motor necesita un controlador externo generalmente un micro controlador para cada controlar sus movimientos.

1.1.2.3.4 Servomotores

Los servomotores contienen una tarjeta electrónica interna que registran el ángulo de giro del motor y un sistema de control interno con sensor de posición que junto con la etapa de potencia y realimentación de la posición se supervisan sus movimientos. Sus ventajas lo hacen ideal para el campo

de la robótica y en este trabajo se usó motores de esta clase que se analizaran más adelante.

1.1.3 Sistema de control

Antes de hablar de sistemas de control debemos tener en cuenta las siguientes definiciones (OGATA):

- **Variable controlada y variable manipulada:** La variable controlada es la cantidad o condición que se mide o se controla. La variable manipulada es la cantidad o condición que el controlador modifica para efectuar el valor de la variable controlada. Por lo común, la variable controlada es la salida (resultado) del sistema. Controlar significa medir el valor de la variable controlada del sistema y aplicar la variable manipulada al sistema para corregir o limitar una desviación del valor medio a partir de un valor deseado.
- **Plantas:** la definición más próxima a planta es ser una parte de un equipo, tal vez un conjunto de las partes de una máquina que funcionan juntas, el propósito de la cual es ejecutar una operación particular.

- **Sistemas:** Un sistema es una combinación de componentes que actúan juntos y realizan un objetivo determinado. Un sistema no necesariamente es físico. El concepto de sistema se aplica a fenómenos abstractos y dinámicos
- **Perturbaciones:** Una perturbación es una señal que tiende a afectar negativamente el valor de la salida de un sistema. Si la perturbación se genera dentro del sistema se denomina interna, en tanto que una perturbación externa se produce fuera del sistema y es una entrada.

Los sistemas de control se pueden definir como un conjunto de componentes o técnicas capaces de controlar las variables controladas o salidas del robot a través de un proceso realizado a las variables manipuladas o entradas del mismo.

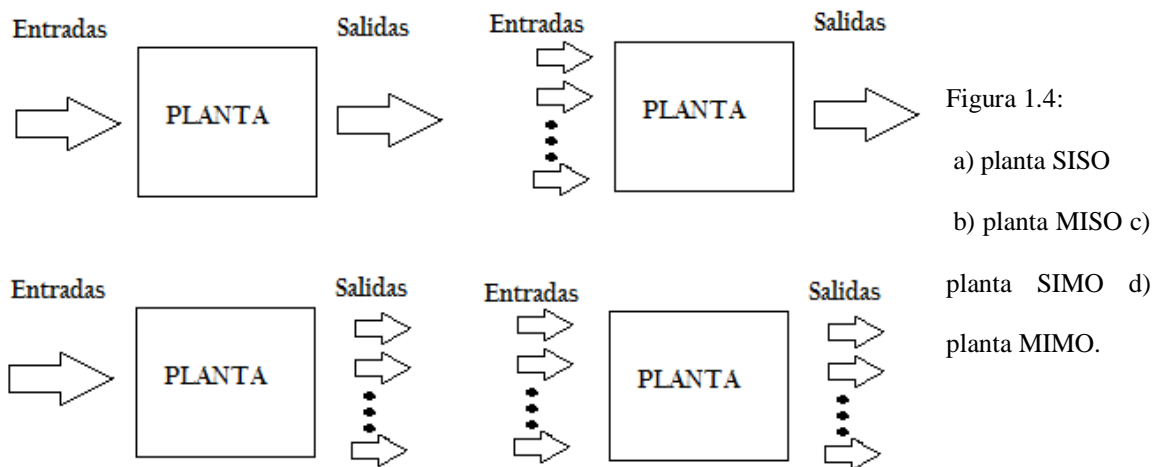


Figura 1.4:

a) planta SISO

b) planta MISO c)

planta SIMO d)

planta MIMO.

Las plantas a controlar tienen cuatro configuraciones en términos de las entradas y salidas. En la figura 1.4a y 1.4c observamos que las plantas son de tipo SISO y SIMO, por sus siglas en inglés Single Input Single Output y Single Input Multiple Output respectivamente.

Aunque todas las plantas tienen perturbaciones y genera una entrada extra, lo que quiere decir que no existe una planta con una sola entrada, algunas de estas perturbaciones son despreciables y se pueden analizar como si tuvieran una sola entrada. En la figura 1.4b y 1.4d observamos las plantas MISO y MIMO, por sus siglas en inglés Multiple Input Single Output y Multiple input Multiple Output. Estas plantas tienen en cuenta más de una entrada y son las que usualmente se trabajan en la industria.

Existen diferentes tipos de controladores que se le pueden aplicar a una planta. Entre las más utilizadas se encuentra:

- **Controlador PID:** Técnica de control para una planta por realimentación que contiene 3 parámetros en su algoritmo y son el proporcional (P), el derivativo (D) y el integral (I).

Ejemplo:

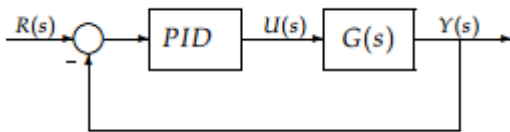


Fig.1.5. Controlador PID. Ogata Katsuhido (2006)

- **Controlador Fuzzy:** Esta clase de controlador es una técnica de razonamiento que trabaja con variables de entrada y reglas limitadas que aproximan a describir el comportamiento de un sistema. Esta tesis utiliza este controlador, así que más adelante se ampliará su información.

CAPITULO 2.-DISEÑO MECÁNICO

El diseño mecánico de la estructura del brazo robótico debe estar sujeto a las tareas que se le asignará. Para cumplir con esto, existen técnicas de desarrollo que proveen orden al trabajo. Para este caso siguiendo la metodología Down-top se definieron las siguientes etapas:

1. Selección de articulaciones
2. Calculo matriz de torques
3. Selección de motores
4. Diseño y construcción del brazo robótico
5. Diseño y construcción de cabina

2.1 Selección de articulaciones.

Uno de los principales requisitos que debe cumplir cualquier brazo robótico es llegar a cualquier punto dentro del espacio de trabajo definido. Por ello se escogen las articulaciones adecuadas que garantice el óptimo desempeño del sistema.

La clasificación de los robots poli-articulados viene dado por el tipo de articulaciones que posee el robot. De esta información se definen los grados de libertad y la manera de desplazarse en el espacio. En la figura 2.1.a se observa un robot de tipo RPP (rotacional, prismático, prismático) y en la figura 2.1b un brazo robótico tipo RRR con tres grados de libertad que se pueden ubicar perfectamente en el espacio xyz . Estas son soluciones para el diseño del robot clasificador. En este trabajo se seleccionó el 2.1b.

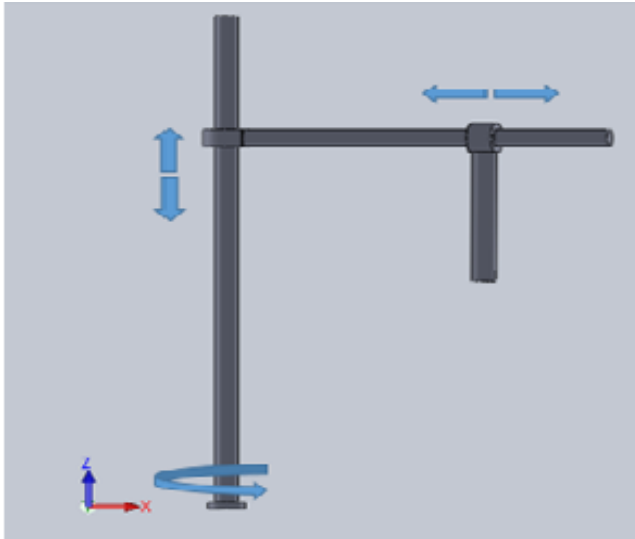


Fig. 2.1: a) Robot tipo RPP

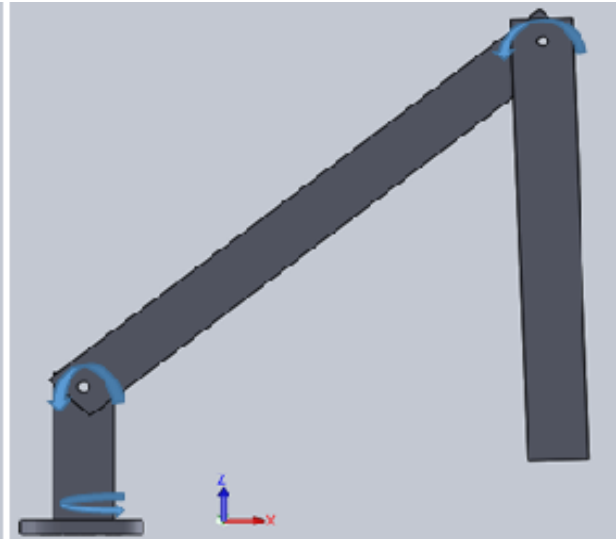


Fig. 2.1b) Robot tipo RRR

2.2 Calculo matriz de torque:

El torque es la fuerza que aplican los motores para mover los eslabones de cada articulación del brazo robótico. Por eso es necesario hacer el cálculo de torques que requiere cada articulación antes de seleccionar un motor en específico. Para realizar este cálculo, primero es necesario recordar:

- a) El torque es la fuerza rotacional con la que gira un objeto sobre un eje central. Y está definida por:

$$T = f * l = w * l$$

$T = \text{Torque}$

$f = \text{fuerza actuante en el objeto}$

$w = \text{peso del objeto}$

$l = \text{longitud entre el punto de rotación y la fuerza}$

- b) El torque necesario de los actuadores depende del ángulo de inclinación entre la carga y el punto de rotación. El ángulo en el que se necesita mayor torque es de 45 grados.
- c) Se asume que la masa de los eslabones está repartida uniformemente. Esto indica que el centro de gravedad está en el centro del eslabon. El torque que se necesita para levantar un eslabón sin fuerzas externas (sin peso extra) está definida por :

$$T = \frac{1}{2} l * w$$

En la figura 2.2 se observa los eslabones y articulaciones del brazo robótico. Donde l_2 es la distancia del eslabón 2 con un peso de w_2 , l_3 es la distancia del eslabón 3 con un peso de w_3 , w_e es el peso del electroiman, w_m es el peso de un motor y w_p es el peso de las piezas.

Por tanto, la matriz de torque está definida por:

$$\text{Torque}_{(\text{motor2}, \text{Motor3})} = \begin{pmatrix} \frac{1}{2}(l_3 w_3) + (l_3 w_e + l_3 w_p) \\ (w_e + w_p)(l_2 + l_3) + \frac{w_2}{2}(2l_2 + l_3) + l_2 w_m + \frac{l_2 w_2}{2} \end{pmatrix}$$

Debido a que el motor 1 (motor de la base) tiene el peso sobre su eje de rotación el torque necesario es menor que el torque del motor 2.

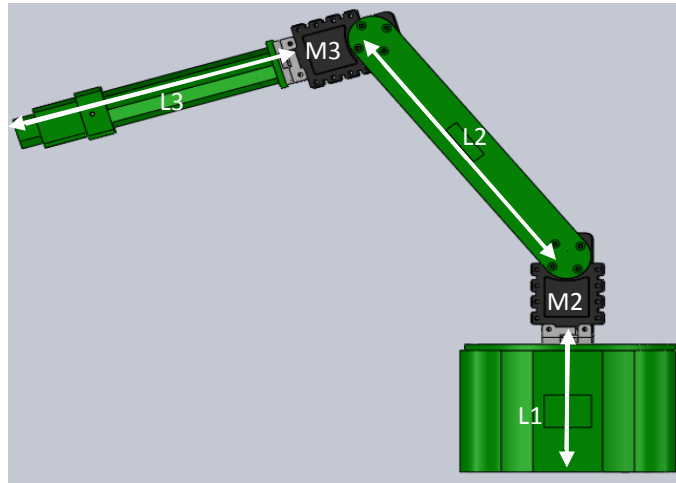


Figura 2.2 Modelo de robot RRR

2.3 Selección de motores.

Actualmente en el mercado hay una gran variedad de motores con características variadas que cumplen con el objetivo del brazo robótico. Entre estos están:

Tabla 2.1

| Nombre | Características generales | Comunicación | Precio | Imagen |
|-------------------------------|---|---------------------------------------|-----------|---|
| Herkulex DRS-0101 | Torque máximo 12 kg-cm Peso 45g Voltaje de operación 7v-12v Angulo de giro 0-180 | FULL-DUPLEX | \$ 125000 |  |
| Servo Power Hd1214th-titanium | Torque máximo 16 kg-cm Peso 57g Voltaje de operación 6v-9v Angulo de giro 0-270 | PWM | \$ 130000 |  |
| Dynamixel ax-12 ^a | Torque máximo 15.3 kg-cm Peso 55g Voltaje de operación 9v-12v Angulo de giro 0-360 | Half Duplex Asynchronous Serial | \$110000 |  |


| | | | | |
|-------------------|--|-----|----------|---|
| Futaba FUTM303 | Torque máximo 3.7 kg-cm Peso 23g Voltaje de operación 4.8v -6 v Angulo de giro 0-180 | PWM | \$260000 |  |
|-------------------|--|-----|----------|---|

Tabla 2.1. Características de motores.

Características como la comunicación, el ángulo de giro, el peso, el diseño y precio determinaron la compra del motor Dynamixel AX-12. Además este motor cuenta con otras características como:

1. Los motores se pueden conectar en serie gracias a su topología Daisy-Chain. Esta característica tiene como ventajas:
 - a) son controlados por protocolos de comunicación digital de paquetes en la red serie.
 - b) se le puede asignar un ID a cada motor y llamarlo cuando sea conveniente.
 - c) reduce el número de cables a utilizar.
2. Se puede configurar simultáneamente los siguientes parámetros: velocidad, torque máximo, corriente máxima y ángulo máximo.
3. Se puede leer directamente del motor los siguientes parámetros: voltaje, temperatura, posición y velocidad.

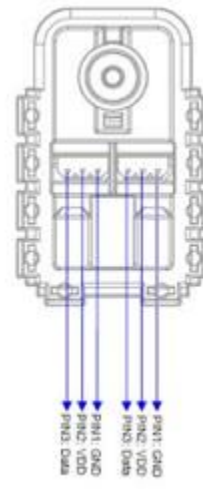
4. Tiene un sistema de alarma que se activa cuando el motor está trabajando fuera de los límites normales.

2.3.1 Comunicación

Los motores Dynamixel AX-12A tienen una topología Daisy-Chain que facilita la comunicación entre dos o más motores Dynamixel y el computador. En la figura 2.3a se muestra dos motores Dynamixel conectados entre sí en serie. Esta característica minimiza el número de cables que debemos utilizar al conectar más de dos motores. En la figura 2.3b se muestra la forma de conectar los motores Dynamixel ax-12a



(a)



(b)

Figura 2..3. a) Conexión motores Dynamixel b) Configuración de comunicación. User's Manual. Robotis (2006)

En la comunicación en serie cada motor tiene un ID, modificable, con el que el usuario puede comunicarse con el motor. Estos motores responden al protocolo Half Duplex Asynchronous Serial que es básicamente la comunicación entre dos o más motores por una sola línea de transmisión de forma asíncrona.

Para comenzar la comunicación se necesita enviar por la línea de comunicación dos Start (0xFF 0xFF) seguido del ID del motor y las instrucciones. En la figura 2.4 se puede ver gráficamente la forma de enviar los datos. Notemos que hay un Checksum que es finaliza la comunicación. Este último parámetro es el que garantiza que la información llegó de forma adecuada y finaliza la comunicación. La fórmula para calcular el Checksum es:

$$\text{Check Sum} = \sim (\text{ID} + \text{Length} + \text{Instruction} + \text{Parameter1} + \dots + \text{Parameter N})$$

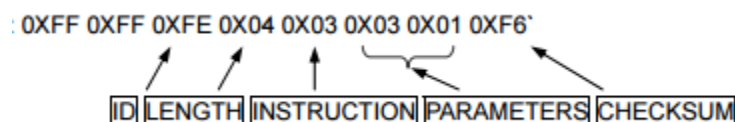


Figura 2.4 Orden de la comunicación *User's Manual. Robotis* (2006), Recuperado de: <http://support.robotis.com/en/>

En la tabla del anexo 3 se muestra las instrucciones que el motor Dynamixel reconoce, la dirección de cada instrucción y el Access de cada instrucción. Así mismo se encuentra las diferentes formas de configurar el Baud Rater para establecer la comunicación. Hay que tener en cuenta que tanto el computador como los motores deben estar enviando o recibiendo datos a la misma velocidad.

En el siguiente link <http://savageelectronics.blogspot.com.co/search/?q=dynamixel> se puede descargar una librería, y su manual de uso, que controla los motores desde el programa Arduino.

Las instrucciones que utilizamos de esta librería para este proyecto son:

- a) Serial.begin (Baud Rate) quien configura la velocidad de transmisión de los motores,
- b) Dynamixel.torqueStatus (ID, State), que se encarga de activar el torque de los motores
- c) Dynamixel.moveSpeed (ID, Angle, Speed), que se encarga de mover el motor a un ángulo determinado con una velocidad determinada

En la figura 2.5 están los pasos que se deben de realizar para entablar la comunicación entre el programa Matlab y los motores Dynamixel usando como puente a Arduino. El código del Programa Arduino y el código de comunicación de Matlab lo encontramos en el anexo 4 y 5 respectivamente.

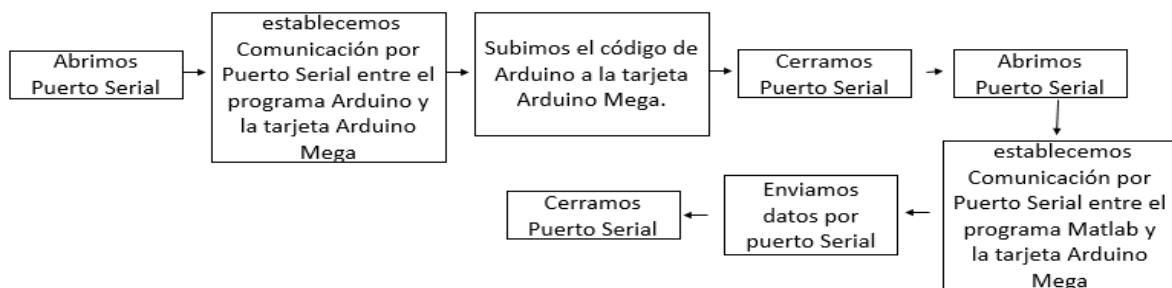


Figura 2.5 Comunicación Matlab y Dynamixel

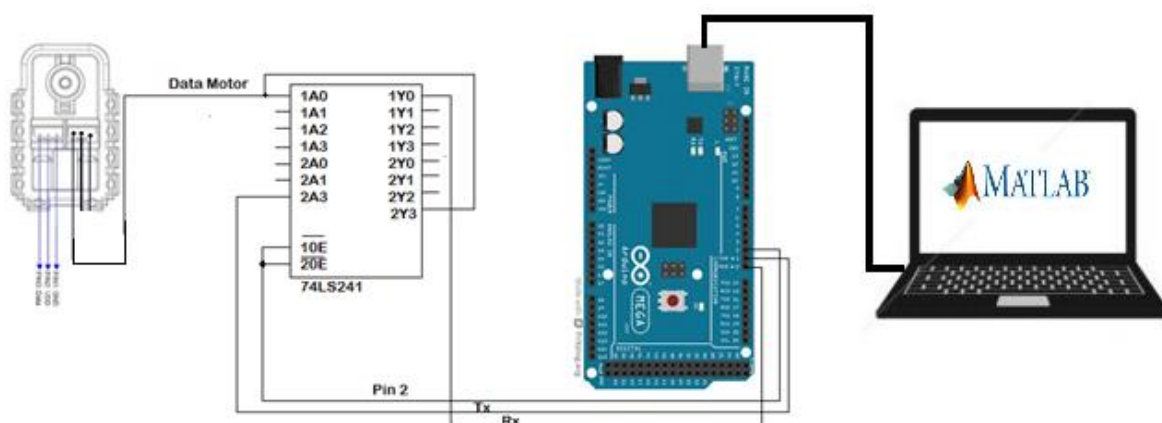


Figura 2.6 Comunicación completa. User's Manual. Robotis (2006),

2.4 Diseño y construcción del brazo robótico:

El diseño de cada pieza de pieza y su ensamble se realizaron en SolidWorks y se muestra en la figura 2.7. Ésta es una herramienta de diseño de modelado que permite crear sólidos en 3D y contiene operaciones de desarrollo de interfaz gráfica.

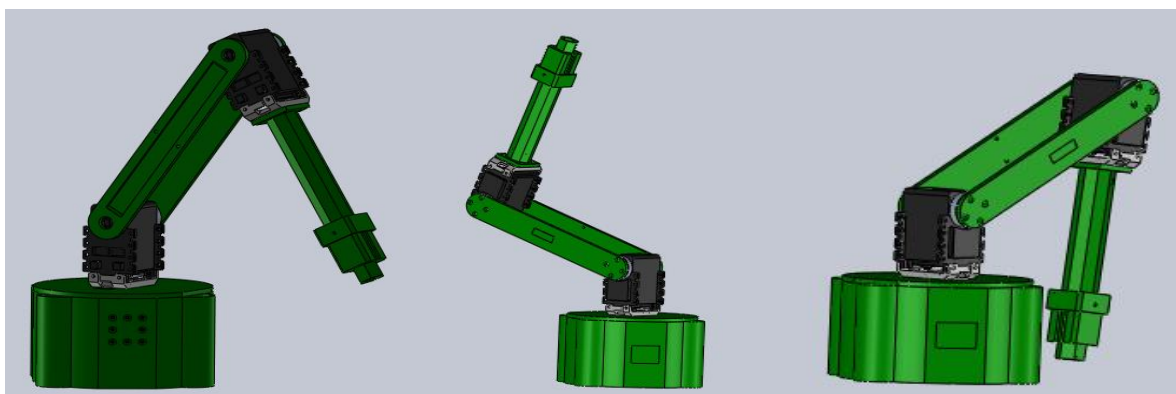
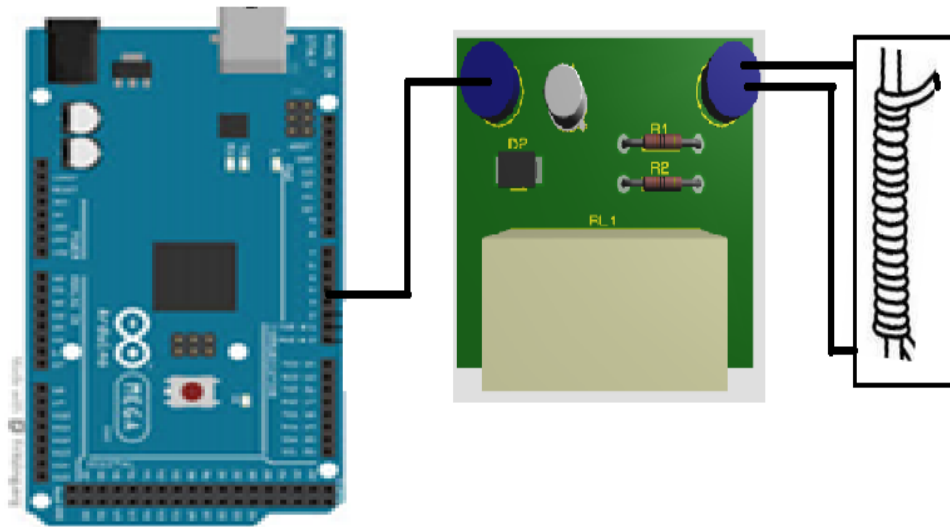
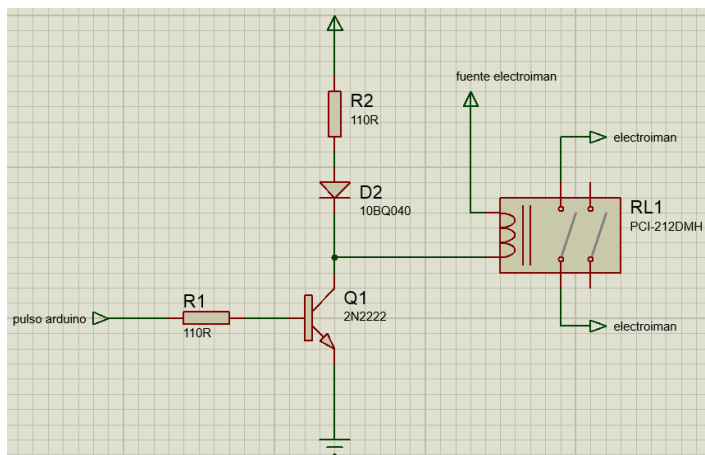


Figura 2.7 Modelado del brazo robótico

Las piezas a clasificar son tuercas, arandelas y tornillos. Su principal característica es que son metálicas, lo que quiere decir que basta con tener un efector que genere a voluntad un campo magnético lo suficientemente potente para poder alzar las piezas. Considerando esta característica se propone que el efector sea un electroimán. En la figura 2.8 y 2.9 se observa la implementación y circuito del electroimán respectivamente.



2.8 Implementación del electroimán



2.9 Circuito electroimán

Ahora se debe establecer las medidas de los eslabones. La selección la longitud de los eslabones es criterio del diseñador. En las tablas 2.2 se pueden observar las medidas de cada eslabón y en la tabla 2.3 se observa el peso de las piezas que interactúan con los eslabones

Tabla 2.2

| Elemento | Peso (gr) | Longitud (cm) |
|-----------|-----------|---------------|
| Base | 35 | 6 |
| Eslabón 2 | 10 | 16 |
| Eslabón 3 | 10 | 10.7 |

Tabla 2.2. Datos de los elementos del brazo

Tabla 2.3

| | |
|-------------|------|
| Electroimán | 25 g |
| Motor | 55 g |
| Tornillos | 5 g |
| Tuercas | 5 g |
| Arandelas | 5 g |

Tabla 2.3. Datos piezas

Aplicando la matriz de torque definida anteriormente se calcula los torques de los motores:

Motor 1 Torque 1.974 Kg

Motor 2 Torque 1.974 Kg

Motor 3 Torque 0.454 kg

Los motores Dynamixel Ax-12A responden satisfactoriamente a las exigencias de las labores asignadas al brazo. Las piezas impresas en 3D resisten las tensiones generadas por los pesos de las piezas, los motores y giros del brazo. El electroimán genera la atracción suficiente para levantar las tuercas, tornillos y arandelas. En la figura 2.9 se muestra el brazo robótico totalmente ensamblado y la construcción y medida de las piezas se encuentran en el anexo 1.

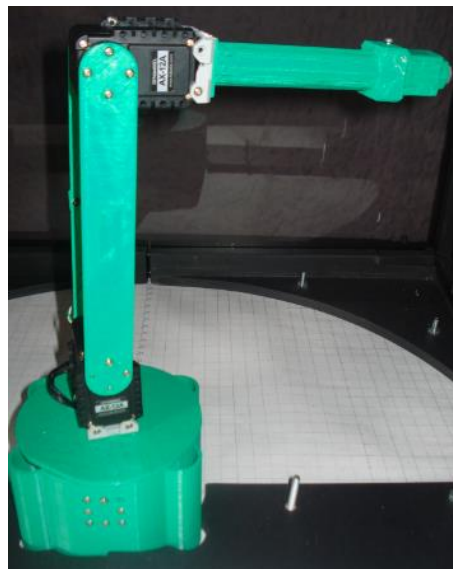


Figura 2.10 Ensamble brazo robotico

2.5 Diseño y construcción de la cabina

El diseño de la cabina debe tener las dimensiones adecuadas para que el robot se desplace con libertad en el área de trabajo y la cámara obtenga una imagen completa del área de trabajo. A partir del diseño del brazo robótico se establece que el área de trabajo es de $A = 1850\text{cm}^2$ lo que corresponde a un semicírculo con radio de 25cm (alcance del robot) menos un semicírculo con radio de 6cm (base del robot) ver figura 2.11. La cámara implementada en el proyecto es la Logitech c170 que necesita una distancia mínima de 71 cm para proyectar la totalidad del área de trabajo establecido.

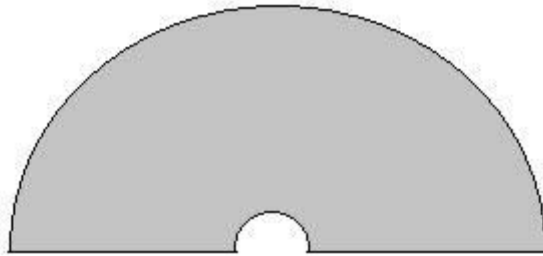


Figura 2.11 Área de trabajo

Teniendo en cuenta la distancia mínima que requiere la cámara y el área de trabajo del robot se diseñó en SolidWorks, ver figura 2.12, la cabina que cubrirá el brazo robótico. El robot está encerrado por una cabina con laterales de vidrio, tiene la base de madera y acrílico como se muestra en la figura 2.13. Las medidas de las piezas se encuentran en anexo 2.

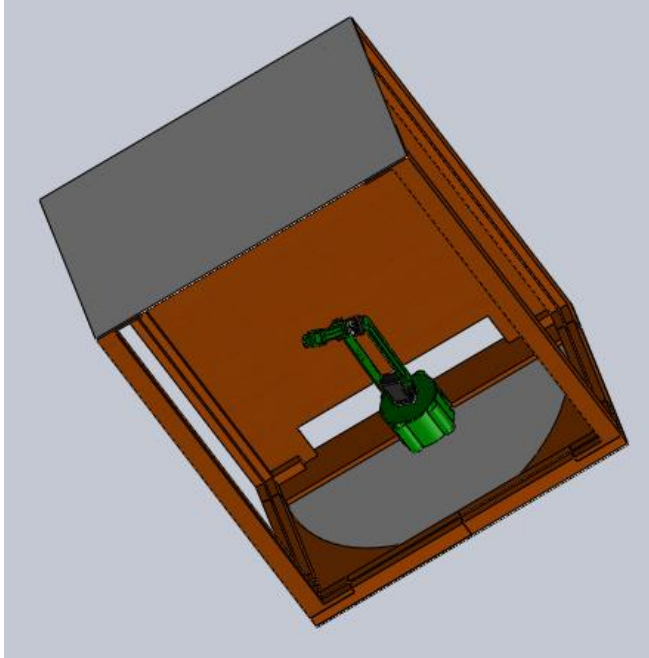


Figura 2.12 Modelado de cabina



Figura 2.13 Cabina y brazo robótico

2.5.1 Iluminación

El brazo robótico es un robot que clasifica y organiza tuercas, arandelas y tornillos. La identificación de estas piezas se desarrolla a partir de la forma de los objetos. Se obtiene una mejor imagen de la forma si la iluminación se hace a contra luz con las piezas. Lo que quiere decir que los bombillos, preferiblemente de luz blanca, están por debajo de la zona de trabajo del brazo robótico iluminando directamente a la cámara.

CAPITULO 3.-CINEMÁTICA Y CONTROL

La cinemática es el estudio de la mecánica clásica que se enfoca en la descripción del movimiento de los cuerpos con respecto a un sistema de referencia fijo. Dicho de otra manera no es más que el estudio geométrico del posicionamiento del efector final del robot. En los brazos robóticos existen dos tipos de problemas a resolver con cinemática:

Por un lado está el problema de **cinemática directa** que se encarga de indicar la posición del efector del robot en el espacio teniendo en cuenta el movimiento de cada articulación del robot, la distancia de los eslabones y el estado inicial del robot. Por el otro lado está el problema de **cinemática inversa** que es la encargada de definir los ángulos necesarios que debe tomar cada articulación para que el efector del robot llegue a una posición deseada en el espacio. Para la solución de estos 2 inconvenientes existe la siguiente herramienta.

3.1 Denavit- Hartenberg (D-H)

Las matrices de transformación propuestas por Denavit y Hartenberg en 1955 es un método sistemático que describe la relación rotacional y traslacional entre una cadena de eslabones unidos por varias articulaciones. El método de D-H establece una matriz de transformación homogénea de 4×4 para cada articulación con el propósito de poder encontrar una matriz compuesta de igual tamaño que indique la posición y orientación del robot con respecto al sistema de coordenadas base. La matriz de transformación para cada articulación está definida por:

$$A_i^{i-1} = Rot_{(z,\theta_i)} Tras_{(z,d_i)} Tras_{(x,a_i)} Rot_{(x,\alpha_i)}$$

$$A_i^{i-1} = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_i & -s\alpha_i & 0 \\ 0 & s\alpha_i & c\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_i^{i-1} = \begin{bmatrix} c\theta_i & -c\alpha_i s\theta_i & s\alpha_i s\theta_i & \alpha_i c\theta_i \\ s\theta_i & c\alpha_i c\theta_i & -s\alpha_i c\theta_i & \alpha_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Donde los parámetros requeridos para hacer las transformaciones en cada articulación vienen dadas por:

θ_i : Es el ángulo que forman los ejes x_{i-1} y x_i medido en un plano perpendicular al eje z_{i-1} , utilizando la regla de la mano derecha. Se trata de un parámetro variable en articulaciones giratorias.

D_i : Es la distancia a lo largo del eje z_{i-1} desde el origen del sistema de coordenadas $(i-1)$ -ésimo hasta la intersección del eje z_{i-1} con el eje x_i . Se trata de un parámetro variable en articulaciones prismáticas.

A_i : Es la distancia a lo largo del eje x_i que va desde la intersección del eje z_{i-1} con el eje x_i hasta el origen del sistema i -ésimo, en el caso de articulaciones giratorias. En el caso de articulaciones prismáticas, se calcula como la distancia más corta entre los ejes z_{i-1} y z_i .

A_i : Es el ángulo de separación del eje z_{i-1} y el eje z_i , medido en un plano perpendicular al eje x_i , utilizando la regla de la mano derecha. (CASTILLO & CABERTA).

La matriz compuesta está definida por:

$T = \prod_{i=1}^n A_i$ Donde n es el número de articulaciones del robot. Al obtener las matrices transformadas de las articulaciones y multiplicarlas entre si obtenemos la matriz compuesta $T = \begin{bmatrix} R_{3x3} & P_{3x1} \\ 0 & 1 \end{bmatrix}$ donde la sub-matriz R_{3x3} nos proporciona las ecuaciones matemáticas de la orientación del robot y el vector P_{3x1} nos brinda las ecuaciones matemáticas que determinan la posición del efector con respecto al sistema cartesiano de referencia.

3.2 Cinemática Directa

Gracias al método propuesto por Denavit – Hartenberg se puede establecer de forma sencilla las ecuaciones que describen la orientación y posición del robot teniendo como variables de entrada los ángulos de cada motor y la distancia de cada eslabón. En la *figura 3.1* se muestra los cuatro sistemas de coordenadas establecidas para las articulaciones, las distancias de cada eslabón y el ángulo de rotación de cada motor.

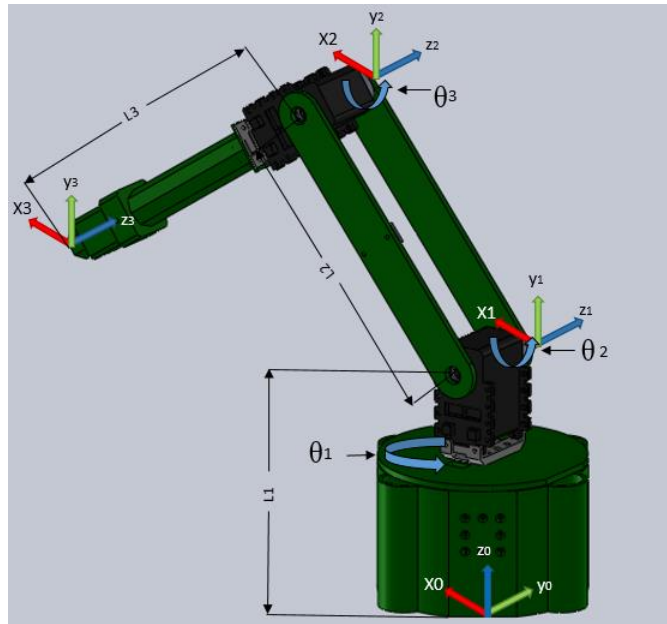


Fig.3.1 Sistema de coordenadas

Estableciendo los parámetros Denavit – Hartenberg se obtiene la siguiente tabla:

Tabla 3.1. Articulaciones

| Articulación | Θ | D | A | A |
|--------------|------------|----------------------|----------------------|------------|
| 1 | Θ_1 | $L_1=105.5\text{mm}$ | 0 | 90° |
| 2 | Θ_2 | 0 | $L_2=138\text{mm}$ | 0 |
| 3 | Θ_3 | 0 | $L_3=162.5\text{mm}$ | 0 |

Ahora se procede a hacer las matrices de transformación para cada articulación:

$${}^0A_1 = \begin{bmatrix} \cos(\theta_1) & 0 & \sin(\theta_1) & 0 \\ \sin(\theta_1) & 0 & -\cos(\theta_1) & 0 \\ 0 & 1 & 0 & 105.5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1A_2 = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & 138 \cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & 138 \sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2A_3 = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & 162.5 \cos(\theta_3) \\ \sin(\theta_3) & \cos(\theta_3) & 0 & 162.5 \sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Por consiguiente la matriz compuesta T esta definida por:

$$T = \begin{bmatrix} c(\theta_2 + \theta_3)c\theta_1 & -s(\theta_2 + \theta_3)c\theta_1 & s\theta_1 & Px \\ c(\theta_2 + \theta_3)s\theta_1 & -s(\theta_2 + \theta_3)s\theta_1 & -c\theta_1 & Py \\ s(\theta_2 + \theta_3) & c(\theta_2 + \theta_3) & 0 & Pz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Donde el vector $P_{(x,y,z)}$ es la posición espacial en el que se encuentra el efector final

(electroimán). El vector está definido por:

$$p_{(x,y,z)} = \begin{pmatrix} c\theta_1(325c(\theta_2 + \theta_3) + 276c\theta_2)/2 \\ s\theta_1(325c(\theta_2 + \theta_3) + 276c\theta_2)/2 \\ (325s(\theta_2 + \theta_3) + 276s\theta_2 + 211)/2 \end{pmatrix}$$

3.3 Cinemática inversa

La solución al problema de cinemática inversa consiste en encontrar un conjunto de ecuaciones que indiquen los ángulos (θ_1 , θ_2 y θ_3) en los que debe estar cada articulación para posicionar el efector o electroimán en una posición deseada en el sistema de coordenadas base. Este cálculo se puede hacer desde varios métodos y por ende las ecuaciones y ángulos pueden variar dependiendo la técnica usada. Nosotros proponemos utilizar el método de D-H para aprovechar las matrices transformadas de las articulaciones establecidas en el problema de cinemática directa.

Recordando que la matriz compuesta que brinda la posición y orientación del robot está definida por: $T = A_1^0 A_2^1 A_3^2$ y las matrices de transformación A_1 , A_2 y A_3 están previamente definidas en el modelo cinemático directo se despeja $A_3^2 = T(A_1^0)^{-1}(A_2^1)^{-1}$ para establecer tres ecuaciones con 3 incógnitas, θ_1 , θ_2 y θ_3 donde las coordenadas P_x , P_y y P_z son variables de entrada.

$$A_3^2 = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & 162.5 \cos(\theta_3) \\ \sin(\theta_3) & \cos(\theta_3) & 0 & 162.5 \sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3^2 = \begin{bmatrix} R_{11} & R_{21} & R_{31} & p_x \\ R_{21} & R_{22} & R_{32} & p_y \\ R_{31} & R_{32} & R_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta_1 & s\theta_1 & 0 & 0 \\ 0 & 0 & 1 & -211/2 \\ s\theta_1 & -c\theta_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta_2 & s\theta_2 & 0 & -138 \\ -s\theta_2 & c\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} O_{3x3} & P_{3x1} \\ 0 & 1 \end{bmatrix}$$

Donde:

O_{3x3}

$$= \begin{bmatrix} n_z S\theta_2 + n_x C\theta_1 C\theta_2 + n_y C\theta_2 S\theta_1 & O_z S\theta_2 + O_x C\theta_1 C\theta_2 + O_y C\theta_2 S\theta_1 & a_z S\theta_2 + a_x C\theta_1 C\theta_2 + a_y C\theta_2 S\theta_1 \\ n_z C\theta_2 - n_x C\theta_1 S\theta_2 - n_y S\theta_1 S\theta_2 & O_z C\theta_2 + O_x C\theta_1 S\theta_2 + O_y S\theta_2 S\theta_1 & a_z C\theta_2 + a_x C\theta_1 S\theta_2 - a_y S\theta_2 S\theta_1 \\ n_x S\theta_1 - n_y C\theta_1 & O_x S\theta_1 - O_y C\theta_1 & a_x S\theta_1 - a_y C\theta_1 \end{bmatrix}$$

$$P_{3x1} = \begin{bmatrix} p_z S\theta_2 - \frac{211}{2} S\theta_2 + p_x C\theta_1 C\theta_2 + p_y C\theta_2 S\theta_1 - 138 \\ p_z C\theta_2 - \frac{211}{2} C\theta_2 - p_x C\theta_1 S\theta_2 - p_y S\theta_1 S\theta_2 \\ p_x S\theta_1 - p_y C\theta_1 \end{bmatrix}$$

Al igualar la cuarta columna de A_3^2 con P_{3x1} obtendremos las siguientes ecuaciones con las incógnitas θ_1 , θ_2 y θ_3 :

$$p_z S\theta_2 - \frac{211}{2} S\theta_2 + p_x C\theta_1 C\theta_2 + p_y C\theta_2 S\theta_1 - 138 = 162.5 \cos(\theta_3)$$

$$p_z C\theta_2 - \frac{211}{2} C\theta_2 - p_x C\theta_1 S\theta_2 - p_y S\theta_1 S\theta_2 = 162.5 \sin(\theta_3)$$

$$p_x S\theta_1 - p_y C\theta_1 = 0$$

Resolviendo este conjunto de ecuaciones tendremos las ecuaciones que nos resuelven el problema de cinemática inversa. La solución es:

$$a = \frac{y^2 + x^2 - 7362.25}{276} \quad x = (p_x C\theta_1 + p_y S\theta_1) \quad y = \left(p_z - \frac{211}{2} \right)$$

$$\theta_1 = \tan^{-1}(p_y/p_x).$$

$$\theta_2 = \sin^{-1}\left(\frac{-(2ay) \pm \sqrt{(2ay)^2 - 4(y^2 + x^2)(a^2 - x^2)}}{2(y^2 + x^2)}\right)$$

$$\theta_3 = \cos^{-1}\left(\frac{p_z S\theta_2 - \frac{211}{2} S\theta_2 + p_x C\theta_1 C\theta_2 + p_y C\theta_2 S\theta_1 - 138}{162.5}\right)$$

3.4 Simulación de cinemática inversa en matlab

Simulink es una herramienta grafica que permite hacer cálculos matemáticos y simular plantas mecánicas entre otras aplicaciones útiles en la robótica. En simulink con la herramienta simMechanics primera generación se puede construir cualquier tipo de planta conectando diagramas de bloques que simulan eslabones (piezas solidas) y articulaciones (ejes de rotación) con la posibilidad de simular el entorno de trabajo del robot. Además de ofrecer herramientas de construcción, simulink también ofrece las herramientas para simular y controlar la planta. En la figura 3.2 se muestra el diagrama general de la construcción de la planta en simulink y el

bloque de cinemática inversa.

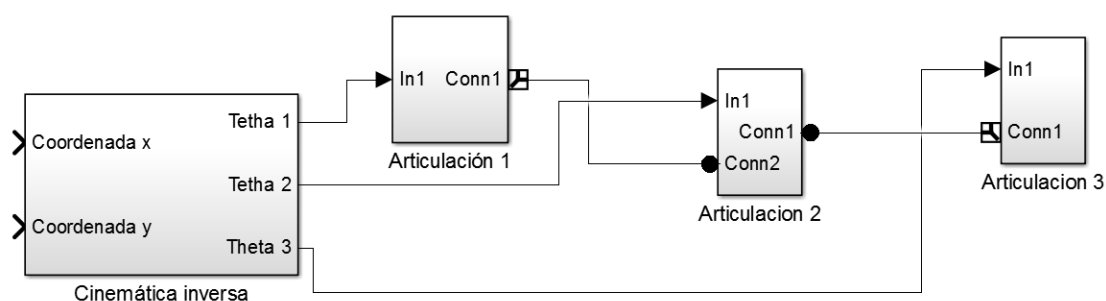


Figura 3.2 Cinemática inversa

En la figura 3.3, 3.4 y 3.5 se muestran los bloques utilizados para construir el brazo robótico de tres grados de libertad. Hay que tener en cuenta, a la hora de definir los parámetros de los diagramas de bloques, la dirección de los motores, las longitudes de cada eslabón, las matrices de posición en el espacio de los eslabones y articulaciones y de ser necesario el torque de los motores.

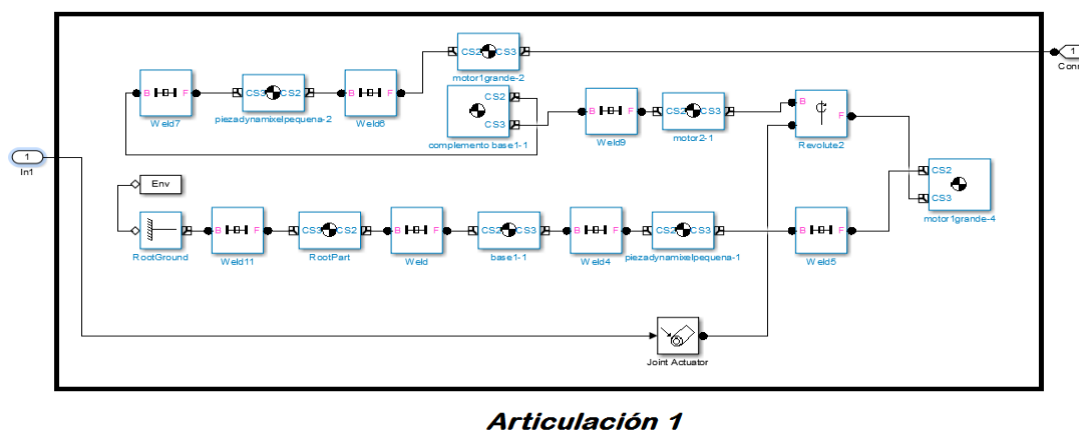


Figura 3.3 Articulación 1

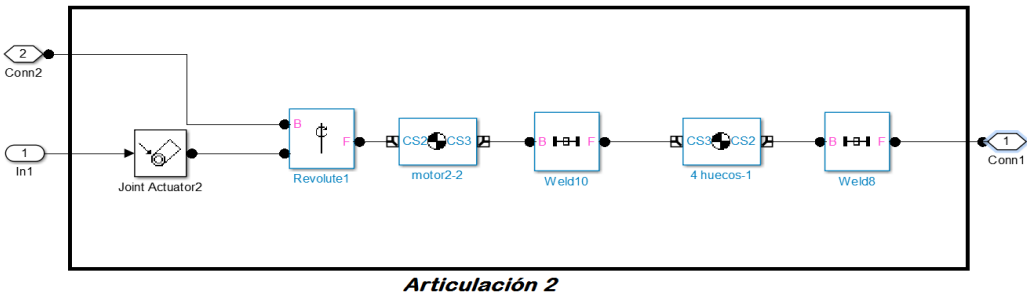


Figura 3.4 Articulación 2

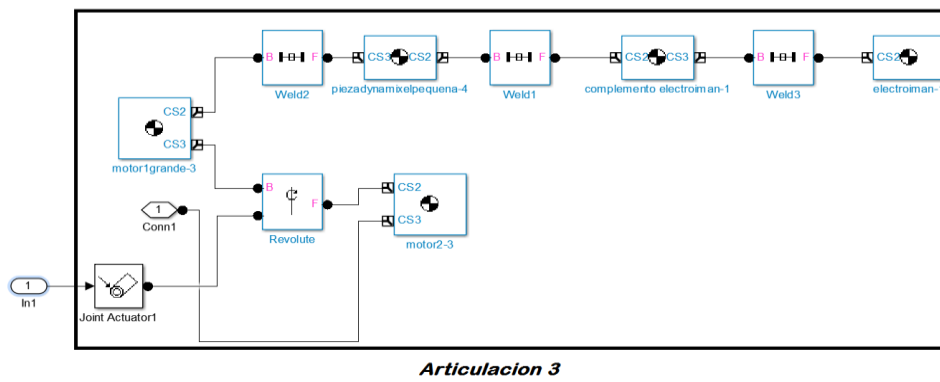


Figura 3.5 Articulación 3

El brazo robótico generado en simulink se muestra en la figura 3.6.

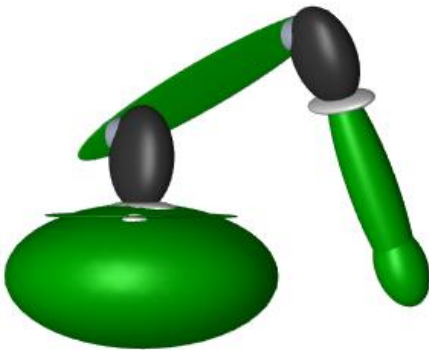


Figura 3.6 Robot en simulink

En la figura 3.7 se muestra el bloque de cinemática inversa que es el encargado de ordenar al robot que se desplace a las coordenadas introducidas por el usuario. En el anexo 6 se muestra el algoritmo de cinemática inversa y en el siguiente link https://www.youtube.com/watch?v=e_aVgffovfQ&feature=youtu.be se encuentra la simulación del robot junto con las gráficas de las entradas sinusoidales y las gráficas del comportamiento de cada una de las salidas. Se debe hacer la aclaración se asumió la coordenada z siempre como cero. El resultado de esta simulación no es del todo satisfactoria ya que hay áreas en las que el robot no se posiciona en las coordenadas dadas. Por tanto la implementación de un algoritmo de cinemática inversa como controlador no es fiable para posicionar el objeto con la precisión que se requiere para recoger tuercas, tornillos y arandelas; así que es necesario la ayuda de un controlador Fuzzy.

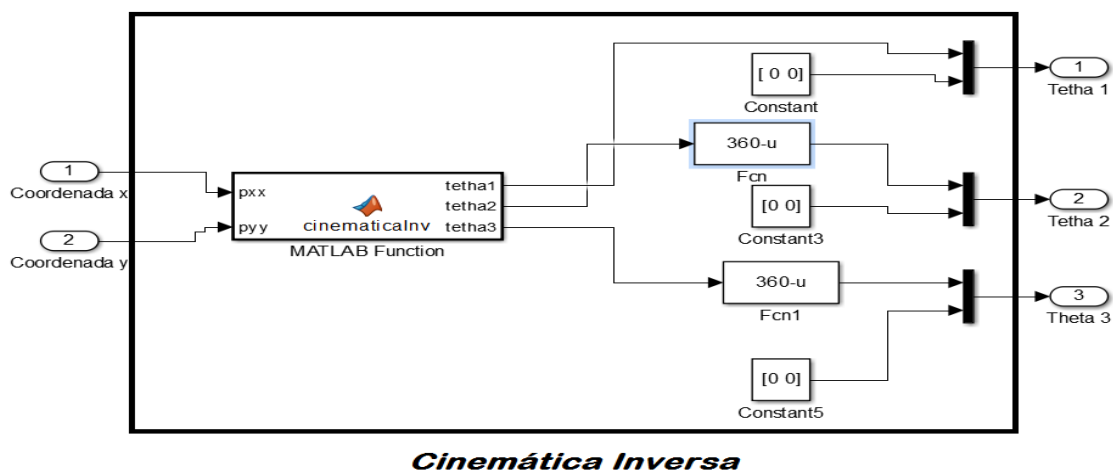


Figura 3.7 Bloque cinemática inversa

El algoritmo de cinemática inversa se encuentra en el anexo 6

3.5 Controlador Fuzzy

La lógica difusa es un método que proporciona una manera simple y elegante de obtener una conclusión a partir de información de entrada, ambigua, imprecisa, o incompleta (LEMUS) .

3.5.1 Estructura de controlador Fuzzy

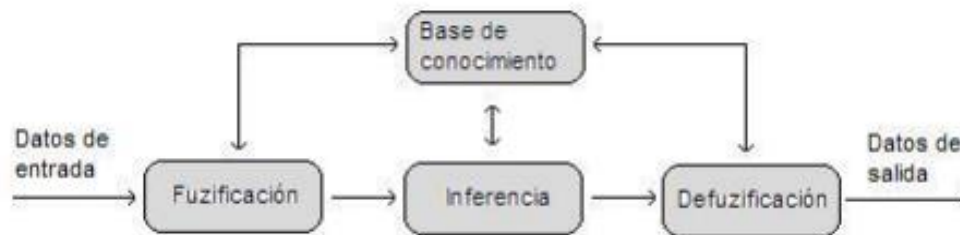


Figura 3.8 Estructura de un controlador Fuzzy. Ogata Katsuhiko (2006) Ingeniería de Control moderna

- Fuzificación: Calcula los datos de entrada.
- Inferencia: relaciona los datos de entrada con los conjuntos determinados del sistema y da un grado de pertenencia.
- Defuzificación: toma de decisiones.
- Base de conocimiento: son el conjunto de reglas determinados por cada diseñador.

3.6. Implementación control Fuzzy

Para este caso las entradas del controlador fuzzy son las coordenadas cartesianas en el que se encuentra el objeto. Las salidas son los ángulos en los que los motores se deben encontrar para

posicionarse encima del objeto. Teniendo en cuenta que las piezas estan posicionadas siempre en el plano bidimensional X - Y y aprovechando el teorema de pitagoras y la razón trigonométrica mostrada en la figura 3.9 a. se reducen el numero de entradas en el controlador Fuzzy a una sola entrada obteniendo las mismas salidas como se muestra en la figura 3.9b.

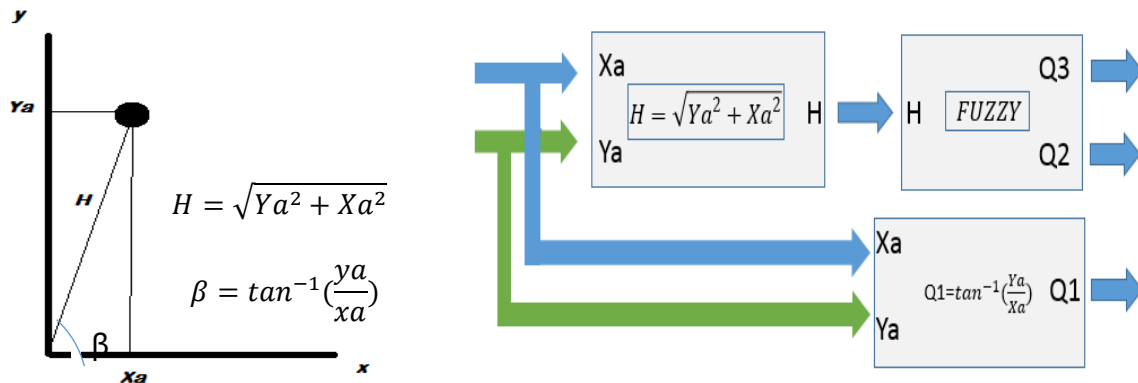


Figura 3.9 a) teorema y razón trigonométrica

Figura 3.9b) esquema general del controlador.

X_a =posición en el eje x . Y_a = posición en el eje y . Q_1 = ángulo motor 1.

Q_2 = ángulo motor 2. Q_3 =ángulo motor 3.

Es indispensable determinar la posición inicial del brazo robótico y el sentido de giro de cada motor para definir los parámetros del controlador Fuzzy.

En la figura 3.10 a se observa el estado inicial del robot en la simulación. Esto es cuando los motores tienen un ángulo de giro de cero grados. Al comparar el estado inicial del robot simulado y el estado inicial del robot real, figura 3.10 a y 3.10b respectivamente se aprecia que el sistema de coordenadas de referencia para el brazo simulado es diferente al

sistema de coordenadas del brazo real. Para que el brazo robotico y la simulacion concidan en la posicion inicial es necesario sumar 50° en el motor 1 y 20° en el motor 3.

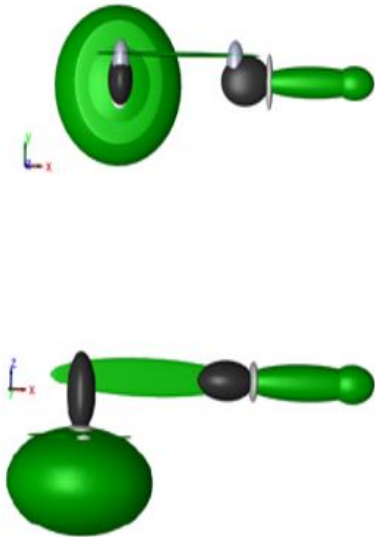


Figura 3.10 a. Estado inicial simulado



Figura 3.10b Estado inicial real

La tabla 3.2 muestra los valores de los angulos que debe tener cada motor para llegar a una determinada coordenada obtenidos de manera experimental. Al comparar la tabla 3.2 con la tabla 3.3 obtenida de las ecuaciones cinematicas inversas se observa que la diferencia entre los angulos obtenidos y simulados se mantienen en un rango pequeño ver tabla 3.4. Estas diferencias de angulos obedece a dos factores principales: El primer factor es el error humano en la tabla experimental y el segundo factor equivale a la diferencia de angulos mencionados en el anterior parrafo.

Tabla 3.2 Angulos de los motores

| Tabla experimental | | | | | | | | | |
|--------------------|-----------------|------------|------------|------------|-----------------|-----------------|---------|---------|---------|
| Coordenada y | coordenada x | Grados | | | Coordenada y | Coordenada x | Grados | | |
| | | motor 1 | motor 2 | motor 3 | | | motor 1 | motor 2 | motor 3 |
| 0 | 6 | 50 | 21 | 140 | 0 | 17 | 50 | 26 | 103 |
| 0 | 7 | 50 | 24 | 138 | 0 | 18 | 50 | 25 | 98 |
| 0 | 8 | 50 | 25 | 135 | 0 | 19 | 50 | 25 | 95 |
| 0 | 9 | 50 | 27 | 134 | 0 | 20 | 50 | 22 | 89 |
| 0 | 10 | 50 | 28 | 130 | 0 | 21 | 50 | 20 | 83 |
| 0 | 11 | 50 | 28 | 126 | 0 | 22 | 50 | 19 | 79 |
| 0 | 12 | 50 | 28 | 123 | 0 | 23 | 50 | 16 | 72 |
| 0 | 13 | 50 | 28 | 120 | 0 | 24 | 50 | 12 | 63 |
| 0 | 14 | 50 | 28 | 116 | 0 | 25 | 50 | 11 | 57 |
| 0 | 15 | 50 | 28 | 112 | | | | | |
| 0 | 16 | 50 | 27 | 108 | | | | | |

Tabla 3.3 Cinematica inversa

| Tabla cinemática inversa | | | | | | | | | |
|--------------------------|-----------------|------------|------------|------------|-----------------|-----------------|---------|---------|---------|
| coordenada Y | coordenada x | Grados | | | Coordenada y | Coordenada x | Grados | | |
| | | motor 1 | motor 2 | motor 3 | | | motor 1 | motor 2 | motor 3 |
| 0 | 6 | 0 | 17 | 116 | 0 | 17 | 0 | 22 | 75 |
| 0 | 7 | 0 | 19 | 112 | 0 | 18 | 0 | 21 | 72 |
| 0 | 8 | 0 | 21 | 107 | 0 | 19 | 0 | 19 | 68 |
| 0 | 9 | 0 | 22 | 103 | 0 | 20 | 0 | 18 | 65 |
| 0 | 10 | 0 | 23 | 99 | 0 | 21 | 0 | 16 | 62 |
| 0 | 11 | 0 | 24 | 96 | 0 | 22 | 0 | 14 | 58 |
| 0 | 12 | 0 | 24 | 92 | 0 | 23 | 0 | 11 | 54 |
| 0 | 13 | 0 | 24 | 88 | 0 | 24 | 0 | 8 | 50 |
| 0 | 14 | 0 | 24 | 85 | 0 | 25 | 0 | 5 | 46 |
| 0 | 15 | 0 | 24 | 82 | | | | | |
| 0 | 16 | 0 | 23 | 78 | | | | | |

Tabla 3.4 Comparacion

| Coord. Y | Coord. X | Error motor 1 | Error motor 2 | Error motor 3 | Coord. Y | Coord. X | Error motor 1 | Error motor 2 | Error motor 3 |
|----------|----------|---------------|---------------|---------------|----------|----------|---------------|---------------|---------------|
| 0 | 6 | 50 | 4 | 24 | 0 | 17 | 50 | 4 | 28 |
| 0 | 7 | 50 | 5 | 26 | 0 | 18 | 50 | 4 | 26 |
| 0 | 8 | 50 | 4 | 28 | 0 | 19 | 50 | 6 | 27 |
| 0 | 9 | 50 | 5 | 31 | 0 | 20 | 50 | 4 | 24 |
| 0 | 10 | 50 | 5 | 31 | 0 | 21 | 50 | 4 | 21 |
| 0 | 11 | 50 | 4 | 30 | 0 | 22 | 50 | 5 | 21 |
| 0 | 12 | 50 | 4 | 31 | 0 | 23 | 50 | 5 | 18 |
| 0 | 13 | 50 | 4 | 32 | 0 | 24 | 50 | 4 | 13 |
| 0 | 14 | 50 | 4 | 31 | 0 | 25 | 50 | 6 | 11 |
| 0 | 15 | 50 | 4 | 30 | | | | | |
| 0 | 16 | 50 | 4 | 30 | | | | | |

La informacion de la tabla 3.2 aporta la informacion necesaria para construir el controlador fuzzy. Teniendo en cuenta que en la tabla 3.2 la coordenada Y siempre es 0, la entrada del controlador fuzzy es la coordenada X (que hace referencia a la distancia l de la figura 3.3) y las salidas son solo los angulos del motor 2 y motor 3. Matlab ofrece la herramienta gráfica Fuzzy Logic Toolbox para la construccion de controladores fuzzy como sigue:

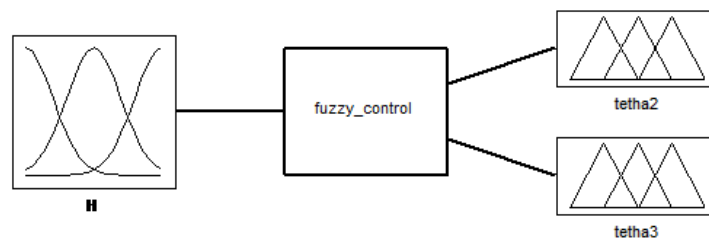


Figura 3.11 Fuzzy_control

Membresía H:

En el anexo 7 esta la tabla con los rangos de cada entrada y la función de la membresía H y en la figura 3.12 la representación gráfica de la membresía H.

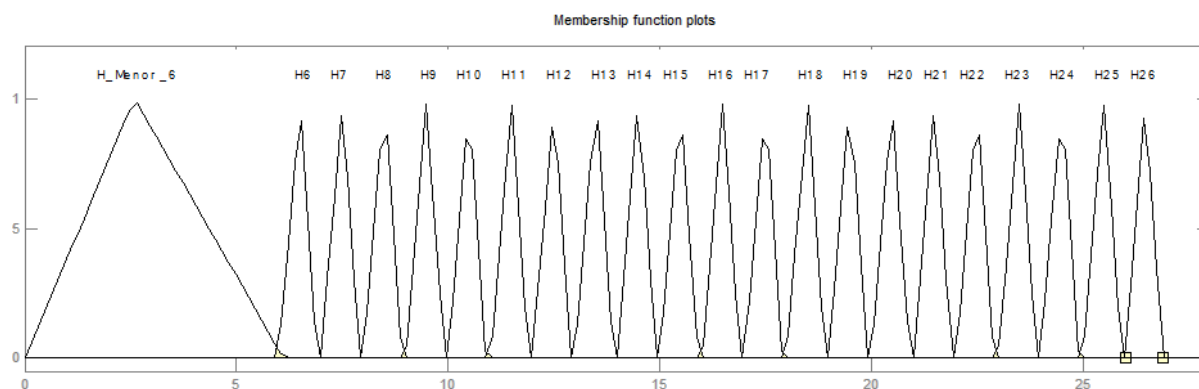


Figura 3.12 Membresía H

Membresía Tetha2:

En el anexo 7 esta la tabla con los rangos de cada entrada y su función de la membresía H. Así mismo la figura 3.13 representa gráficamente la membresía tetha2. Esta corresponde a los posibles ángulos que debe tomar el motor 2.

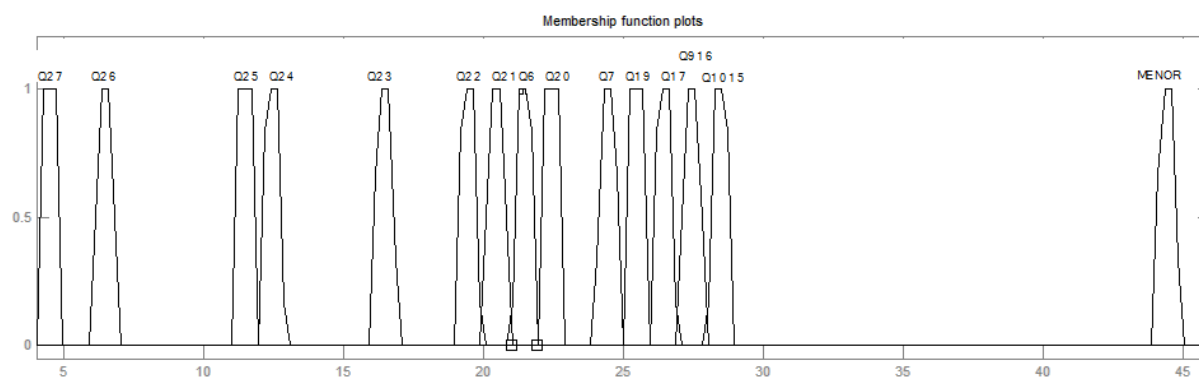


Figura 3.13 Membresía Tetha 2

Membresía Tetha3:

En el anexo 8 se encuentran los rangos de cada entrada y su función de la membresía H. La figura 3.14 representa gráficamente la membresía tetha2 correspondiente a los posibles angulos que debe tomar el motor 3.

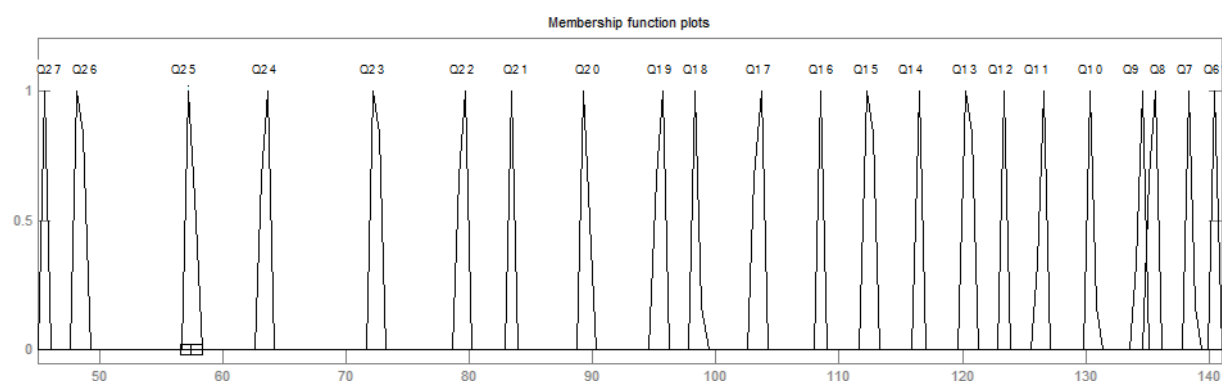


Figura 3.14 Membresía Tetha 3

En la tabla 3.5 se establece la relación de entrada y salida de cada una de las posibles entradas de la membresía H. El comportamiento de tetha 2 y tetha 3 con respecto a la entrada se muestra en la figura 3.15.

Tabla 3.5 Membresía H (in-out)

| H | Theta 2 | Theta 3 | H | Tetha 2 | Tetha 3 |
|-----|---------|---------|-----|---------|---------|
| H6 | Q6 | Q6 | H17 | Q17 | Q17 |
| H7 | Q7 | Q7 | H18 | Q19 | Q18 |
| H8 | Q19 | Q8 | H19 | Q19 | Q19 |
| H9 | Q916 | Q9 | H20 | Q20 | Q20 |
| H10 | Q1015 | Q19 | H21 | Q21 | Q21 |
| H11 | Q1015 | Q11 | H22 | Q22 | Q22 |
| H12 | Q1015 | Q12 | H23 | Q23 | Q23 |
| H13 | Q1015 | Q13 | H24 | Q24 | Q24 |
| H14 | Q1015 | Q14 | H25 | Q25 | Q25 |
| H15 | Q1015 | Q15 | H26 | Mayor | Q26 |
| H16 | Q916 | Q16 | | menor | |

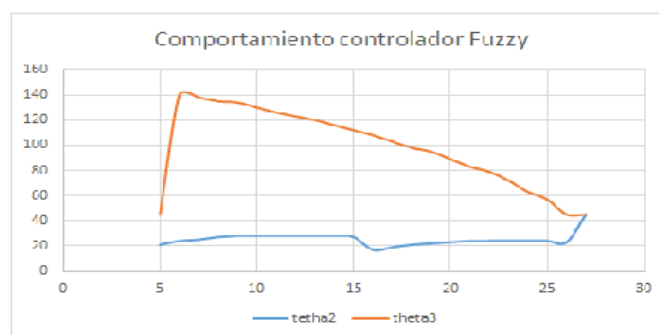


Figura 3.15 Comportamiento del controlador

Por ultimo, para que los angulos arrojados por la ecuacion $\tan^{-1}\left(\frac{y}{x}\right) = Q1$ siempre arroje angulos positivos es necesario aplicar el siguiente diagrama:

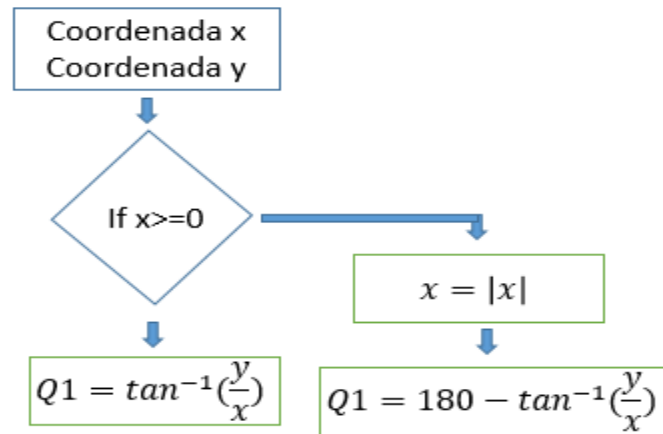


Figura 3.16 Aplicación de ecuacion

3.7 Pruebas del funcionamiento del brazo robotico con el controlador Fuzzy

En la figura 3.17 se aprecia el mapa donde se desplaza el brazo robotico para observar la precision con la cual el robot llega a un punto determinado. El color negro representa las coordenadas que el brazo robotico no puede alcanzar. El color rojo representa que el objeto no se alcanzo. El color azul representa que el objeto fue alcanzado con ayuda de la atracción del electroiman. Y el color verde significa que el brazo se posicionó encima del objeto.

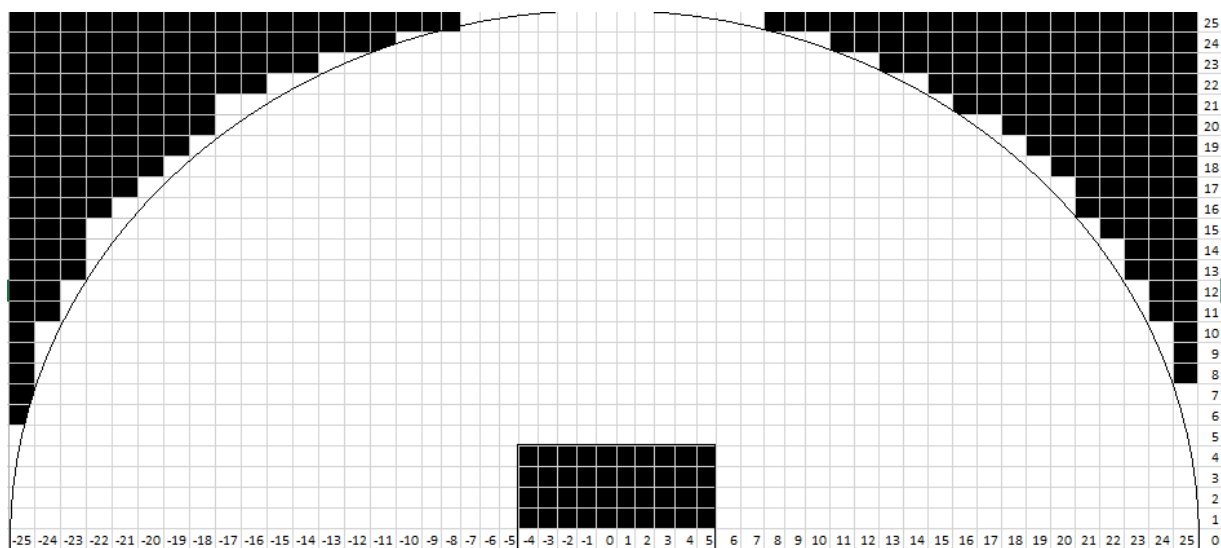


Figura 3.17 Mapa

3.7 Pruebas de funcionamiento del controlador

Los resultados fueron los esperados:

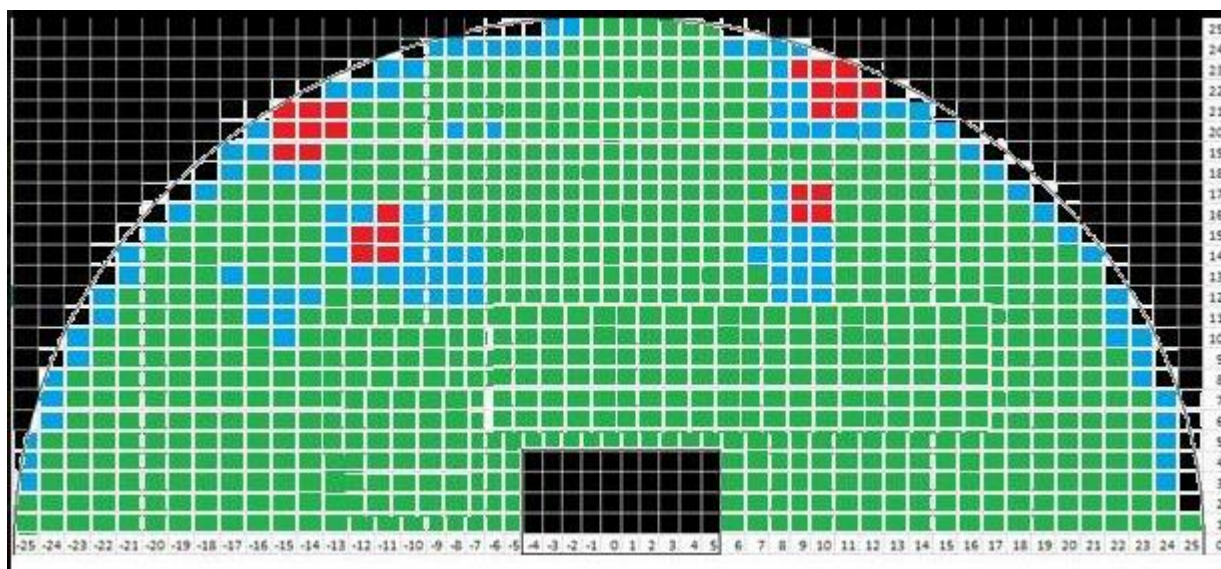


Figura 3.18 Mapa con zona en color

Zona verde: El efector final se posiciona sobre el objeto

Zona rojo: El brazo robótico no pudo alcanzar este objeto

Zona azul: Presenta un poco d dificultad pero con ayuda del electroimán es posible la obtención de la pieza.

Zona negra: espacios que el robot no alcanza

En el siguiente link se aprecia las pruebas de funcionamiento del controlador Fuzzy:

<https://www.youtube.com/watch?v=IqvbWbljY4s&feature=youtu.be>

CAPITULO 4.-VISION ARTIFICIAL

El extraer características de una imagen constituye el objetivo principal de la visión artificial. En la figura 4.1 se muestra de manera general los pasos que se realizaron para la obtención de la información necesaria antes de generar un algoritmo que indique la posición y la clase de los objetos que hay en la zona del trabajo del brazo robótico.

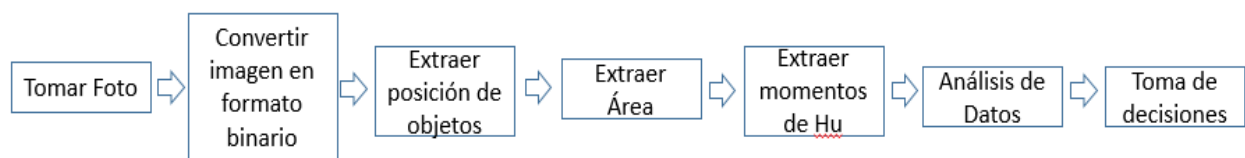


Figura 4.1 Obtención de información

Con apoyo en las herramientas de Matlab se obtiene una foto del área de trabajo del brazo robótico. En el anexo 9, la primera parte del código muestra la forma de instalar una cámara externa en Matlab y la instrucción que se requiere para poder visualizar la imagen que esté tomando la cámara. Así mismo toma una imagen en formato RGB (Red, Green, Blue) que es representada por una matriz general con tres sub-matrices que corresponden a las tres composiciones de color de una imagen.

4.1 Ajuste de cámara

Al tomar la foto es necesario asegurar la cámara para que proyecte con totalidad el área de trabajo del brazo robótico. La matriz para representar numéricamente una imagen en Matlab es de $768 \times 1024 \times 3$ donde 768×1024 es el tamaño de las tres matrices de colores primarios. Por tanto,

si la cámara está muy cerca o con un ángulo de inclinación inadecuado no se proyecta en su totalidad el área de trabajo del robot. Así que antes de empezar a crear o implementar cualquier algoritmo de reconocimiento de objetos, cuadrar de forma manual la cámara para que proyecte en su totalidad el área de trabajo.

En la figura 4.2 se muestran los pasos que se deben hacer para ajustar la cámara. En el anexo 9 se muestra el código y la explicación por bloques del algoritmo que convierte una imagen RGB en una imagen binaria, recorta la imagen binaria y hace la cuadrícula de la misma imagen.

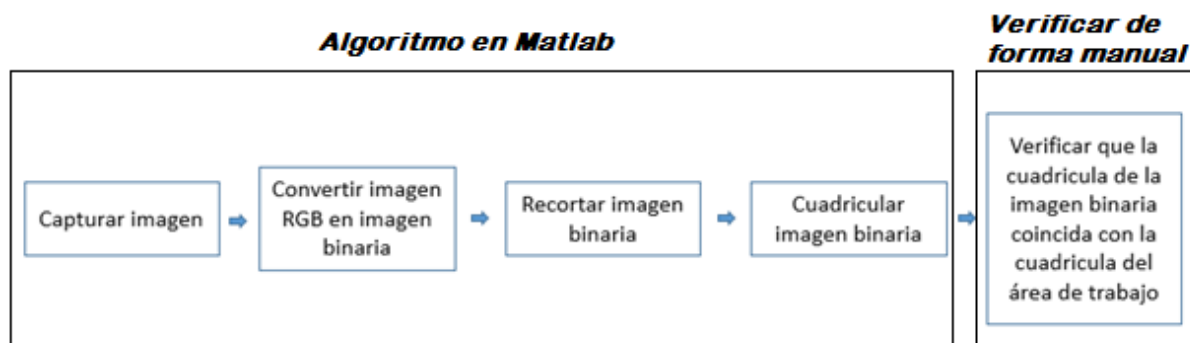


Figura 4.2 Ajuste de cámara

En el anexo 9 el algoritmo siempre entrega una imagen binaria cuadriculada como se muestra en las figuras 4.3 a. De forma manual se verifica que la cuadrícula de la imagen arrojada por matlab tenga una dimensión de 50x25 como se muestra en la figura 4.3b.

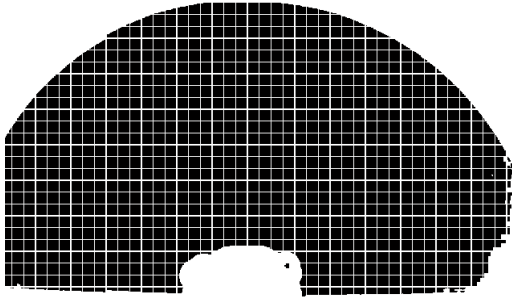


Figura 4.3 a Imagen binaria

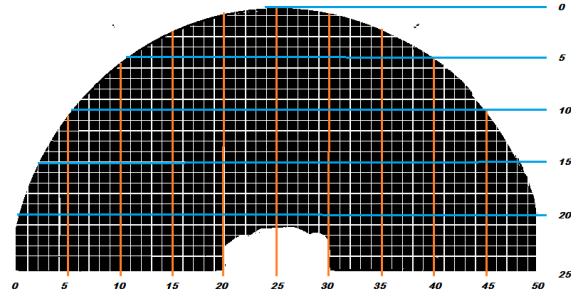


Figura 4.3b Imagen por matlab

4.2 Análisis de piezas

El análisis de las piezas para su identificación y clasificación se hacen a partir del área.

4.2.1. Área:

Como primera medida hacer un estudio previo sobre las características que tienen las tuercas tornillos y arandelas. En la figura 4.4 a. se ven los tres grupos de piezas utilizadas y en la figura 4.4b la imagen en formato binario de las mismas piezas. El volver una imagen RGB a imagen binaria hace que muchas de las piezas sean modificadas en área conservando su forma. En la figura 4.4b en la clase 2 se observa que el primer tornillo tiene menos área que los otros dos tornillos pero mantiene la forma característica de los tornillos. Lo mismo pasa en la clase 3 de la figura 4.4b con la primera tuerca.



Figura 4.4 a. Piezas reales

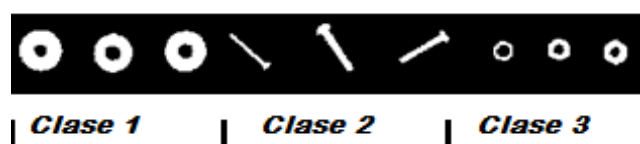


Figura 4.4b Piezas en formato binario

En primer lugar saber si el área de las piezas es un factor definitivo para conocer la clase a la que pertenecen es definitivo. Para ello con ayuda de la función región Props de Matlab se obtuvo el área de los objetos y su posición. El anexo 10 cuenta con el código que determina el área de los objetos. En primera medida fue desplazado una arandela, una tuerca y un tornillo por toda el área de trabajo para obtener el área del objeto a analizar. En la tabla 4.1 y la figura 4.5 hay un ejemplo de la forma en que se hizo la recolección de datos de área. La imagen en formato binario de un solo tornillo está en la figura 4.5 y la tabla 4.1 muestra los datos de área y posición de las figuras que se encuentran en la imagen binaria. El área de 238284 corresponde a todo lo que está fuera del área de trabajo del brazo robótico. Por ende el siguiente objeto que tiene mayor área es el del tornillo con un área de 235 y el resto de áreas son figuras que se generaron a partir de la conversión a formato binario una imagen RGB.

Tabla 4.1 Área y posición

| Fields | Area | Centroid |
|--------|--------|----------------|
| 1 | 238482 | [585.2320 3... |
| 2 | 7 | [39.5714 33... |
| 3 | 8 | [41 403.5000] |
| 4 | 16 | [134.1250 5... |
| 5 | 235 | [490.8511 1... |
| 6 | 50 | [755.5200 5... |
| 7 | 5 | [839.2000 4... |
| 8 | 1 | [888 191] |
| 9 | 3 | [897 294] |

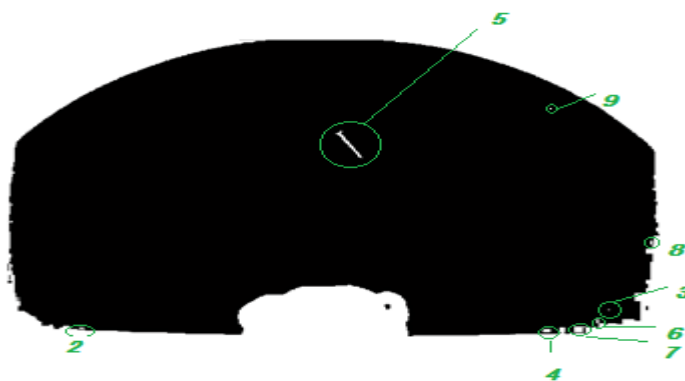


Figura 4.5 Tornillo en formato binario

Los resultados del análisis se muestran en las tablas 4.2, 4.3 y 4.4. Como primera conclusión de este análisis esta que todo aquel objeto que tenga un área mayor a 120 y menor a 1700 casillas pertenece a tuercas tornillos o arandelas. También se puede decir que todo aquel que tenga un área entre 1000 y 1700 casillas es una arandela. Como última conclusión que el área no es un factor preciso para determinar si el objeto es tuerca o tornillo. Para esto se busca otro tipo de características de los objetos que determine con precisión si es tuerca o tornillo.

Tabla 4.2 Área arandela

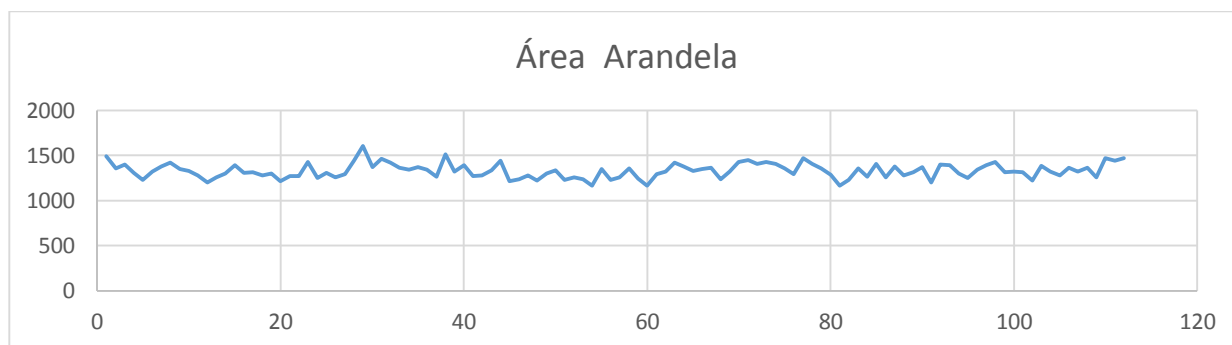


Tabla 4.3 Área tornillo

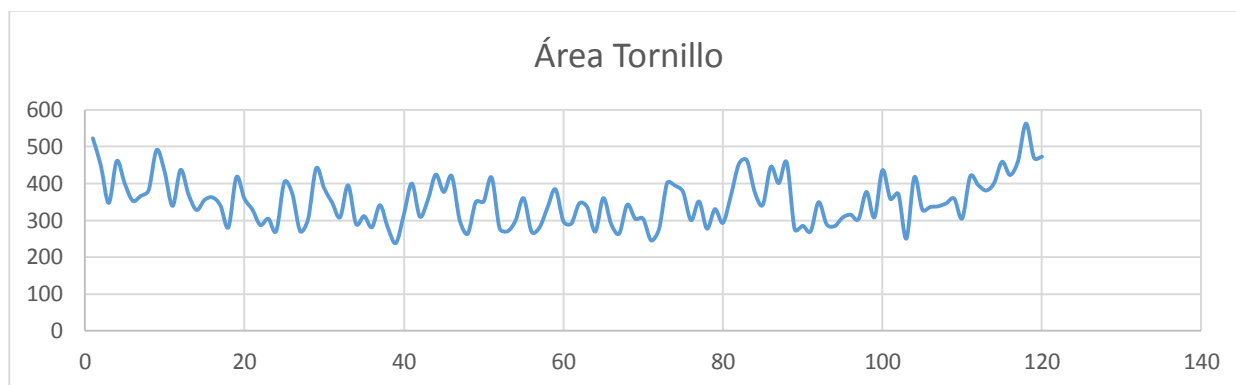
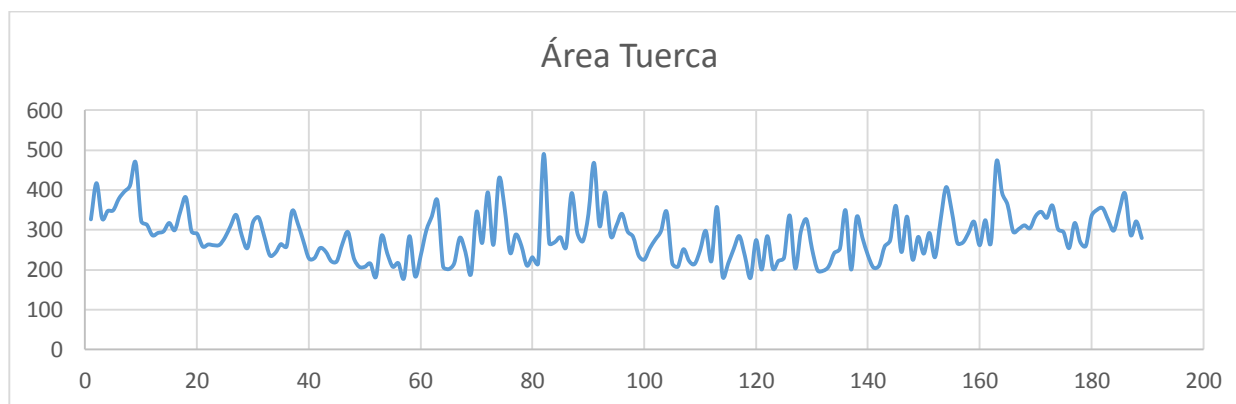


Tabla 4.4 Área tuerca



4.2.2 Momentos invariantes de HU

Los momentos de Hu son momentos invariantes a la traslación, rotación y cambios de escala de una figura bidimensional. Estos momentos parten de los momentos geométricos cuya función es:

$$M_{p,q} = \sum_{x=1}^N \sum_{y=1}^M x^p y^q I(x,y)$$

Donde $M_{p,q}$ es el momento geométrico de la figura de orden $pq \geq 0$ y $I(x,y)$ es el valor del pixel. Es así como el momento geométrico $M_{0,0}$ representa el área de la figura en una imagen binaria.

$$A = M_{0,0}$$

Ahora, antes de establecer los momentos invariantes de HU, se debe calcular el centro de gravedad para una figura, los momentos centrales y los momentos centrales normalizados. El centro de gravedad viene dada por:

$$\bar{x} = \frac{\sum_{x=1}^N \sum_{y=1}^M x I(x,y)}{A}$$

$$\bar{y} = \frac{\sum_{x=1}^N \sum_{y=1}^M y I(x,y)}{A}$$

Los momentos centrales están dadas por:

$$u_{p,q} = \sum_{x=1}^N \sum_{y=1}^M (x - \bar{x})^p (y - \bar{y})^q I(x, y)$$

Y los momentos centrales normalizados se calculan como sigue:

$$n_{p,q} = \frac{\mu_{p,q}}{\frac{\frac{p+q}{2}+1}{A}}$$

Finalmente Hu propuso los siguientes parámetros para hallar 7 momentos invariantes:

$$h_1 = n_{2,0} + n_{0,2}$$

$$h_2 = (n_{2,0} - n_{0,2})^2 + 4(n_{1,1})^2$$

$$h_3 = (n_{3,0} - 3n_{1,2})^2 + (3n_{2,1} - n_{0,3})^2$$

$$h_4 = (n_{3,0} + n_{1,2})^2 + (n_{0,3} + n_{2,1})^2$$

$$h_5 = (n_{3,0} - 3n_{1,2})(n_{3,0} + n_{1,2}) \left[(n_{3,0} + n_{1,2})^2 - 3(n_{0,3} + n_{2,1})^2 \right]$$

$$+ (3n_{2,1} - n_{0,3})(n_{0,3} + n_{2,1}) \left[3(n_{3,0} + n_{1,2})^2 - (n_{0,3} + n_{2,1})^2 \right]$$

$$h_6 = (n_{2,0} - n_{0,2}) \left[(n_{3,0} + n_{1,2})^2 - (n_{0,3} + n_{2,1})^2 + 4n_{1,1}(n_{3,0} + n_{1,2})(n_{0,3} + n_{2,1}) \right]$$

$$h_7 = (3n_{2,1} - n_{0,3})(n_{3,0} + n_{1,2}) \left[(n_{3,0} + n_{1,2})^2 - 3(n_{0,3} + n_{2,1})^2 \right] - (n_{3,0} - 3n_{1,2})(n_{0,3} + n_{2,1}) \left[3(n_{3,0} + n_{1,2})^2 - 3(n_{0,3} + n_{2,1})^2 \right]$$

Los anexos 11 y 12 muestran el código que se implementó para obtener los momentos invariantes de HU. En la figura 4.6 se muestra un tornillo, una tuerca y una arandela a los que se le obtuvo los momentos de HU mostrada en la tabla 4.5

Tabla 4.5 Momentos de HU

| | TORNILLO | ARANDELA | TUERCA |
|----|------------|------------|------------|
| H1 | 2,59074814 | 2,56584782 | 2,26481782 |
| H2 | 5,18149627 | 5,13169564 | 4,52963565 |
| H3 | 8,76248457 | Inf | Inf |
| H4 | 8,76248457 | Inf | Inf |
| H5 | 17,5249691 | Inf | Inf |
| H6 | 11,3532327 | Inf | Inf |
| H7 | Inf | Inf | Inf |



Figura 4.6 Tornillo-tuerca-arandela

Al igual que el análisis de área de objetos se realiza un análisis de los momentos de Hu para el tornillo, arandela y tuerca. Para poder graficar los momentos de Hu el número Infinito (Inf) fue reemplazado por 1000.

En las tablas 4.6, 4.7, 4.8, 4.9, 4.10, 4.11, 4.12 se ve los momentos de Hu de ciento trece muestras de arandelas del mismo tamaño distribuidas por toda el área de trabajo del brazo robótico. Los momentos invariantes H3, H4, H5, H6, H7 no son parámetros confiables para determinar si el objeto es una arandela. Por el momento es confirmado que el momento 1 de Hu está en el rango de 2,53 y 2,59. El segundo momento de Hu está en el rango de 5.05 y 5.2.

Tabla 4.6 HU1-arandela

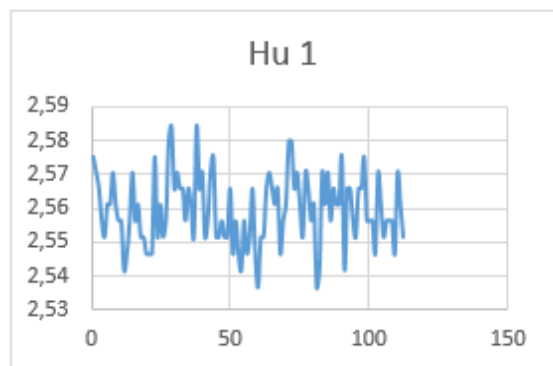


Tabla 4.7 HU2-arandela

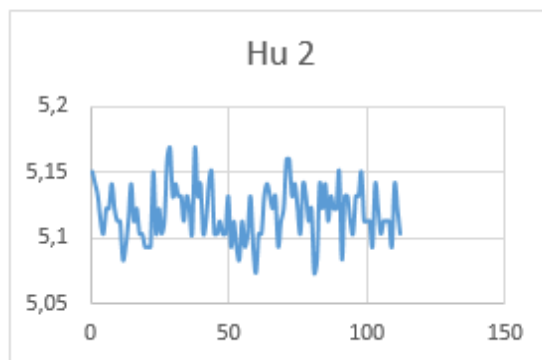


Tabla 4.8 HU3-arandela

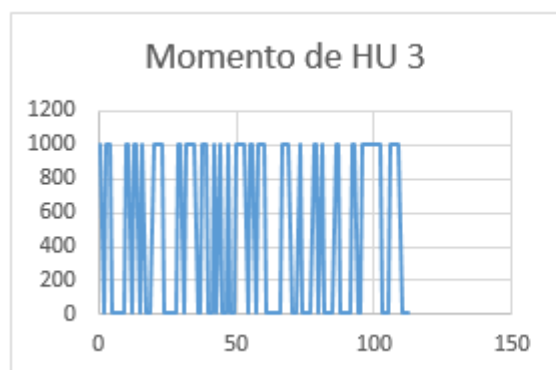


Tabla 4.9 HU4-arandela

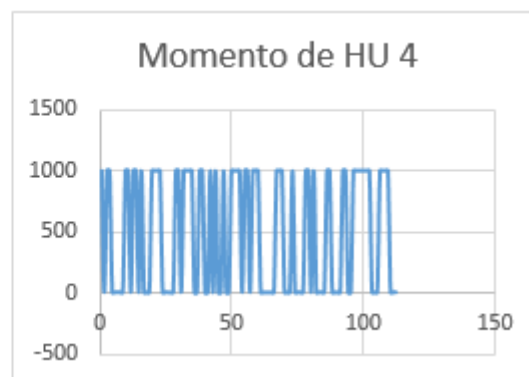


Tabla 4.10 HU5-arandela



Tabla 4.11 HU6-arandela

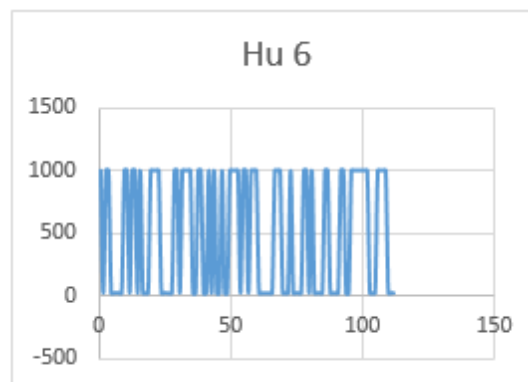
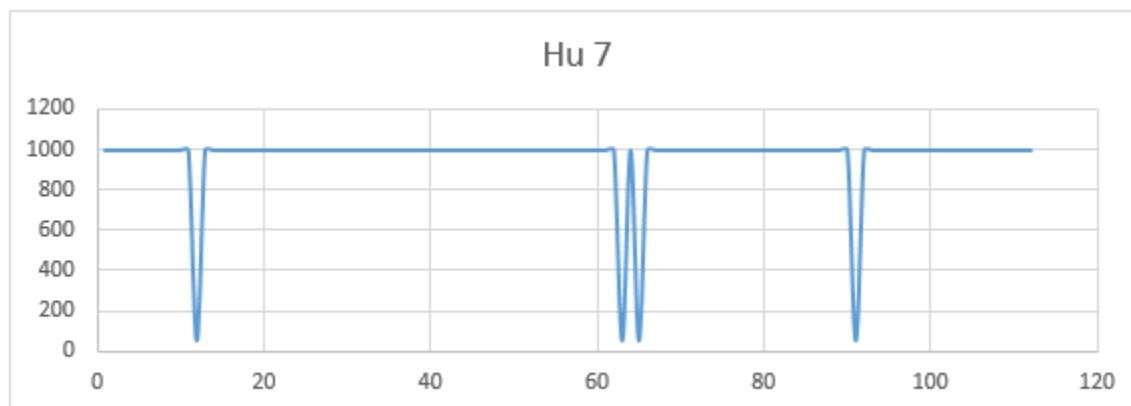


Tabla 4.12 HU7-arandela



En las tablas 4.13, 4.14, 4.15, 4.16, 4.17, 4.18, 4.19 se ven los momentos de Hu de ciento veinte y uno tornillos distribuidos a lo largo del área de trabajo del brazo robótico. El primer momento de Hu de los tornillos está en el rango de 2.5 y 2.75. El segundo momento de HU está en el rango de 5.05 y 5.5. El tercer momento de HU está entre 7 y 11. El cuarto momento de HU está entre 7 y 11. El quinto momento de HU está entre 14 y 22. El sexto momento de HU está entre 10 y 13. El último momento de HU no es un parámetro confiable.

Tabla 4.13 HU1-tornillo

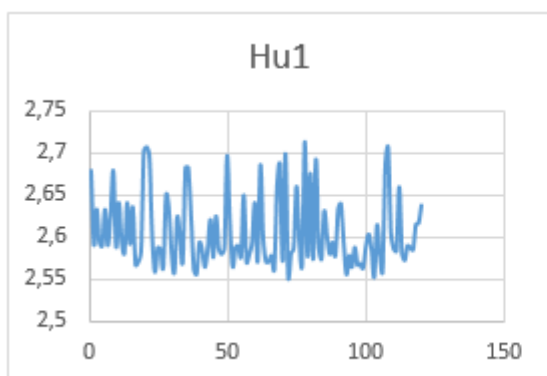


Tabla 4.14 HU2-tornillo

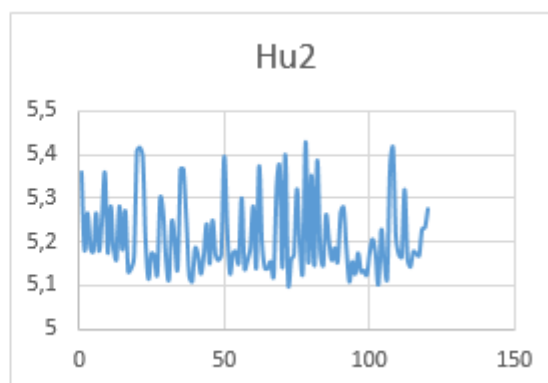


Tabla 4.15 HU3-tornillo

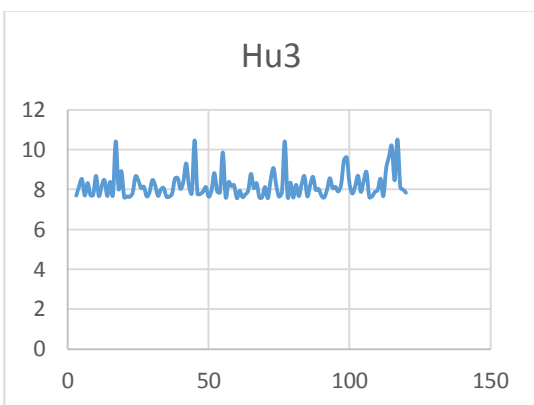


Tabla 4.16 HU4-tornillo

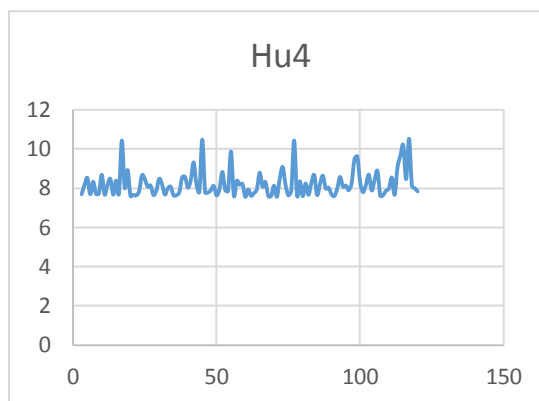


Tabla 4.17 HU5-tornillo

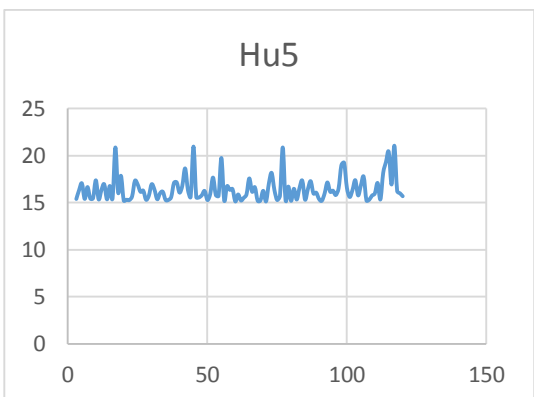


Tabla 4.18 HU6-tornillo

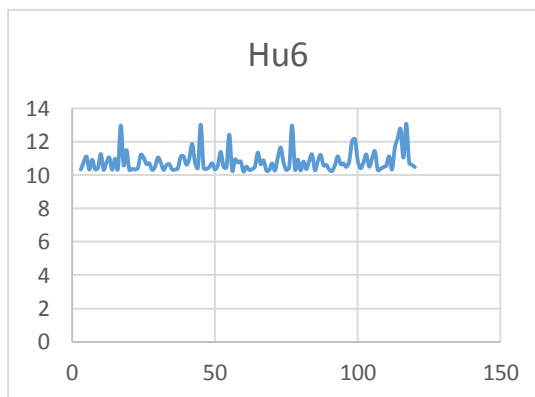
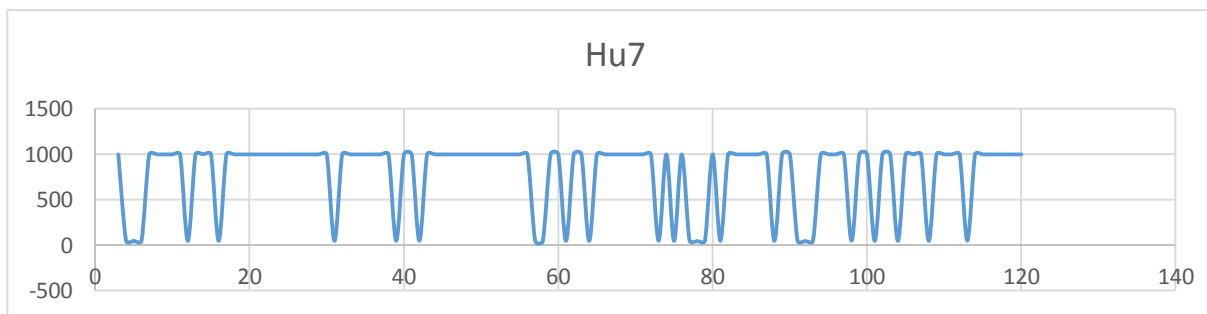


Tabla 4.19 HU7-tornillo



En las tablas 4.20, 4.21, 4.22, 4.23, 4.24, 4.25, 4.26 se ven los momentos de HU de ciento noventa tuercas distribuidos a lo largo del área de trabajo del brazo robótico. Se observa que los momentos de Hu son muy parecidos a los momentos de HU de las arandelas. Esto se debe a que son figuras muy parecidas aunque tengan diferentes áreas

Tabla 4.20 HU1-tuerca

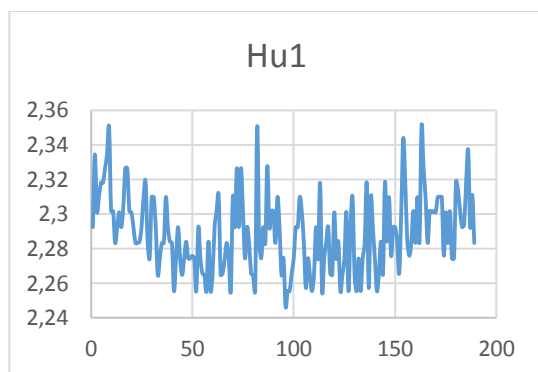


Tabla 4.21 HU2-tuerca

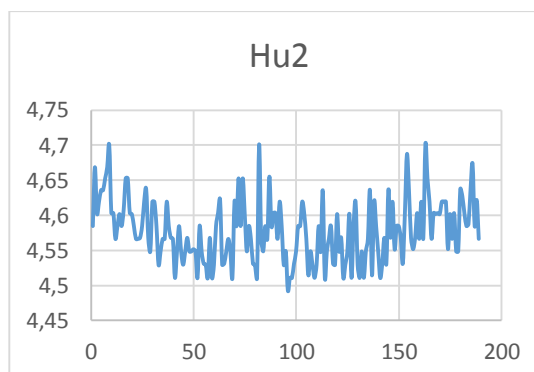


Tabla 4.22 HU3-tuerca

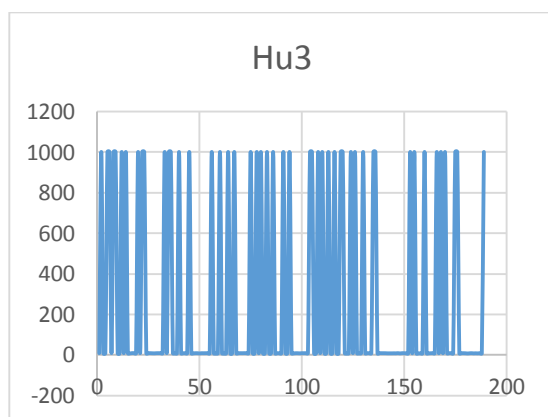


Tabla 4.23 HU4-tuerca

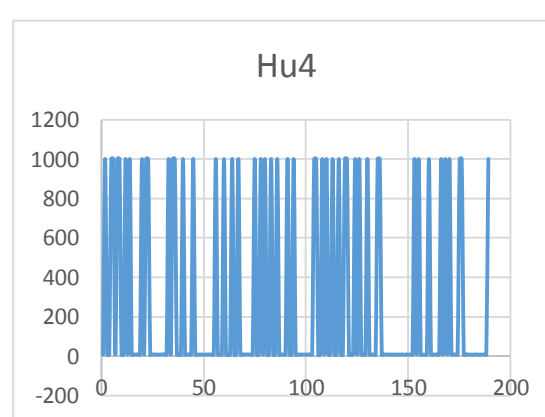


Tabla 4.24 HU5-tuerca

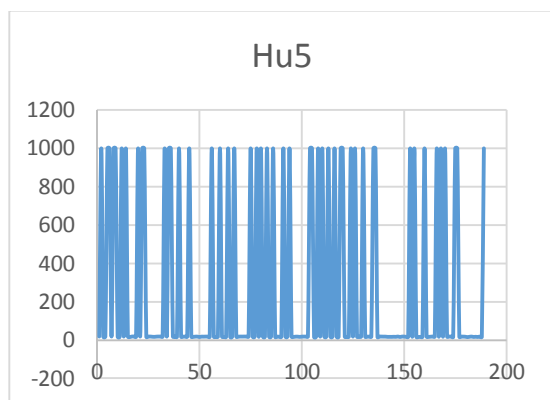


Tabla 4.25 HU6-tuerca

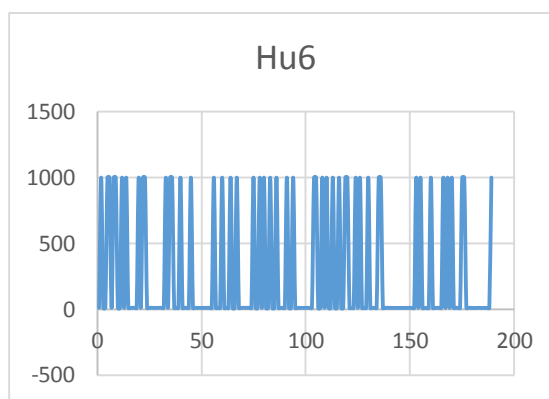
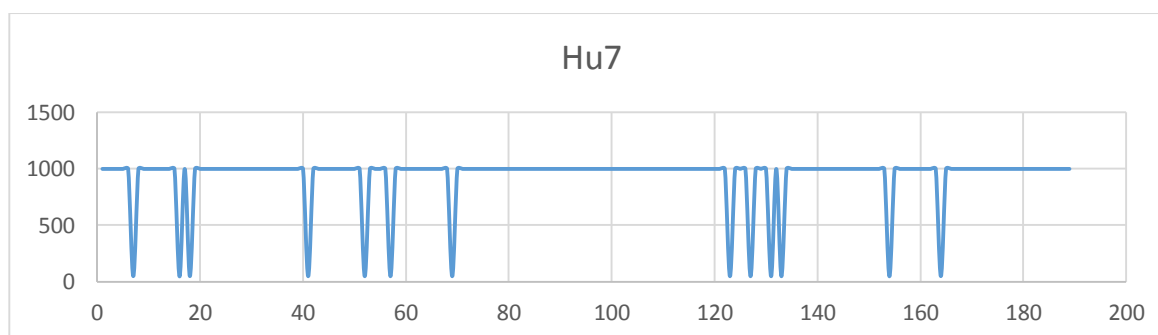


Tabla 4.26 HU7-tuerca



En la tabla 4.27 se resume el análisis de los momentos de HU y el área de las tuercas, tornillos y arandelas.

Tabla 4.27 Análisis momentos de HU

| objeto | Área | HU 1 | HU 2 | HU 3 | HU 4 | HU 5 | HU 6 | HU 7 |
|----------|------|--------|--------|---------|------|------|---------|---------|
| Tornillo | 200 | A 2.5 | A 5.05 | A | 7 | A 15 | A | |
| | 550 | 2.75 | 5.5 | 7 A 11 | 11 | 22 | 10 A 14 | 0 A Inf |
| | 1000 | A 2.53 | A 5.05 | A | 0 | A 0 | A | |
| arandela | 1700 | 5.9 | 5.2 | 0 A Inf | Inf | Inf | 0 A Inf | 0 A Inf |
| | 150 | A 2.24 | A 4.45 | A | 7 | A 15 | A | |
| Tuerca | 550 | 2.36 | 4.7 | 7 A 11 | 11 | 22 | 10 A 14 | 0 A Inf |

4.3 Algoritmo de reconocimiento de objetos

Teniendo en cuenta la tabla 4.27 se dedujo que las arandelas se pueden reconocer por su área mientras que los tornillos y tuercas se reconocen por los momentos de Hu, en especial por el primer momento. En la figura 4.7 se observa el diagrama de flujo que tiene el algoritmo de reconocimiento de imágenes y en el anexo 13 el código en Matlab.

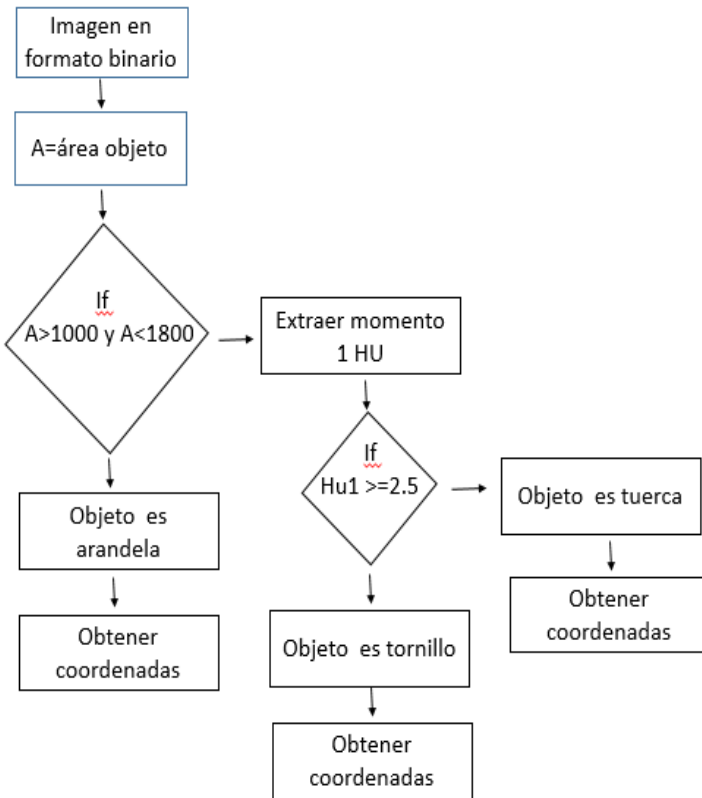


Figura 4.7 Algoritmo de reconocimiento de imágenes

4.4 Pruebas de tratamiento de imágenes

La tabla y la figura muestran el resultado del tratamiento de imagen. El robot solo pudo identificar 5 tuercas pero en el área de trabajo hay 6, esto se debe a que matlab no lo identifico como una imagen tratada.

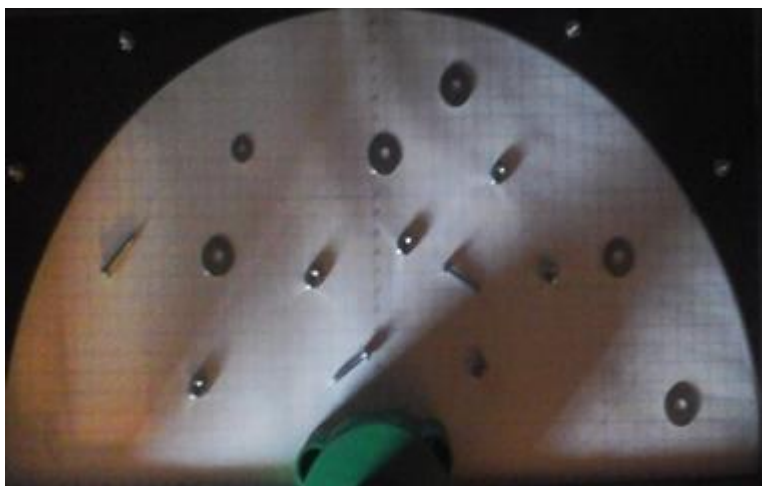


Figura 4.8 Prueba de tratamiento de imagen

Tabla 4.28 Estadística de tratamiento de imagen

| numero | Elemento | Área | hu1 | x | y |
|--------|-----------|------|------|----|----|
| 1 | Tornillo | 400 | 2.55 | 17 | 12 |
| 2 | Tornillo | 395 | 2.65 | 3 | 6 |
| 3 | Tornillo | 430 | 2.52 | 7 | 10 |
| 4 | Tuerca | 301 | 2.22 | 12 | 3 |
| 5 | Tuerca | 220 | 2.3 | -5 | 11 |
| 6 | Tuerca | 210 | 2.26 | 2 | 12 |
| 7 | Tuerca | 276 | 2.31 | 7 | 6 |
| 8 | Tuerca | 253 | 2.23 | 12 | 11 |
| 9 | Arandelas | 1100 | 2.56 | 10 | 12 |
| 10 | Arandelas | 1645 | 2.53 | 1 | 18 |
| 11 | Arandelas | 1390 | 2.45 | 5 | 21 |
| 12 | Arandelas | 1206 | 2.45 | 16 | 11 |
| 13 | Arandelas | 1520 | 2.49 | 20 | 3 |

Estadísticamente la tabla mostro una efectividad de 83.3 % en el algoritmo de tratamiento de imágenes.

CAPITULO 5. IMPLEMENTACION FINAL

Como último paso de este trabajo se integra el hardware y software realizados en capítulos anteriores para hacer las pruebas finales de funcionamiento. En la figura 5.1 se muestra el esquema general de la implementación del hardware requerido para este proyecto y en la figura 5.2 se muestra el esquema general del algoritmo que clasifica y orienta al robot a la posición de la pieza.

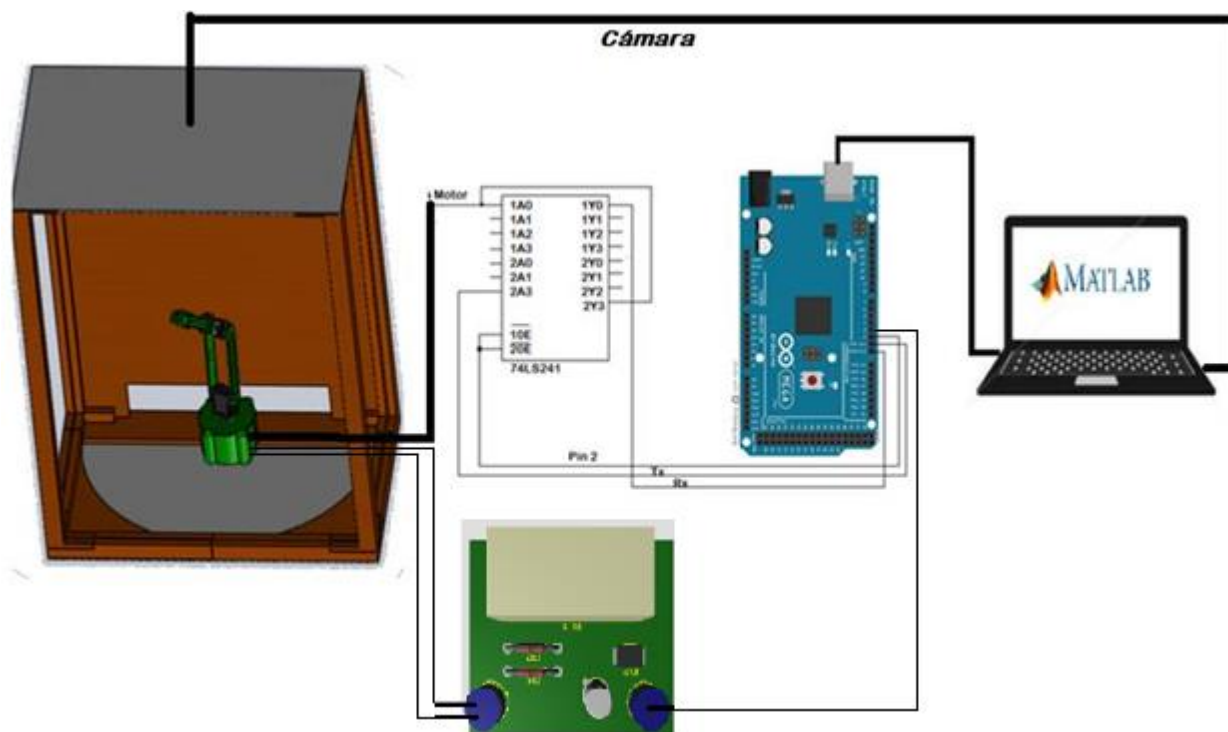


Figura 5.1 Implementación final

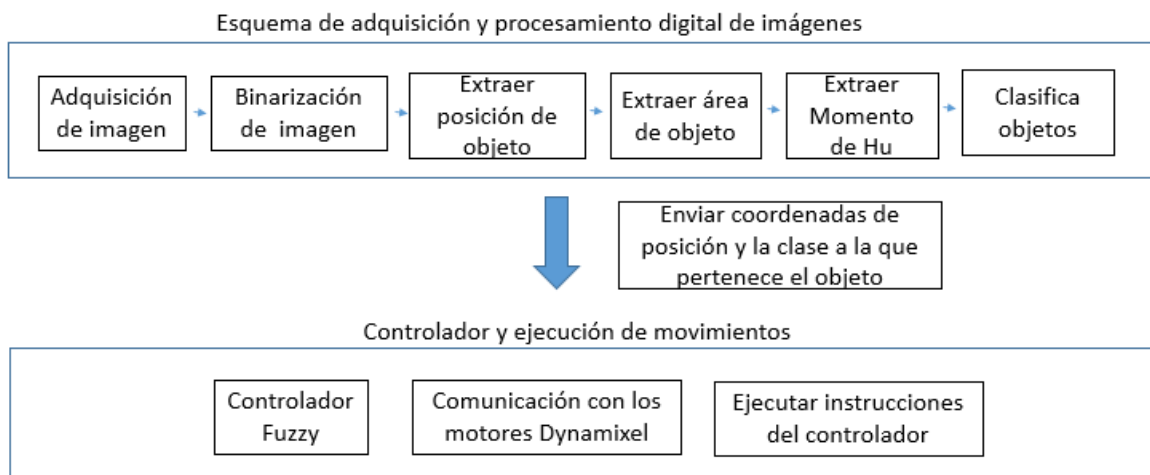


Figura 5.2 Esquema del algoritmo completo

5.1 Pruebas de funcionamiento

En la siguiente tabla 5.1, se muestra el resultado de las pruebas realizadas a 50 tornillos, 50 arandelas y 50 tuercas teniendo la implementación final de la planta. En la figura 5.3 se da un ejemplo de como se repartieron las piezas a través del área de trabajo. La efectividad total del brazo robotico es de 84.6%.

| Pieza | Clasificación efectiva | Descripción de los errores | Causas del error |
|-----------|------------------------|---|--|
| Arandelas | 92% | El robot toma el objeto y va al punto indicado para clasificar la arandela. Pero en el camino suelta la pieza | <ul style="list-style-type: none"> - Error 1 - Error 2 |
| Tornillos | 84% | - El 10% de los errores fueron causados porque el electroiman no se situaba encima del objeto | - Error 3 |
| | | - El 6% restantes se debe a que el electroiman suelta la pieza antes de ubicarlo en el sitio correspondiente para tornillos | <ul style="list-style-type: none"> - Error 1 - Error 2 |
| Tuercas | 78% | - El 8% de los errores fueron causados porque el electroiman no se situaba encima del objeto | - Error 3 |
| | | - el 4% de errores se debe a que no clasifico bien la pieza, el robot asumía que era un tornillo en vez de una pieza | <ul style="list-style-type: none"> - Error 4 - Error 5 |
| | | - El 10% restantes se debe a que el robot soltaba la pieza antes de ubicarla en la posición asignada para tuercas | <ul style="list-style-type: none"> - Error 1 - Error 2 |

Tabla 5.1 Pruebas finales y estadísticas de error

Error 1: El electroimán no generó suficiente atracción hacia la pieza. Esto sucede cuando las baterías están descargadas.

Error 2: Hubo una alteración de estado (de ON a OFF) producida en la tarjeta arduino. Estas alteraciones pueden producirse por malas conexiones y/o ruido de interferencia en la tarjeta Arduino.

Error 3: El controlador Fuzzy no generaba los ángulos de giro adecuados para cada motor

Error 4: La binarización de la imagen afectó el área o forma del objeto.

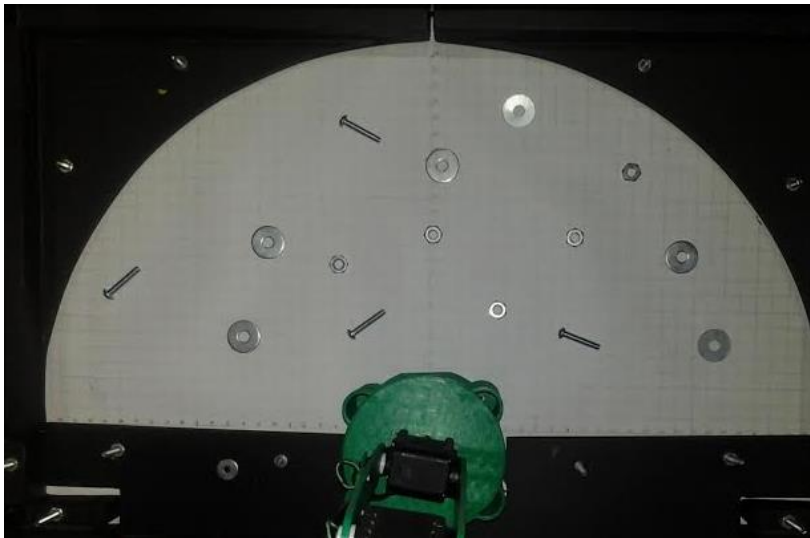


Figura 5.3 repartición de piezas en el área de trabajo

5.2 Inconvenientes y recomendaciones

Las pruebas de funcionamiento plasmaron inconvenientes como interferencia en la tarjeta Arduino, desajuste en la conexión de cables, descarga de baterías en el electroimán que afectaron su fuerza de atracción y fallas en el controlador Fuzzy al determinar los ángulos de giro de cada motor.

Estos inconvenientes se mejoran al supervisar continuamente las conexiones del brazo robótico con las tarjetas, con la cámara y con su alimentación. Igualmente al optimizar el controlador Fuzzy con reglas específicas, conseguir una cámara con resolución de 4 Mpx para arriba e integrar los momentos invariantes de HU en el algoritmo de tratamientos de imágenes; la calidad en la clasificación de objetos del brazo robótico aumenta.

CONCLUSIONES

1. La metodología de trabajo permite obtener resultados satisfactorios en el desarrollo de un proyecto. En este caso apporto orden y argumentación en cada avance de los objetivos planteados.
2. Los conceptos de cinemática inversa y directa y los cálculos matemáticos permitieron obtener parámetros importantes como la posición y el torque de cada una de las articulaciones del brazo robótico.
3. Este trabajo apporto a nuestra formación como Docentes en Electrónica habilidades de análisis en la solución de problemas, herramientas técnicas como el uso de Software de diseño y habilidades matemáticas que se unen a los conocimientos adquiridos en las áreas cursadas en la carrera.
4. Los resultados obtenidos en la simulación de la planta con la obtención de cinemática inversa fueron validados con los resultados obtenidos en la planta real. Lo que quiere decir que la obtención de cinemática inversa si resuelve el problema de posicionamiento del electroimán.

5. El controlador Fuzzy careció de precisión en algunas zonas del área de trabajo. Esto se debe a la falta de membresías y variables de entrada. Se puede mejorar el controlador integrando las entradas x , y ; y obteniendo los resultado de los ángulos de los motores requeridos para cada punto del área de trabajo. Esto trae como consecuencia más costo computacional pero mejora la precisión del controlador.
6. Los momentos de HU son parámetros decisivos del reconocimiento de objetos porque aportan, de forma numérica, información única e invariantes a traslación, rotación o cambio de escala, la forma de los objetos.

Bibliografía

(s.f.). Obtenido de <http://es.slideshare.net/mgarofalo85/robotica-2865379>

ABDALA SALOMON, R. Ñ. (2003). *Caracterizacion de un robot manipulador articulado*. Cuernavaca.

BARRIENTOS, A., Peñin, L., BALAGUER, C., & Aracil, R. (s.f.). *Fundamentos de la robotica*.

CASTILLO, S. A., & CABERTA, R. Ñ. (s.f.). CARACTERIZACIÓN DE UN ROBOT.

CORTÉS, F. R. (2011). *Robótica. Control de robots manipuladores*. México: Alfaomega Grupo Editor.

JARAMILLO, D. A. (2010). *Modelo simulacion y control de un brazo robotico mediante matlab y simulink para soldadura de arco*. Quito, Ecuador.

LEMUS, M. E. (s.f.). *Simulación de sistema difuso para el control de velocidad de un motor C.D.*

M., C. C. (1989). *Curso moderno de máquinas eléctricas rotativas*. Barcelona: Editores Técnicos Asociados .

MARTIN, M. (2002). *Descriptores de imagen*.

OGATA, K. (s.f.). *Ingeniería de control moderna*.

RAMIREZ, J. L. (2012). *Modelamiento matemático de la cinemática directa e inversa de un robot manipulador de tres grados de libertad*. Bogotá, Colombia.

ROBOTIS. (14 de Junio de 2006). Dynamixel AX-12.

ROMERO, A. A. (2012). *Diseño, impresion, montaje y control de un manipulador robotico*. Madrid.

SAHA, S. K. (2010). *Introduccion a la Robotica*. Mexico D.F.: Mc Graw Hill.

SPONG, M. W., & Vidyasagar, M. (1989). *Robot dynamics and control*. John Wiley & Sons.

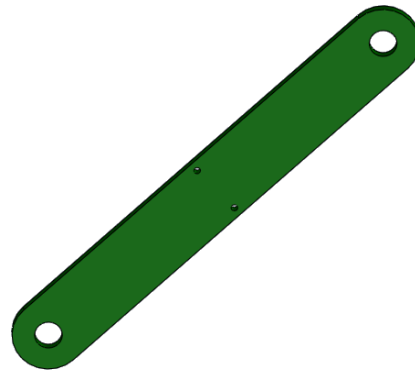
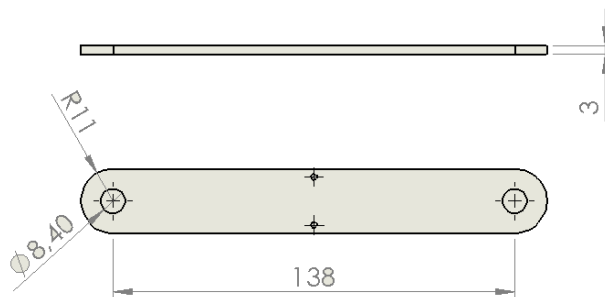
ANEXOS

Anexo No 1 Partes del brazo robótico con sus respectivas medidas

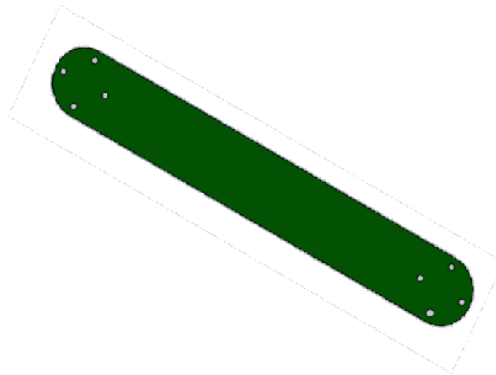
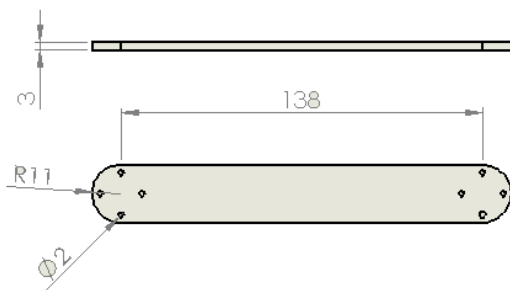
Piezas

BRAZO ROBOTICO

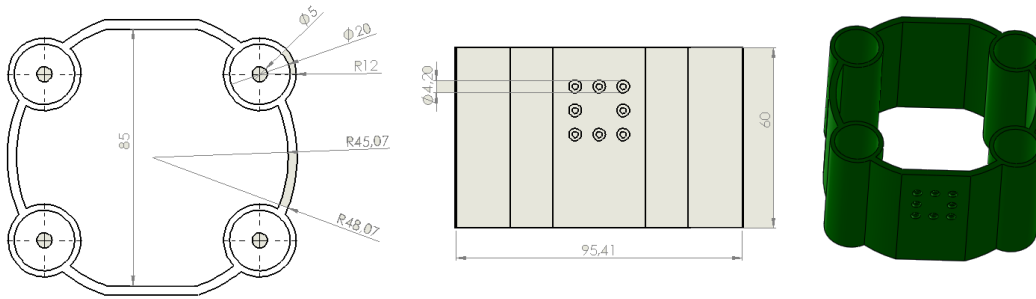
Pieza 1



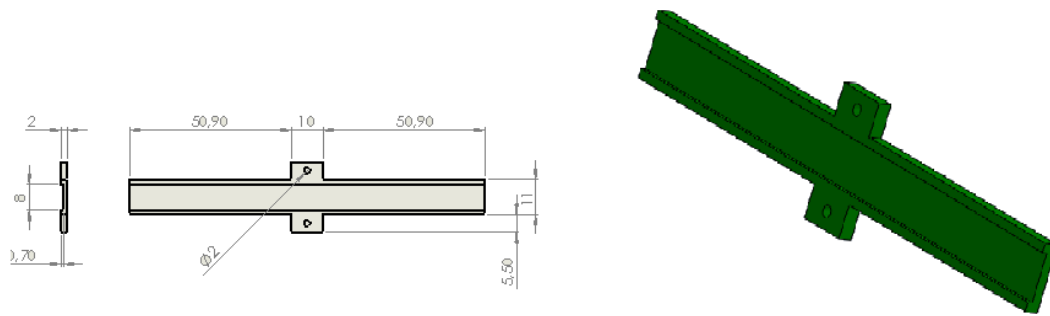
Pieza 2



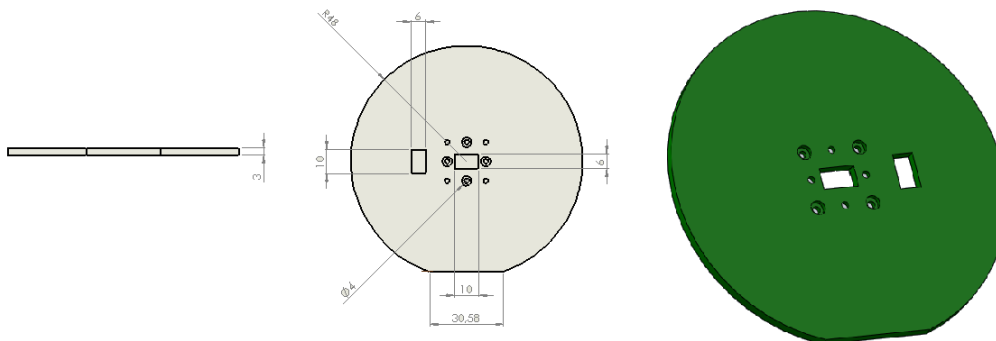
Pieza 3



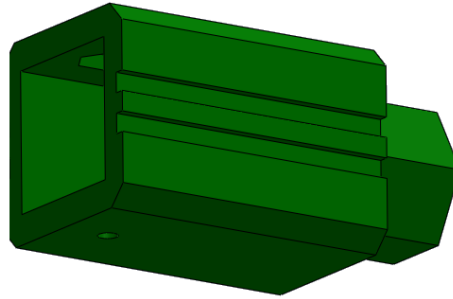
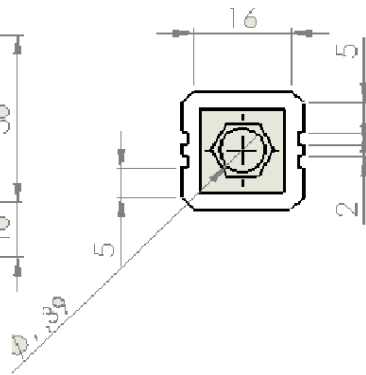
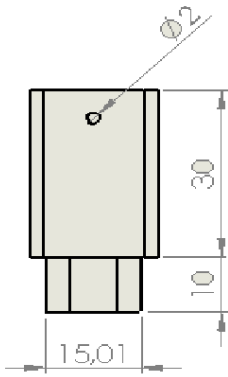
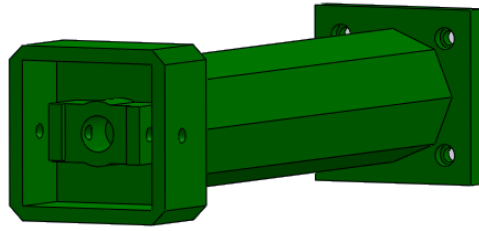
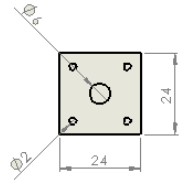
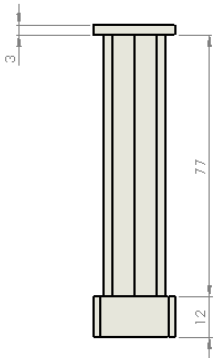
Pieza 4

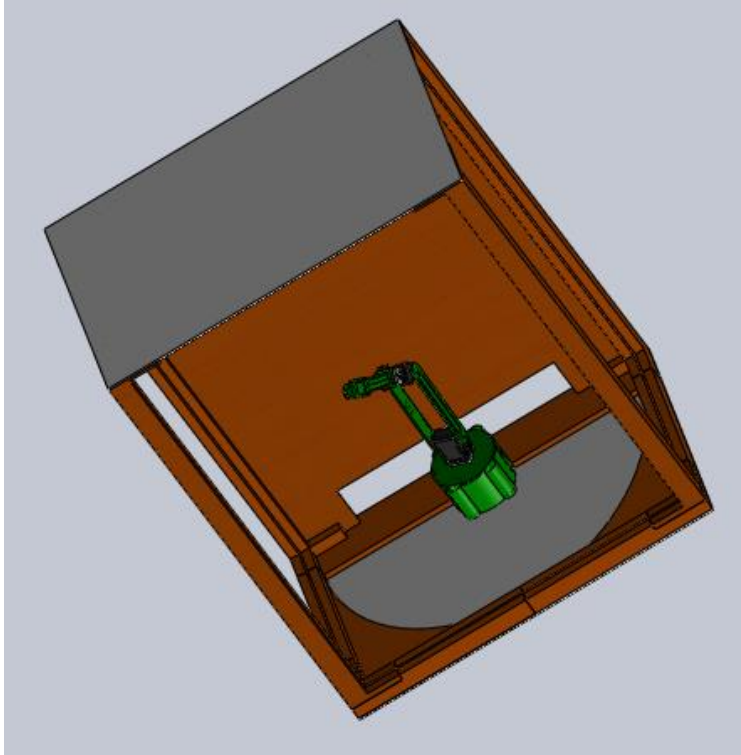


Pieza 5



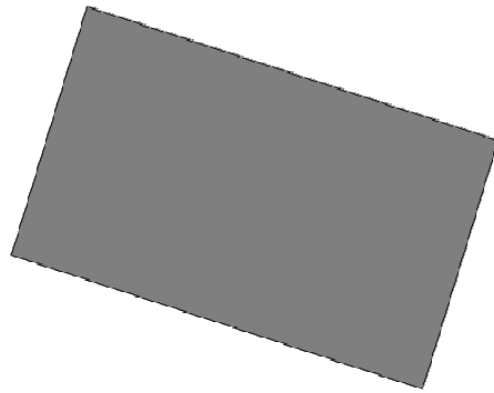
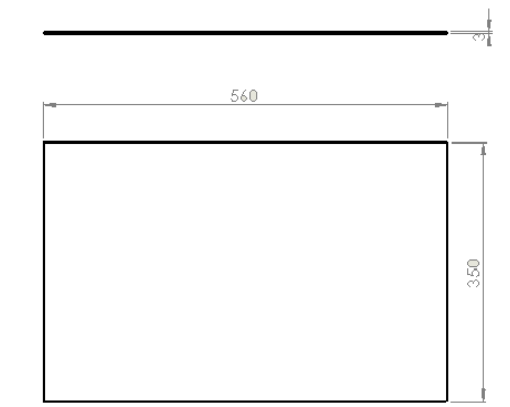
Pieza 6



Anexo No 2 Partes de la cabina con sus respectivas medidas

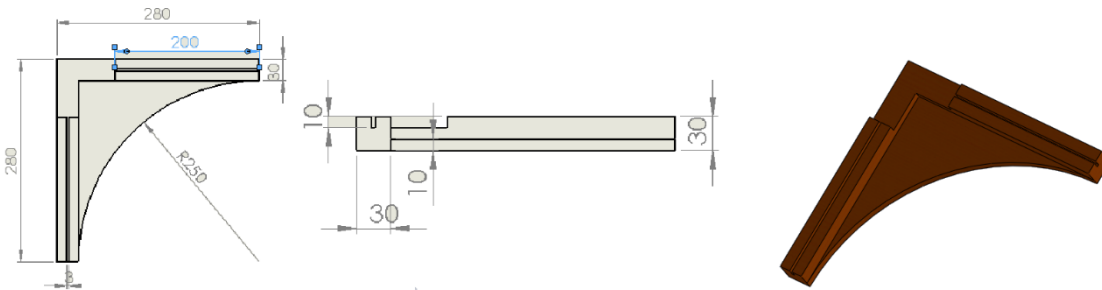
Pieza 1

Esta pieza se repite 2 veces



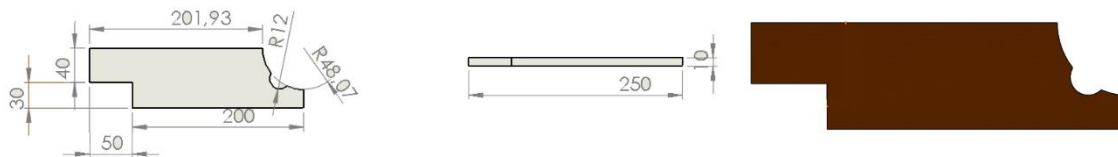
Pieza 2

Esta pieza se repite 2 veces



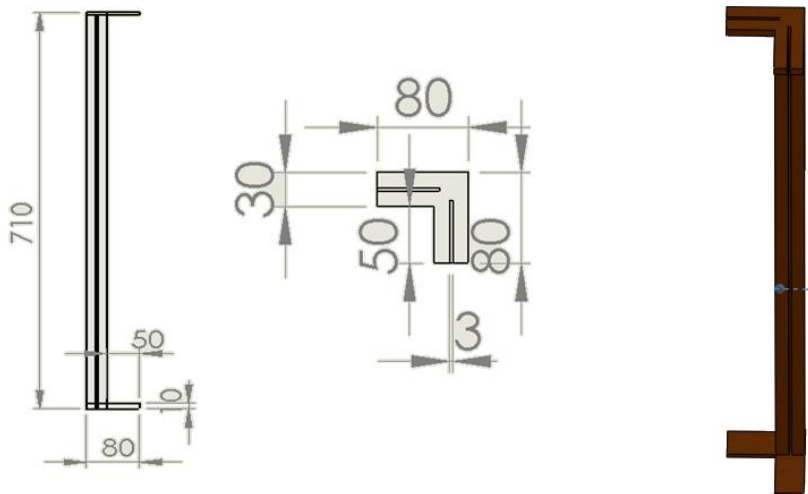
Pieza 3

Esta pieza se repite 2 veces



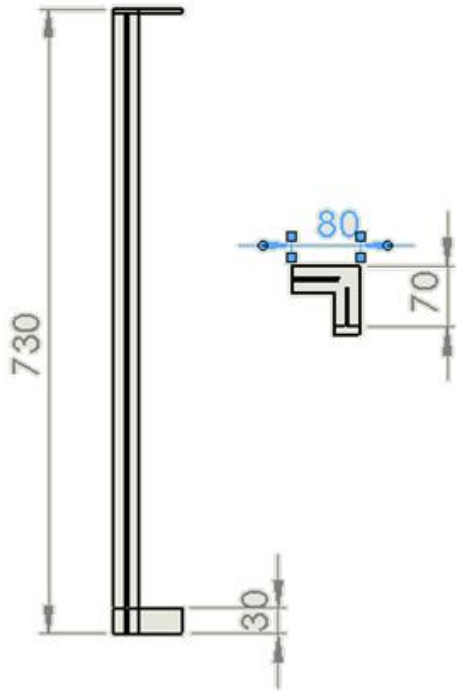
Pieza 4

Esta pieza se repite 2 veces



Pieza 5

Esta pieza se repite 2 veces



Anexo No 3. Tabla para realizar la comunicación entre los Dynamixel.

| | Address | Item | Access | Initial Value |
|-------------|-------------------|-------------------------------|--------|----------------|
| EEPROM Area | 0(0X00) | Model Number(L) | RD | 12(0x0C) |
| | 1(0X01) | Model Number(H) | RD | 0(0x00) |
| | 2(0X02) | Version of Firmware | RD | ? |
| | 3(0X03) | ID | RD,WR | 1(0x01) |
| | 4(0X04) | Baud Rate | RD,WR | 1(0x01) |
| | 5(0X05) | Return Delay Time | RD,WR | 250(0xFA) |
| | 6(0X06) | CW Angle Limit(L) | RD,WR | 0(0x00) |
| | 7(0X07) | CW Angle Limit(H) | RD,WR | 0(0x00) |
| | 8(0X08) | CCW Angle Limit(L) | RD,WR | 255(0xFF) |
| | 9(0X09) | CCW Angle Limit(H) | RD,WR | 3(0x03) |
| | 10(0x0A) | (Reserved) | - | 0(0x00) |
| | 11(0X0B) | the Highest Limit Temperature | RD,WR | 85(0x55) |
| | 12(0X0C) | the Lowest Limit Voltage | RD,WR | 60(0X3C) |
| | 13(0X0D) | the Highest Limit Voltage | RD,WR | 190(0xBE) |
| | 14(0X0E) | Max Torque(L) | RD,WR | 255(0xFF) |
| | 15(0X0F) | Max Torque(H) | RD,WR | 3(0x03) |
| | 16(0X10) | Status Return Level | RD,WR | 2(0x02) |
| | 17(0X11) | Alarm LED | RD,WR | 4(0x04) |
| | 18(0X12) | Alarm Shutdown | RD,WR | 4(0x04) |
| | 19(0X13) | (Reserved) | RD,WR | 0(0x00) |
| | 20(0X14) | Down Calibration(L) | RD | ? |
| | 21(0X15) | Down Calibration(H) | RD | ? |
| | 22(0X16) | Up Calibration(L) | RD | ? |
| 23(0X17) | Up Calibration(H) | RD | ? | |
| RAM Area | 24(0X18) | Torque Enable | RD,WR | 0(0x00) |
| | 25(0X19) | LED | RD,WR | 0(0x00) |
| | 26(0X1A) | CW Compliance Margin | RD,WR | 0(0x00) |
| | 27(0X1B) | CCW Compliance Margin | RD,WR | 0(0x00) |
| | 28(0X1C) | CW Compliance Slope | RD,WR | 32(0x20) |
| | 29(0X1D) | CCW Compliance Slope | RD,WR | 32(0x20) |
| | 30(0X1E) | Goal Position(L) | RD,WR | [Addr36]value |
| | 31(0X1F) | Goal Position(H) | RD,WR | [Addr37]value |
| | 32(0X20) | Moving Speed(L) | RD,WR | 0 |
| | 33(0X21) | Moving Speed(H) | RD,WR | 0 |
| | 34(0X22) | Torque Limit(L) | RD,WR | [Addr14] value |
| | 35(0X23) | Torque Limit(H) | RD,WR | [Addr15] value |
| | 36(0X24) | Present Position(L) | RD | ? |
| | 37(0X25) | Present Position(H) | RD | ? |
| | 38(0X26) | Present Speed(L) | RD | ? |
| | 39(0X27) | Present Speed(H) | RD | ? |
| | 40(0X28) | Present Load(L) | RD | ? |
| | 41(0X29) | Present Load(H) | RD | ? |
| | 42(0X2A) | Present Voltage | RD | ? |
| | 43(0X2B) | Present Temperature | RD | ? |
| | 44(0X2C) | Registered Instruction | RD,WR | 0(0x00) |
| | 45(0X2D) | (Reserved) | - | 0(0x00) |
| | 46(0x2E) | Moving | RD | 0(0x00) |
| | 47(0x2F) | Lock | RD,WR | 0(0x00) |
| | 48(0x30) | Punch(L) | RD,WR | 32(0x20) |
| | 49(0x31) | Punch(H) | RD,WR | 0(0x00) |

Anexo 4 Programación de Arduino para la comunicación entre los motores Dynamixel

```
#include <DynamixelSerial.h>
```

```
int data=0;
```

```
int data2=0;
```

```
int data1=0;
```

```
int data3=0;
```

```
int count=1;
```

```
float motor1=0;
```

```
float motor2=0;
```

```
float motor3=0;
```

```
int control=1;
```

```
void setup() {
```

```
pinMode(4,OUTPUT);
```

```
digitalWrite(ON,HIGH);
```

```
Serial.begin (1000000);
```

```
delay(10);
```

```
Dynamixel.torqueStatus ( 1 , ON );  
Dynamixel.torqueStatus ( 2 , ON );  
Dynamixel.torqueStatus ( 3 , ON );  
delay(10);
```

```
while (control==1){  
data=Serial.read();  
if (data>0){control=2;}  
while (control==2){  
data1=Serial.read();  
if (data1>0){control=3;}  
}  
while (control==3){  
data2=Serial.read();  
if (data2>0){control=4;}  
while (control==4){  
data2=Serial.read();  
if (data3>0){control=5;}  
}  
}
```

```
void loop() {  
if (data3==0){  
digitalWrite(4,LOW);
```

```
    }  
else  
{  
    digitalWrite(4,LOW);}
  
if (data>0 && data1>0 && data2>0){  
  
Dynamixel.moveSpeed(2,500,80);  
delay(400);  
Dynamixel.moveSpeed(3,1000,80);  
delay(2000);  
  
while(1) {  
  
motor1=round(3.333*data);  
motor2=round(3.333*data1 +195);  
motor3=round(3.333*data2 +195);  
Dynamixel.moveSpeed(1,motor1,80);  
delay(2000);  
Dynamixel.moveSpeed(3,motor3,80);  
delay(2000);
```

```

Dynamixel.moveSpeed(2,motor2,80);

delay(2000);

}}}

```

Anexo 5 Código en Matlab de la comunicación d los motores Dynamixel

```

function p=COMUNICACION(motor1,motor2,motor3)

motor1=motor1+1;

motor2=motor2+1;

motor3=motor3+1;

PS=serial('COM3');

set(PS,'Baudrate',1000000); % se configura la velocidad a 9600 Baudios

set(PS,'StopBits',1); % se configura bit de parada a uno

set(PS,'DataBits',8); % se configura que el dato es de 8 bits, debe estar entre 5 y 8

set(PS,'Parity','none'); % se configura sin paridad

set(PS,'Terminator','CR/LF');% “c” caracter con que finaliza el envío

set(PS,'OutputBufferSize',1); % ”n” es el número de bytes a enviar

set(PS,'InputBufferSize' ,1); % ”n” es el número de bytes a recibir

set(PS,'Timeout',5); % 5 segundos de tiempo de espera

fopen(PS);

```

```
pause(2)
fwrite(PS,motor1,'uint8');
pause(0.5);
fwrite(PS,motor2,'uint8');
pause(0.5);
fwrite(PS,motor3,'uint8');
fclose(PS);
delete(PS);
clear PS;
end
```

Anexo 6 Código de cinemática inversa

```

function [Q1,Q2,Q3]=CinematicaInversa(px,py,pz)

l1=105.5;

l2=138;

l3=162.5;

Q1=atan(py/px);

y=pz-(211/2);

x=(px*cos(Q1))+(py*sin(Q1));

a=(y^2+x^2-7362.25)/276;

b1=(-2*a*y);

a1=(2*a*y)^2-(4*(y^2+x^2)*(a^2-x^2))

c1=2*(y^2+x^2)

Q2=asin((b1+sqrt(a1))/(c1));

Q3=acos((pz*sin(Q2)- (211*sin(Q2)/2)+(px*cos(Q1)*cos(Q2))+(py*cos(Q2)*sin(Q2)) -

138)/162.5)

Q1=Q1*180/pi;

```

$Q2=Q2*180/\pi;$

$Q3=Q3*180/\pi;$

end

Anexo No 7 Membresía H

| Distancia en Cm | Función | Rango | Distancia en Cm | Función | Rango |
|-----------------|---------|----------------|-----------------|---------|----------------|
| 6 | Trim | [6 6.5 6.9] | 17 | Trim | [17 17.5 17.9] |
| 7 | Trim | [7 7.5 7.9] | 18 | Trim | [18 18.5 18.9] |
| 8 | Trim | [8 8.5 8.9] | 19 | Trim | [19 19.5 19.9] |
| 9 | Trim | [9 9.5 9.9] | 20 | Trim | [20 20.5 20.9] |
| 10 | Trim | [10 10.5 10.9] | 21 | Trim | [21 21.5 21.9] |
| 11 | Trim | [11 11.5 11.9] | 22 | Trim | [22 22.5 22.9] |
| 12 | Trim | [12 12.5 12.9] | 23 | Trim | [23 23.5 23.9] |
| 13 | Trim | [13 13.5 13.9] | 24 | Trim | [24 24.5 24.9] |
| 14 | Trim | [14 14.5 14.9] | 25 | Trim | [25 25.5 25.9] |
| 15 | Trim | [15 15.5 15.9] | 26 | Trim | [26 26.5 26.9] |
| 16 | Trim | [16 16.5 16.9] | | | |

Membresía tetha 2

| Distancia en Cm | Función | Rango |
|----------------------|---------|---------------------|
| 6 | Trapmf | [21 21.2 21.7 21.9] |
| 7 | Trapmf | [24 24.2 24.7 24.9] |
| 8,18,19 | Trapmf | [25 25.2 25.7 25.9] |
| 9,16 | Trapmf | [27 27.2 27.7 27.9] |
| 10,11,12,13,14,15,16 | Trapmf | [28 28.2 28.7 28.9] |
| 17 | Trapmf | [26 26.2 26.7 26.9] |
| 20 | Trapmf | [22 22.2 22.7 22.9] |
| 21 | Trapmf | [20 20.2 20.7 20.9] |
| 22 | Trapmf | [19 19.2 19.7 19.9] |
| 23 | Trapmf | [16 16.2 16.7 16.9] |
| 24 | Trapmf | [12 12.2 12.7 12.9] |
| 25 | Trapmf | [11 11.2 11.7 11.9] |
| mayor menor | Trapmf | [44 44.2 44.7 44.9] |

Anexo No 8. Valores para Tetha 3

| Distancia en Cm | Función | Rango | Distancia en Cm | Función | Rango |
|--------------------|---------|-----------------------------|--------------------|---------|----------------------------|
| 6 | Trapmf | [140 140.2 140.7 140.9] | 16 | Trapmf | [108 108.2 108.7 108.9] |
| 7 | Trapmf | [138 138.2 138.7 138.9] | 17 | Trapmf | [103 103.2 103.7 103.9] |
| 8 | Trapmf | [135 135.2 135.7 135.9] | 18 | Trapmf | [98 98.2 98.7 98.9] |
| 9 | Trapmf | [134 134.2 134.7 134.9] | 19 | Trapmf | [95 95.2 95.7 95.9] |
| 10 | Trapmf | [130 130.2 130.7 130.9] | 20 | Trapmf | [89 89.2 89.7 8.9] |
| 11 | Trapmf | [126 126.2 126.7 126.9] | 21 | Trapmf | [83 83.2 83.7 83.9] |
| 12 | Trapmf | [123 123.2 123.7 123.9] | 22 | Trapmf | [79 79.2 79.7 79.9] |
| 13 | Trapmf | [120 120.2 120.7 120.9] | 23 | Trapmf | [72 72.2 72.7 72.9] |
| 14 | Trapmf | [116 116.2 116.7 116.9] | 24 | Trapmf | [63 63.2 63.7 63.9] |
| 15 | Trapmf | [112 112.2 112.7 112.9] | 25 | Trapmf | [57 57.2 57.7 57.9] |

Anexo No 9 Ajuste de cámara

El software utilizado es matlab. Antes de poner en marcha el código es necesario la parte recta del semicírculo debe estar arraz con la parte inferior de la imagen. Además, se deben colocar tornillos en diferentes puntos y evaluar (con la cuadrícula que se implementa) si están en los puntos correctos.

```
% %%

clc,clear all;

imaqhwinfo

cam=imaqhwinfo;

cam.InstalledAdaptors

vid=videoinput('winvideo',1);

preview(vid); %visualizar camara

img=getsnapshot(vid);      %foto instantanea

im_g=rgb2gray(img);

umb=graythresh(im_g);

bn=~(im2bw(im_g,umb));

imshow(bn)

%%

[f c]=size(bn);

recorte=154; % cambie el valor y mire como se comporta bn.

% este valor debe colocarlo en la variable recorte del script completo

bn1=bn(recorte:f,:);
```

Anexo No 10 Código de área de las piezas

```
recorte=154;

Etiqueta_Area=0;

imaqhwinfo

cam=imaqhwinfo;

cam.InstalledAdaptors

vid=videoinput('winvideo',1);

img=getsnapshot(vid);

im_g=rgb2gray(img);

umb=graythresh(im_g)

bn=~(im2bw(im_g,umb));

[f c]=size(bn);

bn=bn(recorte:f,:);

imshow(bn)

%% sacar el area del arandela y su posicion

[Etiquetas, Num_Etiquetas]=bwlabel(bn,8);

Propiedades = regionprops(Etiquetas,'all');

i= 1;

% Seleccionamos las arandelas

for n=1:1:Num_Etiquetas

    if (Propiedades(n).Area>1000 && Propiedades(n).Area<1700)

        Etiqueta_Area(i) = n;

        i=i+1;
```

```
    end
end
% _____
[f c]=size(Etiqueta_Area)
for l=1:c
    arandela(l)=Propiedades(Etiqueta_Area(l)).Area;
end
Etiqueta_Area=0;
i= 1;
% seleccionamos tuerca y tornillos
for n=1:1:Num_Etiquetas
    if (Propiedades(n).Area>150 && Propiedades(n).Area<550)
        Etiqueta_Area(i) = n;
        i =i+1;
    end
end
end

[f c]=size(Etiqueta_Area)
for l=1:c
    tuercaytornillo(l)=Propiedades(Etiqueta_Area(l)).Area;
end
arandela
tuercaytornillo
```

Anexo 11 Momentos invariantes de HU (I)

```
clc

clear all

recorte=154;

Etiqueta_Area=0;

imaqhwinfo

cam=imaqhwinfo;

cam.InstalledAdaptors

vid=videoinput('winvideo',1);

img=getsnapshot(vid);      %foto instantanea

im_g=rgb2gray(img);

umb=graythresh(im_g)

bn=~(im2bw(im_g,umb));

[f c]=size(bn);

bn=bn(recorte:f,:);

imshow(bn)

[Etiquetas, Num_Etiquetas]=bwlabel(bn,8);

Propiedades = regionprops(Etiquetas,'all');

i= 1;

for n=1:1:Num_Etiquetas

    if (Propiedades(n).Area>100 && Propiedades(n).Area<2000)

        Etiqueta_Area(i) = n;

        i=i+1; end ;end
```

Anexo 12 Momentos invariantes de HU(II)

```

% guardamos en una variable toda la info de las arandelas

i=0;

[f c]=size(Etiqueta_Area);

for l=1:1:c

    Tornillo_Mediano(l,1)=Propiedades(Etiqueta_Area(l)).Area;

end

[f c]=size(Etiqueta_Area);

for n=1:1:c

    IMMomentos = Etiquetas == Etiqueta_Area(n);

    Propiedades1 = regionprops(IMMomentos,'BoundingBox');

    a = floor(Propiedades1.BoundingBox);

    N_Imagen = IMMomentos(a(2):(a(2)+a(4)),a(1):(a(1)+a(3)));

    phi = invmoments(size(N_Imagen));

    %escalado

    THu(n,:) = abs(log10(abs(phi)));

%   figure, imshow(N_Imagen)

end

[fi co]=size(THu);

% convertimos los Inf en 1000

for i=1:1:fi

    for j=1:1:co

        if(THu(i,j)==Inf)

```

```
    THu(i,j)=1000;
```

```
end
```

```
end
```

```
end
```

Anexo 13 Código Matlab de algoritmo d reconocimiento de objetos

```

function phi = invmoments(F)

F = double(F);

phi = compute_phi(compute_eta(compute_m(F)));

%-----%

function m = compute_m(F)

[M,N] = size(F);

[x,y] = meshgrid(1:N,1:M);

x = x(:);

y = y(:);

F = F(:);

m.m00 = sum(F);

if (m.m00 == 0)

    m.m00 = eps;

end

m.m10 = sum(x.*F);

m.m01 = sum(y.*F);

m.m11 = sum(x.*y.*F);

m.m20 = sum(x.^2.*F);

m.m02 = sum(y.^2.*F);

m.m30 = sum(x.^3.*F);

m.m03 = sum(y.^3.*F);

m.m12 = sum(x.*y.^2.*F);

```

```

m.m21 = sum(x.^2.*y.*F);

%-----%

function e = compute_eta(m)

xbar = m.m10 /m.m00;

ybar = m.m01 /m.m00;

e.eta11 = (m.m11 - ybar*m.m10)/m.m00^2;

e.eta20 = (m.m20 - xbar*m.m10)/m.m00^2;

e.eta02 = (m.m02 - ybar*m.m01)/m.m00^2;

e.eta30 = (m.m30 - 3*xbar*m.m20 + 2*xbar^2*m.m10)/m.m00^2.5;

e.eta03 = (m.m03 - 3*ybar*m.m02 + 2*ybar^2*m.m01)/m.m00^2.5;

e.eta21 = (m.m21 - 2*xbar*m.m11 - ybar*m.m20+2*xbar^2*m.m01)/m.m00^2.5;

e.eta12 = (m.m12 - 2*ybar*m.m11 - xbar*m.m02+2*ybar^2*m.m10)/m.m00^2.5;

%-----%

function phi = compute_phi (e)

phi(1) = e.eta20 + e.eta02;

phi(2) = (e.eta20 - e.eta02)^2 + 4*e.eta11^2;

phi(3) = (e.eta30 - 3*e.eta12)^2 + (3*e.eta21 - e.eta03)^2;

phi(4) = (e.eta30 + e.eta12)^2 + (e.eta21 + e.eta03)^2;

phi(5) = (e.eta30 - 3*e.eta12)*(e.eta30+e.eta12)*...

        ((e.eta30 + e.eta12)^2 - 3*(e.eta21 + e.eta03)^2)+...

        (3*e.eta21 - e.eta03)*(e.eta21 + e.eta03)*...

        (3*(e.eta30 + e.eta12)^2 - (e.eta21 + e.eta03)^2);

phi(6) = (e.eta20 - e.eta02)*((e.eta30+e.eta12)^2 - ...

```

$$(e.eta21 + e.eta03)^2 + \dots$$

4*e.eta11*(e.eta30 + e.eta12)*(e.eta21 + e.eta03); %diferente posicion de parentesis - No

afecta resultado

$$\text{phi}(7) = (3*e.eta21 - e.eta03)*(e.eta30 + e.eta12)*\dots$$

$$((e.eta30 + e.eta12)^2 - 3*(e.eta21 + e.eta03)^2) + \dots$$

$$(3*e.eta12 - e.eta30) * (e.eta21 + e.eta03) * \dots$$

(3*(e.eta30 + e.eta12)^2 - (e.eta21 + e.eta03)^2); %diferente posicion de parentesis - No

afecta resultado

13.2 reconocimiento de objetos y obtención de coordenadas

```
clc,clear all;
```

```
recorte=154;
```

```
Etiqueta_Area=0;
```

```
imaqhwinfo
```

```
cam=imaqhwinfo;
```

```
cam.InstalledAdaptors
```

```
vid=videoinput('winvideo',1);
```

```
img=getsnapshot(vid);
```

```
im_g=rgb2gray(img);
```

```
umb=graythresh(im_g)
```

```
bn=~(im2bw(im_g,umb));
```

```
[f c]=size(bn);
```

```
bn=bn(recorte:f,:);
```

```
imshow(bn)
```

```

%% sacar el area del arandela y su posicion
[Etiquetas, Num_Etiquetas]=bwlabel(bn,8);
Propiedades = regionprops(Etiquetas,'all');
i= 1;
% Seleccionamos las arandelas
for n=1:1:Num_Etiquetas
    if (Propiedades(n).Area>1000 && Propiedades(n).Area<1700)
        Etiqueta_Area(i) = n;
        i =i+1;
    end
end
% extraemos posicion de las arandelas
[f_Etiqueta_Area c_Etiqueta_Area]=size(Etiqueta_Area);
for l=1:1:c_Etiqueta_Area
    arandela(l,:)=round((Propiedades(Etiqueta_Area(l)).Centroid)*4/88);
end
[f_arandela c_arandela]=size(arandela);
for y=1:1: f_arandela
    if arandela(y,1)<21
        arandela(y,1)=(-21+arandela(y,1));
    elseif arandela(y,1)==21
        arandela(y,1)=0;
    else

```

```

    arandela(y,1)= arandela(y,1)-21;
end

arandela(y,2)=25-arandela(y,2);

end

%_____

[f c]=size(Etiqueta_Area)

for l=1:c

arandela(l)=Propiedades(Etiqueta_Area(l)).Area;

end

Etiqueta_Area=0;

i= 1;

% seleccionamos tuerca y tornillos

for n=1:1:Num_Etiquetas

if (Propiedades(n).Area>150 && Propiedades(n).Area<550)

    Etiqueta_Area(i) = n;

    i =i+1;

end

end

[f c]=size(Etiqueta_Area)

for l=1:c

tuercaytornillo(l)=Propiedades(Etiqueta_Area(l)).Area;

end

[f c]=size(Etiqueta_Area);

```

```

for n=1:1:c
    IMMomentos = Etiquetas == Etiqueta_Area(n);
    Propiedades1 = regionprops(IMMomentos,'BoundingBox');
    a = floor(Propiedades1.BoundingBox);
    N_Imagen = IMMomentos(a(2):(a(2)+a(4)),a(1):(a(1)+a(3)));
    phi = invmoments(size(N_Imagen));
    THu(n,:) = abs(log10(abs(phi)));
end

mi=1;
mo=1;

% clasificamos lo que es tuerca y lo que es tornillo
[fi co]=size(THu);
for g=1:1:fi
    if (Thu(fi,1)<2.4)
        tuerca(mi,:)=round((Propiedades(Etiqueta_Area(1)).Centroid)*4/88);
        mi=mi+1;
    else
        tornillo(mo,:)=round((Propiedades(Etiqueta_Area(1)).Centroid)*4/88);
    end
end

% extraemos posicion tuercas y tornillos
[f_tuerca c_tuerca]=size(tuerca);
for y=1:1: f_tuerca

```

```
if tuerca(y,1)<21
    tuerca(y,1)=(-21+tuerca(y,1));
elseif tuerca(y,1)==21
    tuerca(y,1)=0;
else
    tuerca(y,1)= tuerca(y,1)-21;
end

tuerca(y,2)=25-tuerca(y,2);
end

[f_tornillo c_tornillo]=size(tornillo);
for y=1:1: f_tuerca
    if tornillo(y,1)<21
        tornillo(y,1)=(-21+tornillo(y,1));
    elseif tornillo(y,1)==21
        tornillo(y,1)=0;
    else
        tornillo(y,1)=tornillo(y,1)-21;
    end

    tornillo(y,2)=25-tornillo(y,2);
end

% _____

clc,clear all;
```

```
% COMUNICACION(128,90,90);

% pause(5)

recorte=154;

Etiqueta_Area=0;

imaqhwinfo

cam=imaqhwinfo;

cam.InstalledAdaptors

vid=videoinput('winvideo',1);

img=getsnapshot(vid);

im_g=rgb2gray(img);

umb=graythresh(im_g)

bn=~(im2bw(im_g,umb));

[f c]=size(bn);

bn=bn(recorte:f,:);

imshow(bn)

%% sacar el area del arandela y su posicion

[Etiquetas, Num_Etiquetas]=bwlabel(bn,8);

Propiedades = regionprops(Etiquetas,'all');

i= 1;

% Seleccionamos las arandelas

for n=1:1:Num_Etiquetas

    if (Propiedades(n).Area>1000 && Propiedades(n).Area<1700)

        Etiqueta_Area(i) = n;
```

```

        i=i+1;
    end
end
% extraemos posición de las arandelas
[f_Etiqueta_Area c_Etiqueta_Area]=size(Etiqueta_Area);
for l=1:1:c_Etiqueta_Area
    arandela(l,:)=round((Propiedades(Etiqueta_Area(l)).Centroid)*4/88);
end
% [f_arandela c_arandela]=size(arandela);
for y=1:1: f_arandela
    if arandela(y,1)<21
        arandela(y,1)=(-21+arandela(y,1));
    elseif arandela(y,1)==21
        arandela(y,1)=0;
    else
        arandela(y,1)= arandela(y,1)-21;
    end
    arandela(y,2)=25-arandela(y,2);
end
% evaluamos con el controlador fuzzy
fuzzy_c=readfis('fuzzy_control');
for r=1:1:f_arandela
    [Q1 xa]=Pre_fuzzy(arandela(r,1),arandela(r,2));

```

```

Q23=evalfis(xa,fuzzy_c);
Q1=round(Q1);
%enviamos informacion a los motores Dynamixel
COMUNICACION(Q1+38,round(Q23(1,1)),round(Q23(1,2))+92);
pause(10);
COMUNICACION(Q1+38,90,90);
    pause(6);
COMUNICACION(238,70,210); % posicion de las arandelas clasificadas
pause(10);
end
% _____
[f c]=size(Etiqueta_Area)
for l=1:c
    arandela(l)=Propiedades(Etiqueta_Area(l)).Area;
end
Etiqueta_Area=0;
i= 1;
% seleccionamos tuerca y tornillos
for n=1:1:Num_Etiquetas
    if (Propiedades(n).Area>150 && Propiedades(n).Area<550)
        Etiqueta_Area(i) = n;
        i =i+1;
    end
end

```

```

end

[f c]=size(Etiqueta_Area)

for l=1:c

tuercaytornillo(l)=Propiedades(Etiqueta_Area(l)).Area;

end

[f c]=size(Etiqueta_Area);

for n=1:1:c

    IMMomentos = Etiquetas == Etiqueta_Area(n);

    Propiedades1 = regionprops(IMMomentos,'BoundingBox');

    a = floor(Propiedades1.BoundingBox);

    N_Imagen = IMMomentos(a(2):(a(2)+a(4)),a(1):(a(1)+a(3)));

    phi = invmoments(size(N_Imagen));

    THu(n,:) = abs(log10(abs(phi)));

end

mi=1;

mo=1;

% clasificamos lo que es tuerca y lo que es tornillo

[fi co]=size(THu);

for g=1:1:fi

if (Thu(fi,1)<2.4)

tuerca(mi,:)=round((Propiedades(Etiqueta_Area(l)).Centroid)*4/88);

mi=mi+1;

else

```

```
tornillo(mo,:)=round((Propiedades(Etiqueta_Area(l)).Centroid)*4/88);

end

end

% extraemos posicion tuercas y tornillos

[f_tuerca c_tuerca]=size(tuerca);

for y=1:1: f_tuerca

    if tuerca(y,1)<21

        tuerca(y,1)=(-21+tuerca(y,1));

    elseif tuerca(y,1)==21

        tuerca(y,1)=0;

    else

        tuerca(y,1)= tuerca(y,1)-21;

    end

    tuerca(y,2)=25-tuerca(y,2);

end

[f_tornillo c_tornillo]=size(tornillo);

for y=1:1: f_tuerca

    if tornillo(y,1)<21

        tornillo(y,1)=(-21+tornillo(y,1));

    elseif tornillo(y,1)==21

        tornillo(y,1)=0;

    else

        tornillo(y,1)=tornillo(y,1)-21;
```

```

    end

    tornillo(y,2)=25-tornillo(y,2);

end

%_____

% evaluamos con el controlador fuzzy

fuzzy_c=readfis('fuzzy_control');

for r=1:1:f_arandela

    [Q1 xa]=Pre_fuzzy(tornillo(r,1),tornillo(r,2));

    Q23=evalfis(xa,fuzzy_c);

    Q1=round(Q1);

    %enviamos informacion a los motores Dynamixel

    COMUNICACION(Q1+38,round(Q23(1,1)),round(Q23(1,2))+92);

    pause(10);

    COMUNICACION(Q1+38,90,90);

    pause(6);

    COMUNICACION(270,70,210); % posicion de las arandelas clasificadas

    pause(10);

end

% evaluamos con el controlador fuzzy

fuzzy_c=readfis('fuzzy_control');

for r=1:1:f_tuerca

    [Q1 xa]=Pre_fuzzy(tuerca(r,1),tuerca(r,2));

    Q23=evalfis(xa,fuzzy_c);

```

```
Q1=round(Q1);  
%enviamos informacion a los motores Dynamixel  
COMUNICACION(Q1+38,round(Q23(1,1)),round(Q23(1,2))+92);  
pause(10);  
COMUNICACION(Q1+38,90,90);  
    pause(6);  
COMUNICACION(238,70,210); % posicion de las arandelas clasificadas  
pause(10);  
end  
COMUNICACION(0,90,270);
```