

**DISEÑO E IMPLEMENTACIÓN DE UNA CELDA ROBOTIZADA
CLASIFICADORA UTILIZANDO UN ROBOT DELTA 3RSS Y UN SISTEMA
DE VISIÓN ARTIFICIAL**

JAVIER ALFONSO ZAPATA MONTOYA

**UNIVERSIDAD DE PAMPLONA
FACULTAD DE INGENIERÍAS Y ARQUITECTURA
DEPARTAMENTO DE INGENIERÍA MECÁNICA, MECATRONICA E
INDUSTRIAL
INGENIERÍA MECATRÓNICA
PAMPLONA N.S. COLOMBIA
2019**

**DISEÑO E IMPLEMENTACIÓN DE UNA CELDA ROBOTIZADA
CLASIFICADORA UTILIZANDO UN ROBOT DELTA 3RSS Y UN SISTEMA
DE VISIÓN ARTIFICIAL**

JAVIER ALFONSO ZAPATA MONTOYA

Cód: 1094279075

Tel: 3007601673

Correo electrónico: Javier.alfonso28@hotmail.com

**Trabajo de grado presentado como requisito para optar por el título de
ingeniero en mecatrónica**

Director: CÉSAR AUGUSTO PEÑA CORTÉS

Ph.D. AUTOMÁTICA Y ROBÓTICA

Correo electrónico: cesarapc@unipamplona.edu.co

**UNIVERSIDAD DE PAMPLONA
FACULTAD DE INGENIERÍAS Y ARQUITECTURA
DEPARTAMENTO DE INGENIERÍA MECÁNICA, MECATRONICA E
INDUSTRIAL
INGENIERÍA MECATRÓNICA
PAMPLONA N.S. COLOMBIA**

2019

Dedicatoria

Primeramente a Dios, por estar siempre a mi lado.

A mis padres y toda mi familia por todo el apoyo y ánimo que me han brindado para llegar hasta acá.

A mis amigos, compañeros y docentes.

AGRADECIMIENTOS

A mi tutor CESAR AUGUSTO PEÑA por el apoyo, la dedicación y disponibilidad que tuvo lograr mi objetivo.

En general a todos mis docentes por compartir su conocimiento y experiencia para lograr el resultado final.

RESUMEN

Las células robotizadas se han convertido en una parte importante de cualquier industria, estas han hecho mucho más fácil y rápido el trabajo industrial como pintar, soldar o paletizar, teniendo un muy papel importante en el desarrollo. Este tipo de implementación tienden a ser muy costosos, por esta razón las instituciones no siempre cuentan con este tipo de instalaciones para preparar a sus estudiantes en este campo de aplicación.

En este proyecto se presenta de forma general la planeación, diseño, construcción y desarrollo de una celda robotizada implementando un proceso pick and place, la cual es capaz de detectar objetos que se encuentran en movimiento por una banda transportadora, a través de un sistema de visión artificial, que serán sujetadas por un robot paralelo tipo delta.

Inicialmente se diseñó la estructura física de la celda junto con su respectivo sistema electrónico y eléctrico, teniendo en cuenta parámetros de seguridad como los interruptores de seguridad, barreras materiales. Se procedió al diseño y construcción de las bandas transportadora y por último se procedió al rediseño de un robot paralelo tipo delta, implementando el análisis cinemático y un algoritmo de optimización con el fin de encontrar las medidas más óptimas de sus eslabones para posteriormente realizar simulaciones del robot verificando su alcance, teniendo en cuenta sus respectivas restricciones como lo son mecánicas y puntos singulares.

Se desarrolló un sistema de visión artificial con el objetivo de reconocer y localizar los objetos o piezas presentes en la banda implantando un algoritmo detectando los círculos por medio de la transformada de Hough. Y un sistema de control cinemático encargado de la planeación y ejecución de trayectorias para que el robot pueda trasladarse a las posiciones deseadas en un tiempo dado.

Por último se sincronizaron los sistemas con el fin de ejecutar la tarea, localizando y trasladando los objetos que se encuentran en movimiento.

ABSTRACT

Robotized cells have become an important part of any industry, they have made industrial work much easier and faster like painting, welding or palletizing, having a very important role in development. This type of implementation depends on being very expensive, for this reason the institutions do not always have this type of facilities to prepare their students in this field of application.

In this project, the planning, design, construction and development of a robotic cell is presented in a general way, implementing a pick and place process, the quality is able to detect objects that are moving through a conveyor belt, through a system of artificial vision, which will be held by a parallel delta robot.

Initially, the physical structure of the cell was designed together with its respective electronic and electrical system, taking into account safety parameters such as safety switches, material barriers. The design and construction of the conveyor belts was carried out and finally a parallel delta-type robot was redesigned, implementing the kinematic analysis and an optimization algorithm in order to find the most optimal measurements of its links and recently performed simulations of the robot to verify its scope, taking into account their respective restrictions such as mechanical and singular points.

Verify an artificial vision system with the aim of recognizing and locating the objects or pieces present in the band by implanting an algorithm detected the circles through the transformation of Hough. And a kinematic control system in charge of the planning and execution of trajectories so that the robot can move to the desired positions in a given time.

Finally, the systems were synchronized in order to execute the task, locating and moving the objects that are in motion.

CONTENIDO

| | |
|---|----|
| 1. PRELIMINARES..... | 14 |
| 1.1 DESCRIPCIÓN DEL PROBLEMA..... | 14 |
| 1.2 JUSTIFICACIÓN | 14 |
| 1.3 SOLUCIÓN AL POBLEMA..... | 15 |
| 1.4 OBJETIVOS..... | 15 |
| 1.4.1 Objetivo genera..... | 15 |
| 1.4.2 Objetivos específicos..... | 16 |
| 2. MARCO TEÓRICO Y ESTADO DEL ARTE..... | 17 |
| 2.1 ROBOTS PARALELOS..... | 17 |
| 2.1.1 Robot Manipulador tipo Delta..... | 17 |
| 2.1.2 Aplicaciones..... | 18 |
| 2.1.2.1 Aplicaciones médicas | 18 |
| 2.1.2.2 Aplicaciones industriales..... | 19 |
| 2.2 CINÉMÁTICA..... | 20 |
| 2.3 VISIÓN ARTIFICIAL..... | 21 |
| 2.3.1 Métodos de captación de las imágenes digital..... | 22 |
| 2.4 TRANSFORMADA DE HOUGH..... | 23 |
| 2.5 DISEÑO Y CONTROL DE UNA CÉLULA ROBOTIZADA..... | 25 |
| 2.5.1 Características del sistema de control de la célula de trabajo..... | 25 |
| 2.5.2 Medidas de seguridad a tomar en la fase de diseño de la célula robotizada..... | 26 |
| 2.5.3 Normas internacionales..... | 27 |
| 3. DISEÑO CELDA ROBOTIZADA..... | 29 |
| 3.1 FUNCIONAMIENTO..... | 29 |
| 3.2 DISEÑO DE BANDAS..... | 30 |
| 3.3 DISEÑO ESTRUCTURA..... | 33 |
| 3.3.1 Barreras materiales..... | 34 |

| | | |
|---------|--|----|
| 3.3.2 | Interruptores de seguridad..... | 34 |
| 3.4 | SISTEMA ELECTRÓNICO..... | 37 |
| 3.4.1 | Elementos Electrónicos..... | 37 |
| 3.5 | DISEÑO ROBOT DELTA..... | 40 |
| 3.5.1 | Cinemática inversa | 40 |
| 3.5.2 | Espacio de trabajo del robot..... | 42 |
| 3.5.2.1 | Limitantes mecánicas..... | 42 |
| 3.5.2.2 | Optimización de tamaño del robot..... | 44 |
| 3.5.2.3 | Optimización por enjambres de partículas..... | 44 |
| 3.5.3 | Diseño articulaciones | 46 |
| 3.5.3.1 | Eslabón 1..... | 46 |
| 3.5.3.2 | Eslabon 2..... | 47 |
| 3.5.4 | Construcción delta..... | 48 |
| 3.5.5 | Efector final..... | 50 |
| 3.5.6 | Actuadores robot delta..... | 51 |
| 3.5.7 | USB2Dynamixel..... | 53 |
| 3.6 | CONTROL CINEMÁTICO ROBOT..... | 53 |
| 3.6.1 | Planeación trayectoria..... | 55 |
| 3.7 | DESARROLLO DEL SISTEMA DE VISIÓN ARTIFICIAL..... | 60 |
| 3.7.1 | Procesamiento de imagen | 60 |
| 3.8 | ACOPLAMIENTO DE SISTEMA..... | 65 |
| 4 | RESULTADOS..... | 67 |
| 4.1 | PRUEBAS DE VELOCIDAD..... | 67 |
| 4.2 | CALIBRACIÓN MOTORES..... | 68 |
| 4.3 | PRUEBAS DE SUJETACION DE OBJETOS..... | 68 |
| 4.4 | TIEMPOS DE EJECUCIÓN DE FUNCIONES..... | 73 |
| 4.5 | Máximo de piezas alcanzadas en un minuto..... | 73 |
| | RECOMENDACIONES..... | 77 |
| | CONCLUSIONES..... | 77 |
| | RERENCIAS..... | 79 |
| | ANEXOS..... | 81 |

LISTADO DE TABLAS

| | |
|---|----|
| Tabla 1 Medidas optimas..... | 45 |
| Tabla 2 Parámetros cámara..... | 59 |
| Tabla 2 Datos de tiempos de ejecución vs estimados..... | 71 |
| Tabla 3..... | 73 |
| Tabla 4 Tiempo de ejecución de funciones..... | 73 |

LISTADO DE FIGURAS

| | |
|--|----|
| Figura 1 Robot Manipulador Paralelo tipo Delta. | 18 |
| Figura 2 Corte de tres esferas | 18 |
| Figura 3 SurgiScope. Robot Delta en aplicaciones médicas. | 20 |
| Figura 4 IRB 340. Robot con arquitectura delta | 20 |
| Figura 5 Figura 2.12: Planteamiento cinemático Delta. (a) Esquema isométrico del Robot Delta. (b) Vista superior del Robot Delta. | 21 |
| Figura 6 pixel imagen binaria y escala de grises. | 23 |
| Figura 7 Puntos bordes de un círculo | 25 |
| Figura 8 Espacio acumulador círculo | 25 |
| Figura 9 Ejemplo de lay-out de una célula robotizada de fabricación de calzado. | 26 |
| Figura 10 Célula robotizada de paletizado con vallas de protección para impedir el acceso..... | 29 |
| Figura 11 Diagrama lay-out..... | 31 |
| Figura 12 Planos banda transportadora 1. | 32 |
| Figura 13 Plano banda transportadora 2. | 33 |
| Figura 14 Bandas transportadores ensambladas | 33 |
| Figura 15 Servomotor MG995..... | 34 |
| Figura 16 Diseño CAD de la estructura de la celda. | 34 |
| Figura 17 Estructura con la bandas trasportadoras dentro. | 35 |
| Figura 18 Botón paro de emergencia..... | 36 |
| Figura 19 Botón de inicio | 36 |
| Figura 20 Botón de paro | 36 |
| Figura 21 Botón de reset | 36 |
| Figura 22 Esquema sistema de control..... | 37 |
| Figura 23 Conexiones eléctricas..... | 37 |
| Figura 24 Diseño PCB. | 38 |
| Figura 25 Fuente de poder ATX..... | 38 |
| Figura 26 Arduino nano. | 39 |
| Figura 27 Modulo HC-05..... | 40 |
| Figura 28 Modelo geométrico robot delta. | 41 |
| Figura 29 Rotula Fuente Catalogo ikko..... | 43 |
| Figura 30 Robot delta | 43 |
| Figura 31 Diagrama abertura de rotula | 44 |
| Figura 32 Abertura rotula | 44 |
| Figura 33 Diagrama de optimización | 46 |
| Figura 34 Medidas más óptimas para los eslabones obtenidos | 46 |
| Figura 35 Evaluación espacio de trabajo | 47 |
| Figura 36 Diseño CAD eslabón 1..... | 48 |
| Figura 37 Plano de vistas eslabón 1 | 48 |
| Figura 38 Rotula ikko LHSA 4..... | 48 |
| Figura 39 Tomado de catálogo iko..... | 49 |
| Figura 40 Articulación 2 | 49 |

| | | |
|-----------|--|----|
| Figura 41 | Ensamble de un brazo del robot..... | 49 |
| Figura 42 | Robot antes del redimensionamiento..... | 50 |
| Figura 43 | Robot después del redimensionamiento..... | 50 |
| Figura 44 | Minisolenoide electroiman | 52 |
| Figura 45 | Minisolenoide ensamblado | 52 |
| Figura 46 | servomotor AX-12A..... | 53 |
| Figura 47 | Conexión de USB2Dynamixel con servos AX-12 ^a | 54 |
| Figura 48 | Esquema control robot delta | 55 |
| Figura 49 | Esquema de generación de trayectoria cartesiana | 56 |
| Figura 50 | Trayectoria a seguir por el robot | 57 |
| Figura 51 | Recorrido articular de cada actuador | 58 |
| Figura 52 | Velocidades calculadas para cada actuador..... | 59 |
| Figura 53 | Graficas de posiciones articulares estimadas vs ejecutadas por el robot..... | 59 |
| Figura 54 | Imagen captura por cámara web | 60 |
| Figura 55 | Imagen recortada..... | 61 |
| Figura 56 | Imagen binarizada | 61 |
| Figura 57 | Eliminación de pixeles pequeños (menor a 20 pixeles) | 61 |
| Figura 58 | Erosión..... | 62 |
| Figura 59 | Imagen con los objetos detectados | 62 |
| Figura 60 | Coordenadas en pixeles con referencia a la cámara..... | 63 |
| Figura 61 | Equivalencia de medidas pixeles y cm | 63 |
| Figura 62 | Distancia entre ejes coordenados del robot y la cámara | 64 |
| Figura 63 | Objetos reordenados en orden de prioridad para el robot | 64 |
| Figura 64 | Posiciones recalculadas con referencia al robot..... | 64 |
| Figura 65 | Acoplamiento de los sistemas | 65 |
| Figura 66 | prueba de velocidad del actuador sin carga | 67 |
| Figura 67 | Prueba de velocidad del actuador son carga..... | 68 |
| Figura 68 | Constante promedio sin carga | 68 |
| Figura 69 | Constante promedio con carga..... | 69 |
| Figura 70 | Tiempos de ejecución de trayectoria de los actuadores | 70 |
| Figura 71 | Tiempos de ejecución de trayectoria de los actuadores | 70 |
| Figura 72 | Tiempos de ejecución de trayectoria de los actuadores | 71 |
| Figura 73 | Tiempos de ejecución de trayectoria de los actuadores | 71 |
| Figura 74 | Efector sujetando la pieza..... | 72 |
| Figura 75 | Separación entre objetos | 73 |
| Figura 76 | Número de piezas sujetadas en cada prueba..... | 75 |
| Figura 77 | Estructura completa | 75 |

INTRODUCCIÓN

La manufactura ha sido practicada durante miles de años tanto por artesanos como sus aprendices, tallando madera, metales, entre otros materiales.

Con la llegada de la segunda revolución industrial la manufactura se benefició con la tecnología de las computadoras y diversos dispositivos electrónicos, haciendo posible el desempeño de tareas con una mayor eficiencia, mayor control y administración, teniendo un bajo costo. Las celdas de manufactura son usadas con frecuencia en las industrias manufactureras, optimizando sus procesos, con el fin de lograr eficiencia en la producción. La robótica inició formalmente en el siglo XVIII en la industria textil que se encontraba en pleno crecimiento con la creación de una máquina que realizaba tareas textiles, era programada por tarjetas perforadas, después de este artefacto empezó la revolución industrial, el primer brazo robótico en la industria se introdujo en el mercado en 1960 y fue llamado “Unimate” que estaba basado en la transferencia de artículos y fue ingresado al mundo industrial por The Ford Motor Company (Robotic Spot, 2004). El uso de robots industriales junto con los sistemas de diseño asistidos por computadora (CAD), y los sistemas de fabricación asistidos por computadora (CAM), son la última tendencia en automatización de los procesos de fabricación (Ramos, Gonzáles, & Melendez, 2008).

Hoy, las industrias manufactureras e incluso laboratorios de manufactura en entidades educativas como escuelas, colegios y universidades técnicas, tienden a implementar sus laboratorios de producción con maquinarias de menos costo y que se adapten con más facilidad al entorno de trabajo, sean estos espacios pequeños. La robótica ocupa en la actualidad un papel preponderante en el proceso de modernización e innovación de las industrias. Es conocido que ésta contribuye fundamentalmente a la calidad del producto y al mejoramiento de la productividad. A nivel internacional existen empresas que cuentan con celdas de manufactura robotizadas de soldadura, mecanizado, paletizado, corte con plasma o láser, etc., aplicadas en sectores automotrices, metalmecánicos, áreas

de diseño, medicina, industria alimentaria, maderera, metalúrgica, del plástico y otros.

Entre los robots paralelos, el que más impacto industrial ha tenido hasta la fecha es el robot DELTA. Este robot se caracteriza por ser muy liviano y veloz, además de poseer una configuración cinemática sencilla basada en paralelogramos. A nivel industrial este robot se utiliza en aplicaciones “pick and place”, es decir donde se requiere reposicionar objetos, una aplicación muy común a nivel industrial y llevada a cabo en la producción de alimentos, componentes electrónicos, farmacéuticos, manipulación de materiales, empaques, entre otros. En este ámbito se centra el presente trabajo, en el cual se busca diseñar, construir e implementar una celda robotizada que consta con un robot delta él encarga de trasladar las piezas, unas bandas transportadoras y un sistema de visión artificial encargado de brindar la información de posición de los objetos.

1. PRELIMINARES

1.1 DESCRIPCIÓN DEL PROBLEMA

Debido a los grandes avances de las distintas áreas industriales dentro de la ingeniería que se están produciendo en la actualidad se necesita de un alto desempeño en la producción y lograr una mayor eficiencia, ya que el ser humano tiene límites a la hora de ejecutar una tarea ya sea su velocidad de ejecución precisión y continuidad. La introducción de la robótica en la industria ha sido la clave para lograr dichos objetivos, así como para reducir el riesgo de accidentes para los operarios. De acuerdo a su aplicación, se especifica la configuración morfológica y los parámetros característicos del robots a implementar.

Para una institución es importante contar con el equipo apropiado para preparar a sus estudiantes con situaciones con los que se podrá encontrar a futuro, muchas veces estos equipos tienden a tener un alto costo, lo que hace difícil su obtención. Por lo cual se plantea el siguiente interrogante. ¿Es posible diseñar e implementar una celda robotizada de bajo costo y materiales asequible, que cuente con un sistema de visión artificial, un robot manipulador delta y una bandas trasportadoras con fines educativos en el programa de ingeniera mecatrónica?

1.2 JUSTIFICACIÓN

El programa de mecatrónica no cuenta con una celda robotizada clasificadora y su adquisición es costosa, estas celdas pueden ser de gran utilidad en el programa debido a las múltiples aplicaciones que tienen en el ámbito industrial como la farmacéutica, alimenticia, automotriz, electrónica entre otras.

Es importante que los estudiantes tengan la oportunidad de interactuar un poco con este tipo de procesos, incentivando la investigación dentro del ámbito académico contribuyendo a la formación integral del ingeniero mecatrónico, explorando los distintos campos de aplicación que existen para este tipo de robots paralelos.

1.3 SOLUCIÓN AL PROBLEMA

Como solución se elaboró una celda robotizada clasificadora apoyada con un sistema de visión artificial con el fin de que el programa pueda contar con una de estas, lo que traería beneficios ya que se podrá profundizar en estas áreas de estudios. Elaborando una estructura con sus respectivos accesorios como las bandas transportadoras, sus elementos de control como lo es botón de inicio, paro y paro de emergencia y una estructura robótica paralela encargada de sujetar las piezas y transportarlas.

1.4 OBJETIVOS

1.4.1 Objetivo general

Desarrollar e integrar un sistema de celda robotizada para la clasificación y posicionamiento de piezas según sus características, a partir de la sincronización de funciones de un robot paralelo delta y un sistema de visión artificial.

1.4.2 Objetivos específicos

- Diseñar y/o implementar los elementos de la celda de manufactura y el robot tipo delta.
- Realizar el análisis cinemático de la estructura paralela.
- Desarrollar un sistema de visión para la clasificación de piezas u objetos y para la generación de referencias del robot.
- Desarrollar un algoritmo de control cinemático para las articulaciones del robot.
- Sincronizar las funciones del sistema de visión y del robot paralelo delta en una tarea de clasificación y posicionamiento de pieza actividad.

2 MARCO TEÓRICO Y ESTADO DEL ARTE

2.1 ROBOTS PARALELOS

Las primeras máquinas que se diseñaron con arquitecturas paralelas fueron para realizar pruebas de neumáticos (S. G. Whitehall, Gough, 1962) y simuladores de vuelo como la estructura de Stewart (Stewart D, 1965). En años recientes, robots con este tipo de arquitecturas comenzaron a llamar la atención, no solo de investigadores sino también de la industria. Si se toma únicamente la definición expuesta previamente para definir un robot paralelo se estaría considerando también los mecanismos redundantes, los cuales cuentan con menos grados de libertad que actuadores. Lo que aquí interesa es una definición formal como la que propone. (Jean-Pierre Merlet, 2000)

Robot Paralelo. Es aquél que está compuesto por una base móvil de n grados de libertad generados con n actuadores simples y una base fija. Dichas bases se encuentran unidas por al menos dos cadenas cinemáticas independientes.

2.1.1 Robot Manipulador tipo Delta

El Robot Manipulador tipo Delta fue diseñado y patentado por R. Clavel en 1991, con la idea de contar con un robot capaz de alcanzar grandes velocidades y aceleraciones en procesos de manipulación de objetos. El diseño original está constituido por tres cadenas cinemáticas paralelas unidas a una base móvil, cada una de estas cadenas son dirigidas por tres actuadores de revoluta que se encuentran acoplados a una base fija del robot. El movimiento de la base móvil se logra por la combinación del movimiento de los brazos y transmitido a la base móvil por los antebrazos (sistema de barras paralelas) a través de dos rótulas. La combinación de las contribuciones de las tres cadenas cinemáticas y de los antebrazos provoca que la base móvil siempre se mantenga paralela a la base

fija. Idealmente su espacio alcanzable se encuentra en la intersección de las tres esferas provocadas por los tres brazos. (Figura 2).

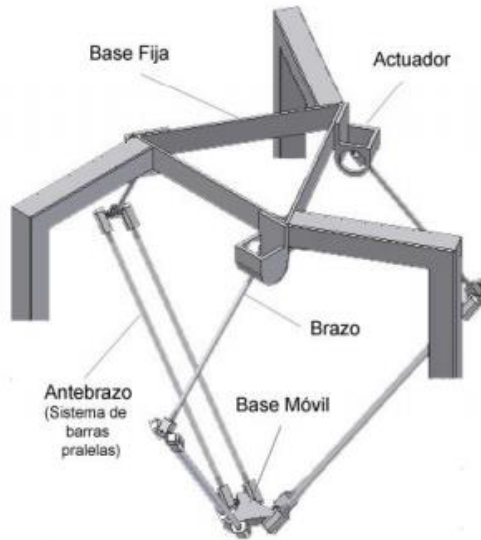


Figura 1 Robot Manipulador Paralelo tipo Delta.

Fuente: CORONADO, Control de un Robot Paralelo Tipo Delta basado en Manipulacion en Espacio de Cámara Lineal

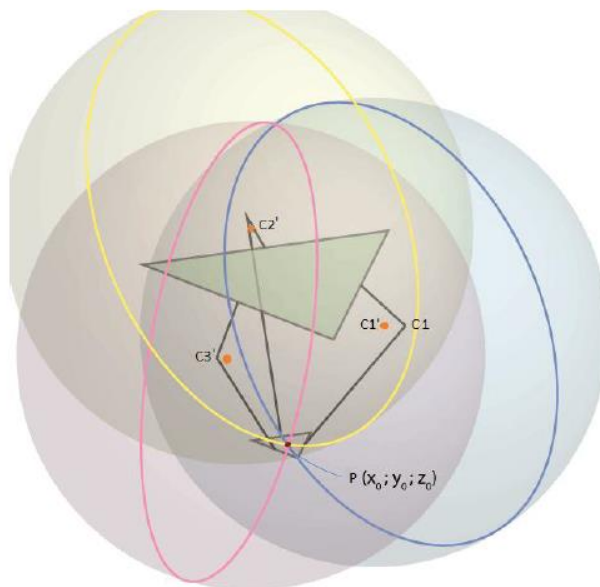


Figura 2 Corte de tres esferas

Fuente: EDISON MARTINEZ, Diseño y construcción de un Robot Paralelo para operaciones de “Tomar y Colocar”, 2011

2.1.2 Aplicaciones

El robot Delta fue introducido al mercado por Demarex, cuyo objetivo era comercializar el robot para la industria del empaquetamiento (la primera aplicación del robot Delta fue el empaquetamiento de galletas); y por ABB bajo el nombre de IRB 340 FlexPicker. Ahora bien, el robot Delta ofrece diferentes soluciones tanto en aplicaciones industriales como en muchas otras áreas.

2.1.2.1 Aplicaciones médicas

El robot Delta está siendo utilizado como un contenedor para microscopio. Este robot tiene el nombre de SurgiScope (Figura 3). El microscopio pesa aproximadamente 20Kg y el robot SurgiScope es el encargado de cargarlo mientras se realizan las cirugías. Se utiliza principalmente en neurocirugías. El robot está siempre bajo el control del cirujano y su equipo de trabajo.

2.1.2.2 Aplicaciones industriales

Como se mencionó antes el robot Delta fue diseñado para aplicaciones de manipulación de objetos. Así, el campo fuerte de este tipo de robot son las operaciones llamadas "fast pick-and-place". ABB ha diseñado robots Delta para incorporarse a las industrias alimenticias, farmacéuticas y electrónicas. Su principal modelo es el IRB 340 flexpicker (Figura 4).

También existen aplicaciones para el mecanizado, un ejemplo es el robot desarrollado por el grupo Krause and Máuser. Ellos desarrollaron el Quickstep, una fresadora CNC con tres grados de libertad. Además de ellos, la compañía Hitachi Seiki manufactura robots Delta de pequeñas dimensiones para labores de empaquetamiento y perforado. Esta licencia fue concedida por Demarex.

Así como estos, se pueden enumerar varios ejemplos en la industria de robots con arquitecturas paralelas. Se puede observar que los diseños están hechos para mucho más que solamente aplicaciones de "pick-and-place".



Figura 3 SurgiScope. Robot Delta en aplicaciones médicas.

Fuente: CORONADO, Control de un Robot Paralelo Tipo Delta basado en Manipulacion en Espacio de Cámara Lineal.



Figura 4 IRB 340. Robot con arquitectura delta una solución de ABB para operaciones de “pick-and-place”.

Fuente: ABB robotics

2.2 CINEMÁTICA

La cinemática del robot estudia el movimiento del mismo con respecto a un sistema de referencia. Se interesa principalmente por la descripción analítica del movimiento espacial del robot como una función del tiempo, y en particular por las relaciones entre la posición y la orientación de la herramienta del robot con los valores que toman las coordenadas de sus articulaciones [4].

Se conocen dos planteamientos de modelos cinemáticos: modelo cinemático directo y modelo cinemático inverso. El primero consiste en determinar la posición y orientación del efector final con respecto al sistema de la base del robot a partir de conocer los valores de las articulaciones y los parámetros geométricos. Mientras que el modelo cinemático inverso consiste en determinar la posición que deben adaptar las articulaciones para que el efector final llegue a una posición conocida.

Una de estas ventajas de los robots paralelos con respecto a los seriales es la simpleza en el análisis de su cinemática inversa, no así para la directa.

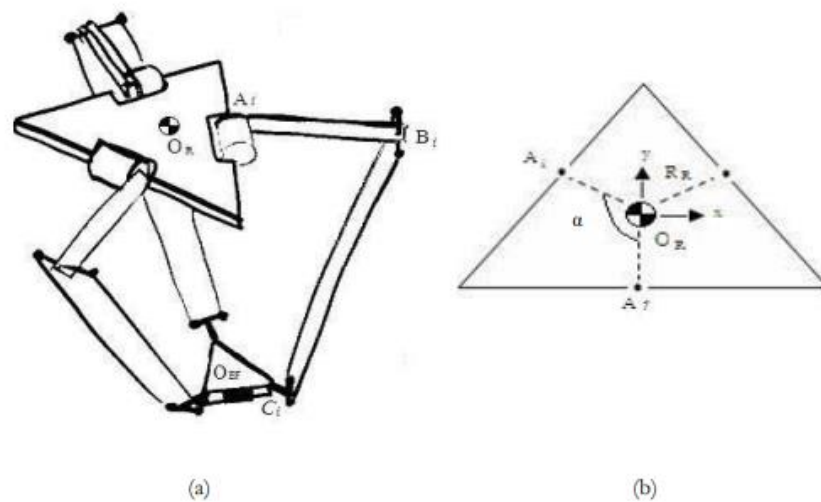


Figura 5 Figura 2.12: Planteamiento cinemático Delta. (a) Esquema isométrico del Robot Delta. (b) Vista superior del Robot Delta.

Fuente: PÉREZ. Implementación de un Control Supervisorio Difuso para el Control de Posición de un Robot Manipulador Delta-Edición Única. Pp 19 – 22.

2.3 VISIÓN ARTIFICIAL

La visión artificial es una tecnología industrial aplicable a diferentes sectores y fases de producción. Es de los métodos automatizados e inteligentes más efectivo e innovador para adquirir, procesar y analizar imágenes en los procesos de producción.

Las principales aplicaciones de la visión artificial en la industria actual son:

- Identificación e inspección de objetos.
- Determinación de la posición de los objetos en el espacio.
- Establecimiento de relaciones espaciales entre varios objetos (guiado de robots).
- Determinación de las coordenadas importantes de un objeto. Realización de mediciones angulares.
- Mediciones tridimensionales.

2.3.1 Métodos de captación de las imágenes Digital.

Una imagen digital es una imagen $f(x,y)$ que se ha discretizado tanto en coordenadas espaciales como en brillo. Por lo tanto una imagen digital se puede considerar como una matriz cuyos índices de fila y columna identifican un unto en la imagen y el valor correspondiente del elemento de la matriz indica el nivel de gris en ese punto. Los elementos de una distribución digital de este tipo se denominan elementos de la imagen, o más comúnmente píxeles o pels (abreviatura de picture elements).

La función obtenida tras el resultado de la medida o muestreos realizados a intervalos de tiempo espaciados regularmente, siendo el valor de dicha función un número positivo y entero. Los valores que esta función toma en cada punto dependen del brillo que presenta en esos puntos la imagen original.

- **Píxel:** Una imagen digital se considera como una cuadrícula. Cada elemento de esa cuadrícula se llama Píxel (Picture element). La

resolución estándar de una imagen digital se puede considerar de 512x484 Pixel.

- **Nivel de grises:** Cuando una imagen es digitalizada, la intensidad del brillo en la escena original correspondiente a cada punto es cuantificada, dando lugar a un número denominado “nivel de gris”. Figura 6.
- **Imagen binaria:** Es aquella que sólo tiene dos niveles de gris: negro y blanco. Cada píxel se convierte en negro o blanco en función del llamado nivel binario o umbral. Figura 6.
- **Escena:** Es un área de memoria donde se guardan todos los parámetros referentes a la inspección de un objeto en particular: Cámara utilizada, imágenes patrón memorizadas, tolerancias, datos a visualizar, entradas y salidas de control, etc.
- **Window** (ventana de medida): Es el área específica de la imagen recogida que se quiere inspeccionar.

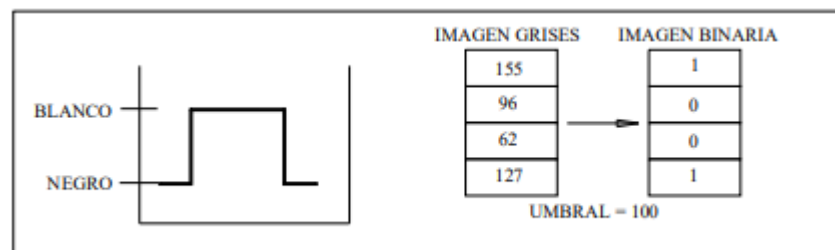


Figura 6 pixel imagen binaria y escala de grises.

Fuente: ROJAS, Teddy V.; SANZ, Wilmer; ARTEAGA, Francisco. Sistema de visión por computadora para la detección de objetos esféricos a través de la transformada de Hough, 2008.

2.4 TRANSFORMADA DE HOUGH

La transformada de Hough, según Nixon y Aguado (2012), es una técnica que localiza formas en imágenes definiendo una implementación eficiente de coincidencia de patrones para patrones binarios. Concretamente, se usa para extraer formas simples como líneas, círculos o elipses, así como formas arbitrarias más complejas. Su principal ventaja es que da el mismo resultado que la técnica conocida como “template matching” (búsqueda de patrones), pero más rápido. Es por tanto, un algoritmo muy potente usado en procesamiento de imágenes que se encuentra entre las técnicas más usadas para extracción de formas.

Para estos mismos autores, la transformada de Hough se basa en la detección de bordes para conseguir segmentar la imagen en regiones. La implementación de esta transformada define un mapeo de los puntos de la imagen en un espacio acumulador llamado espacio de Hough. El mapeo es logrado de una manera computacionalmente eficiente, basada en la función que describe la forma del objetivo. Este mapeo requiere menos recursos computacionales que la búsqueda de patrones.

Esta transformada define un grupo de puntos centrados en un origen „ (x_0, y_0) ” y con radio „ r ”. Cada punto de borde en la imagen define un conjunto de círculos en el espacio, los cuales están definidos por todos los valores posibles del radio y están centrados en las coordenadas del punto de borde. Los círculos se definen para un valor de radio dado. Se realiza, por tanto, un barrido por toda la imagen considerando los tres parámetros mencionados: la posición „ (x, y) ” de cada píxel y el radio máximo del círculo. El barrido se hace para todos los píxeles de borde detectados

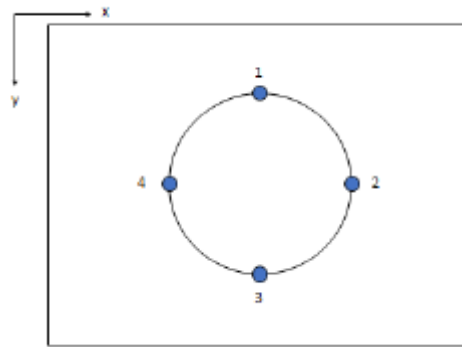


Figura 7 Puntos bordes de un círculo

Fuente: ROJAS, Teddy V.; SANZ, Wilmer; ARTEAGA, Francisco. Sistema de visión por computadora para la detección de objetos esféricos a través de la transformada de Hough, 2008.

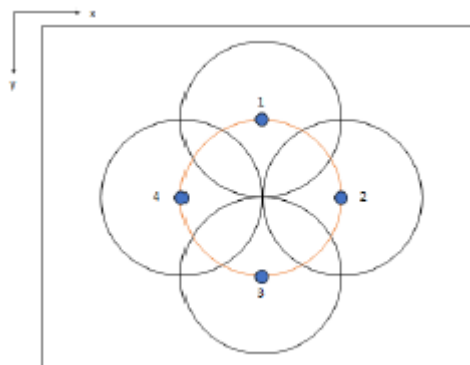


Figura 8 Espacio acumulador círculo

Fuente: ROJAS, Teddy V.; SANZ, Wilmer; ARTEAGA, Francisco. Sistema de visión por computadora para la detección de objetos esféricos a través de la transformada de Hough, 2008.

2.5 DISEÑO Y CONTROL DE UNA CÉLULA ROBOTIZADA

El proyecto e implantación de un sistema robotizado implica la consideración de un gran número de factores, que van desde el posible rediseño del producto, hasta la definición detallada del *lay-out* o plano de implantación del sistema.

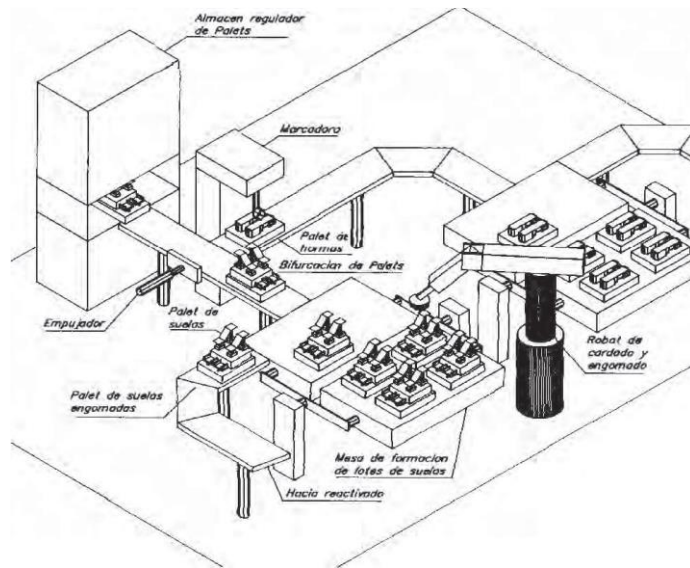


Figura 9 Ejemplo de lay-out de una célula robotizada de fabricación de calzado.

Fuente: BARRIENTOS, *Fundamentos de robótica*, 2007

Junto con la selección más adecuada del robot para la aplicación, habrá que definir, e incluso diseñar, los elementos periféricos pasivos (mesas, alimentadores, utillajes, etc.) o activos (manipuladores secuenciales, máquinas CN, etc.) que intervienen en la célula, y situarlos físicamente en el sistema. [6]

2.5.1 Características del sistema de control de la célula de trabajo

Una célula robotizada debe responder a las premisas de flexibilidad y automatización que justifican su empleo. Para ello es imprescindible el establecimiento de un buen sistema de control que deberá realizar, entre otras, las siguientes funciones. (BARRIENTOS, 2007)

- **Control individual** de cada una de las máquinas, transportes y demás dispositivos, incluidos robots, que compongan la célula.
- **Sincronización** del funcionamiento de los diferentes dispositivos entre sí.
- **Detección, tratamiento y recuperación** si es posible de las situaciones anómalas de funcionamiento que puedan presentarse.

- **Optimización del funcionamiento** conjunto de los dispositivos de la célula, distribuyendo si es posible las funciones de manera dinámica, para así evitar paradas por espera o acciones innecesarias.
- **Interfaz con el usuario**, mostrando la información adecuada para que en todo momento se conozca con el detalle necesario el estado del sistema, así como permitiendo que el operador acceda, con las restricciones pertinentes, al funcionamiento del mismo y a las utilidades de mantenimiento y ajuste del sistema.
- **Interfaz con otras células**, para permitir la sincronización entre ellas, optimizando el funcionamiento de un sistema de fabricación flexible compuesto por varias células. (BARRIENTOS, 2007)
- **Interfaz con un sistema de control superior** que realiza básicamente funciones de supervisión y actualización de programas cuando se diese un cambio en la producción. (BARRIENTOS, 2007).

2.5.2 Medidas de seguridad a tomar en la fase de diseño de la célula robotizada

En el establecimiento del *lay-out* de la célula se ha de considerar la utilización de barreras de acceso y protecciones en general que intenten minimizar el riesgo de aparición de un accidente.

De forma general se pueden citar las siguientes:

- **Barreras de acceso a la célula:** Se dispondrán barreras en torno a la célula, que impidan el acceso a personas (parada inmediata al entrar en la zona de trabajo) (Figura 10).

- **Dispositivos de intercambio de piezas:** En caso de que el operador deba poner/recoger piezas situadas dentro del área de trabajo del robot, se utilizarán dispositivos que permitan realizar estas acciones a distancia, utilizando, por ejemplo, mesas giratorias (Figura 9).

- **Movimientos condicionados:** En el caso de que durante el funcionamiento de la célula el operario deba entrar en determinados momentos dentro del campo de acción del robot (para alimentar de nuevas piezas al robot, por ejemplo), se programará a éste de manera que no efectúe movimiento alguno durante estos instantes.

Zonas de reparación: Se preverá la existencia de zonas de reparación y mantenimiento. Estas zonas, dentro del campo de acción del robot, estarán fuera de su zona de trabajo. En ellas se asegurará mediante diferentes dispositivos que el robot no realizará movimientos de manera automática. (BARRIENTOS, 2007)

- **Condiciones adecuadas en la instalación auxiliar:** Sistema eléctrico con protecciones, aislamientos, sistemas neumáticos o hidráulicos correctos. [6]

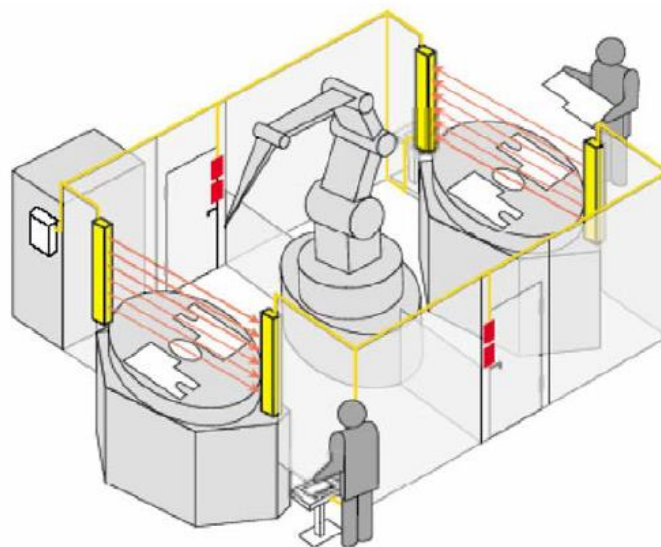


Figura 10 Célula robotizada de paletizado con vallas de protección para impedir el acceso.

Fuente: Guía técnica de seguridad en robótica. Departamento de hacienda, economía y empleo.

2.5.3 Normativas Internacionales

Las normativas internacionales más relevantes en la temática que nos ocupa son:

Normativa ISO 10218-1 (Internacional)

“Requerimientos de seguridad en entornos con Robots Industriales”

- Realizada por el Organismo Internacional de Normalización (ISO), está dedicada al análisis de la seguridad, la definición de riesgos y la identificación de posibles fuentes de peligros o accidentes.

- Contiene además una sección sobre diseño y fabricación, que dedica un breve análisis al diseño de sistemas robotizados, teniendo en cuenta aspectos mecánicos, ergonómicos y de control.

Normativa ANSI / RIA R15.06 (Americana en primera instancia, convertida luego en estándar Internacional).

“Requerimientos de seguridad para Robots Industriales y Sistemas Robots”

- Proporciona especificaciones para la fabricación de robots industriales, la integración/instalación de sistemas de robots, y los métodos para implementar la seguridad del personal asociado con el empleo de robots y sistemas robots.

Normativa EN 775 y española UNE-EN 775

“Robots Manipuladores. Seguridad”

Esta normativa, aprobada por el Comité Europeo de Normalización, ha sido anulada y reemplazada por la ISO 10218-1:2006.

Norma EN ISO 13.849-1

“Seguridad de las máquinas. Partes del sistema de mando relativas a la seguridad.”

- Parte 1: Principios generales para el diseño.

Esta normativa abarca todo tipo de tecnologías (eléctrica – electrónica – neumática – hidráulica – mecánica – etc.). Reemplaza la norma EN 954-1 basada en las categorías de seguridad, incorporando sus conceptos con requisitos adicionales de seguridad.

Norma EN IEC 62.061

“Seguridad de máquinas – Seguridad funcional de sistemas eléctricos, electrónicos y electrónicos programables relacionados a la seguridad”.

Conforme se observa en su título, a diferencia de la norma anterior no abarca la totalidad de tecnologías industriales disponibles.

3 DISEÑO CELDA ROBOTIZADA

3.1 FUNCIONAMIENTO

El funcionamiento de la celda pick and place comienza con la pieza principal que es dirigida por la banda transportadora, la cual mediante sensores encargados (cámara) de detectar la pieza y su ubicación, las coordenadas son procesadas y enviadas al robot delta para que este pueda tomarla y llevarla a otra banda y así de manera cíclica.

El diseño y construcción de la celda se dividió en tres partes principales la estructura física, bandas transportadoras y el robot delta.

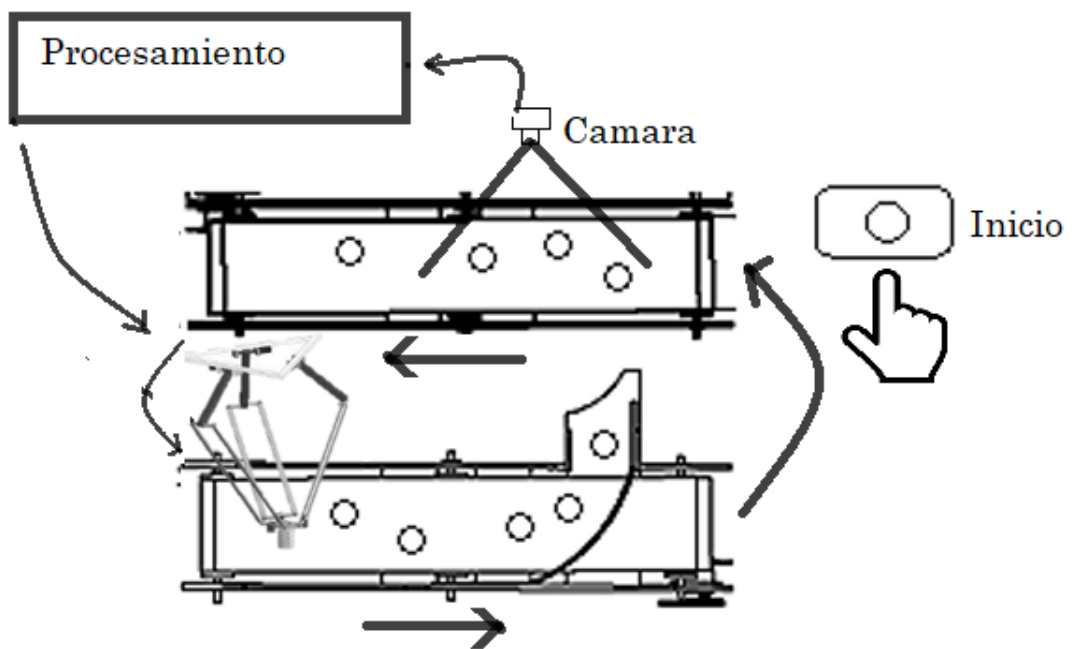


Figura 11 Diagrama lay-out
Fuente autor

3.2 DISEÑO DE BANDAS

El diseño de las bandas se realizó inicialmente en una herramienta de software cad autodesk inventor, teniendo en cuenta el tamaño de la cinta y un buen torque en la selección de motor para que fuese capaz de mover objetos medianamente pequeños.

Las bandas transportadoras tienen como objetivo transportar los objetos del proceso en la cual una es la que recibe los objetos del robot delta, este mismo tiene una pequeña estructura que retorna los objetos a la otra banda (Figura 12). La banda numero dos transporta los objetos hasta un rango del robot delta para que puedan ser sujetados y el ciclo se repite. (Figura 13). En la figuras 12 y 13 se muestra los planos correspondientes de las bandas trasportadoras con sus respectivas vistas.

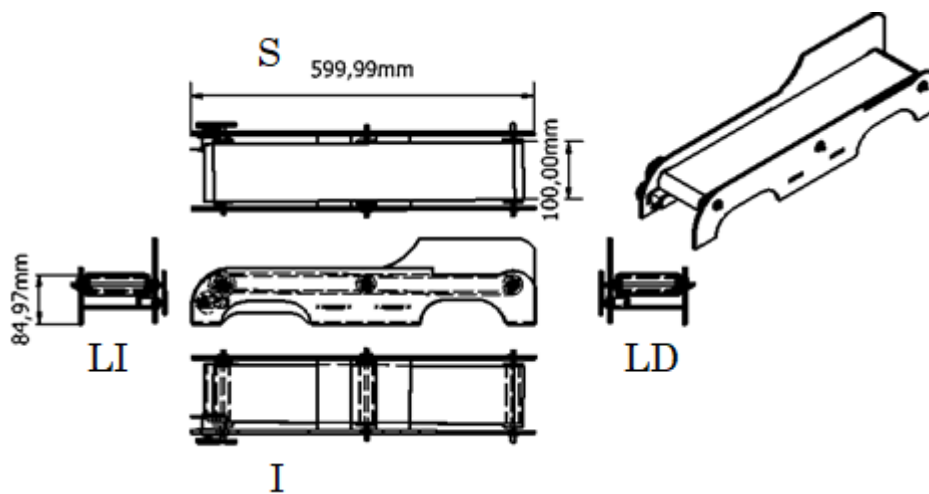


Figura 12 Planos banda transportadora 1. Fuente: Autor

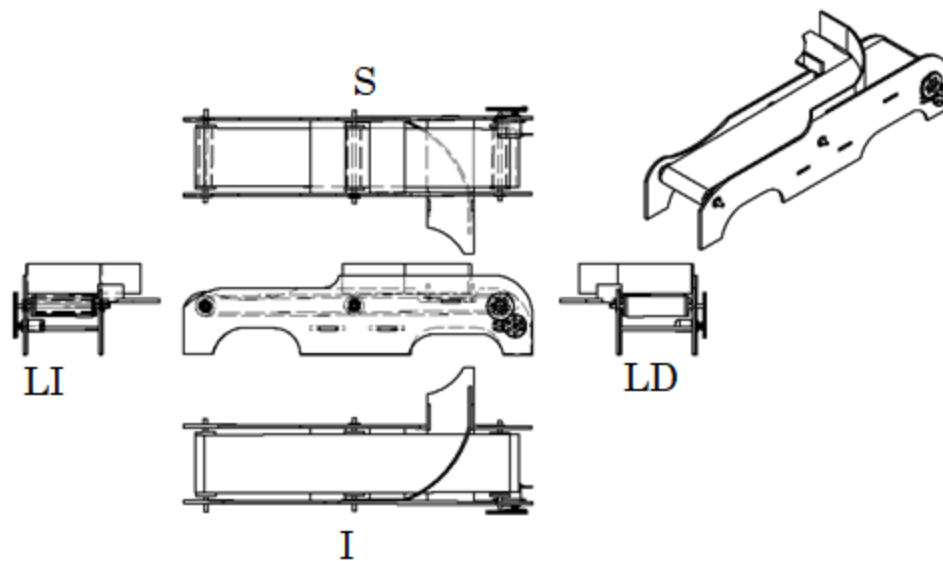


Figura 13 Plano banda transportadora 2. Fuente: Autor

Las bandas transportadoras fueron ensambladas en el software de diseño con el fin de ver su aspecto y verificar que estas tuvieran dimensiones correctas antes de proseguir a la construcción real. Figura 15.

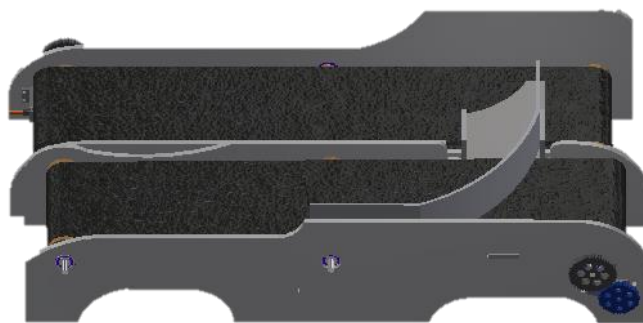


Figura 14 Bandas transportadores ensambladas. Fuente: Autor



Figura 15 Servomotor MG995

Para el actuador se eligió un motor tipo servo de giro continuo (MG995), se caracterizan por ser capaces de girar 360 grados, es decir, una rotación completa. Su funcionamiento es similar al de un motor convencional, pero con las características propias de un servo. Esto quiere decir que podemos controlar su posición y velocidad de giro en un momento dado.

Característica:

Par de torsión: 9.4kg / cm (4.8v)

Velocidad máxima de operación: 0.17sec / 60 grados (4.8V sin carga)

Voltaje de funcionamiento: 4.8 ~ 6.6v

3.3 DISEÑO ESTRUCTURA

Para el diseño de la estructura se realizó un modelo CAD para observar cuales serían las dimensiones más adecuadas para la construcción de la estructura. (Figura 16)



Figura 16 Diseño CAD de la estructura de la celda. Fuente: Autor

3.3.1 Barreras materiales

En el diseño también se tiene en cuenta parámetros de seguridad como lo las barreras materiales, estos elementos de seguridad estarían dentro de lo que se podría llamar seguridad positiva y consiste, en un diseño por el cual se trata de proteger al operador de los riesgos, tratando de impedir que el trabajador acceda a la zona de peligro del robot. También la barrera dispone proporcionar suficiente espacio dentro de la célula de trabajo. (Figura 17)

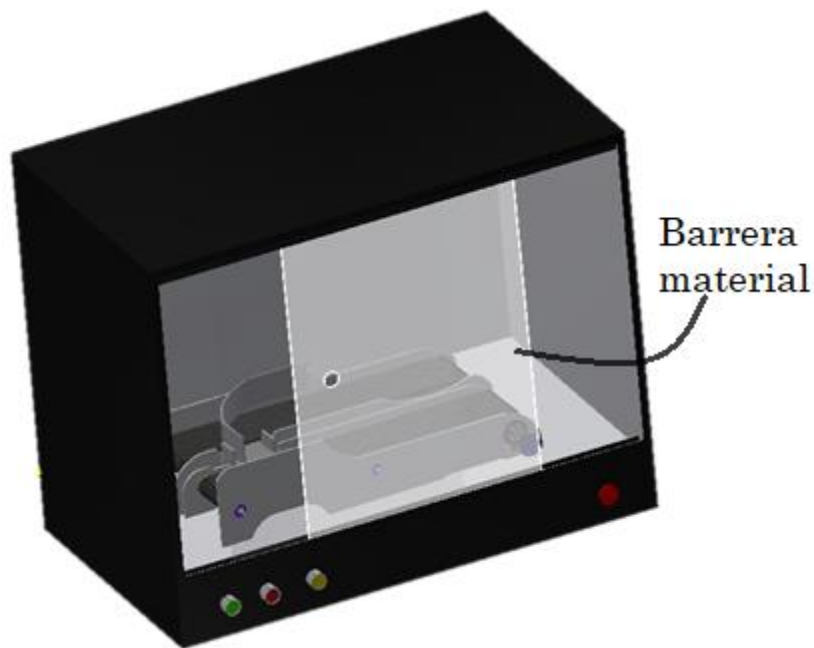


Figura 17 Estructura con la bandas trasportadoras dentro. Fuente: Autor

3.3.2 Interruptores de seguridad

En el diseño de la estructura se tiene en cuenta un espacio para el panel de control el cual está a fácil acceso del usuario y fuera del espacio de trabajo de la célula.

Paro de emergencia: Este botón es de suma importancia y según la norma EN ISO 13850, tiene como función prevenir situaciones que puedan poner en peligro a las personas, para evitar daños en la máquina o en trabajo. (Figura 18)



Figura 18 Botón paro de emergencia. Fuente: Autor

Inicio: Este tiene como función iniciar todo el proceso al ser presionado. Figura 18.



Figura 19 Botón de inicio

Paro: Tiene como función poner en pausa el proceso, esto quiere decir que puede continuar donde se había detenido. Figura 20.



Figura 20 Botón de paro

Reset: Permite regresar a la maquina a sus condiciones iniciales sin importar en que parte de la secuencia se encuentra. Figura (21)



Figura 21 Botón de reset

3.4 SISTEMA ELECTRÓNICO

En el diseño del sistema electrónico se tuvo en cuenta factores importantes como lo es el control de los actuadores de las bandas, las señales de entrada del panel (pulsadores), el control del efector del robot y él envió y entrada de datos desde el PC. Figura 22.

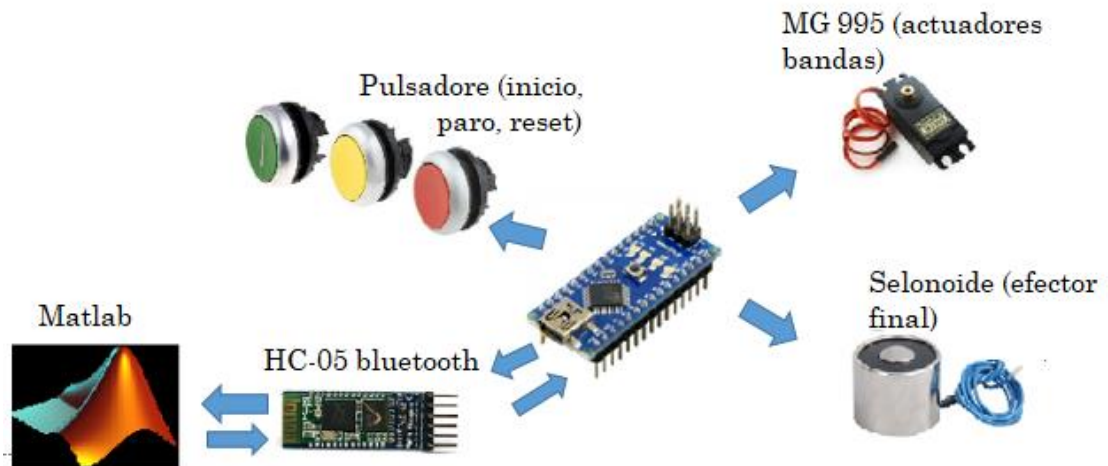


Figura 22 Esquema sistema de control. Fuente: Autor

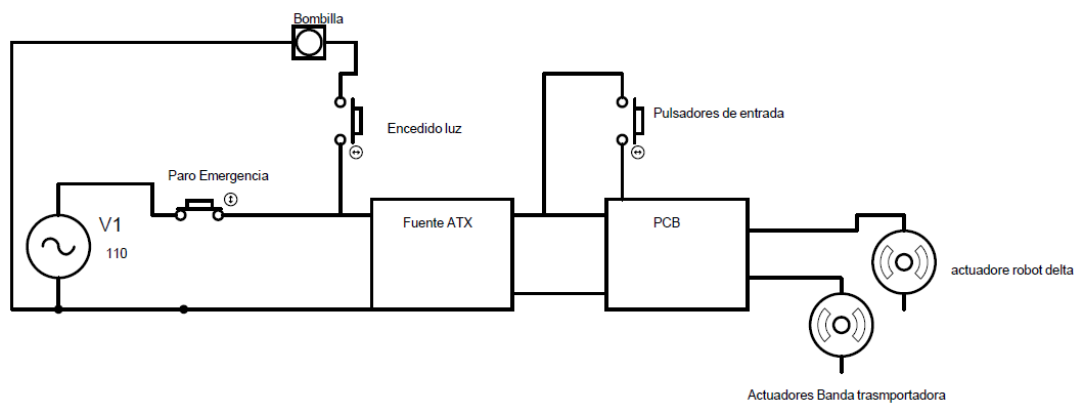


Figura 23 Conexiones eléctricas. Fuente: Autor

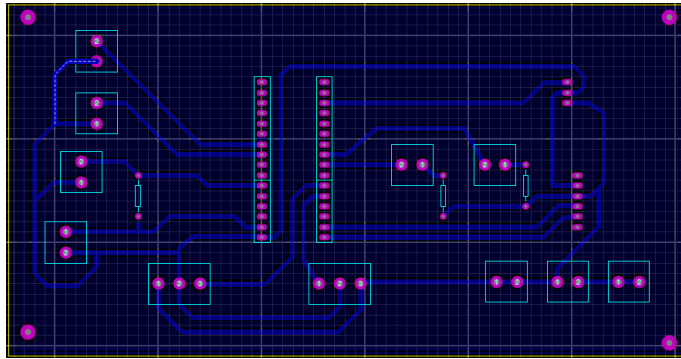


Figura 24 Diseño PCB. Fuente: Autor

3.4.1 Elementos Electrónicos

Para la alimentación de todos los componentes se usó una fuente ATX ya que estas proporcionan los voltajes de salida necesarios (5v y 12v en DC).

Las fuentes *ATX* son un dispositivo que se instala internamente en la caja del ordenador, y se encarga de transformar la corriente alterna de la línea eléctrica comercial en corriente directa; la cuál es utilizada por los elementos electrónicos y eléctricos.



Figura 25 Fuente de poder ATX. Fuente: Autor

Arduino nano

Arduino es una plataforma de creación de electrónica de código abierto, la cual está basada en hardware y software libre, flexible y fácil de utilizar para los creadores y desarrolladores. Esta plataforma permite crear diferentes tipos de microordenadores de una sola placa a los que la comunidad de creadores puede darles diferentes tipos de uso.

El arduino nano es una placa de desarrollo de tamaño compacto, completo y compatible con protoboar, basada en el micro controlador ATmega328P. Tiene 14 pines de entrada/salida digital (de los cuales 6 pueden ser usando con PWM), 6 entradas analógicas, un cristal de 16Mhz, conexión Mini-USB, terminales para conexión ICSP y un botón de reseteo. En este trabajo la placa tiene la función de realizar tareas mandadas principalmente desde el PC que es el realiza las operaciones principales.



Figura 26 Arduino nano. Fuente: Autor

Características:

Micro controlador: ATmega328

Voltaje de operación: 5V

Voltaje de alimentación (Recomendado): 7-12V

I/O Digitales: 14 (6 son PWM)

Memoria Flash: 32KB

EEPROM: 1KB

Frecuencia de trabajo: 16MHz

Módulo Hc-05

El Bluetooth HC-05 es un pequeño modulo transmisor/receptor TTL diseñado para ser controlado a través de RS232. Este módulo permite tanto como transmitir como recibir datos a través de tecnología bluetooth sin conectar cables a los dispositivos a comunicar. Es un dispositivo muy fácil de usar y compacto este se controla mediante comandos AT por el puerto serie. Tiene compatibilidad con Arduino o cualquier micro controlador con UART. Además esta referencia permite la configuración tanto maestro como esclavo.

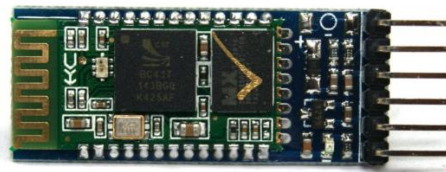


Figura 27 Modulo HC-05. Fuente: Autor

Características

Funciona como dispositivo maestro y esclavo bluetooth

Configurable mediante comandos AT

Bluetooth V2.0+EDR

Frecuencia de operación: 2.4 GHz Banda ISM

Modulación: GFSK (Gaussian Frequency Shift Keying)

Potencia de transmisión: ≤ 4 dBm, Class 2

Sensibilidad: ≤ -84 dBm @ 0.1% BER

Perfiles Bluetooth: Puerto serie bluetooth.

Distancia de hasta 10 metros en condiciones óptimas del ambiente

3.5 DISEÑO ROBOT DELTA

Para el diseño del robot primero se optó por dibujar el robot delta con sus respectivas cinemáticas y se desarrolló el algoritmo de la cinemática inversa del robot.

3.5.1 Cinemática inversa

En la robótica el estudio de la posición entre el efector final y la de los actuadores se le conoce como cinemática. Donde la cinemática inversa se conoce la posición final del efector y a partir de este se calcula la posición de los actuadores.

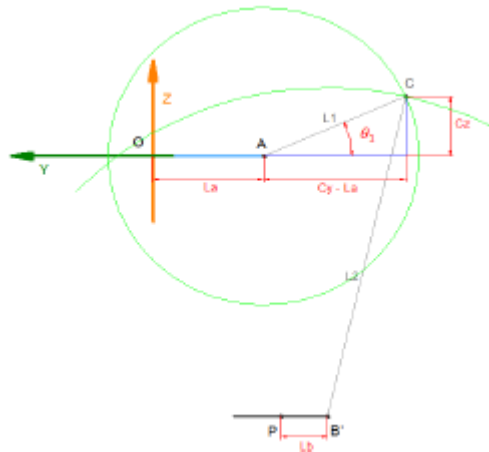


Figura 28 Modelo geométrico robot delta Fuente: Diseño y construcción de un Robot Paralelo para operaciones de “Tomar y Colocar”, Edison Martínez.

“Debido a la restricción de la junta A el brazo describe una circunferencia de radio L_1 , y con respecto al punto B, el punto antebrazo puede describir una esfera de Radio L_2 , la intersección de la circunferencia y la esfera se produce en dos puntos, se toma como solución el punto con menor valor en la coordenada Y. Al determinar la posición del punto C se puede obtener el ángulo θ_1 del actuador” *5+, como se muestra en la figura 39.

Coordenadas del Punto **P**, **B**, **A** y **B'**.

$$P(x_0, y_0, z_0); \quad B(x_0, y_0 - L_b, z_0); \quad A(0, -L_a, 0); \quad B'(0, y_0 - L_b, z_0)$$

“Con las coordenadas de los puntos descritos anteriormente, se plantea un sistema de dos ecuaciones no lineales que permita encontrar la posición del punto C, con la cual se puede calcular el ángulo que forma el brazo con el plano horizontal, obteniendo así la solución esperada” [5].

$$(C_y - L_a)^2 + (C_z - A_z)^2 = L_1^2$$

$$(C_y - B'_y)^2 + (C_z - B'_z)^2 = L_2^2 - x_0^2$$

De las anteriores ecuaciones, se presenta la siguiente solución cuadrática.

$$aC_y^2 + bC_y + c = 0$$

Donde:

$$a = \left(1 + \frac{L_a - y_0 + L_b^2}{z_0} \right)$$

$$b = \left(2 \left(\frac{L_a - y_0 + L_b}{z_0} \right) \left(\frac{L_2^2 - x_0^2 - z_0^2 - L_1^2 - L_a^2}{2z_0} \right) - 2L_a \right)$$

$$c = \left(\frac{L_2^2 - x_0^2 - z_0^2 - L_1^2 - L_a^2}{2z_0} - L_a^2 - L_1^2 \right)$$

$$C_{y_{1-2}} = \frac{-b \mp \sqrt{b^2 - 4ac}}{2a}$$

El ángulo de la primera articulación se calcula con:

$$\theta_1 = \arctan\left(\frac{C_{z1}}{L_a - C_{y1}}\right)$$

Gracias a la triple simetría para hallar la solución en los otros brazos se usa la matriz de rotación con un ángulo de 120° para el brazo 2 y 240° para el 3. Esta matriz de rotación permite girar el sistema de coordenadas de manera que se pueda usar la solución descrita para el cálculo de los restantes ángulos.

3.5.2 Espacio de trabajo del robot

Una vez estudiada las cinemáticas del robot se describe el espacio de trabajo del robot paralelo. Se analiza el espacio de trabajo del robot definido por las limitantes geométricas del mecanismo, obteniéndose así una función implícita que define un volumen tridimensional dentro del cual el análisis de posicionamiento es válido. Así mismo, también se analiza el espacio de trabajo del robot definido por las limitantes mecánicas del robot; específicamente, los límites de giro impuestos por las articulaciones esféricas.

3.5.2.1 Limitantes mecánicas

Las limitantes mecánicas del robot se deben a las restricciones de giro impuestas por las articulaciones esféricas, las cuales fueron consideradas como ideales al analizar las limitantes geométricas. Dichas limitantes restringen aún más el espacio de trabajo siendo preciso también determinar si la localización deseada de la plataforma forzará las piezas mecánicas del robot o no, para así evitar deformaciones o fracturas.



Figura 29 Rotula Fuente Catalogo ikko

Para analizar esta restricción es necesario primero expresar la orientación de las rótulas y los aros con respecto al marco de referencia dada una localización de la plataforma que es el eje principal.

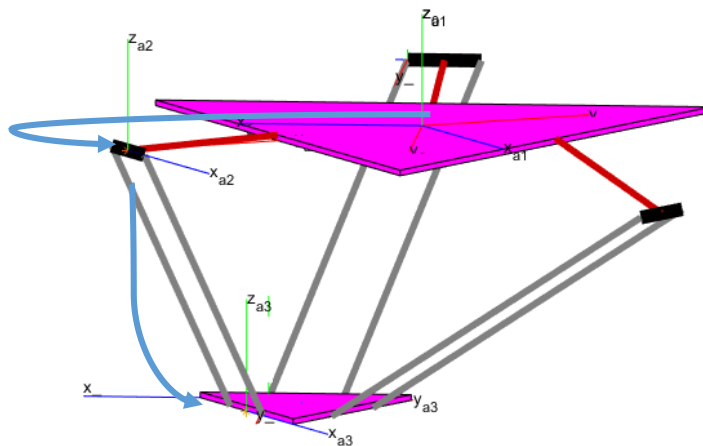


Figura 30 Robot delta

Fuente: Autor

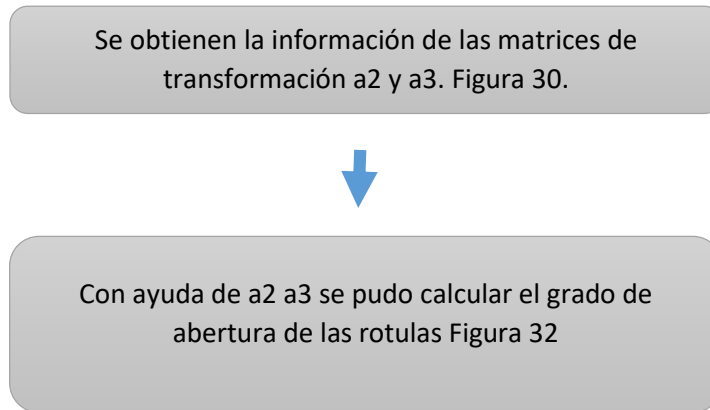


Figura 31 Diagrama abertura de rotula

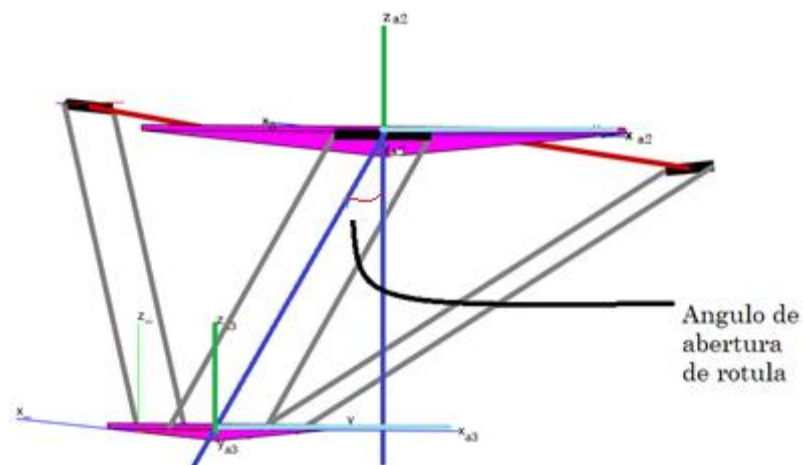


Figura 32 Abertura rotula

3.5.2.2 Optimización de tamaño del robot

Ya con el estudio cinemático del robot (inversa) y sus respectivas restricciones se aplicaron algoritmos para la optimización de las longitudes del robot adquiriendo de esta manera las dimensiones más adecuadas para la célula robotizada.

El método por el que se optó para calcular las dimensiones más apropiadas se llama optimización por enjambres de partículas.

3.5.2.3 Optimización por enjambres de partículas

Este tipo algoritmos permite optimizar un problema a partir de una población de soluciones candidatas, denotadas como "partículas", moviendo éstas por todo el espacio de búsqueda según reglas matemáticas que tienen en cuenta la posición y la velocidad de las partículas. El movimiento de cada partícula se ve influido por su mejor posición local hallada hasta el momento, así como por las mejores posiciones globales encontradas por otras partículas a medida que recorren el espacio de búsqueda. El fundamento teórico de esto es hacer que la nube de partículas converja rápidamente hacia las mejores soluciones.

Para el cálculo de las longitudes más óptimas de las articulaciones del robot se implementó un algoritmo en matlab usando funciones propias del programa. Para esto se definen un conjunto de límites inferior y superior en las variables de longitudes tanto para el eslabón 1 como para el eslabón 2, de modo que se encuentre una solución en los rangos especificados.

MaxL1 = 23; Longitud máxima de eslabón 1

MinL1 = 9; Longitud mínima de eslabón 1

MaxL2 = 23; Longitud máxima de eslabón 2

MinL2 = 5; Longitud mínima de eslabón 2

Max_h_bandas = -15; Distancia máxima desde base del robot a la cita transportadora

Min_h_bandas = -40*5; Distancia mínima desde base del robot a la cita transportadora.

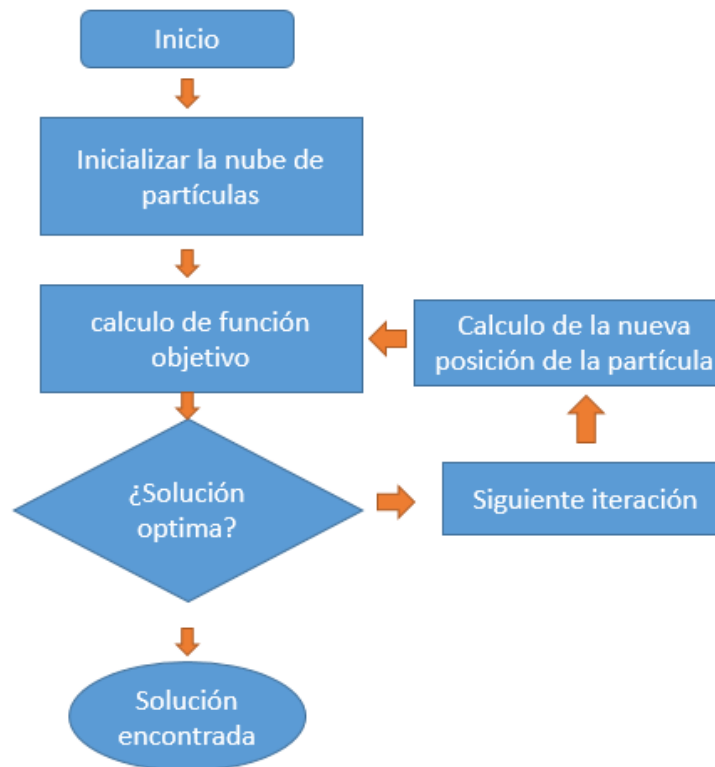


Figura 33 Diagrama de optimización

x =
 9.0000 30.5358 -37.2087

Figura 34 Medidas más óptimas para los eslabones obtenidos

Como resultado del cálculo del algoritmo obtenemos que las dimensiones más óptimas son las siguientes.

| | |
|-------------------------------------|-----------------|
| Eslabón 1 | 9 |
| Eslabón 2 | 30.5358 |
| Altura bandas trasportadoras | -37.2087 |

Tabla 1 Medidas optimas

Con la dimensiones obtenidas se realizo una prueba de verificación del espacio de trabajo del robot, visualizando todos los puntos que pueden ser alcanzados por el robot delta, comprobando que los valore obtenidos efectivamente abarcan todo el campo requerido para este proceso. Figura 35.

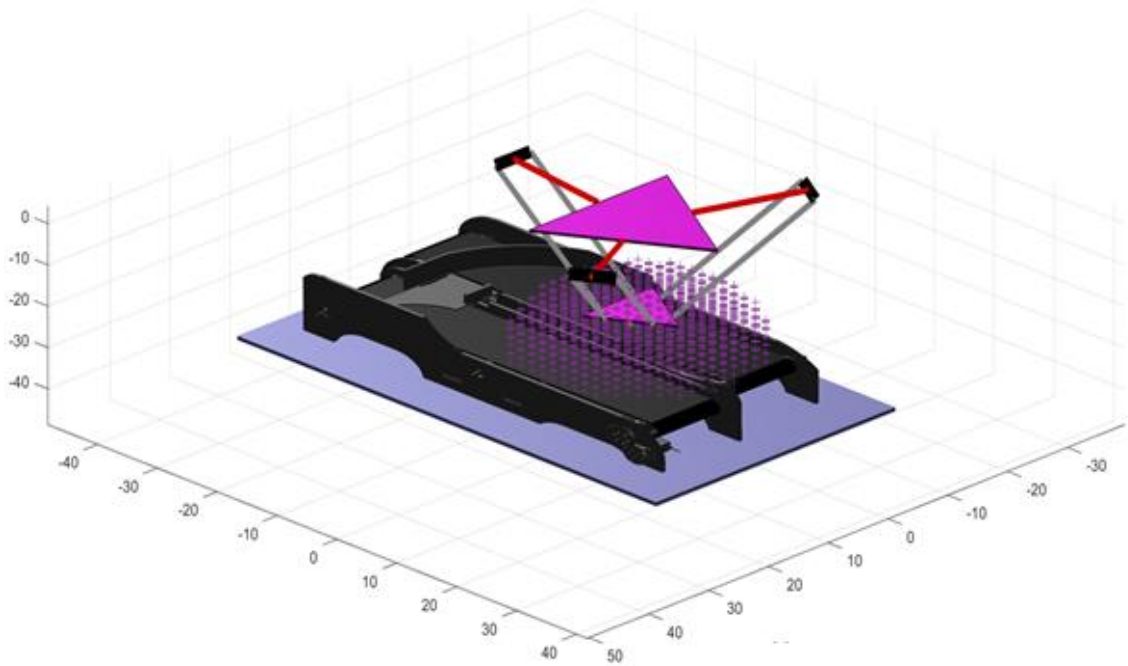


Figura 35 Evaluación espacio de trabajo

3.5.3 Diseño articulaciones

3.5.3.1 Eslabón 1

Con la optimización de las medidas de las articulaciones del robot y la verificación del espacio de trabajo el siguiente paso en el diseño del robot es diseñar las dos articulaciones, para esto se diseñó en un software CAD figura 36 y se procedió a imprimirla en impresión 3d. Teniendo en cuenta que los puntos de rotación tuviesen una longitud de 9 cm.

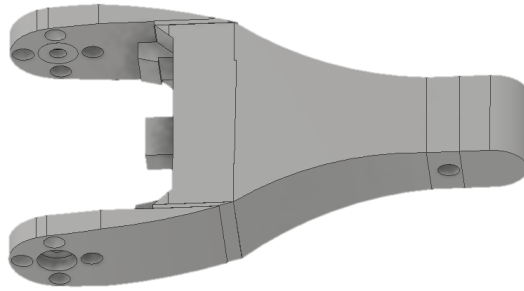


Figura 36 Diseño CAD eslabón 1

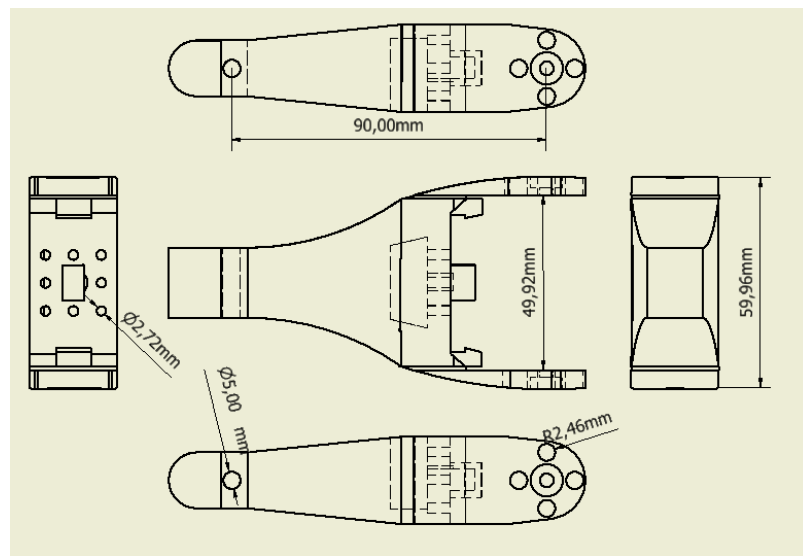


Figura 37 Plano de vistas eslabón 1

3.5.3.2 Eslabón 2

Para la articulación 2 sé tuvo en cuenta que ya se contaba con algunos materiales que son juntas rotacionales de referencia IKO 4.



Figura 38 Rotula ikko LHA 4

| 呼び番号 | 質量 (参考) g | 主要寸法 mm | | | | | | | | | | | | |
|----------|-----------------|----------|----|----|------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|------|------|
| | | ねじ S | V | D | L | L ₁ | L ₂ | l ₁ | W ₁ | W ₂ | D ₁ | D ₂ | ℓ | P |
| LHSA 4 | 11 | M 4×0.7 | *4 | 14 | 25 | 18 | 8 | 4 | 8 | — | 8 | 10 | 19.5 | *5.5 |
| LHSA 5 | 27 | M 5×0.8 | 5 | 17 | 38.5 | 30 | 16 | 5 | 10 | — | 10 | 12 | 32.5 | 8 |
| LHSA 6 | 27 | M 6×1 | 6 | 19 | 39.5 | 30 | 16 | 5 | 10 | — | 10 | 12 | 32.5 | 8 |
| LHSA 8 | 64 | M 8×1.25 | 8 | 24 | 48 | 36 | 19 | 5 | 14 | 14 | 13 | 16 | 41.5 | 10 |
| LHSA 10 | 106 | M10×1.25 | 10 | 28 | 57 | 43 | 23 | 6.5 | 17 | 17 | 15 | 19 | 49 | 12 |
| LHSA 10M | 106 | M10×1.5 | 10 | 28 | 57 | 43 | 23 | 6.5 | 17 | 17 | 15 | 19 | 49 | 12 |
| LHSA 12 | 180 | M12×1.75 | 12 | 34 | 67 | 50 | 27 | 6.5 | 19 | 19 | 17.5 | 22 | 64 | 14 |
| LHSA 14 | 260 | M14×2 | 14 | 38 | 76 | 57 | 30 | 8 | 22 | 22 | 20 | 25 | 72 | 17 |

Figura 39 Tomado de catálogo iko

En base a estas juntas se prosiguió a elegir unas varillas para la fabricación de la articulación 2, estas tienen que ser resistentes y livianas. Se eligió varillas de fibra de vidrio estas se acoplaron perfectamente a las juntas rotacionales.



Figura 40 Eslabón 2

Fuente: Autor

3.5.4 Construcción delta

Una vez los materiales y piezas han sido conseguidos e impresos se procede a la construcción del robot brazo por brazo.



Figura 41 Ensamble de un brazo del robot

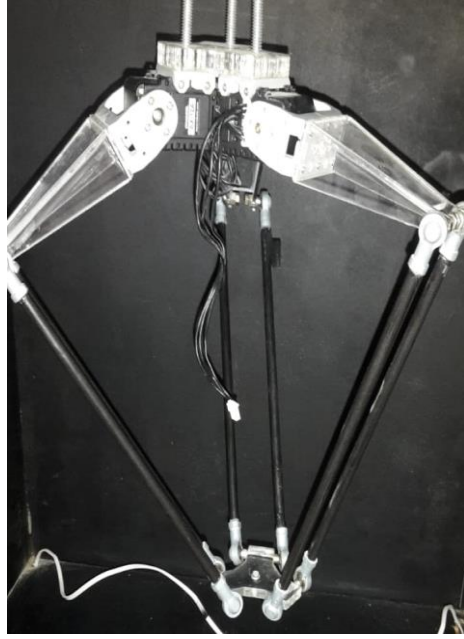


Figura 42 Robot antes del redimensionamiento



Figura 43 Robot después del redimensionamiento

3.5.5 Efecto final

El efecto final representa la herramienta especial que permite al robot de uso general realizar una aplicación particular. Esta herramienta especial debe diseñarse específicamente para la aplicación.

Los efectos finales pueden dividirse en dos categorías: pinzas y herramientas. Las pinzas se utilizarían para tomar un objeto, normalmente la pieza de trabajo, y sujetarlo durante el ciclo de trabajo del robot. Hay una diversidad de métodos de sujeción que pueden utilizarse, además de los métodos mecánicos obvios de agarrar la pieza entre dos o más dedos. Estos métodos suplementarios incluyen el empleo de casquetes de sujeción, imanes, ganchos, y cucharas.

El peso bajo para permitir tener un manejo de la carga más útil, aumento de las aceleraciones, tiempo de ciclo en disminución. Se deben considerar dimensiones mínimas dispuestas por el tamaño de la pieza de trabajo, y despachos de aduanas de área de trabajo.

Para este robot se seleccionó como herramienta de sujeción un electroimán estos son de tipo de accionamiento eléctrico lo cual son perfectamente útiles ya que las piezas son ferromagnéticas.

Algunas ventajas de los electroimanes son:

- Capacidad de activarse o desactivarse: Podemos activar y desactivar el electroimán cuantas veces sea necesario, solo basta con interrumpir o no la corriente eléctrica.
- La velocidad de accionamiento esta van desde los 20 a los 300 ms lo da una buena reacción.

Datos técnicos

Voltaje de entrada: 5V 12V 24V DC

Consumo de energía: 2W

Fuerza de retención: 2.5kg

Material: Metal

Color: plateado

Diámetro: 20 mm

Diámetro central: 8 mm

Altura: 15 mm



Figura 44 Minisolenoides electroimán



Figura 45 Minisolenoides ensamblado

3.5.6 Actuadores robot delta

Los servomotores Dynamixel AX-12A están diseñados específicamente para aplicaciones de robótica y minirobótica. Cada servomotor está microcontrolado de tal manera que se puede tener acceso a diferentes variables: velocidad rotacional, temperatura, carga y voltaje.

Estos servomotores están protegidos contra sobre voltaje, sobrecalentamiento además de condiciones predefinidas de error.

- Especificaciones:
- Peso: 53.5g (AX-12/AX-12), 54.6g (AX-12A)
- Dimensiones: 32mm * 50mm * 40mm
- Relación de reducción en el engranaje: 254:1
- Torque máximo: 15kgf.cm(12.0V, 1.5A)
- Velocidad sin carga: 59rpm (12V)
- Rango de movimiento: 0 a 300°
- Giro continuo
- Temperatura de trabajo: -5°C hasta 70°C
- Voltaje de trabajo: 9-12V (se recomiendan 11.1V)
- Señal de comando: Paquete de datos digitales.
- Tipo de protocolo: Comunicación serial asíncrona tipo half duplex (8bits, 1 stop, sin paridad)
- Identificador: 0 a 253
- Velocidad de comunicación: desde 7343bps hasta 1Mbps
- Retroalimentación: Posición, temperatura, carga, voltaje de entrada, etc.



Figura 46 servomotor AX-12A

Fuente: AX-12ª manual

3.5.7 USB2Dynamixel

Controla cualquier red de actuadores Dynamixel desde un PC como controlador, estableciendo la comunicación con el bus Dynamixel a través del puerto USB, sin pasar por el controlador (CM-5, CM-510 o CM-700). Con librerías de soporte para programar en los entornos Matlab, Microso Visual Studio, Java, C/C++, C#, Visual Basic, Python y LabVIEW.

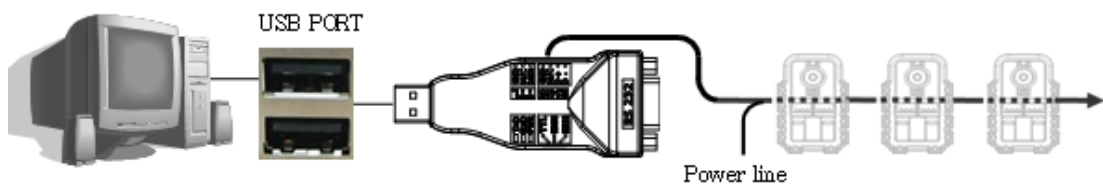


Figura 47 Conexión de USB2Dynamixel con servos AX-12^a

3.6 CONTROL CINEMÁTICO ROBOT

El control cinemático tiene como función principal establecer las trayectorias que debe seguir cada articulación del robot delta a lo largo del tiempo para lograr los objetivos fijados por el sistema de visión, (punto de destino, trayectoria cartesiana del efector final del robot, tiempo invertido en ejecutar la tarea, etc.). Estas trayectorias se seleccionan atendiendo a las restricciones físicas propias de los accionamientos y a ciertos criterios de calidad de trayectoria, como suavidad o precisión de la misma.

Existen múltiples tipos de trayectorias para que un robot puede llegar a los puntos objetivos o deseados, en este trabajo se seleccionó las trayectorias isócronas o coordinadas en el cual el sistema del robot delta realiza la interpolación entre los puntos especificados, de forma tal que, que le sea posible realizar el control de movimientos para que el robot pase por todos los puntos objetivos.

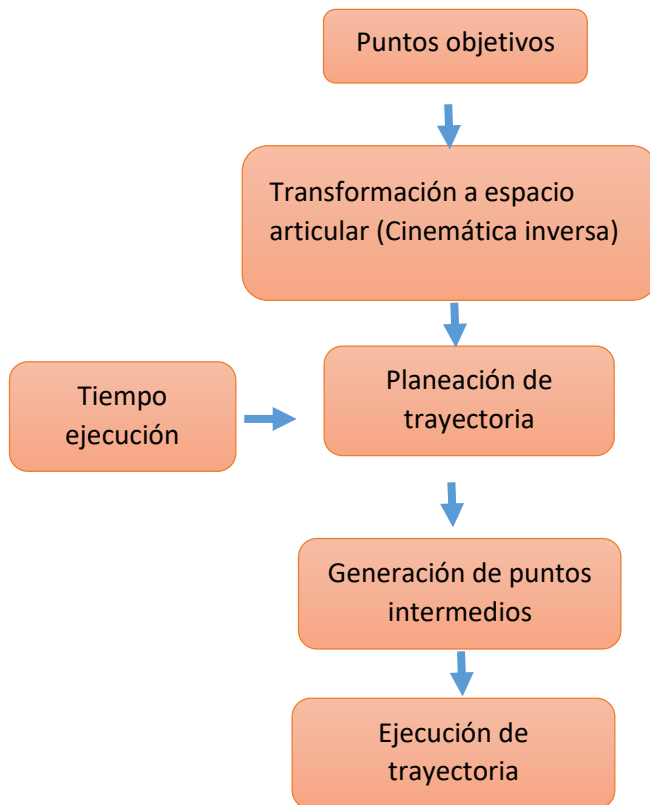


Figura 48 Esquema control robot delta

Las trayectorias punto a punto se pueden especificar en tres: movimiento eje a eje, movimiento de ejes simultáneo y coordinado o isócrono.

Como la trayectoria a implementar será ejecutada por un robot el cual es capaz de alcanzar puntos en el espacio siempre y cuando los actuadores de las articulaciones estén sincronizadas, se optó por que la trayectoria punto a punto sea de tipo coordinada o isócrona, las cuales permiten que todas articulaciones inician el movimiento y terminen al mismo tiempo. Esto se logra disminuyendo la velocidad aquellas articulaciones que tienen un recorrido más corto con respecto a las otras.

3.6.1 Planeación trayectoria

Inicialmente se tienen el punto objetivo que corresponde a la localización del objeto que es calculado por el sistema de visión, se le asignan los puntos que corresponden al recorrido que efectuara el robot (estos puntos son pre-terminados) y se interpolan linealmente entre cada punto consiguiendo el muestreo intermedio entre puntos.

$$f_1(x) = \frac{x - x_0}{x_1 - x_0} f(x_0) + \frac{x_1 - x}{x_1 - x_0} f(x_1)$$

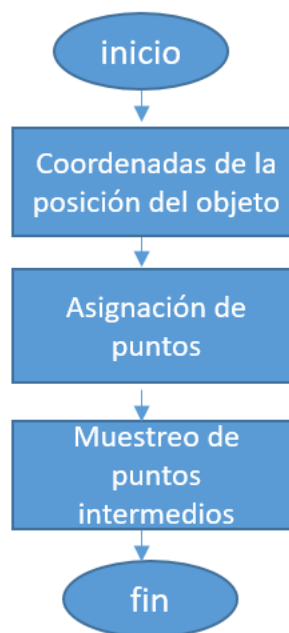


Figura 49 Esquema de generación de trayectoria cartesiana

Fuente: Autor

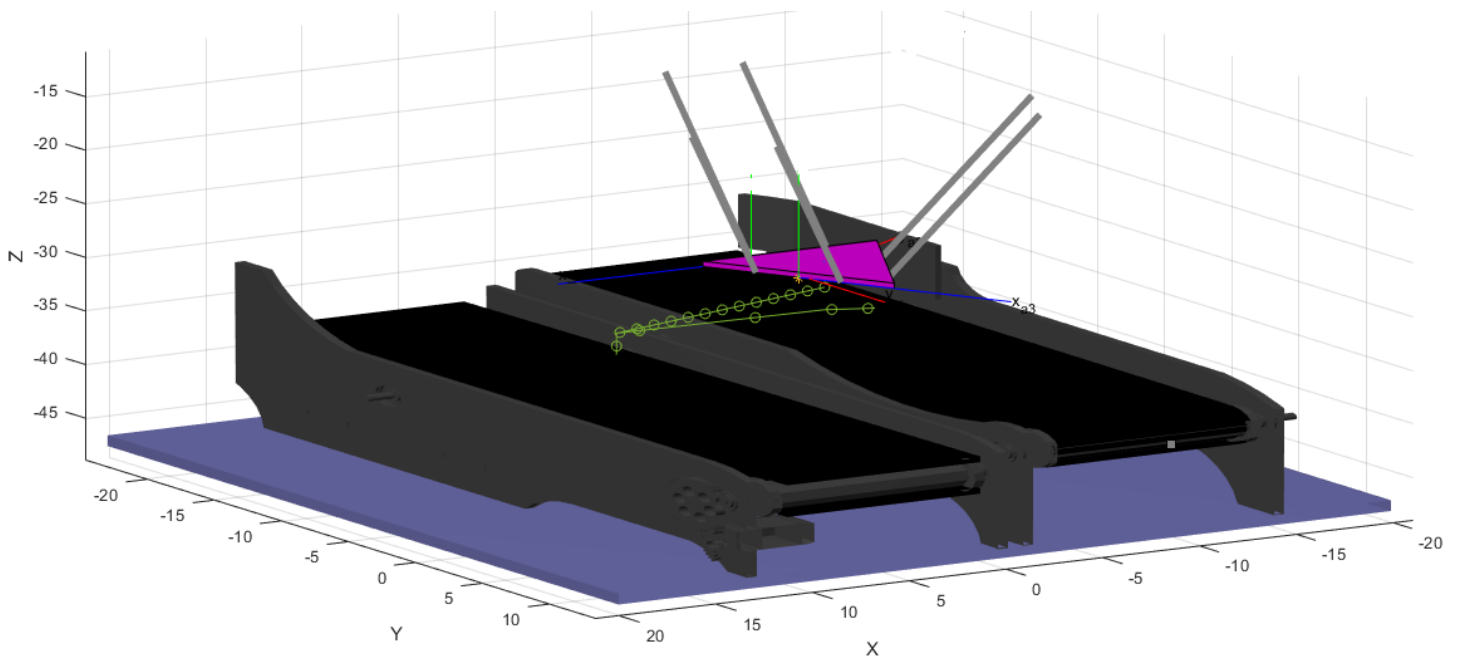
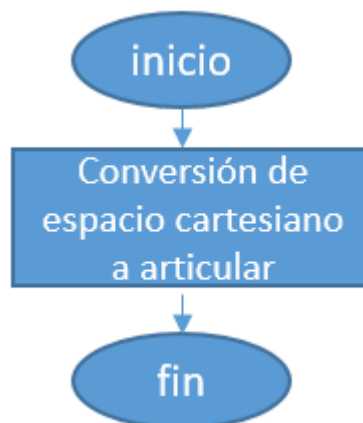


Figura 50 Trayectoria a seguir por el robot

Fuente: Autor

Una vez se tienen los puntos cartesianos se procede a realizar la conversión a espacio articular mediante la cinemática inversa que ya ha sido estudiada en capítulos anteriores. Obteniendo de esta manera todos los ángulos de abertura de cada uno de los actuadores. Figura 51.



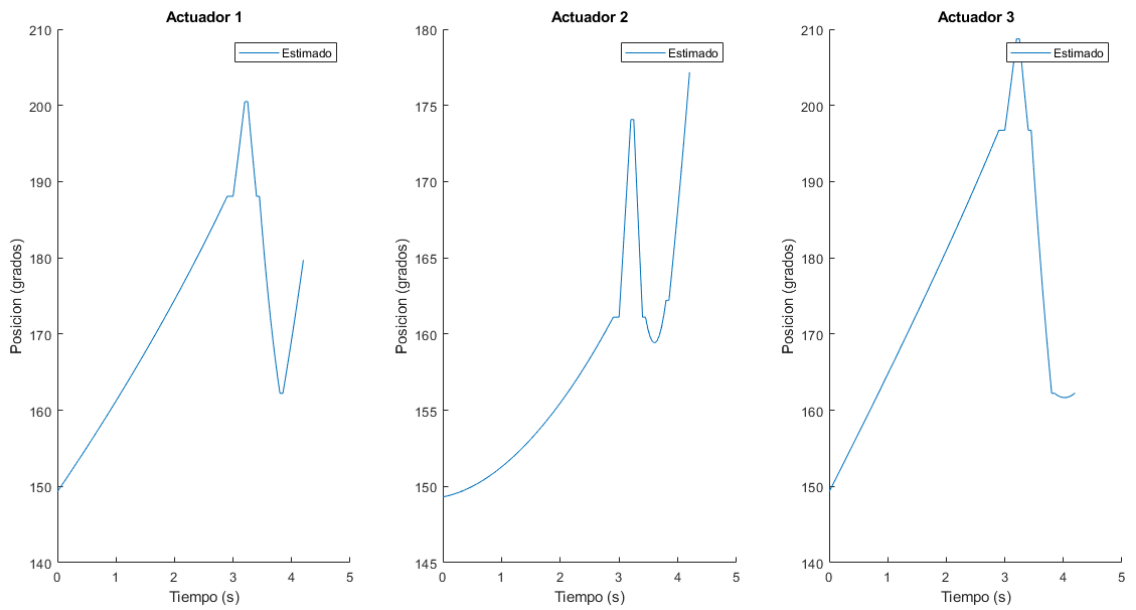


Figura 51 Recorrido articular de cada actuador
Fuente: Autor

Una vez obtenido los puntos articulares se procede al cálculo de las velocidades entre cada punto articular. Con de que el robot pueda cumplir los tiempo y poder tomar el objeto que se encuentra en movimiento. Figura 52.

$$q(t) = (q^i - q^{i-1}) \frac{t - t^{i-1}}{T} + q^{i-1} \quad t^{i-1} < t < t^i$$

$$T = t^i - t^{i-1}$$

Fuente: Barrientos

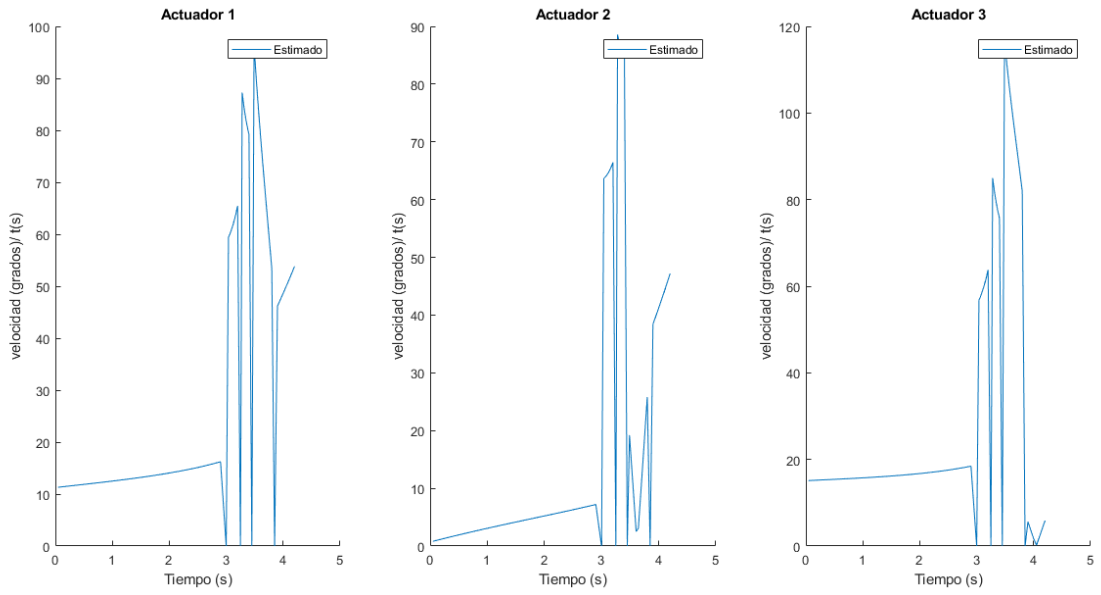


Figura 52 Velocidades calculadas para cada actuador
 Fuente: Autor

Con los valores de posición y velocidad articular calculados se procede a ejecutar la trayectoria, que corresponde a sujetar la pieza y transportarla a la siguiente banda. Figura 53.

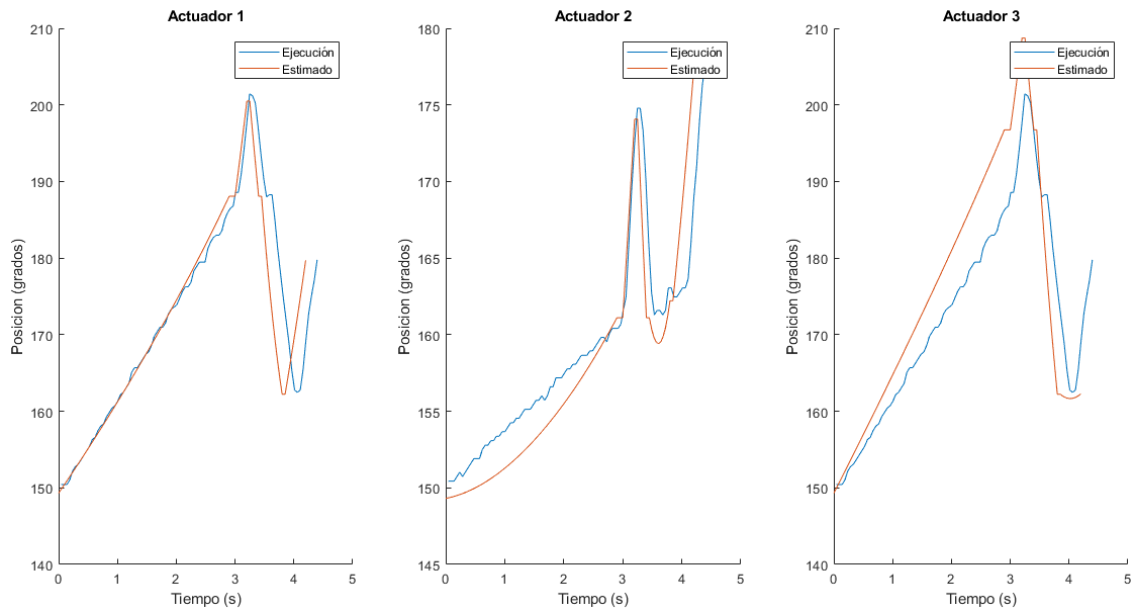


Figura 53 Graficas de posiciones articulares estimadas vs ejecutadas por el robot

3.7 DESARROLLO DEL SISTEMA DE VISIÓN ARTIFICIAL

Este sistema tiene como objetivo reconocer los objetos que se encuentran en la banda transportadora, analizar cuantos hay y en que posiciones se encuentran localizados con respecto al robot delta mediante visión artificial.

La toma de imágenes se realiza mediante una cámara web a través del software MATLAB. Donde primero se realizaron configuraciones de los parámetros de la cámara para así obtener una mejor captura. Entre los parámetros se encuentran:

| | |
|-----------------------------------|---|
| Resolución de la cámara | 640x480 pixeles |
| Canal de entrada del video | 2 (el canal 1 corresponde a la cámara integrada del pc) |
| Formato de color de entrada | RGB |
| Intervalos de captura de cuadros. | 1 (cuadro a cuadro) |

Tabla 2 Parámetros cámara
Fuente: Autor

3.7.1 Procesamiento de imagen

El proceso inicia con la captura de la imagen, una vez tomada la imagen se realiza un recorte de la imagen de tal forma que solo queda el espacio de la cinta transportadora, como la cámara se encuentra en una posición fija ayuda a que el recorte sea mucho más sencillo. Figura 50.

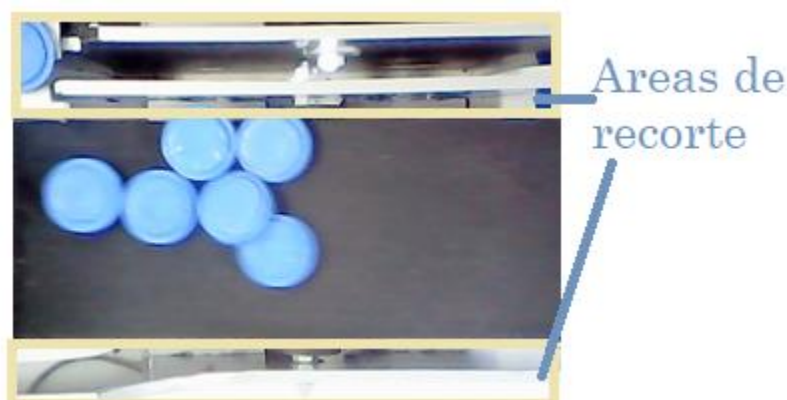


Figura 54 Imagen captura por cámara web
Fuente: Autor

```
rgb=rgb(112:372,,:);
```

 Se eliminan las áreas mencionadas en la figura 50

Figura 55 Imagen recortada

Fuente: Autor

Una vez la imagen es recortada se realizan unos procesos para poder obtener las posiciones de los objetos y cuantos hay en la banda. La imagen es transformada en una imagen binaria dejando solo dos tonos (blanco y negro), los más claros en este caso los objetos se tornan blancos y los más oscuros de color negro. Figura 44, se remueven los pequeños objetos o ruidos de la imagen (menor a 20 pixeles) teniendo en cuenta su cantidad de pixeles permitiendo así una imagen más limpia.

```
E=im2bw(rgb,0.5);
```

 Binarización de imagen.

Figura 56 Imagen binarizada

Fuente: Autor

```
E = bwareaopen(E,20);
```

 %remueve los objetos con menos de 20 pixeles

Figura 57 Eliminación de píxeles pequeños (menor a 20 píxeles). Fuente: Autor

Se crean los elementos estructurantes en forma de disco ya que los objetos son circulares, especificando el tamaño del radio en pixeles, y se realiza el proceso de erosión de imagen contrayendo un poco los bordes, dejando un poco más definido los límites entre cada objeto. Figura 54.

```
se = strel('disk',10); % Elemento estructurante
E = imerode(E,se); % Erosiona Imagen
```



Figura 58 Erosión

Fuente: Autor

Por último se localizan los círculos y sus centroides con una función de matlab llamada `imfindcircle` este tiene un algoritmo basado en la transformada de Hough, esta transformada encuentran los pixeles candidatos que forman un patrón de círculo con un radio fijo, obteniendo sus centros y el número total del objetos que se encuentren en la imagen, la función retorna toda esta información en una matriz. Figura 55. `[centersBright,radiiBright] = imfindcircles(E,[24 50], ...`

```
'ObjectPolarity','bright','Sensitivity',0.90);
f3=figure(3)
set(f3, 'color', 'w');
title(sprintf('Imagen con centro localizados'))
figure(3)
imshow(rgb)
hBright = viscircles(centersBright, radiiBright,'Color','b');
```

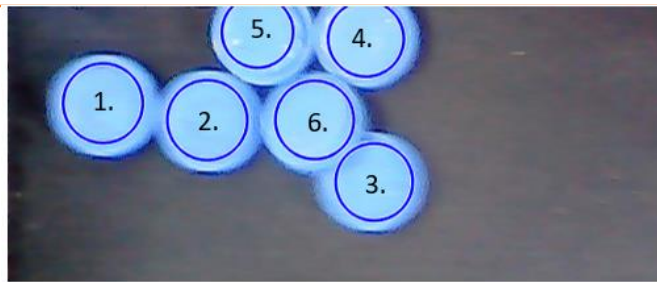


Figura 59 Imagen con los objetos detectados

Fuente: Autor

```

coor_pix =
      81.5486   80.3349
     170.1913   94.6834
     312.0643  147.2797
     305.1320   26.1781
     211.8709   19.9351
     261.4122   94.5144

```

Figura 60 Coordenadas en pixeles con referencia a la cámara.
Fuente: Autor

La matriz retornada tiene coordenadas en pixeles se procede a hacer una conversión a medidas longitudinales (cm) usando regla de tres, esto se logra gracias a la posición fija de la cámara, ya que esta siempre capta el mismo espacio (Figura 57), hecha la transformación de medidas, se transforman las medidas con respecto al sistema de referencia del robot y se reordenan de tal forma que las más cercanas al delta se ubican de primero en la matriz además de esto se calcula el tiempo aproximado que en estará en un rango accesible al robot.

$$680 \text{ pix} = 26 \text{ cm} \text{ Entonces } X_{cm} = \frac{26 \text{ cm} * X_{pix}}{680 \text{ pix}}$$

$$X \text{ pix} = X_{cm}$$

$$240 \text{ pix} = 10 \text{ cm} \text{ Entonces } Y_{cm} = \frac{10 \text{ cm} * Y_{pix}}{240 \text{ pix}}$$

$$Y \text{ pix} = Y_{cm}$$

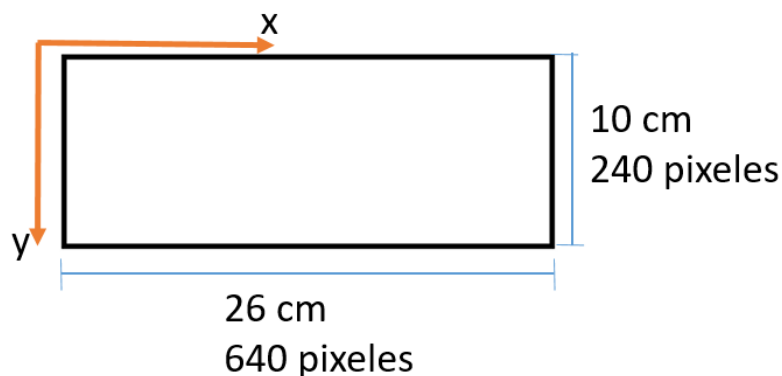


Figura 61 Equivalencia de medidas pixeles y cm
Fuente: Autor

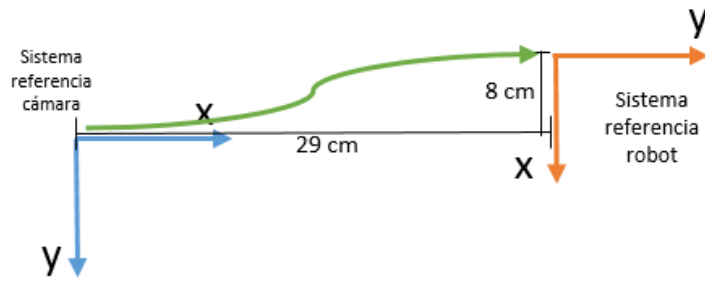


Figura 62 Distancia entre ejes coordenados del robot y la cámara
Fuente: Autor

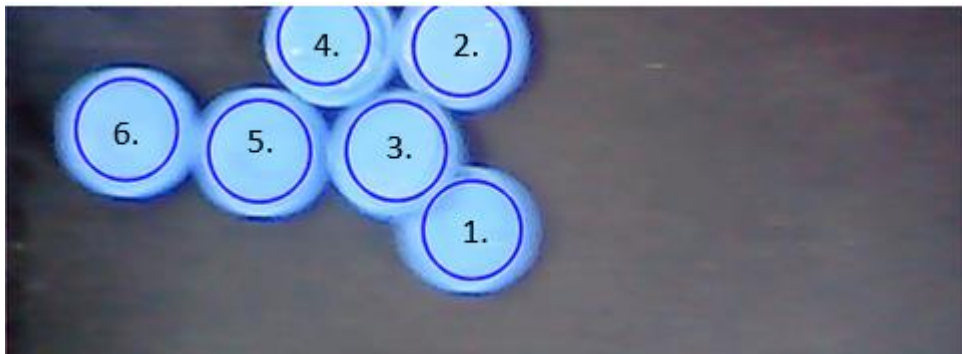


Figura 63 Objetos reordenados en orden de prioridad para el robot
Fuente: Autor

| pos | x | y | tiempo |
|-----|--------|----------|--------|
| 1 | 9.8448 | -16.5538 | 3.6106 |
| 2 | 5.4500 | -16.8192 | 3.6935 |
| 3 | 7.9300 | -18.4928 | 4.2165 |
| 4 | 5.2235 | -20.3893 | 4.8092 |
| 5 | 7.9361 | -21.9849 | 5.3078 |
| 6 | 7.4154 | -25.3782 | 6.3682 |

Figura 64 Posiciones recalculadas con referencia al robot
Fuente: Autor

3.8 ACOPLAMIENTO DE SISTEMA

Al integrar todos los sistemas para el funcionamiento de la célula, se optó por la realización del siguiente diagrama de flujo del sistema. A continuación se explica paso a paso lo que en cada bloque se cumple.

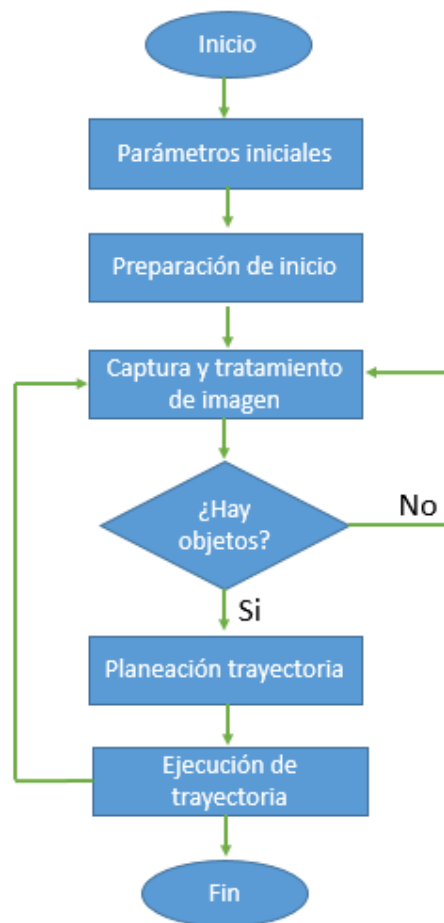


Figura 65 Acoplamiento de los sistemas

Fuente: Autor

Parámetros iniciales: En este proceso se cargan los parámetros respectivamente de cada sistema como lo es los parámetros y librerías para controlar el robot delta, los parámetros e inicio de la cámara web, la comunicación con el micro controlador vía bluetooth por medio del módulo HC-05.

Preparación de inicio: Una vez cargadas las configuraciones el robot se prepara para iniciar moviéndose automáticamente a la posición de inicio, la persona a cargo debe activar la iluminación manualmente, entonces el programa manda un mensaje que está listo para comenzar.

Captura y tratamiento de imagen: Una vez se presiona inicio, el microcontrolador pone en marcha las bandas y la cámara comienza con la captura de imágenes hasta identificar los objetos, luego se realiza el respectivo

procesamiento donde identifica cuantos hay, en que posiciones se encuentran toda esta información es enviada al robot.

Planeación de trayectoria: Se reciben las coordenadas con respecto al robot y estas se transforman a coordenadas articulares, se traza la ruta a seguir en tiempo calculados para que sea posible sujetar las piezas a tiempo.

Ejecución trayectoria: Una vez tiene las rutas trazadas el robot se encarga de seguirla llegando a los puntos donde se encuentran los objetos y trasladándolos a la banda siguiente, cuando esta al alcanza de la pieza u objeto el programa manda una señal al micro controlador para activar el efector final y así mismo cuando para soltarla.

4. RESULTADOS

4.1 PRUEBAS DE VELOCIDAD

A la hora de generar las trayectorias a ejecutar hay que tener algunas consideraciones una de estas son las velocidades máximas de los actuadores del robot delta, ya que se puede dar el caso en que las velocidades calculadas son mayores a las del actuador.

Se realizaron unas pruebas experimentales para encontrar las velocidades máximas, estas pruebas consistían en mover el actuador de una posición inicial a una final que son fijas, variando la velocidad. El actuador recorrió un ángulo total de 180 grados, aumentando gradualmente su velocidad entre recorrido realizado, hasta el punto en el que el actuador no realizar recorrido en menor tiempo, como resultado se obtuvo que la velocidad máxima es de 305 grados/s.

Las pruebas experimentales se ejecutaron con carga (300 gramos) figura 51 y sin carga (libre) figura 50. Alimentados a 12V fuente ATX, el actuador se probó de manera individual es decir que este no se encontraba ensamblado en el momento de la prueba ya que los ángulos alcanzados podían causar destrozos a la estructura paralela.

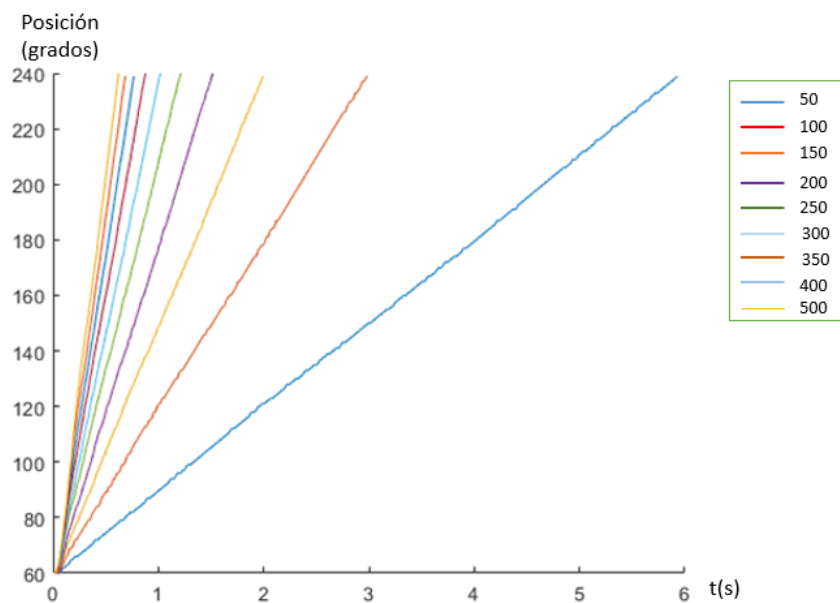


Figura 66 prueba de velocidad del actuador sin carga Fuente: Autor

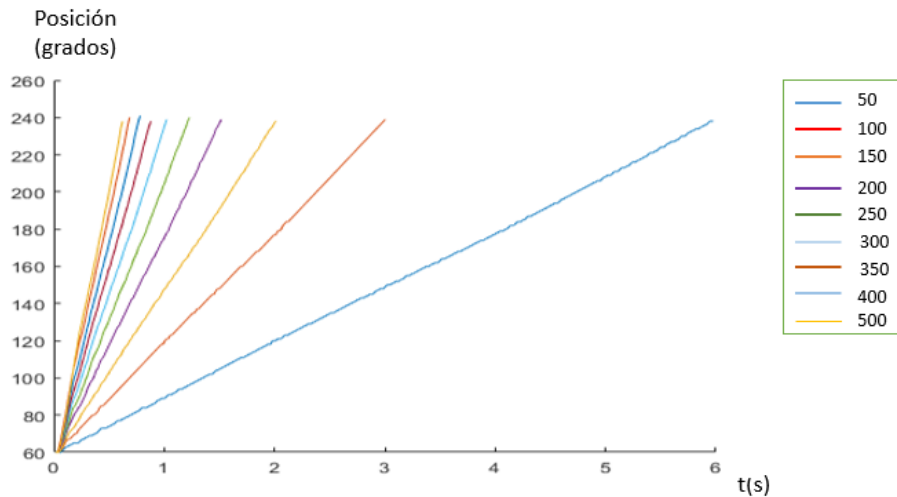


Figura 67 Prueba de velocidad del actuador son carga

Fuente: Autor

4.2 CALIBRACIÓN MOTORES

Para una mayor precisión a la hora de ejecutar las trayectorias fue necesario calibrar los actuadores del robot delta, para esto se calculó una constante la cual puede variar dependiendo del voltaje de alimentación.

Esta constante relaciona el valor de registro que tiene un rango de 0 a 1023 con la velocidad de movimiento 0 a 300 rad/s. Las figuras 68 y 69 muestran los valores que tomo la constante a lo largo de cada recorrido.

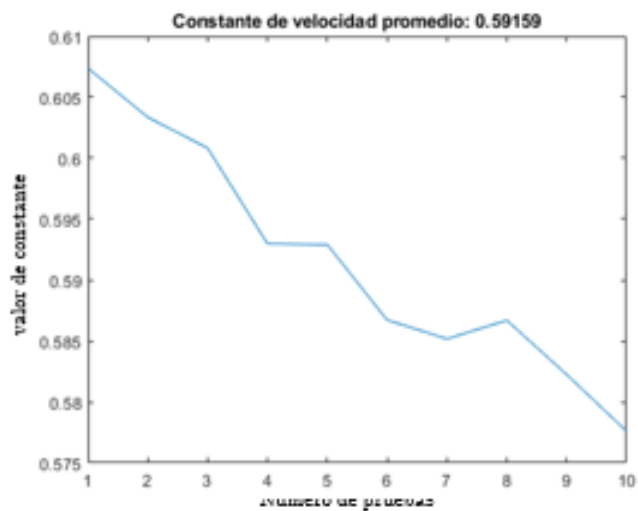


Figura 68 Constante promedio sin carga

Fuente: Autor

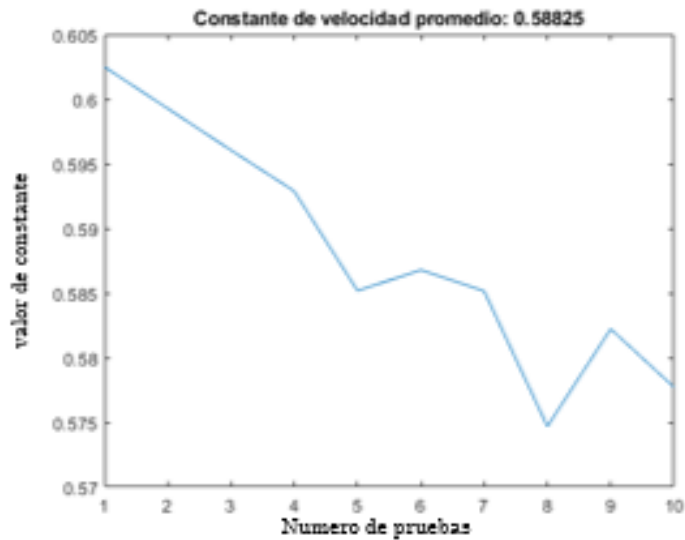


Figura 69 Constante promedio con carga

Fuente: Autor

4.3 PRUEBAS DE SUJETACION DE OBJETOS

Se realizaron una serie de pruebas para comprobar cuantas piezas es capaz de transportar, teniendo en cuenta los tiempos en que debería ejecutarlos y los tiempos reales, comenzando con una sola pieza luego dos hasta el máximo que pueda tomar.

Un solo objeto

Se inició realizando pruebas con una pieza en la banda para observar si los sistemas estaban correctamente sincronizados las siguientes figuras muestran el comportamiento estimado y el de ejecución de cada actuador al ejecutar la tarea de sujetar la pieza.

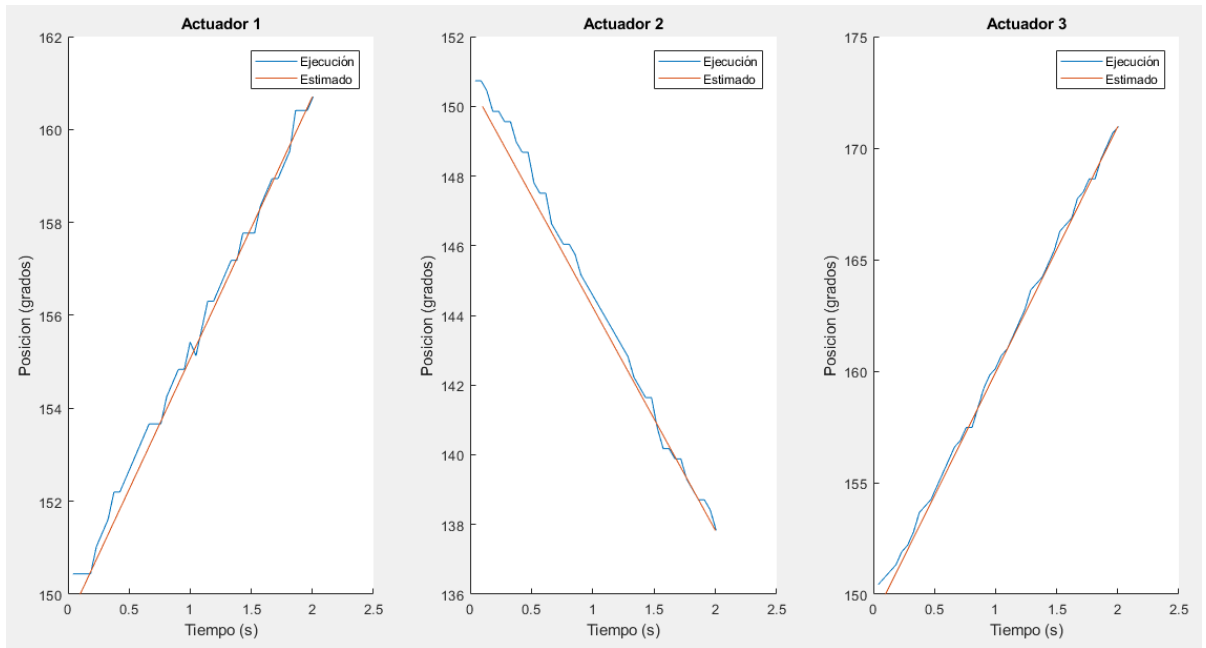


Figura 70 Tiempos de ejecución de trayectoria de los actuadores

Fuente: Autor

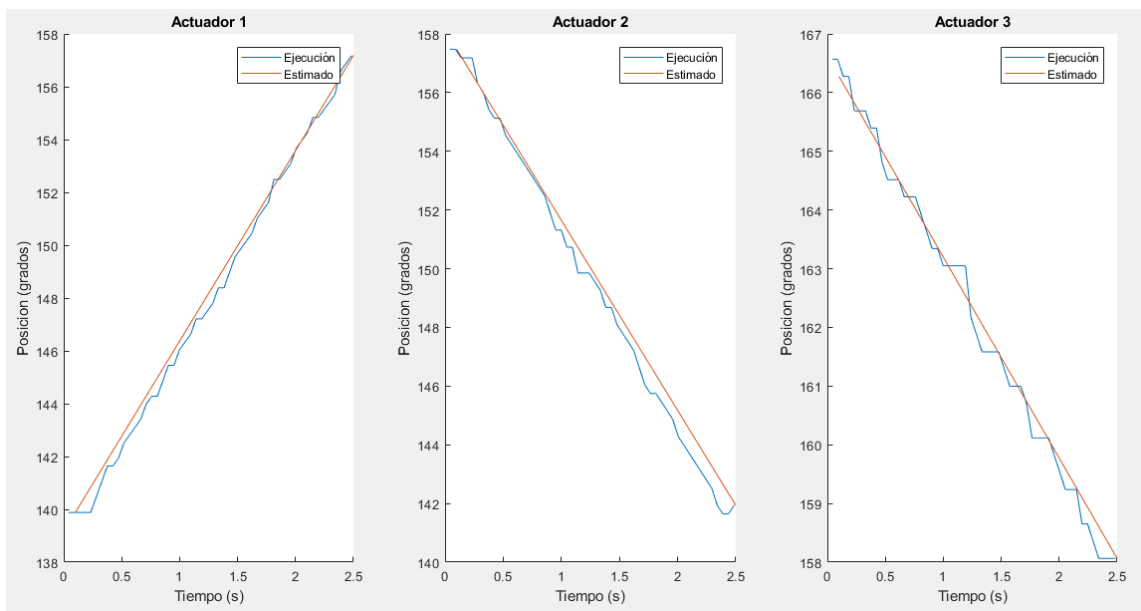


Figura 71 Tiempos de ejecución de trayectoria de los actuadores

Fuente: Autor

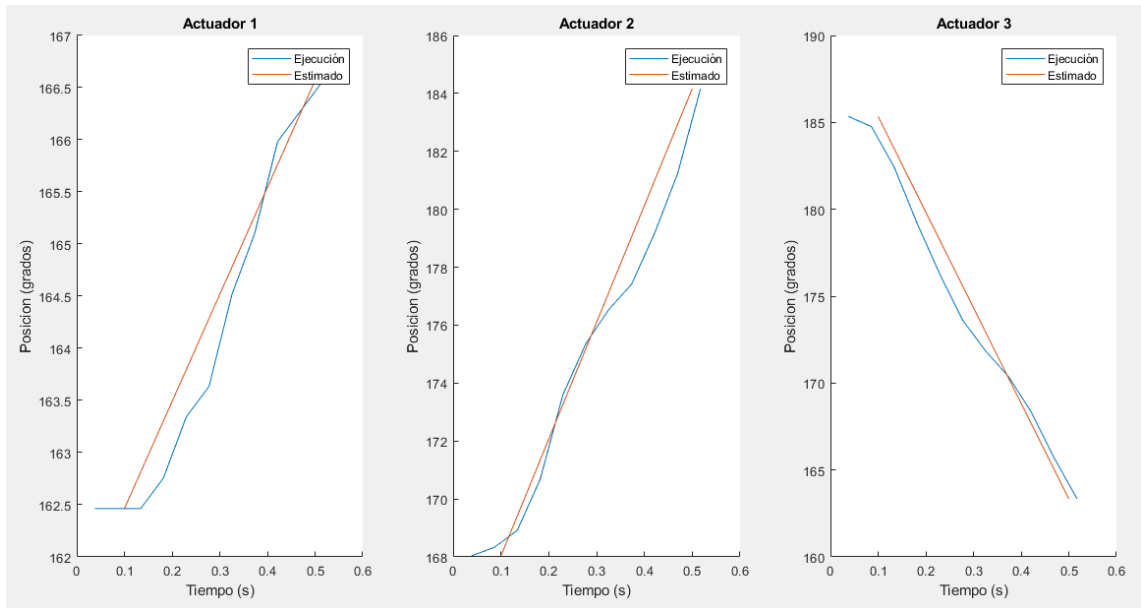


Figura 72 Tiempos de ejecución de trayectoria de los actuadores
Fuente: Autor

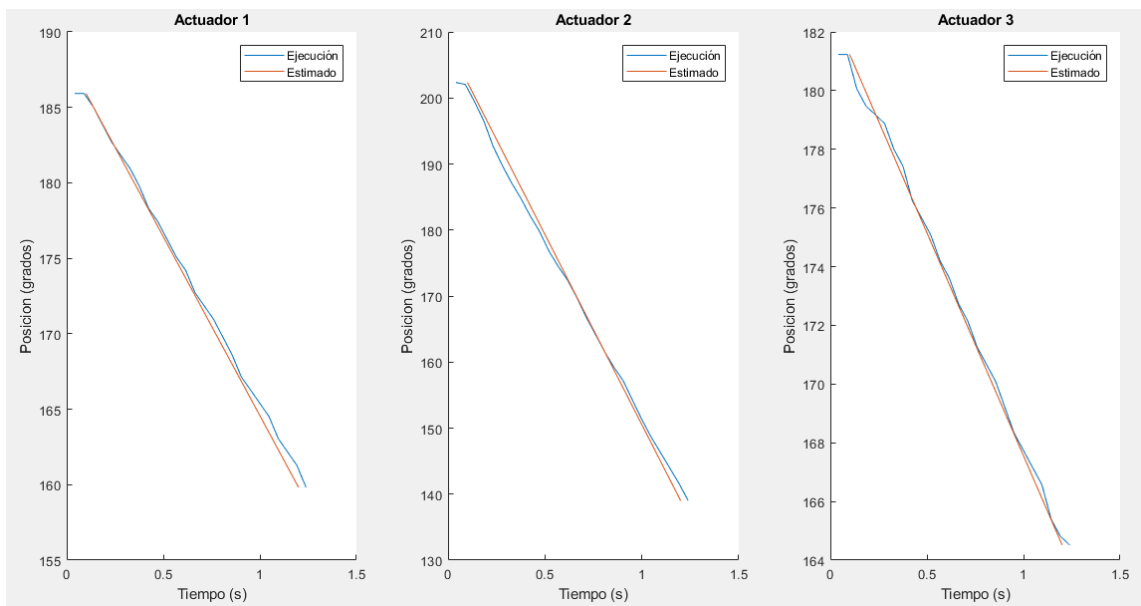


Figura 73 Tiempos de ejecución de trayectoria de los actuadores

Fuente: Autor

Las anteriores figuras (70 71 72 73) muestran los resultados obtenidos de cada actuador del robot tras ejecutar la trayectoria planificada, dando constancia de que el control si esta ejerciendo.

| Tiempos de ejecución | Tiempos estimados |
|----------------------|-------------------|
| 2.006 seg | 2 seg |
| 2.4849 seg | 2.5 seg |
| 0.5173 | 0.5 seg |
| 1.239 | 1.2 |

Tabla 3 Datos de tiempos ideales vs reales

Fuente: Autor

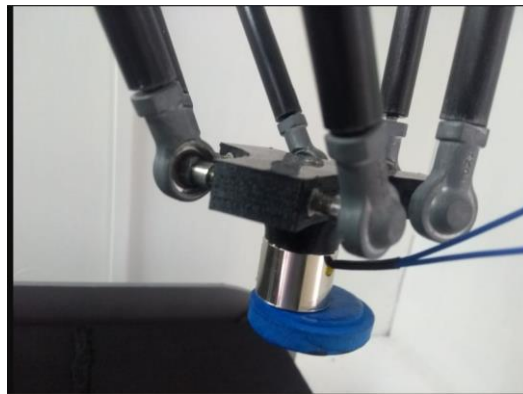


Figura 74 Efector sujetando la pieza

Fuente: Autor

Con dos o más objetos

Se realizaron pruebas con varios objetos aumentándolos gradualmente y se comenzaron a percibir que la dificultad aumento a partir de cuatros objetos cuando se encuentran muy cercanos. Obteniendo datos de la separación mínima que debe haber entre objetos para que realice la tarea con éxito, esta medida de 3 cm entre centros de cada pieza.

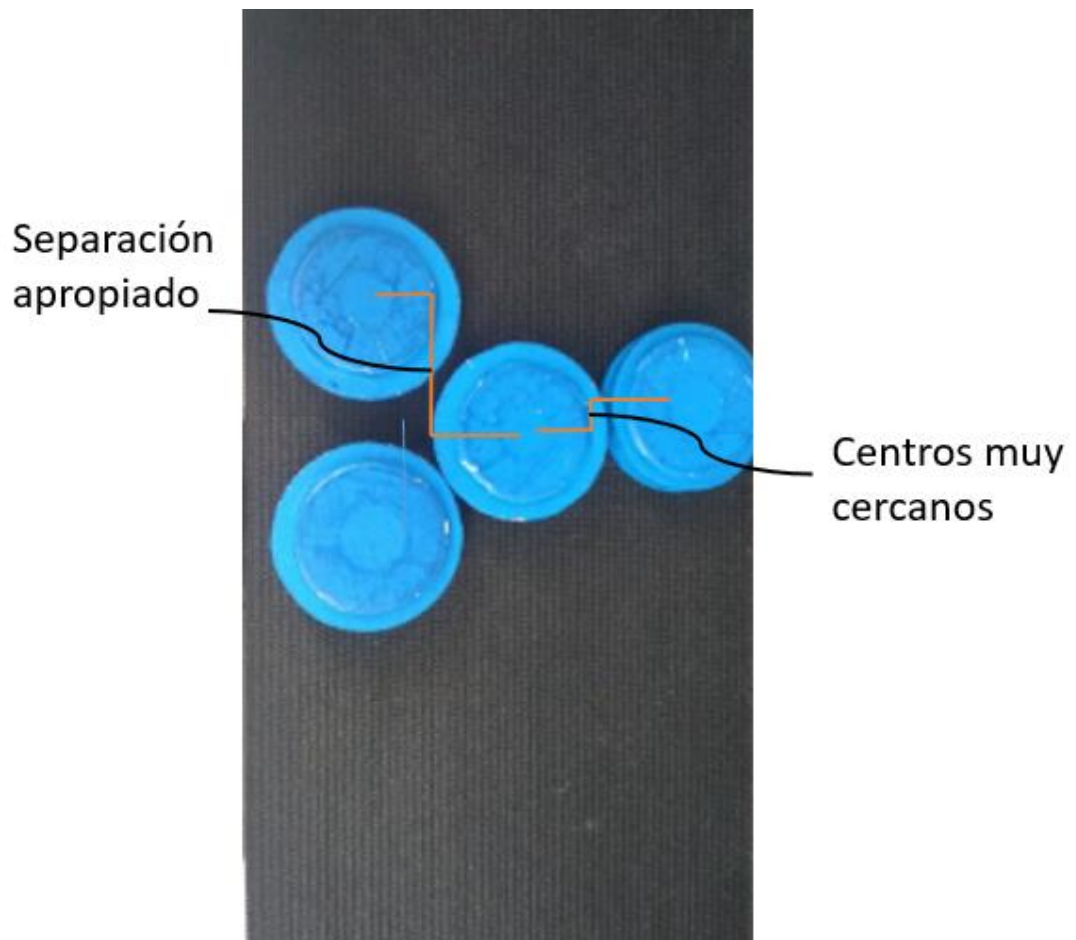


Figura 75 Separación entre objetos
Fuente: Autor

Se realizó múltiples ensayos con 4 objetos con distintas separaciones en cada ensayo obteniendo cuantas no lograban ser alcanzadas por el robot

| Distancia entre objetos | No acertadas |
|-------------------------|--------------------------|
| 1 cm | 3 objetos |
| 1.5 cm | 2 objetos con dificultad |
| 2 cm | 2 objetos |
| 2.5 cm | 1 objetos con dificultad |
| 3 cm | Ninguno |
| 3.5 | Ninguno |

Tabla 4

Fuente: Autor

4.4 TIEMPOS DE EJECUCIÓN DE FUNCIONES

Una consideración muy importante en este proyecto es tener en cuenta el tiempo de ejecución de las funciones encargadas ya estas pueden retrasar un poco la ejecución debido a esto se dio la tarea de ejecutar función por función y conocer el tiempo promedio en finalizar cada función. Tabla 5.

Estos tiempos se tiene en cuenta en la planeación de trayectoria del robot

| <i>Nombre Funciones</i> | <i>Tiempo</i> |
|---------------------------------|---------------|
| crd_cin_dir_delta | 0.012 |
| crd_cin_inv_delta | 0.044 |
| crd_posicion_objetos | 0.243 |
| crd_proceso_imagen | 0.108 |
| crd_verificar_velocidad_motores | 0.0189 |
| crd_verificar_mecanica | 0.017 |
| getsnapshot | 0.044 |

Tabla 5 Tiempo de ejecución de funciones

Fuente: Autor

4.5 MÁXIMO DE PIEZAS ALCANZADAS EN UN MINUTO

Se realizaron pruebas del máximo de piezas que puede sujetar en un minuto. Tomando los datos de cuantas lograban ser alcanzadas por el robot, este proceso se realizó repetidamente hasta tener un estimo promedio. Figura 72.



Figura 76 Número de piezas sujetadas en cada prueba

Fuente: Autor



Figura 77 Estructura completa

Fuente: Autor

5 RECOMENDACIONES

- El proceso puede mejorar implantando el control de variación de velocidad en las bandas en vez de una velocidad fija, de esta manera poder incrementar la velocidad de las bandas cuando detecte pocos objetos o disminuirla cuando se detecte cantidades grandes.

CONCLUSIONES

- En el diseño y construcción de la celda robotizada, se aplican varios conocimientos los cuales en conjunto permiten el desarrollo óptimo de este tipo de máquinas, por lo cual es importante el saber interconectar y poner en funcionamiento distintos sistemas y elementos como actuadores, sistemas digitales y electrónicos, los cuales toman importancia al interactuar entre todos como sucede en la mayoría de maquinarias e industrias.
- Se puede resaltar la vital importancia del software de diseño tipo CAD en el momento de diseñar y crear todos los elementos utilizados para realizar un prototipo de la celda robótica, ya que estas permiten visualizar el aspecto, teniendo en cuenta sus medidas reales, posibles impactos o errores que se pueden llegar a presentar durante el ensamblaje y manejo.
- Se desarrolló un sistema de visión artificial a partir de la detección de elementos de distintas características, este tipo de sistema es de vital importancia en los procesos de detección, ya que estos proporcionan la información necesaria para que el robot pueda ejecutar la tarea de manera correcta, utilizando procesos de tratamiento de imágenes, en este caso haciendo uso de la transformada de Houg, que permite el rastreo de los objetos circulares.
- El uso de algoritmos de optimización de espacio en el diseño de estructuras robóticas fue de vital importancia ya que permitió calcular las dimensiones más apropiadas de las articulaciones del robot, cumpliendo así las exigencias del área de trabajo.
- Una de las características no ventajosas de los robots paralelos es el reducido espacio de trabajo con que se cuenta, debido a las restricciones

de tipo mecánico y la presencia de singularidades al momento de llegar a ciertas posiciones.

- En ese trabajo se implementó el modelo cinemático inverso del manipulador delta y se implementó con éxito en el prototipo real. Este fue de utilidad para mover Los mismos fueron de utilidad para mover el efector final del manipulador en el espacio cartesiano. Es decir, se logró mover las tres articulaciones al mismo tiempo al ingresar la posición espacial final deseada.

REFERENCIAS

- [1] Jean-Pierre Merlet. *Parallel Robots*. Kluwer Academic-Publishers, 2000.
- [2] S. G. Whitehall, Gough. Universal tire test machine. In *Proceedings 9th Int. Technical Congress F.I.S.I.T.A.* Vol. 117, pp. 117-135. Mayo, 1962.
- [3] Stewart D. A plataforma with 6 degrees of freedom. En *Proc. of the Institution of mechanical engineers*, 180(Parte 1,15):371-386, 1965.
- [4] S.B. Niku. *Introduction to Robotics: Analysis, Systems, Applications*. Prentice Hall, New Jersey, 2001.
- [5] PÉREZ, Pedro Armando Juárez. Implementación de un Control Supervisorio Difuso para el Control de Posición de un Robot Manipulador Delta-Edición Única. Pp 19 – 22.
- [6] BARRIENTOS, Antonio. *Fundamentos de robótica*. e-libro, Corp., 2007. Pp. 401-421
- [7] BUENO SÁNCHEZ, Cristian Alejandro; MALDONADO SORIA, Alejandro Emmanuel. *Diseño, construcción e implementación de una plataforma tipo Stewart para control de una aplicación bola en plataforma a través de visión artificial*. 2017. Tesis de Licenciatura. Universidad de las Fuerzas Armadas ESPE. Carrera de Ingeniería en Mecatrónica.
- [8] NOPE, Sandra E.; LOAIZA, Humberto; CAICEDO, Eduardo. Estudio Comparativo de Técnicas para el Reconocimiento de gestos por Visión Artificial. *Avances en Sistemas e Informática*, 2008, vol. 5, no 3, p. 127-134.
- [9] SIERRA ÁLVAREZ, Santiago, et al. *Sistema de medición de objetos basado en visión artificial*. 2012. Tesis de Licenciatura. Universidad EAFIT.
- [10] ELERA, Juárez; EDUARDO, Juan; RODRÍGUEZ GONZALES, Luis Gustavo. Implementación de un Algoritmo para la Detección de Cuerpos Extraños de las Leguminosas de Grano Seco Utilizando la Visión Artificial. 2017.
- [11] GARNICA, Aucapiña, et al. *Diseño e implementación de un sistema de visión artificial para la manipulación, ensamblaje y control de calidad de piezas a través del*

uso de un brazo robótico para el Laboratorio de Mecatrónica. 2019. Tesis de Licenciatura. Universidad de las Fuerzas Armadas ESPE. Carrera de Ingeniería en Mecatrónica.

[12] JARRÍN VIVAR, Jorge Darío; GALLEGOS LÓPEZ, Cristian Fernando. *Diseño e implementación de un módulo didáctico para clasificación, empaqueo e inspección de huevos aplicando PLC*. 2013. Tesis de Licenciatura.

[13] SANTAMARÍA, Albert Lester Balmaceda; CASTAÑEDA, Eduardo Castillo. Metodología de diseño de un robot paralelo tipo Delta de 3-GDL en función de un espacio de trabajo prescrito. *Nexo Revista Científica*, 2013, vol. 26, no 1, p. 24-33.

[14] ROJAS, Teddy V.; SANZ, Wilmer; ARTEAGA, Francisco. Sistema de visión por computadora para la detección de objetos esféricos a través de la transformada de Hough. *revista ingeniería UC*, 2008, vol. 15, no 1, p. 77-87.

[15] CANSECO, Mauricio Cirilo Méndez. Diseño de un robot paralelo. *ITECKNE: Innovación e Investigación en Ingeniería*, 2009, vol. 6, no 2, p. 29-35.

[16] Guía técnica de seguridad en robótica. Departamento de hacienda, economía y empleo.

[17] ALBERTO R. KREMER. Seguridad en la celda robotizada. Universidad tecnológica nacional, 2018, p. 1-48

[18] RODRÍGUEZ ESPINOSA, María Jesús, et al. Detección automática de glóbulos rojos mediante la transformada de Hough. 2018.

[19] APOLO, Acosta; QUINALUISA, Caizalitín; ALEJANDRO, Edwin. *Diseño e implementación de una celda de manufactura robotizada, para la réplica de piezas 3d, mediante fresado en el Laboratorio de Robótica Industrial de la ESPE-EL*. 2014. Tesis de Licenciatura. Universidad de las Fuerzas Armadas ESPE Extensión Latacunga. Carrera de Ingeniería Mecatrónica.

[21] VERGARA, Diego Édison Flórez; RIVEROS, Fabián Camilo Castro; ESTEPA, Ricardo Andrés Castillo. planeación y ejecución de trayectorias en un robot Delta. *Scientia et technica*, 2017, vol. 22, no 2, p. 183-192.

- [22] CALVO RUIZ, Rubén, et al. Diseño de célula de fabricación flexible mediante la utilización de herramientas software de digitalización 3D. 2019.
- [23] FERNÁNDEZ YÁNEZ, Lorena Alejandra; SOTOMAYOR REINOSO, Luisa Fernanda. *Análisis cinemático inverso y directo del robot paralelo*. 2016. Tesis de Maestría. Quito, 2016.
- [24] CORONADO, Enrique, et al. Control de un Robot Paralelo Tipo Delta basado en Manipulacion en Espacio de Cámara Lineal.
- [25] GARCÍA, José Javier Doria; CACHAY, Fabián Leonardo Melo; CHAVEZ, John Jairo Chavez. Implementación de proceso de Pick and Place mediante Grúa-Robot aplicado a celda de manufactura flexible. *Scientia et technica*, 2013, vol. 18, no 4, p. 633-639.
- [25] CAMELO, Muñoz, et al. Diseño mecánico de un robot de arquitectura paralela tipo delta de 3 DOF. 2014.
- [26] BELTRÁN NOVA, ALEJANDRO. *Sistema de control de robot delta basado en visión artificial*. 2015. Tesis Doctoral.
- [27] SU, Tingting, et al. Time-optimal trajectory planning for delta robot based on quintic pythagorean-hodograph curves. *IEEE Access*, 2018, vol. 6, p. 28530-28539.

Anexos

Algoritmos

Algoritmos de optimización

Este programa se encargó de encontrar las dimensiones más óptimas de los eslabones del robot.

```
clear, close all, clc
MaxL1 = 23;    %Longitud maxima de articulacion 1
MinL1 = 9;    %Longitud minima de articulacion 1
MaxL2 = 35;   %Longitud maxima de articulacion 2
MinL2 = 5;    %Longitud maxima de articulacion 2
Max_h_bandas = -15; %Distancia maxima desde base del robot a la cita
transportadora
Min_h_bandas = -40*5; %Distancia minima desde base del robot a la
cita transportadora
FitnessFunction = @crd_funcion_obj_optimizacion_delta;
numberOfVariables = 3;
rng default
[x, fval, exitflag, output] = particleswarm(FitnessFunction,...
    numberOfVariables, [MinL1, MinL2, Min_h_bandas],...
    [MaxL1, MaxL2, Max_h_bandas] );
f2=figure(2);, cla;
set(f2, 'color', 'w');
plot(x, fval,'x');
```

Para que el programa de optimización conociera que las longitudes halladas eran correctas, se verificaban en una función llamada verificación1, esta tiene la tarea de probar el espacio de trabajo del robot con las dimensiones halladas. Si eran apropiadas envía un 1 si no un 0.

```
function objetivo=crd_funcion_obj_optimizacion_delta(X)

L1 = X(1);    %Longitud articulacion 1
L2 = X(2);    %Longitud articulacion 2
h_bandas = X(3);    %Distancia desde base del robot a la cinta de la
banda
v = crd_verificacion1(L1,L2, h_bandas); %funcion de verificacion de si
es posible alcanzar los puntos    %con las medidas dadas
if v == 1
    objetivo = L1+L2;
else
    objetivo = 10e7;
end
function v = crd_verificacion1(L1,L2, h_bandas)
DIBUJAR_ROBOT_AL_VERIFICAR = 0;% 0: No dibuja 1: si dibuja
v = 0;
% Parametros del robot delta
delta.f = 20;    % lado del triangulo base del robot
delta.rf = L1;    % longitud primer eslabon
```

```

delta.re = L2;      % longitud segundo eslabon
delta.e = 9;       % lado del triangulo efector movil
delta.eab=2;
aber_rotula=20;   %grado maximo que abre la rotula

% Coordenadas del efector final referidas a la base del robot
vec1=linspace(-2,2,3);           % Creación del vector coordenadas
eje Y
vec2=linspace(-10,10,4);        % Creación del vector coordenadas
eje X
h_bandas=[h_bandas];          %Creacion del Vector coordenadas eje Z
[n f_vec1 ]=size(vec1);        %tamaño de los vectores
[n f_vec2]=size(vec2);
[n f_h_bandas]=size(h_bandas);
M=zeros(f_vec1*f_vec2*f_h_bandas,3); % Matriz de ceros con las
dimensiones dadas
c=0;
% Llenado de la matriz con los puntos de coordenadas
for i=1:f_h_bandas
    for j=1:f_vec2
        for k=1:f_vec1
            c=c+1;
            M(c,:)=[vec2(j) vec1(k) h_bandas(i)];
        end
    end
end
[fil, col]=size(M);
for i = 1:fil
    [vec_theta,alfa]= cin_inv_delta(M(i,1) , M(i,2), M(i,3) , delta)
    for o=1:3
        if (alfa(o)>=aber_rotula)
            warning('No es posible de alcanzar el punto %d con este
robot - rotulas',i);
            alfa;
            return;
        end
    end
    if (sum(abs(imag(vec_theta))>0))>0
        warning('No es posible de alcanzar el punto %d con este
robot',i);
        return;
    else
        if DIBUJAR_ROBOT_AL_VERIFICAR
            f1=figure(1);, clf;, set(f1,'Color', [1,1,1]);
            hold on
            axis([-50 50 -50 40 -70 20]);
            crd_dibujar_bandas_transportadoras3d(h_bandas(1));
            crd_dibujar_objetos(M);
            crd_dibujar_paralelo_delta_L1L2_P(L1, L2, M(i,:));
            %view (-180,11);
            view (175,25);
            pause(0.1);
        end
    end
end
end

v = 1; % el robot logro todos los puntos

```

Algoritmos parámetros iniciales

Ante de iniciar el proceso es necesarios cargar las configuraciones y parámetros de los sistemas. Que se dividen comunicación bluetooth, parámetros robot delta y parámetros cámara.

La comunicación bluetooth consiste en enlazar la matlab con el modlo HC-05 y abrir el comunicación serial.

```
bluetooth=Bluetooth('HC-05',1);
fopen(bluetooth);
pause(1);
disp('Bluetooth conectado');
```

Para un correcto funcionamiento de la cámara es necesario establecer parámetros tales como formato de color, abrir canal de video, intervalo de captura de cuadros

```
imagreset; %reseteo parametros de imagen
video=videoinput('winvideo',1,'YUY2_640x480'); %% esto crea el canal
de video a la camara web
preview(video); %abrimos una ventana de preview para ver a donde
apuntamos con la camara
video.ReturnedColorSpace = 'rgb'; % Formato de color de la entrada de
video
video.FrameGrabInterval = 1;
start(video); %inicializamos el canal de Video
pause(1); %puasa mientras carga configuracion de la camara
disp('inicio camara');
```

En los parámetros del robot se cargan las variables relacionadas con medidas del robot tanto de sus articulaciones como de sus bases, grados de abertura entre otros. Esto con el fin de tener un modelo virtual aproximado del real y poder realizar simulaciones.

```
L1 = 9; % longitud primer eslabon
L2 = 30.54; % longitud segundo eslabon
h_bandas = -40.21; % distancia de las bandas con respecto al robot
% (negativo) dado que el eje z apunta hacia arriba
% Parametros del robot delta
delta.f = 23; % lado del triangulo base del robot
delta.rf = L1; % longitud primer eslabon
delta.re = L2; % longitud segundo eslabon
delta.e =11.7; % lado del triangulo efector movil
delta.eab=2;
aber_rotula=30; %grado maximo que abre la rotula
Vel_bandas=29/27;
disp('parametros del robot delta cargados');
pause(1);
```

Algoritmos de sistema de visión

El sistema de visión artificial consta básicamente de dos funciones que se explican a continuación.

`Crdd_proceso_imagen` se encarga de tomar la imagen capturada y realizar el proceso de encontrar el número de piezas y sus localizaciones. La imagen binarizada, filtrada, erosionada, y por último la localización de círculos. Retornando una matriz con las posiciones en pixeles.

```
function centersBright=crd_proceso_imagen(rgb)
dibujar_centros=0;
rgb=rgb(112:372, :, :);
E=im2bw(rgb,0.5);
E = imfill(E, 'holes');
E = bwareaopen(E,20); %remueve los objetos con menos de 20 pixeles
se = strel('disk',10); % Elemento estructurante
E = imerode(E,se); % Erosiona Imagen
[centersBright,radiiBright] = imfindcircles(E,[24 50], ...
    'ObjectPolarity','bright','Sensitivity',0.90);
if dibujar_centros
    f3=figure(3)
    set(f3, 'color', 'w');
    title(sprintf('Imagen con centro localizados'))
    figure(3)
    imshow(rgb)
    hBright = viscircles(centersBright, radiiBright,'Color','b');
end
```

`crd_posicion_objetos` se encarga de recibir la matriz generada por `crd_proceso_imagen` y convertir las coordenadas de pixeles a espaciales con respecto al robot, también calcula un valor estimado del tiempo en llegar o estar al alcance del robot.

```
function coor_cm=crd_posicion_objetos(coor_pix)
global Vel_bandas;
coor_pix;
[fil col]=size(coor_pix);
coor2_pix=sort(coor_pix,1,'descend'); %ordeno vector auxiliar
coor2_pix=coor2_pix(:,1);
for i=1:fil %%ordeno en forma ascendente dejando de primero los mas
    aux=find(coor_pix(:,1)==coor2_pix(i)); %%cercanos
    coor_poxel(i,:)=coor_pix(aux,:);
    y(i)=(28.5-(24.5*coor_poxel(i,1)/640))*-1;
    x(i)=(9*coor_poxel(i,2)/248)+4.5;%valor de la
    t(i)= abs(y(i)/Vel_bandas); %%velocidad constante de la banda
    coor_cm(i,:)=[x(i),y(i),t(i)];
end
```

Algoritmos control celda

A continuación se anexa el programa que realiza todo el proceso de la celda, este inicia cargando todos los parámetros y moviendo el robot a la posición de inicio, cuando cada parámetro es cargado manda un mensaje de notificando. Una vez cargado que a la espera de recibir la señal de inicio, una vez es presionado inicia el proceso captura imágenes hasta que detecte objetos presentes en la banda, el robot recibe una matriz con las coordenadas para sujetar la primera pieza, se realiza los respectivos cálculos de las trayectoria y realiza el recorrido, al terminar el recorrido de la primera recalcula la posición de las siguiente pieza, cuando al terminar todas las posiciones de la matriz vuelve a capturar la imagen y el proceso se repite de manear cíclica.

```
clear all,clc, close all
%% carga de parametros, variables, librerias
ciclo_0=1;
id=[101, 102, 103];
crd_parametros_camara;
CargarLibreriasDynamixel;
crd_parametros_delta;
crd_parametros_bluetooth;
crd_home;
disp('Presiona reset y luego inicio para empezar');
while da~=2
flushinput(bluetooth);
da= fscanf (bluetooth, '%d');
reset=1;
end
flushinput(bluetooth);

%%
while reset==1
    paro=0;
    inicio=0;
    flushinput(bluetooth);
    da= fscanf (bluetooth, '%d');
    %%
    if(da==3)
        inicio=1;
        %%
        while inicio==1
            aux=1;
            rgb=getsnapshot(video); %% se captura la imagen
            obj=crd_proceso_imagen(rgb); %% identifica los objetos en la
imagen
            n = numel(obj);
            if(n>0) %verifica si hay objetos
                Tt=0;
                pos=crd_posicion_objetos(obj); %%calcula las posiciones de
los objetos
```

```

[fil col]=size(pos);
for i=1:fil
    if i==1
        if(pos(1,3)>1)
            tr=pos(1,3)-1;
            pause(tr-0.8)
        else
            tr=0;
        end
        Tt=crd_trayectoria_pieza(pos(i,1:2),abs(tr-
pos(i,3)),id);
        Tt=Tt+pos(1,3);
    end
    if i>=2

        pos(:,2:3)=crd_actualiza_posi2(Tt,pos(:,2:3),i);
        if (pos(i,3)>1)
            tr=pos(i,3)-1;
            pause(tr-0)
        else
            tr=0;
        end
        Tt=crd_trayectoria_pieza(pos(i,1:2),abs(tr-pos(i,3)),id);
        Tt=Tt+pos(i,3);
    end
end
else
    disp('No hay objetos, pulsar stop para detener banda');
    pause(5);
    flushinput(blueetooth)
    da= fscanf(blueetooth,'%d');
    if da==4
        inicio=0;
        paro=1;
    end
end
end

flushinput(blueetooth)
da= fscanf(blueetooth,'%d');
if da==4
    inicio=0;
    paro=1;
end
end

disp('presiona inicio para comenzar')
flushinput(blueetooth);
da= fscanf(blueetooth,'%d');
if da==2 & paro==1
    reset=0;
end
end
end
%%
end
fclose(blueetooth);
DescargarLibreriasDynamixel;

```

Dentro del programa de control existen funciones de verificación que se encargan de indicar si el robot es capaz de ejecutar la tarea o no. `v=crd_verificar_velocidad_motores` restringe que el robot exceda su velocidad máxima.

```
function v=crd_verificar_velocidad_motores(qp)
    [fil col]=size(qp);
    for k=1:fil
        for i=1:3
            if(qp(k,i)>300)
                v=0;
                warning('No es posible alcanzar la velocidad %d el motor
%d',qp(k,i),i);
                return;
            else
                v=1;
            end
        end
    end
end
```

`crd_verificar_mecanica` esta función restringe al robot de ejecutar movimientos que sean imposibles mecánicamente ya sea por los posiciones articulares, o por el grado de abertura de la rótula este último está limitado

```
function v=crd_verificar_mecanica(alfa,aber_rotula,vec_theta);

for o=1:3
    if (alfa(o)>=aber_rotula)
        warning('No es posible de alcanzar el punto %d con este
robot - rotulas',o)
        alfa;
        v=0
        return;
    end
end

if (sum(abs(imag(vec_theta))>0))>0
    v=0;
    warning('No es posible de alcanzar este punto con este
robot')
    return;

end

v=1;
```

La siguiente función tiene como ejecutar la trayectoria del robot, esta recibe la información de la coordenada de la pieza y el tiempo de ejecución. Al finalizar envía los datos de cuanto tardo en ejecutar las acciones.


```

function Tt=crd_trayectoria_pieza(pos,t,id)
global bluetooth;

    data=crd_mover_robot_p_t([pos(1,1),0,-32],t-0.2,id);
    fprintf(bluetooth,'1');
    data=crd_mover_robot_p_t([pos(1,1),0,-
34],0.2,id);%%sujeta la pieza
    data=crd_mover_robot_p_t([pos(1,1),0,-32],0.2,id);
    Tt=data.Vec_time(end);
    data=crd_mover_robot_p_t([0,0,-30],0.4,id); %%vuelve
al home
    pause(0.007);
    Tt=data.Vec_time(end)+Tt;
    data=crd_mover_robot_p_t([-4,-2,-31.5],0.3,id);%%
suelta la pieza
    pause(0.007);
    fprintf(bluetooth,'0');
    Tt=data.Vec_time(end)+Tt;
    CargarLibreriasDynamixel; %%%
    data=crd_mover_robot_p_t([-1,0,-29],0.3,id);
    pause(0.007);
    Tt=data.Vec_time(end)+Tt;
end

```

crd_actualiza_posi se encarga de actualizar las posiciones de todos los objetos capturados por la cámara, recalcula las nuevas coordenadas con referencia al robot y el tiempo para que puedan ser sujetadas.

```

function pos=crd_actualiza_posi2(Tt,pos,i)
global Vel_bandas;
pos_aux=pos;
y=Vel_bandas*(Tt+1.1);
y2=0;
[fil col]=size(pos);
    for j=i:fil
        pos(j,1)=y+(pos(j,1))+y2;
        if(pos(j,1)<-5)
            pos(j,2)=(abs(pos(j,1))/Vel_bandas)-0.9;
        else
            pos(j,2)=0.9;
        end
    end
end

```

La función crd_mover_robot_pi_pf_cumple el rol de mover el robot en el espacio cartesiano en el tiempo deseado este recibe las posiciones iniciales, finales y el tiempo parámetros.

```

function data=crd_mover_robot_pi_pf_t(pi,pf,t,id)
global delta;

```

```

global aber_rotula;
[vec_theta,alfa] =crd_cin_inv_delta(pi(1), pi(2), pi(3), delta);
%Calculo angulos delta
[vec_theta2,alfa] =crd_cin_inv_delta(pf(1), pf(2), pf(3), delta);
%Calculo angulos delta
q1=vec_theta; q2=vec_theta2;
v=crd_verificar_mecanica(alfa,aber_rotula,vec_theta); %verifico
que se pueda llegar a dicha posición;
qp=abs(q2-q1)/t;
data=0;
% data=crd_mover_robot_q_qp(q,qp,id);
end

```

La función `crd_mover_robot_pi_pf_` cumple el rol de mover el robot en el espacio cartesiano, a diferencia del anterior este no recibe la posición inicial, solo recibe la posición final y el tiempo de ejecución, esto quiere decir que detecta su posición inicial y se traslada a la posición indicada en el tiempo que el usuario o el programa desee.

```

function data=crd_mover_robot_p_t(p,t,id)
global delta;
global aber_rotula;

[vec_theta,alfa] =crd_cin_inv_delta(p(1), p(2), p(3), delta);
%Calculo angulos delta
q=vec_theta;
vm=crd_verificar_mecanica(alfa,aber_rotula,vec_theta);
%verifico que se pueda llegar a dicha posición;
qa=leer_posicion_motores_id(id); %leo posicion actual
qp=abs(q-qa)/t;
vv=crd_verificar_velocidad_motores(qp);
data=crd_mover_robot_q_qp(q,qp,id); % funcion de verificar max
velocida
end

```

Programa tarjeta arduino nano

El código de la tarjeta de control arduino tiene la tarea de recibir las señales de los interruptores y enviarlas a matlab por bluetooth, además de esto poner en marcha las bandas y activar o desactivar el efector final cuando reciba la orden del programa principal.

```

#include <Servo.h>
#include <SoftwareSerial.h>
int TxD;
int RxD;

```

```

int data;
int paro = 0;
int inicio = 0;
int reset = 0;
int paroest=0,inicioest=0,resetest=0;
int vc=0;
SoftwareSerial bluetooth(TxD, RxD);
Servo myservo2 ; // crea el objeto servo
Servo myservo3 ; // crea el objeto servo

void setup() {
  Serial.begin(9600);
  bluetooth.begin(9600);
  myservo2.attach(2); // vincula el servo al pin digital 2
  myservo3.attach(3); // vincula el servo al pin digital 3
  pinMode(A2,INPUT);
  pinMode(A3,INPUT);
  pinMode(A6,INPUT);
  pinMode(11,OUTPUT);
  myservo2.write(90);
  myservo3.write(90);
}

void loop() {
  //servo parado angulo 90º //servo 100% angulo 180º //servo 100% a angulo 0º
  reset = analogRead(A6);
  inicio = analogRead(A2);
  paro = analogRead(A3);
  if (reset>=1010)
  {
    resetest=1;
    Serial.println("2");
    paroest=1;
    inicioest=0;
  };

  if ((inicio>=1010) && (resetest==1))
  {
    inicioest=1;
    paroest=0;
    Serial.println("3");
  };
  if (paro>=1010)
  {
    paroest=1;
    inicioest=0;
    Serial.println("4");
  };
  if( (resetest==1) && (inicioest==1) && (paroest==0) )
  {

```

```

    if(vc==0)
    {
        myservo2.write(110); //banda recolectora fondo
        myservo3.write(95); //banda
        vc=1;
    }
    else if(vc==1)
    {
        myservo2.write(110); //banda recolectora fondo
        myservo3.write(96); //banda
        vc==0;
    }
}
if( (resetest==1) && (inicioest==0) && (paroest==1) )
{
    myservo2.write(90);
    myservo3.write(90);
    //Serial.println(1);
}

if(blueetooth.available() > 0)
{
    data = blueetooth.read();

    if(data == '1')
    {
        digitalWrite(11, HIGH);
    }

    else if(data == '0')
    {
        digitalWrite(11, LOW);
    }
}
Serial.println('9');
delay(1); }

```

Manual de usuario

En este anexo se muestra el uso correcto de la celda robótica.

Configuración inicial

- Lo primero en realizar es conectar la celda a la energía eléctrica.
- Encender interruptor de iluminación de la célula.
- Conectar la cámara a un puerto USB del computador,
- Conectar el robot al computador con el módulo USB2dynamixel por puerto USB.
- Se retira el paro de emergencia ubicado.
- Abrir programa de Matlab.
- Se cargan los parámetros de todo el sistema.

Parámetros robot delta: Estos corresponden a las medidas del robot.

Parámetros cámara: Carga las configuraciones de la cámara y una vez finalizado, inicia la transmisión de video.

LibreríasDynamixel: Cargan las librerías de los actuadores dynamixel con sus respectivos registros.

Parámetros bluetooth: Realiza la conexión por comunicación bluetooth matlab y la tarjeta de control arduino.

- Una vez se cargan los parámetros el robot delta se desplaza a la posición de inicio y manda un mensaje al usuario "Presione inicio para comenzar".

Proceso

- El usuario debe presionar el reset antes del botón inicio, sino este no se pondrá en marcha, ya que el paro de emergencia estuvo activo al principio.
- Presionar inicio pondrá en marcha el proceso, las bandas transportadoras comenzaran a moverse, matlab recibe una señal iniciando todo el proceso de captura de imagen y control del robot.

- Presionar paro detendrá inmediatamente las bandas transportadoras, si el robot se encuentra ejecutando una trayectoria esta la terminara y luego se detendrá.
- Presionar inicio después del paro continuara el proceso con normalidad.
- El botón reset reinicia todos los parámetros a su configuración de inicio.
- El paro de emergencia detiene todo el proceso inmediatamente.

Manejo individual

Usuario puede usar los sistemas de manera individual.

Bandas transportadoras: Estas pueden ponerse en marcha sin que el sistema de visión o del robot estén conectados, presionando los botones de inicio y paro.

Visión artificial: EL sistema de visión puede usar sede manera independiente para identificar los objetos que encuentran en la banda, y sus ubicaciones ya sean con referencia al robot a la cámara.

Robot delta: El sistema del robot al igual que los anteriores mencionados, puede usarse de manera individual para ejecutar trayectorias ingresando el punto hacia donde quiere que se dirija el robot en el espacio cartesiano y que tiempo quiere que lo realice. Con esto se puede apreciar el funcionamiento de la cinemática inversa. Si en dado caso el robot no alcanza el punto dado este mandara un aviso diciendo que no es posible alcanzarlo.

Apagar sistema

A la hora de apagar la celda se deben tener las siguientes consideraciones.

- Detener el proceso presionando el botón de paro, si se encuentra en ejecución la célula.
- Descargar librerías Dynamixel.
- Cerrar puerto de comunicación bluetooth.

- Desconectar los puertos USB de la cámara web y el módulo USB2Dynamixel de la computadora.
- Apagar el interruptor de la iluminación de la estructura.
- Presionar paro de emergencia.
- Desconectar de la eléctrica.

Fotografías

Estructura completa



Brazo robótico



Bandas transportadoras



Interruptores de control



Paro de emergencia

