



R E P Ú B L I C A D E L E C U A D O R

# Escuela Politécnica Nacional

"E S C I E N T I A H O M I N I S S A L U S "

La versión digital de esta tesis está protegida por la Ley de Derechos de Autor del Ecuador.

Los derechos de autor han sido entregados a la “ESCUELA POLITÉCNICA NACIONAL” bajo el libre consentimiento del (los) autor(es).

Al consultar esta tesis deberá acatar con las disposiciones de la Ley y las siguientes condiciones de uso:

- Cualquier uso que haga de estos documentos o imágenes deben ser sólo para efectos de investigación o estudio académico, y usted no puede ponerlos a disposición de otra persona.
- Usted deberá reconocer el derecho del autor a ser identificado y citado como el autor de esta tesis.
- No se podrá obtener ningún beneficio comercial y las obras derivadas tienen que estar bajo los mismos términos de licencia que el trabajo original.

El Libre Acceso a la información, promueve el reconocimiento de la originalidad de las ideas de los demás, respetando las normas de presentación y de citación de autores con el fin de no incurrir en actos ilegítimos de copiar y hacer pasar como propias las creaciones de terceras personas.

*Respeto hacia sí mismo y hacia los demás.*

# **ESCUELA POLITÉCNICA NACIONAL**

## **FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA**

### **DISEÑO Y CONSTRUCCIÓN DE UN PROTOTIPO DE ROBOT DELTA PARA APLICACIONES PICK & PLACE**

### **PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA Y CONTROL**

**WLADIMIR CHRISTIAN PACHACAMA OÑA**  
**(wladimirc123@hotmail.com)**

**JUAN FERNANDO SALAZAR MUZO**  
**(juansa\_m@hotmail.com)**

**DIRECTOR: ING. MARCO ANTONIO HERRERA GARZÓN, MSc.**  
**(marco.herrera@epn.edu.ec)**

**CODIRECTOR: ING. NELSON GONZALO SOTOMAYOR OROZCO, MSc.**  
**(nelson.sotomayor@epn.edu.ec)**

**Quito, Diciembre 2016**

## **DECLARACIÓN**

Nosotros, Wladimir Christian Pachacama Oña y Juan Fernando Salazar Muzo declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

---

Wladimir Christian Pachacama Oña

---

Juan Fernando Salazar Muzo

## **CERTIFICACIÓN**

Certifico que el presente trabajo fue desarrollado por Wladimir Christian Pachacama Oña y Juan Fernando Salazar Muzo, bajo nuestra supervisión.

---

**Ing. Marco Herrera, MSc.  
DIRECTOR DEL PROYECTO**

---

**Ing. Nelson Sotomayor, MSc.  
CO-DIRECTOR DEL PROYECTO**

## **AGRADECIMIENTO**

Agradezco a Dios y a la virgen maría por la salud, por permitirme llegar al final de la meta planteada.

Agradezco a mis padres por darme la vida, en especial a mi madre Fabiola quien me ha enseñado que las adversidades de la vida son oportunidades para saber que se puede ser mejor.

Agradezco a mis familiares y a mi hermana quienes me han alentado para cumplir con este objetivo.

Agradezco a la vida por permitirme conocer a varias personas que se convirtieron en grandes amigos y formaron parte del día a día de mi paso por la vida universitaria.

Agradezco a los profesores de la Escuela Politécnica Nacional en especial al departamento de Automatización y Control Industrial, quienes brindaron sus mayores esfuerzos por enseñarnos sus conocimientos teóricos y sus experiencias.

Agradezco Al Ing. Marco Herrera y al Ing. Nelson Sotomayor, por guiarnos en la última etapa de esta formación académica.

Agradezco a mi amigo Juan, por formar parte de la culminación de esta etapa.

## **DEDICATORIA.**

Este trabajo lo dedico a mis padres quienes me han acompañado y guiado en la vida.

Wladimir Pachacama

## **AGRADECIMIENTO**

Agradezco a Dios por la oportunidad que me brinda cada día de poder abrir los ojos y permitirme luchar por los objetivos propuestos, aunque el tiempo no fue el planeado.

A mi familia que nunca ha dejado de creer en mí, que me ha brindado su apoyo y comprensión, que con sus ejemplos me han demostrado que rendirse no es una opción y que de los problemas más grandes se puede salir cuando se pone dedicación.

A la Escuela Politécnica Nacional por los valiosos conocimientos adquiridos a lo largo de la carrera, también por esos momentos de alegría indescriptible que significaba superar una materia difícil en la primera matrícula y por los momentos de tensión únicos que significaban los supletorios que ayudaron a que se formen los grupos de amigos y se solidifique una amistad que perdurara para siempre. Es en la universidad en donde aprendí que existen personas con capacidades increíbles, que cada quien es bueno para una cosa y que en conjunto se pueden generar grandes resultados.

A todas las personas que he conocido durante la carrera, porque gracias a la diversidad que se tiene he logrado aprender cosas muy interesantes que de ahora en adelante forjaran mi camino.

A los Ing. Marco Herrera e Ing. Nelson Sotomayor por el tiempo que le prestaron a este trabajo, por los consejos y por la ayuda que nos brindaron al terminar este proyecto.

A Wladimir(parch) mi compañero y amigo de tesis, por la buenas y malas decisiones que hemos tomado a lo largo de este proyecto pero que a la final logramos concluirlo. Lorena gracias por creer en mí y por ese apoyo incondicional que me has brindado.

Juan Salazar

## **DEDICATORIA**

Quiero dedicar este trabajo al esfuerzo de mis padres, que nunca han dejado de creer en mí, que a pesar de los momentos incomodos vividos el bienestar de sus hijos siempre ha sido lo más importante, por esas enseñanzas de sacrificio para poder obtener algo.

A mis hermanos y familiares porque a pesar de no comprender el fondo de este trabajo siempre preguntaban ¿Cómo estás? , haciéndome saber la importancia de mí ser en sus vidas.

A mis familiares en el exterior por el apoyo ¡MUCHAS GRACIAS!

Juan Salazar

## CONTENIDO

<b>RESUMEN .....</b>	xi
<b>PRESENTACIÓN.....</b>	xii
<b>CAPÍTULO 1 .....</b>	1
<b>MARCO TEÓRICO.....</b>	1
1.1 ROBÓTICA.....	1
1.1.1 INTRODUCCIÓN[1].....	1
1.1.2 COMPONENTES DE UN ROBOT[2] .....	2
1.2 ROBOTS MANIPULADORES[2].....	3
1.2.1 CLASIFICACIÓN DE LOS ROBOTS MANIPULADORES .....	3
1.2.1.1 Cadena cinemática abierta .....	4
1.2.1.1 Cadena cinemática cerrada .....	4
1.3 ROBOTS PARALELOS .....	5
1.3.1 INTRODUCCIÓN[5][4] .....	5
1.3.2 DEFINICIÓN DE ROBOT PARALELO .....	6
1.3.2.1 Características de los robots paralelos[6] .....	6
1.3.2.1.1 <i>Ventajas</i> .....	6
1.3.2.1.2 <i>Desventajas</i> .....	7
1.3.3 CLASIFICACIÓN DE ROBOTS PARALELOS[4][6][7] .....	7
1.3.3.1 Robots paralelos planares .....	7
1.3.3.1.1 <i>Dos GDL</i> .....	7
1.3.3.1.2 <i>Tres GDL</i> .....	8
1.3.3.2 Robots paralelos de movimientos espaciales .....	9
1.3.3.2.1 <i>Tres GDL</i> .....	9
1.3.3.2.2 <i>Cuatro GDL</i> .....	9
1.3.3.2.3 <i>Cinco GDL</i> .....	10
1.3.3.2.4 <i>Seis GDL</i> .....	10
1.4 ROBOT TIPO DELTA .....	10
1.4.1 CINEMÁTICA DEL ROBOT DELTA .....	11

1.4.1.1 Cinemática directa .....	12
1.4.1.2 Cinemática inversa .....	12
1.4.2 ESPACIO DE TRABAJO .....	13
1.4.2.1 Método geométrico[16] .....	13
1.4.2.2 Método de discretización[16] .....	13
1.5 HERRAMIENTAS MATEMÁTICAS .....	13
1.5.1 REPRESENTACIÓN DE LA POSICIÓN .....	14
1.5.2 REPRESENTACIÓN DE LA ORIENTACIÓN[2] .....	14
1.5.2.1 Matrices de rotación.....	14
1.5.2.2 Ángulos de euler.....	16
1.5.2.3 Par de rotación.....	17
1.5.2.4 Cuaterniones .....	17
1.5.3 MATRICES DE TRANSFORMACIÓN HOMOGENEA .....	18
1.6 VISIÓN ARTIFICIAL[10][11][12].....	20
1.6.1 PROCESO DE LA VISIÓN ARTIFICIAL .....	20
1.6.1.1 Adquisición de imágenes .....	21
1.6.1.2 Digitalización de las imágenes .....	21
1.6.1.3 Procesado .....	21
1.6.1.4 Interpretación de los resultados .....	22
1.6 APLICACIONES DEL ROBOT DELTA .....	22
 <b>CAPÍTULO 2 .....</b>	 24
<b>DISEÑO E IMPLEMENTACIÓN DEL PROTOTIPO .....</b>	24
2.1 ANÁLISIS CINEMÁTICO .....	24
2.1.1 GEOMETRÍA DEL ROBOT DELTA[15][16].....	25
2.1.2 DESARROLLO DE LA CINEMÁTICA INVERSA[16].....	28
2.1.3 DESARROLLO DE LA CINEMÁTICA DIRECTA[16].....	32
2.1.4 ANÁLISIS DE SINGULARIDADES[2][4][17] .....	35
2.1.4.1 Matriz Jacobiana .....	36
2.1.4.2 Clasificación de singularidades[17][18].....	40
2.1.4.2.1 <i>Singularidades de la cinemática inversa</i> .....	41

2.1.4.2.2 <i>Singularidades de la cinemática directa</i> .....	42
2.1.4.3 Destreza del robot delta[4][19] .....	42
2.1.5 ANÁLISIS DEL ESPACIO DE TRABAJO[16] .....	43
2.2 DISEÑO MECÁNICO .....	47
2.2.1 PARÁMETROS DE DISEÑO .....	47
2.2.1.1 Dimensiones del prototipo .....	47
2.2.1.1.1 <i>Algoritmo para obtener las dimensiones</i> .....	47
2.2.1.1.2 <i>Aplicación del algoritmo</i> .....	49
2.2.1.1.3 <i>Verificación del espacio de trabajo</i> .....	52
2.2.1.2 Tipo de material .....	56
2.2.2 DISEÑO DE LAS PARTES DEL PROTOTIPO.....	58
2.2.2.1 Descripción de los elementos diseñados .....	60
2.2.2.1.1 <i>Base fija</i> .....	60
2.2.2.1.2 <i>Eslabón L<sub>1</sub></i> .....	61
2.2.2.1.3 <i>Eslabón L<sub>2</sub></i> .....	62
2.2.2.1.4 <i>Junturas esféricas</i> .....	62
2.2.2.1.5 <i>Base móvil</i> .....	63
2.2.2.1.6 <i>Estructura de soporte</i> .....	64
2.2.2.2 Análisis estructural.....	64
2.3 COMPONENTES ELECTRÓNICOS.....	71
2.3.1 ACTUADORES .....	71
2.3.1.1 Dynamixel AX-12A [24][25].....	72
2.3.1.1.1 <i>Conexión de los Dynamixel</i> .....	73
2.3.1.1.2 <i>Protocolo de comunicación</i> .....	74
2.3.2 ARDUINO .....	75
2.3.3 CONVERTIDOR DE SEÑAL TTL A USB .....	77
2.3.4 FUENTE DE ALIMENTACIÓN.....	77
2.4 MONTAJE DEL PROTOTIPO .....	78
2.4 COSTO DEL PROTOTIPO .....	82

<b>CAPÍTULO 3 .....</b>	84
<b>DESARROLLO DEL PROGRAMA DE CONTROL .....</b>	85
3.1 PROGRAMA DE CONTROL PARA EL PROTOTIPO .....	85
3.1.1 PROGRAMA DE ENLACE PC-ROBOT, ROBOT-PC .....	85
3.1.2 PROGRAMA DE CONTROL PRINCIPAL .....	89
3.1.2.1 Comunicación MATLAB-Arduino .....	89
3.1.2.2 Acondicionamiento de datos para los actuadores .....	90
3.1.2.3 Ejecución de trayectorias continuas .....	92
3.1.2.4 Ejecución de trayectorias punto a punto[1].....	94
3.1.2.5 Adquisición de puntos mediante visión artificial.....	96
3.1.2.6 Controlador proporcional, integral, derivativo (PID)[26].....	100
3.2 DISEÑO DE LA INTERFAZ.....	105
3.2.1 VENTANA DEL ESPACIO DE TRABAJO .....	106
3.2.2 VENTANA DE LA SECCIÓN DEMOSTRATIVA .....	107
3.2.3 VENTANA DE TAREAS .....	109
<b>CAPÍTULO 4 .....</b>	112
<b>PRUEBAS Y RESULTADOS .....</b>	112
4.1 PRUEBAS ELECTRÓNICAS .....	112
4.2 FUNCIONAMIENTO DE LA CINEMÁTICA DEL ROBOT .....	112
4.2.1 LIMITES DE MOVIMIENTO DE LA ESTRUCTURA.....	112
4.2.1.1 Posición de los actuadores y alcance en los ejes coordinados .....	113
4.2.1.2 Verificación del espacio de trabajo .....	116
4.2.2 EJECUCIÓN DE TRAYECTORIAS .....	117
4.3 PRUEBA DE TAREAS PICK & PLACE .....	118
4.3.1 MODO AUTOMÁTICO .....	119
4.3.1.1 Prueba de detección de objetos .....	119
4.3.1.2 Ejecución de tareas .....	120
4.3.1.2.1 <i>Apilamiento</i> .....	120
4.3.1.2.2 <i>Ordenamiento</i> .....	121
4.3.2 MODO MANUAL .....	124
4.4 CARACTERÍSTICAS TÉCNICAS DEL PROTOTIPO .....	125

<b>CAPÍTULO 5 .....</b>	127
<b>CONCLUSIONES Y RECOMENDACIONES .....</b>	127
5.1 CONCLUSIONES .....	127
5.2 RECOMENDACIONES .....	129
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	131

## **ANEXOS**

- ANEXO A-PLANOS MECÁNICOS
- ANEXO B-PLANOS ELÉCTRICOS
- ANEXO C-MANUAL DE USUARIO

## ÍNDICE DE FIGURAS

<b>Figura 1.1.</b> Estructura de un robot [3] .....	2
<b>Figura 1.2.</b> Distintos tipos de articulaciones [2].....	2
<b>Figura 1.3.</b> Robots de cadenas cinemáticas abiertas [2].....	4
<b>Figura 1.4.</b> Robot paralelo Plataforma Gough [4] .....	5
<b>Figura 1.5.</b> a).Robot paralelo 6 GDL b). Robot paralelo delta [4] .....	6
<b>Figura 1.6.</b> Robot paralelo planar de 5 barras con su espacio de trabajo [12] .....	8
<b>Figura 1.7.</b> Tipos de cadenas cinemáticas en un robot paralelo planar de 3 GDL [4] .....	8
<b>Figura 1.8.</b> (a)Robot tipo delta, (b) Manipulador de rotación [4].....	9
<b>Figura 1.9.</b> Partes de un robot paralelo tipo delta.....	11
<b>Figura 1.10.</b> Diagrama de relación de la cinemática directa e inversa [2] .....	12
<b>Figura 1.11.</b> Orientación sistema OUVW, con el eje U coincidente con el eje X [2] .....	15
<b>Figura 1.12.</b> Ángulos de Euler secuencia ZXZ y sistema final OU”V”W” [2] .....	16
<b>Figura 1.13.</b> Orientación par de rotación $Rot(k, \theta)$ . [2] .....	17
<b>Figura 1.14.</b> Sistema móvil y sistema fijo asociado a un punto en común .....	18
<b>Figura 1.15.</b> Robot delta utilizado en el proceso de empaquetamiento de galletas [8] .....	22
<b>Figura 1.16.</b> Máquina de medición HCCM desarrollado por la universidad de Florida y Perry automation. [13] .....	23
<b>Figura 1.17.</b> Robot delta para cirugías. [8] .....	23
<b>Figura 2.1.</b> Convención de parámetros para la representación D-H. [2] .....	24
<b>Figura 2.2.</b> Modelo geométrico del robot delta .....	26
<b>Figura 2.3.</b> Descripción de los ángulos $\theta_{1i}, \theta_{2i}, \beta_i$ y $\emptyset_i$ .....	26
<b>Figura 2.4.</b> Representación vectorial de la primera cadena cinemática .....	29
<b>Figura 2.5.</b> (a) Proyección del robot delta en el plano $x_i z_i$ , (b) Proyección de $L_2$ en el plano $x_i z_i$ .....	37
<b>Figura 2.6.</b> a) Mínimo valor del ángulo $\theta_{2i}$ , b) Máximo valor del ángulo $\theta_{2i}$ .....	44
<b>Figura 2.7.</b> Diagrama de flujo cálculo del espacio de trabajo y destreza global.....	46

<b>Figura 2.8.</b> Espacio de trabajo deseado.....	49
<b>Figura 2.9.</b> Diagrama de flujo de la optimización combinatoria bajo restricciones ....	51
<b>Figura 2.10.</b> (a)Espacio de trabajo robot delta, (b) Vista Superior, (c) Vista Plano YZ, (d) Vista Plano XZ .....	52
<b>Figura 2.11.</b> (a)Plano XY espacio deseado (verde), espacio de trabajo robot delta (azul), (b) Vista Plano XZ, (c) Vista Plano YZ .....	53
<b>Figura 2.12.</b> Espacio de trabajo recalculado. (a)Plano XY espacio deseado (verde), espacio de trabajo robot delta (azul), (b) Vista Plano XZ, (c) Vista Plano YZ .....	55
<b>Figura 2.13.</b> Destreza local plano z=-250[mm] .....	55
<b>Figura 2.14.</b> Destreza local plano z=-400[mm] .....	56
<b>Figura 2.15.</b> Prototipo diseñado y partes principales .....	59
<b>Figura 2.16.</b> Prototipo robot delta diseño final.....	59
<b>Figura 2.17.</b> Diseño CAD de la base fija .....	62
<b>Figura 2.18.</b> Diseño CAD del soporte del Dynamixel AX-12A.....	60
<b>Figura 2.19.</b> Diseño CAD del eslabón $L_1$ .....	61
<b>Figura 2.20.</b> Acople eje del servomotor – eslabón $L_1$ .....	61
<b>Figura 2.21.</b> Rodamiento y chumacera .....	62
<b>Figura 2.22.</b> Diseño CAD del eslabón $L_2$ .....	62
<b>Figura 2.23.</b> Diseño CAD de la juntura esférica.....	63
<b>Figura 2.24.</b> Diseño CAD del efecto final.....	63
<b>Figura 2.25.</b> (a) Brazo de sujeción, (b) plancha inferior.....	64
<b>Figura 2.26.</b> Análisis de fuerzas sobre el eje de acople .....	66
<b>Figura 2.27.</b> Desplazamientos en el eje de acople .....	67
<b>Figura 2.28.</b> Análisis de fuerzas sobre el eslabón $L_1$ .....	67
<b>Figura 2.29.</b> Desplazamientos del eslabón $L_1$ .....	68
<b>Figura 2.30.</b> Análisis de fuerzas del brazo de sujeción estática.....	68
<b>Figura 2.31.</b> Desplazamientos del brazo de sujeción estática .....	69
<b>Figura 2.32.</b> Tensión de Von-Mises en el eje de acople. .....	69
<b>Figura 2.33.</b> Tensión de Von-Mises del eslabón $L_1$ .....	70
<b>Figura 2.34.</b> Tensión de Von-Mises del brazo de sujeción estática. .....	71

<b>Figura 2.35.</b> Servomotor Dynamixel AX-12A [25] .....	72
<b>Figura 2.36.</b> Conexión en cadena margarita de los Dynamixel AX-12A [25] .....	74
<b>Figura 2.37.</b> Conexión UART-Dynamixel [25] .....	74
<b>Figura 2.38.</b> Estructura de los paquetes utilizados por el actuador Dynamixel .....	75
<b>Figura 2.39.</b> Distribución de pines de la placa Arduino .....	76
<b>Figura 2.40.</b> Conversor de TTL a USB .....	77
<b>Figura 2.41.</b> Soporte del prototipo con cables de funcionamiento .....	79
<b>Figura 2.42.</b> Plataforma fija acoplada a la base .....	79
<b>Figura 2.43.</b> (a) Acople eje-actuador, (b) Acople actuador-plataforma fija .....	80
<b>Figura 2.44.</b> Acople eslabón $L_1$ al eje del actuador .....	80
<b>Figura 2.45.</b> (a) Acople eslabón $L_1$ -eslabón $L_2$ , (b) Acople eslabón $L_2$ -plataforma móvil .....	81
<b>Figura 2.46.</b> (a) Prototipo terminado, (b) Gabinete de placas electrónicas .....	81
<b>Figura 3.1.</b> Secuencia programa de control del prototipo .....	85
<b>Figura 3.2.</b> Diagrama de flujo del programa de enlace .....	86
<b>Figura 3.3.</b> Diagrama de flujo para establecer la conexión PC-Arduino .....	90
<b>Figura 3.4.</b> Interpretación de los ángulos calculados en la cinemática inversa .....	91
<b>Figura 3.5.</b> Ángulos de la cinemática inversa y ángulos de los servomotores Dynamixel.....	91
<b>Figura 3.6.</b> Curva de la resolución del servomotor Dynamixel .....	92
<b>Figura 3.7.</b> Diagrama de flujo para ejecución de trayectorias continuas .....	93
<b>Figura 3.8.</b> Diagrama de flujo para ejecución de trayectorias punto a punto .....	96
<b>Figura 3.9.</b> Cámara VGA de una laptop HP-G62.....	97
<b>Figura 3.10.</b> Enfoque de la cámara (a) Efector final sin posicionarse, (b) Efector final posicionado.....	97
<b>Figura 3.11.</b> Imagen previa a la detección de círculos .....	98
<b>Figura 3.12.</b> Imagen que muestra figuras circulares encontradas.....	99
<b>Figura 3.13.</b> Representación de un pixel.....	99
<b>Figura 3.14.</b> Representación de los puntos localizados en el plano cartesiano .....	100
<b>Figura 3.15.</b> Diagrama de bloques controlador PID .....	100
<b>Figura 3.16.</b> Diagrama de bloques del sistema de control para los actuadores .....	102

<b>Figura 3.17.</b> Respuesta actuador 1 sin acción de control .....	103
<b>Figura 3.18.</b> Respuesta actuador 2 sin acción de control .....	103
<b>Figura 3.19.</b> Respuesta actuador 3 sin acción de control .....	103
<b>Figura 3.20.</b> Respuesta del controlador en el actuador 1 .....	104
<b>Figura 3.21.</b> Respuesta del controlador en el actuador 2 .....	104
<b>Figura 3.22.</b> Respuesta del controlador en el actuador 3. ....	105
<b>Figura 3.23.</b> Ventana del espacio de trabajo del robot delta .....	106
<b>Figura 3.24.</b> Ventana de la sección demostrativa.....	107
<b>Figura 3.25.</b> Ventana de tareas .....	110
<b>Figura 4.1.</b> Medida del actuador1 y altura del efecto final .....	115
<b>Figura 4.2.</b> Medida del actuador1 y altura del efecto final corregido por software .	115
<b>Figura 4.3.</b> Prototipo del robot delta recorriendo el espacio de trabajo .....	116
<b>Figura 4.4.</b> Ejecución de trayectorias en el plano xy.....	117
<b>Figura 4.5.</b> Enfoque del espacio de trabajo .....	119
<b>Figura 4.6.</b> Detección de círculos con un diámetro determinado .....	119
<b>Figura 4.7.</b> Ubicación de los centros en el espacio de trabajo .....	120
<b>Figura 4.8.</b> (a)Objetos detectados, (b) Ubicación de los objetos en el plano cartesiano, (c) Objetos apilados por el prototipo en el punto (0; 0) mm, visto desde la cámara .	121
<b>Figura 4.9.</b> Ordenamiento formando un cuadrado desde el punto (0; 100) mm ....	122
<b>Figura 4.10.</b> Ordenamiento horizontal desde el punto (-50,100) mm.....	123
<b>Figura 4.11.</b> Ordenamiento vertical desde el punto (100,100) mm .....	123
<b>Figura 4.12.</b> Imagen con centros adaptados y con la posición inicial del mouse ....	124
<b>Figura 4.13.</b> Vista superior del área de trabajo radio=200 mm .....	125
<b>Figura4.14.</b> Vista frontal del área de trabajo cilindro de radio=200 mm y altura=158 mm .....	125

## ÍNDICE DE TABLAS

<b>Tabla 2.1.</b> Parámetros robot IRB-360-ABB .....	47
<b>Tabla 2.2.</b> Coordenadas del espacio de trabajo deseado .....	50
<b>Tabla 2.3.</b> Destreza global para diferentes valores de los eslabones $L_1$ y $L_2$ .....	54
<b>Tabla 2.4.</b> Propiedades de los materiales del prototipo .....	58
<b>Tabla 2.5.</b> Características técnicas del servo Dynamixel AX-12A .....	73
<b>Tabla 2.6.</b> Especificaciones técnicas de Arduino Mega .....	76
<b>Tabla 2.7.</b> Características de la Fuente de Alimentación .....	78
<b>Tabla 2.8.</b> Detalles del costo del proyecto.....	82
<b>Tabla 3.1.</b> Constantes de los controladores PID del sistema .....	104
<b>Tabla 4.1.</b> Comparación de los valores teóricos y reales de los actuadores .....	113
<b>Tabla 4.2.</b> Errores en los actuadores .....	114
<b>Tabla 4.3.</b> Representación de los puntos extremos del prototipo .....	115
<b>Tabla 4.4.</b> Especificaciones técnicas .....	126

## RESUMEN

En el presente proyecto se muestra el diseño y construcción de un prototipo de robot delta de tres grados de libertad, que es capaz de ejecutar diferentes tipos de acciones como son el seguimiento de trayectorias, tareas de apilamiento y ordenamiento de objetos, todo esto sobre la base del estudio de la cinemática directa e inversa determinado a través de la geometría del robot.

Para determinar las dimensiones del prototipo se realiza un análisis de la destreza del robot en un espacio de trabajo definido, realizando un estudio de las singularidades.

El programa de control esta embebido en un computador, que mediante una tarjeta de datos permitirá manipular el prototipo de robot delta a través de una comunicación serial. Con el objetivo de darle mayor autonomía al prototipo se le añadió una cámara, que permite visualizar los objetos a ser desplazados dentro del espacio de trabajo.

La interfaz se desarrolló empleando el software de MATLAB, el que permite al usuario interactuar fácilmente para controlar los movimientos del sistema robótico de forma manual y automática.

## PRESENTACIÓN

Este trabajo tiene como propósito el diseño y la construcción de un prototipo de robot paralelo tipo delta poco conocido en nuestro país pero con muchas aplicaciones a nivel mundial en especial para aplicaciones pick and place. Para el cumplimiento del objetivo el trabajo se desarrolla de la siguiente manera:

En el capítulo 1, se desarrolla los conocimientos teóricos de la robótica, los manipuladores, la configuración mecánica serial y paralela, la configuración tipo delta, las herramientas matemáticas para el análisis de la cinemática, la visión artificial y las aplicaciones del robot delta.

En el capítulo 2, se realiza el análisis geométrico del robot delta para la determinación de la cinemática directa e inversa se realiza el análisis de singularidades, el estudio del espacio de trabajo y la destreza los que permiten determinar las dimensiones del robot que se diseñaran en un software CAD, donde se analiza los esfuerzos de las piezas principales del mecanismo. Además se hace una breve revisión de elementos electrónicos como la tarjeta de adquisición de datos Arduino, los actuadores Dynamixel y la fuente switching de alimentación. Finalmente se detalla los pasos para ensamblar el prototipo con un desglosé de todos los componentes y sus respectivos costos que permiten estimar el valor total invertido para el desarrollo del proyecto.

En el capítulo 3, se detalla el programa de control del sistema donde se la ha dividido en tres secciones, primero se detallan los diagramas de flujo del programa que va en la tarjeta Arduino, que permite la relación Pc-Arduino-Dynamixel. Luego se describe el algoritmo del programa principal realizado en MATLAB, para la ejecución de las trayectorias continuas, las trayectorias punto a punto y la adquisición de puntos mediante la visión artificial. Finalmente se describe la interfaz realizado en el GUIDE de MATLAB, que permite operar fácilmente el robot delta.

En el capítulo 4, se presentan los resultados de todas las pruebas realizadas con el robot en el espacio de trabajo definido, tanto para el seguimiento de trayectorias como para la ejecución de tareas de apilamiento y ordenamiento de forma manual y automática. También se detalla las características técnicas obtenidas del prototipo robot delta.

Finalmente en el capítulo 5, se presentan las conclusiones y recomendaciones que se pudieron obtener luego de realizar este trabajo, también se plantea las posibles mejoras para sacar el mejor provecho del mecanismo diseñado.

# CAPÍTULO 1

## MARCO TEORICO

### 1.1. ROBÓTICA

#### 1.1.1. INTRODUCCIÓN [1]

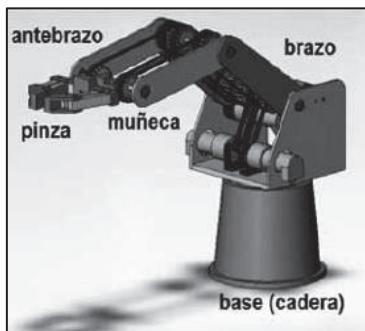
La palabra robot aparece en una obra teatral en 1921 R.U.R (Rossum's Universal Robots) del novelista checo Karel Capek, en donde la palabra robota hacía referencia a la fuerza de trabajo o servidumbre, desde entonces a cualquier mecanismo autónomo programable se le llama robot. La robótica es una disciplina que se encarga del estudio y diseño de robots, para ser incursionados en procesos industriales peligrosos que tengan ambientes con altas temperaturas o ambientes contaminados y para procesos repetitivos que necesiten precisión.

Los robots a nivel industrial se han convertido en un elemento clave en el proceso de automatización debido a la reducción de costos, incremento de la productividad, mejoramiento de la calidad, y reducción de problemas al trabajar en lugares que impliquen peligro para el ser humano.

El funcionamiento de un robot nace de la convergencia de tecnologías del control automático y de la aplicación de sistemas computacionales en tiempo real, el control automático permite gobernar un proceso sin la intervención de agentes externos como la mano del hombre, el problema que se presenta aquí es el seguimiento de una señal de referencia que se soluciona mediante servosistemas, estos generan señales de control que tratan de anular la diferencia entre la señal de referencia y la señal del proceso mediante realimentación. Otro problema es la regulación automática que pretende mantener la señal de referencia aunque existan perturbaciones que traten de sacar al sistema de las condiciones deseadas.

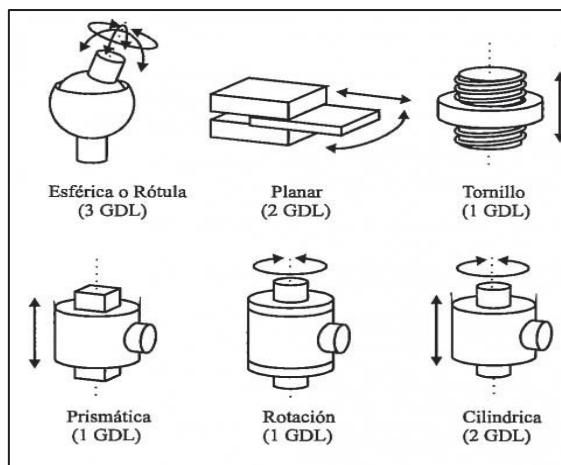
### 1.1.2. COMPONENTES DE UN ROBOT [2]

Un robot está formado por una estructura mecánica que posee una serie de eslabones que se unen mediante articulaciones que permiten un movimiento relativo cada dos eslabones consecutivos, en su mayoría los robots industriales se asemejan al brazo humano y en ocasiones para hacer referencia a la estructura se utilizan términos como: cuerpo, brazo, codo y muñeca. (Figura 1.1.)



**Figura1.1.** Estructura de un robot [3]

Existen diferentes tipos de articulaciones (Figura1.2) que pueden generar movimientos de giro, de desplazamiento o una combinación de ambas. Cada uno de los movimientos independientes que puede realizar una articulación con respecto a la anterior se denomina grado de libertad.



**Figura 1.2.** Distintos tipos de articulaciones [2]

Para generar movimiento en las articulaciones se necesitan de actuadores que se accionan de acuerdo a las órdenes de la unidad de control, los actuadores pueden emplear energía eléctrica, neumática e hidráulica. Las órdenes de la unidad de control son creadas de acuerdo a la tarea seleccionada, al estado de su entorno y a su propio estado, para ello se utilizan sensores que pueden ser de presencia, posición y velocidad.

Por ultimo para realizar la tarea asignada el robot posee un efecto final que es el encargado de interactuar con el entorno, a través de un elemento de agarre o sujeción, también puede realizar tareas con herramientas como fresadoras, soldadoras, etc.

## **1.2. ROBOTS MANIPULADORES [2]**

Los robots manipuladores son los que dominan la industria moderna, su definición es muy difícil de establecer debido a la diferencia conceptual del mercado japonés y el euro-americano.

La definición más aceptada es la establecida por la R.I.A. (Robot Institute of America), según la cual "un robot industrial es un manipulador multifuncional reprogramable, capaz de mover materias, piezas, herramientas, o dispositivos especiales, según trayectorias variables, programadas para realizar tareas diversas".

Mientras que para los japoneses un robot industrial es cualquier dispositivo mecánico dotado de articulaciones móviles destinado a la manipulación.

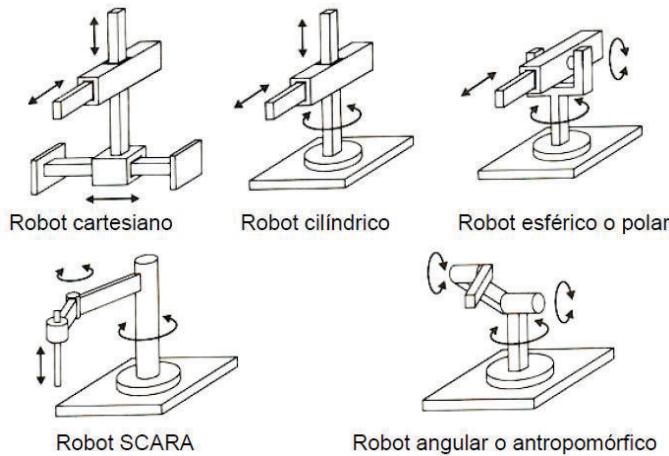
### **1.2.1. CLASIFICACIÓN DE LOS ROBOTS MANIPULADORES**

A los manipuladores se los puede clasificar según la cadena cinemática que poseen, siendo una cadena cinemática la unión de eslabones conectados por articulaciones.

### 1.2.1.1. Cadena cinemática abierta

Se dice que la cadena cinemática es abierta cuando un eslabón se encuentra conectado con cualquier otro de una sola forma, estos pueden ser:

- Cartesiano, cuyo posicionamiento en el espacio se lleva a cabo mediante articulaciones lineales.
- Cilíndrico, posee una articulación rotacional sobre una base y articulaciones lineales para el movimiento en altura y en radio.
- Polar, cuenta con dos articulaciones rotacionales y una lineal.
- Esférico, tres articulaciones rotacionales.
- Mixto, que posee varios tipos de articulaciones, combinaciones de las anteriores. Es destacable la configuración SCARA (Selective Compliance Assembly Robot Arm).



**Figura 1.3.** Robots de cadenas cinemáticas abiertas [2]

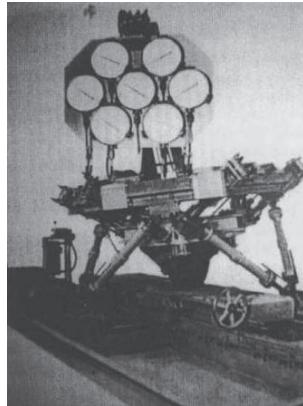
### 1.2.1.2 Cadena cinemática cerrada

Una cadena cinemática cerrada es aquella en la que un eslabón se encuentra conectado con otro de dos maneras diferentes, por lo menos. Estos manipuladores son conocidos como robots paralelos.

## 1.3. ROBOTS PARALELOS

### 1.3.1. INTRODUCCION [5] [4]

Los robots paralelos se concibieron hace más de 50 años. La primera plataforma industrial fue creada en 1947 y se la utilizó para comprobación de neumáticos por la empresa Dunlop Rubber Co., Inglaterra. Se trataba de un robot paralelo hexápodo con longitud de brazos variables el cual simulaba el proceso de aterrizaje de un avión.



**Figura 1.4.** Robot Paralelo Plataforma Gough [4]

Posteriormente en 1965 Stewart presentó una estructura paralela utilizada como simulador de vuelo en donde la conjunción de las diversas cadenas cinemáticas del mecanismo proveía los complejos movimientos a la cabina del piloto.

McCallion y Pham en 1979 fueron los primeros que propusieron usar una plataforma Stewart como un manipulador paralelo en un sistema de ensamble, basados en que la posición del efecto final es menos sensible a las producidas por las articulaciones de los robots seriales. Con el pasar del tiempo los robots paralelos han hecho su incursión en variadas aplicaciones tecnológicas como: posicionamiento de antenas parabólicas, robots trepadores utilizados en tareas de mantenimiento, robots paralelos submarinos y en el campo industrial. En este último sobresalen dos estructuras paralelas, los que

tienen 6 grados de libertad (Figura 1.5a) y los robot delta (Figura 1.5b) que poseen 3; propuesto por Clavel en 1989, el cual tiene una difusión industrial importante.



(a)



(b)

**Figura 1.5.** a).Robot paralelo 6 GDL b). Robot paralelo delta [4]

### 1.3.2. DEFINICION DE ROBOT PARALELO

“Un robot paralelo se compone de un efecto final con n grados de libertad, y de una base fija unidos entre sí por al menos dos cadenas cinemáticas independientes. La actuación se lleva a cabo a través de n actuadores simples.” [4]

#### 1.3.2.1. Características de los robots paralelos [6]

El campo de estudio sobre manipuladores siempre ha estado en continua expansión teniendo como primicia la estructura serie, sin embargo los robots paralelos en constante crecimiento en la actualidad, son utilizados en determinadas tareas donde sus características presentan ventajas frente a los robots seriales.

##### 1.3.2.1.1 Ventajas:

- La relación carga potencia es alta ya que los accionamientos de potencia van directamente conectados desde la base del robot hasta el efecto final.

- Alta rigidez y bajo peso, lo cual se traduce en mayor precisión respecto a los robots serie.
- Elevadas velocidades de operación.

#### *1.3.2.1.2 Desventajas:*

- La cinemática de los mecanismos paralelos es más complicada.
- El espacio de trabajo suele ser pequeño en comparación al de un robot serie.
- Presenta problemas de singularidad y debe resolverse específicamente para cada topología.
- El modelo dinámico no puede ser generalizado, lo que dificulta el desarrollo de algoritmos de control de carácter general.

### **1.3.3. CLASIFICACION DE ROBOTS PARALELOS [4] [6] [7]**

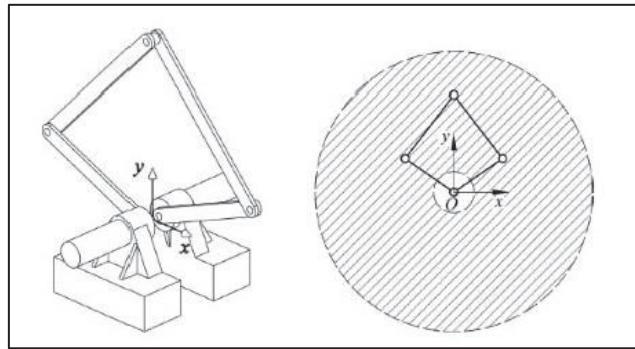
La combinación del número y tipos de cadenas cinemáticas, la restricción en el movimiento de las articulaciones, etc., hacen difícil una clasificación sistemática de las estructuras posibles en robótica paralela. Sin embargo una forma de clasificar a este tipo de robots es de acuerdo al tipo de movimiento que pueden realizar y estos pueden ser robots paralelos planares y robots paralelos de movimientos espaciales.

#### **1.3.3.1. Robots paralelos planares.**

Los robots paralelos planares son aquellos en los que su movimiento de traslación se reduce a un solo plano, por lo tanto puede tener 2 o 3 grados de libertad (GDL) y un movimiento de rotación sobre el eje perpendicular.

##### *1.3.3.1.1. Dos GDL*

Está formado por 4 eslabones y una base unidos por 5 articulaciones rotacionales que solo tiene movimiento en el plano x e y (Figura 1.6).

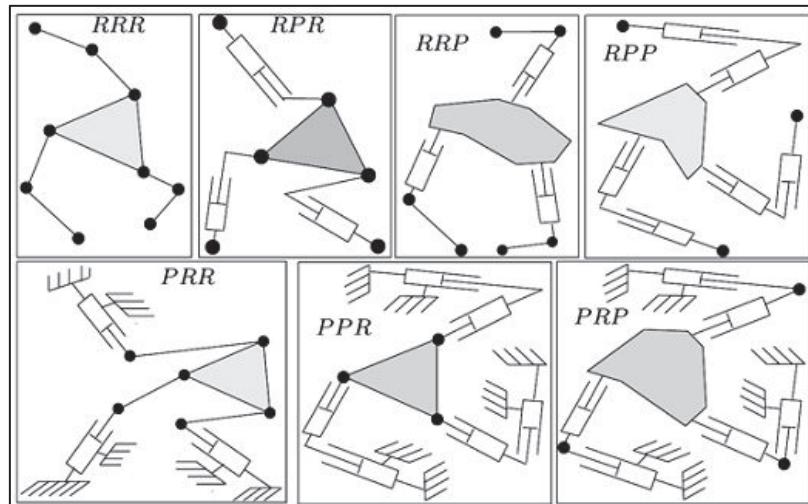


**Figura 1.6.** Robot paralelo planar de 5 barras con su espacio de trabajo [7]

#### 1.3.3.1.2. Tres GDL

Este tipo de robot está formado por tres cadenas cinemáticas independientes accionadas por tres actuadores, sus tres grados de libertad se traducen en un movimiento de su efecto final por el plano x e y, y la rotación a través de un ángulo alrededor del eje z, perpendicular al plano.

Cada una de las cadenas cinemáticas está constituida por dos cuerpos rígidos unidos entre sí por una articulación teniendo un total de tres articulaciones, las cadenas pueden presentar las siguientes secuencias: RRR, RPR, RRP, RPP, PRR, PPR, PRP, PPP (Figura 1.7); donde R, representa una articulación rotacional, y P es una articulación prismática.



**Figura 1.7.** Tipos de cadenas cinemáticas en un robot paralelo planar de 3GDL.[4]

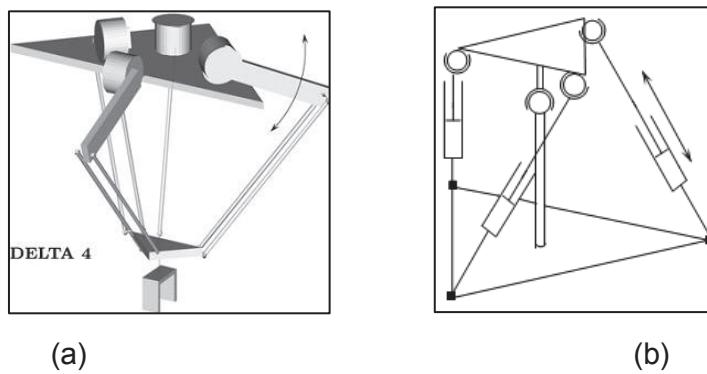
### 1.3.3.2. Robots paralelos de movimientos espaciales.

Este tipo de robot son los que evolucionan dentro del espacio tridimensional pudiendo tener hasta 6 grados de libertad de los cuales 3 son de traslación y 3 son de rotación. En todas las topologías de robots paralelos de movimientos espaciales las uniones de los accionamientos con el efecto final y la base se realizan con juntas universales o esféricas que tienen 2 y 3 grados de libertad respectivamente.

Una forma de clasificar a este tipo de robots es de acuerdo a sus grados de libertad como se detallan a continuación:

#### 1.3.3.2.1. Tres GDL

En esta clasificación se tiene manipuladores de traslación y manipuladores de rotación, el primero es ampliamente usado para aplicaciones pick & place y operaciones de maquinado, el robot más conocido con este tipo de configuración es el tipo delta (Figura 1.8a). El segundo grupo de manipuladores permite rotaciones alrededor de un punto con un movimiento similar al de la muñeca (Figura 1.8b).



**Figura 1.8.** (a) Robot tipo delta, (b) Manipulador de rotación. [4]

#### 1.3.3.2.2. Cuatro GDL

Este tipo de diseño tendrá que depender ya sea de un mecanismo de restricción pasiva, una geometría específica de las cadenas cinemáticas, cadenas cinemáticas

de diferente tamaño, menos de cuatro cadenas cinemáticas o un diseño mecánico específico.

#### *1.3.3.2.3. Cinco GDL*

Los robots con 5 grados de libertad también dependerán de mecanismos de restricción pasiva, geometrías o diseños específicos. Este tipo de estructura es de interés en el campo de las máquinas y herramientas para los llamados cinco ejes de mecanizado.

#### *1.3.3.2.4. Seis GDL*

La construcción de manipuladores de seis grados de libertad se basa en el uso de seis generadores de movimiento tipo D, estos son RRPS, RPRS, PRRS, RRRS. Estos generadores son equivalentes entre sí con respecto a la movilidad de la plataforma ya que el orden de las articulaciones no tiene ninguna importancia.

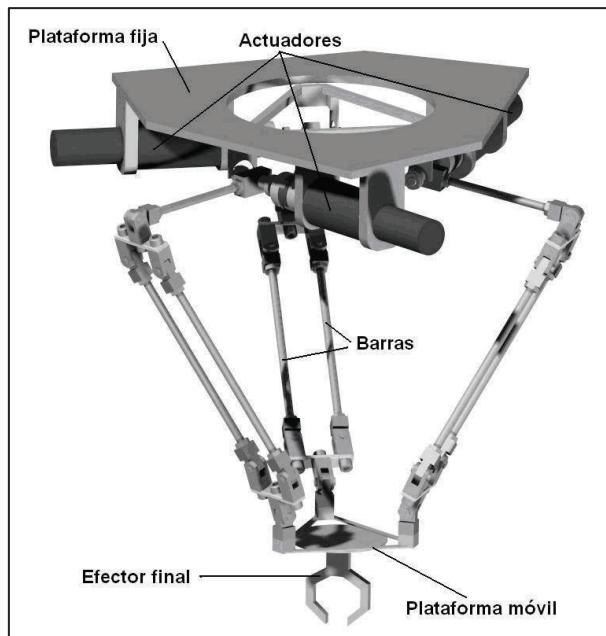
En este tipo de configuraciones el de mayor aplicación es la plataforma Stewart o simulador de vuelo que tienen cadenas de tipo UPS (Universal - Prismatic -Spherical) y SPS (Spherical – Prismatic - Spherical).

### **1.4. ROBOT TIPO DELTA**

En los años 80, Reymond Clavel, profesor en EPFL (École Polytechnique Fédérale de Lausanne) inicia con el uso de paralelogramos para construir una estructura paralela robusta con tres grados de libertad de traslación y un rotatorio. Contrario de las publicaciones científicas hasta el momento, su inspiración era verdaderamente original y no venía de un mecanismo paralelo patentado por Willard L. Pollard en 1942. A esta estructura se le dio el nombre de robot Delta, sin sospechar que al final del siglo, sería uno de los diseños paralelos más acertados con varios cientos de estructuras activas en todo el mundo.

En la Figura 1.9, se muestra un esquema mecánico del robot Delta en donde se puede observar las tres cadenas cinemáticas idénticas, que unen la plataforma fija y la plataforma móvil. [4]

La idea básica del uso de tres de estos paralelogramos restringe totalmente la orientación de la plataforma móvil, que permanece con sólo tres grados de libertad. Los acoplamientos de entrada de los tres paralelogramos están montados en ejes rotativos y estos ejes operaran con la ayuda de motores rotacionales o con actuadores lineales. Finalmente en la plataforma móvil se coloca un efecto final que puede ser de varios tipos dependiendo de la tarea que se realiza o del material con el que se trabaja; este puede ser un electroimán, una ventosa de succión o una garra mecánica.[8]



**Figura 1.9.** Partes de un robot paralelo tipo delta.[28]

#### 1.4.1. CINEMÁTICA DEL ROBOT DELTA.

La cinemática estudia el movimiento de un mecanismo con respecto a un sistema sin tomar en cuenta las fuerzas que intervienen. Así, la cinemática se interesa por la descripción analítica del movimiento espacial del robot como función del tiempo y en

particular por las relaciones entre la posición de la plataforma móvil con los valores que toman sus coordenadas articulares. [2]

#### 1.4.1.1. Cinemática Directa

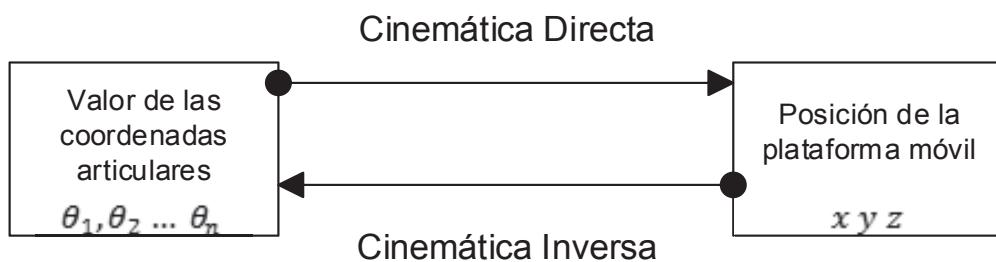
Consiste en determinar cuál es la posición de la plataforma móvil del robot, con respecto a un sistema de coordenadas que se toma como referencia.

El estudio sobre diferentes métodos para el cálculo de la cinemática directa fue realizado por (Merlet 1993). [4]

#### 1.4.1.2. Cinemática Inversa

La cinemática inversa determina la coordenadas articulares de cada actuador para una posición de la plataforma móvil conocida. Existen varios métodos para el cálculo de la cinemática inversa. Merlet (2006) propone dos métodos: el método analítico y el método geométrico, Tsai (1999) hace una breve comparación entre el método de Denavit-Hartenberg y el método geométrico. Ambos autores coinciden en que el método geométrico facilita el análisis de la cinemática inversa de mecanismo cuando se componen de más de un lazo cerrado. [7]

Una forma de entender la relación existente entre la cinemática directa e inversa se puede observar en el diagrama de bloques que se presenta en la Figura 1.10.



**Figura 1.10.** Diagrama de relación entre la cinemática directa e inversa. [2]

#### **1.4.2. ESPACIO DE TRABAJO**

“El espacio de trabajo o campo de acción es el volumen al que puede llegar el extremo del robot.” [9]. El espacio de trabajo de un robot paralelo puede estar restringido por diferentes factores: limitantes mecánicas sobre las junturas pasivas, colisión entre los elementos del robot, limitaciones de los actuadores y variedad de singularidades que pueden dividir el espacio de trabajo en componentes separados [4].

Existen varios enfoques para determinar el espacio de trabajo de un robot paralelo que se resumen a continuación:

##### **1.4.2.1. Método geométrico [16]**

Este método consiste en obtener un objeto geométrico que describa todas las posibles posiciones del efecto final y que además satisfagan las restricciones del brazo del robot. Se debe obtener un objeto geométrico para cada brazo del robot y el espacio de trabajo será la intersección de los objetos geométricos obtenidos.

##### **1.4.2.2. Método de discretización [16]**

En este método el espacio de trabajo está cubierto por una rejilla regular de nodos, ya sean cartesianas o polares. Cada nodo es validado para verificar si pertenece al espacio de trabajo, entonces su límite está formado por todos los nodos validados.

Este método presenta la ventaja de poder tomar en cuenta todas las restricciones mecánicas.

### **1.5. HERRAMIENTAS MATEMÁTICAS**

Para la manipulación de un objeto por parte del robot, este debe mover espacialmente su extremo lo que hace necesario conocer su posición y orientación con respecto a la

base del robot, para llevar a cabo esto se cuenta con diferentes herramientas matemáticas que permiten su análisis, dichas herramientas permiten establecer relaciones espaciales entre distintos objetos con el robot.

### **1.5.1. REPRESENTACIÓN DE LA POSICIÓN.**

Para representar la posición de un objeto en el espacio se necesita un sistema de referencia que tiene ejes perpendiculares entre sí con un origen. El número de ejes depende de la dimensión en la que se quiera trabajar; 2 ejes para dos dimensiones y 3 ejes cuando se trabaja en un espacio tridimensional.

Un cuerpo se puede ubicar en un sistema de referencias mediante coordenadas y estas pueden ser de varios tipos como cartesianas, polares, cilíndricas y esféricas.

### **1.5.2. REPRESENTACIÓN DE LA ORIENTACIÓN. [2]**

Un cuerpo considerado como partícula queda definido en el espacio por su posición, lo que no sucede con un cuerpo sólido ya que para ser definido totalmente también se necesita de su orientación en el sistema de referencias.

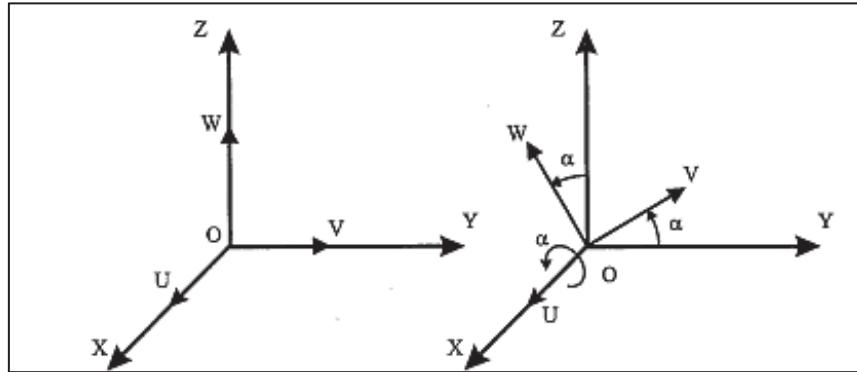
Para poder definir la orientación de un cuerpo sólido en el espacio se necesitan tres componentes linealmente independiente, se parte de un sistema de referencia fijo al cual se le asocia un nuevo sistema de referencia que se moverá junto con el cuerpo sólido y luego se determina la relación espacial existente entre los dos sistemas.

Los métodos para representar la orientación son: matrices de rotación, ángulos de Euler, par de rotación y cuaternios.

#### **1.5.2.1. Matrices de rotación**

Es un método ampliamente utilizado al emplear álgebra matricial en la descripción de orientaciones. Transforma las coordenadas de un vector en un sistema a las de otro,

donde una matriz  $R$  define la orientación del sistema OUVW con respecto al sistema OXYZ, en la Figura 1.11 se puede observar un ejemplo de rotación del sistema OUVW con el eje U y X coincidentes.



**Figura 1.11** Orientación sistema OUVW, con el eje U coincidente con el eje X. [2]

Sistema de dos dimensiones

$$\begin{bmatrix} Px' \\ Py' \end{bmatrix} = R \begin{bmatrix} Px \\ Py \end{bmatrix} \quad (1.1)$$

$$R = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \quad (1.2)$$

Sistema de tres dimensiones

$$\begin{bmatrix} Px' \\ Py' \\ Pz' \end{bmatrix} = R \begin{bmatrix} Px \\ Py \\ Pz \end{bmatrix} \quad (1.3)$$

Nuevo sistema girado sobre el eje 'z'

$$R = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.4)$$

Nuevo sistema girado sobre el eje 'y'

$$R = \begin{bmatrix} \cos \alpha & 0 & \sin \alpha \\ 0 & 1 & 0 \\ -\sin \alpha & 0 & \cos \alpha \end{bmatrix} \quad (1.5)$$

Nuevo sistema girado sobre el eje 'x'

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix} \quad (1.6)$$

$\alpha$  = ángulo de giro del eje

### 1.5.2.2. Ángulos de Euler

En este método se usan tres componentes a diferencia de los nueve que se utilizan en las matrices de rotación. Este método dice, que un sistema OUVW puede definirse con respecto a un sistema OXYZ mediante tres ángulos  $\phi, \theta, \psi$  denominados ángulos de Euler, se necesita saber los ejes sobre los cuales se va a realizar los giros y el valor de los ángulos, de esta manera se obtienen secuencias de giro que no cumplen con la ley commutativa y existen alrededor de 24 secuencias formalmente definidas, siendo una de las más comunes la secuencia ZXZ como se muestra en la Figura 1.12.

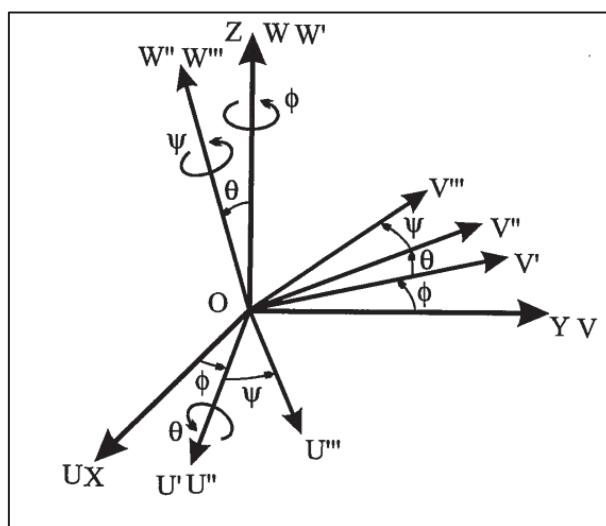
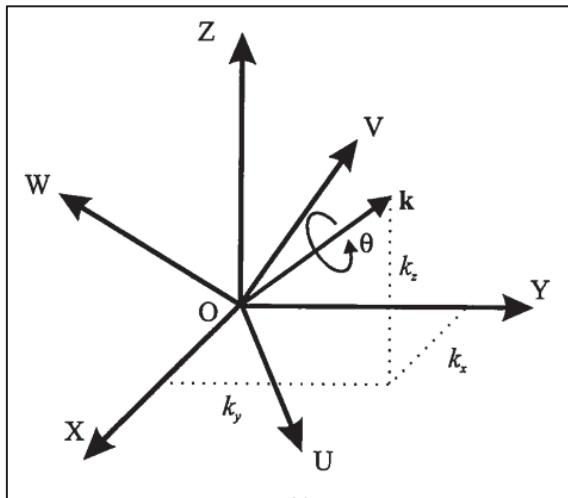


Figura 1.12. Ángulos de Euler secuencia ZXZ y sistema final OU'''V'''W''''. [2]

### 1.5.2.3. Par de rotación

Para representar la orientación de un sistema OUVW respecto a un sistema fijo OXYZ mediante el par de rotación es necesario definir 2 parámetros, un vector  $k(k_x, k_y, k_z)$  y un ángulo de giro  $\theta$ , el vector  $k$  nace del origen común de los dos sistemas y el sistema OUVW gira respecto al sistema fijo tomando como eje dicho vector como se muestra en la Figura 1.13.



**Figura 1.13.** Orientación par de rotación  $Rot(k, \theta)$ . [2]

### 1.5.2.4. Cuaternios

Los cuaternios permiten describir las rotaciones en el espacio tridimensional, debido a esto se los utiliza en gráficos por computadora, videojuegos y en robótica. Son una extensión de los números reales como los imaginarios con la diferencia que los cuaternios tienen la adición de tres unidades imaginarias.

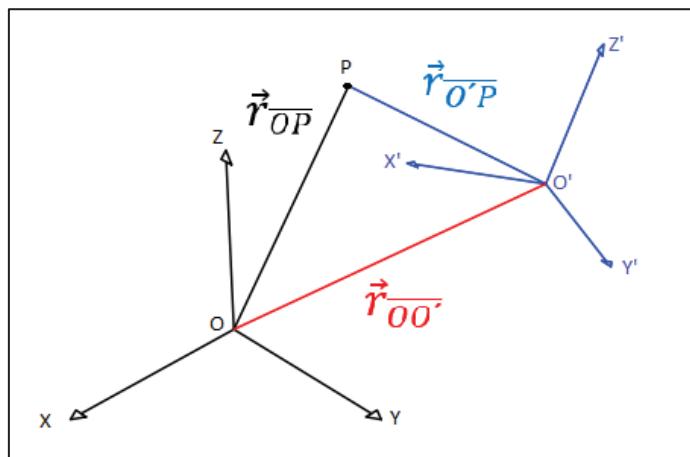
Consta de dos partes una escalar y una vectorial, para este caso la parte escalar será el ángulo de giro, y la vectorial el eje de giro del sistema.

$$Q = Rot(k, \theta) = (\cos \frac{\theta}{2}, k * \sin \frac{\theta}{2}) \quad (1.7)$$

### 1.5.3. MATRICES DE TRANSFORMACIÓN HOMOGÉNEA.

Dentro del análisis cinemático de un robot y en este caso un manipulador, se suelen colocar diferentes sistemas de referencia equivalente al número de articulaciones existentes, dichos sistemas se encuentran trasladados y girados con respecto a un sistema de base fija, por lo tanto se necesita una herramienta que permita transformar la posición y la orientación de los sistemas móviles con respecto a un sistema fijo.

En la Figura 1.14 se puede observar un punto P que está definido en un sistema móvil  $O'X'Y'Z'$ , este se encuentra a una determinada distancia del sistema fijo  $OXYZ$ , mediante suma de vectores se puede encontrar la distancia del punto P con respecto al sistema fijo.



**Figura 1.14.** Sistema móvil y sistema fijo asociado a un punto en común.

El vector  $\overrightarrow{OP}_f$  puede ser escrito como una suma de los vectores  $\overrightarrow{OO'}_f$  y  $\overrightarrow{O'P}_f$  referidos al sistema fijo.

$$\overrightarrow{OP}_f = \overrightarrow{OO'}_f + \overrightarrow{O'P}_f \quad (1.8)$$

Donde  $\overrightarrow{O'P}_f$  puede ser escrito de una manera tal que se tome en cuenta la rotación del sistema móvil con respecto al fijo.

$$\overrightarrow{O'P_f} = R_{m_f} * \overrightarrow{O'P_m} \quad (1.9)$$

Quedando la ecuación de la siguiente manera:

$$\overrightarrow{OP_f} = \overrightarrow{OO'}_f + (R_{m_f} * \overrightarrow{O'P_m}) \quad (1.10)$$

$\overrightarrow{OP_f}$  = Posición  $P$  respecto al sistema fijo

$\overrightarrow{OO'}_f$  = Vector de traslación del sistema móvil

$(R_{m_f})$  = Cambio de orientación sistema móvil respecto al sistema fijo

$(\overrightarrow{O'P_m})$  = Posición  $P$  respecto al sistema móvil

La definición de matriz de transformación homogénea satisface el problema de encontrar un operador compacto en el que conste la traslación del sistema móvil y el cambio de orientación con respecto a un sistema fijo.

Para definir la matriz de transformación homogénea es necesario introducir un término conocido como coordenada homogénea, la cual para mostrar la localización espacial de un sólido en un sistema de dimensión ( $n$ ) usa coordenadas de un espacio de dimensión ( $n + 1$ ); por ejemplo para representar el vector  $P(P_x, P_y, P_z)$  en coordenadas homogéneas se lo hará de la siguiente manera  $P(\lambda P_x, \lambda P_y, \lambda P_z, \lambda)$ , donde el valor de  $\lambda$  puede ser cualquier valor diferente de cero y se denomina valor de escalamiento. Para el caso de la robótica el valor del escalamiento es uno por conveniencia.

$$P = \begin{bmatrix} \lambda P_x \\ \lambda P_y \\ \lambda P_z \\ \lambda \end{bmatrix} = \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} \quad (1.11)$$

"La matriz de transformación homogénea se define como una matriz de dimensión 4x4 que representa la transformación de un vector de coordenadas homogéneas de un sistema a otro". [1]

$$T = \begin{bmatrix} Rotación_{3x3} & Traslacion_{3x1} \\ Perspectiva_{1x3} & Escala_{1x1} \end{bmatrix}$$

Para el caso de la robótica la perspectiva es un vector nulo y la escala es unitaria, por lo tanto la matriz de transformación homogénea es:

$$T = \begin{bmatrix} R_{mf}_{3x3} & \overrightarrow{OO'}_{3x1} \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \quad (1.12)$$

## 1.6. VISIÓN ARTIFICIAL [10][11][12]

El prototipo construido tiene como objetivo coger y colocar objetos dentro de un espacio de trabajo establecido, entonces es en este punto donde surge un inconveniente al no saber cómo ubicar los objetos para que el efecto final se coloque en su posición. Una manera es colocar los objetos en puntos determinados para que el robot opere siempre en esas posiciones, otra manera es ubicar y localizar los objetos por medio de visión artificial.

La visión artificial se puede definir como un campo de la Inteligencia artificial, que mediante la utilización de técnicas adecuadas se puede obtener información especial a través de imágenes digitales y de esta manera tomar alguna acción similar a lo que hace el conjunto vista humana-cerebro.

### 1.6.1. PROCESO DE LA VISIÓN ARTIFICIAL

Los procesos para poder realizar el análisis de imágenes son:

- Adquisición de imágenes.
- Digitalización de imágenes.
- Procesado
- Interpretación de los resultados.

#### **1.6.1.1. Adquisición de imágenes.**

Adquirir una imagen es obtener la información real de la escena mediante la proyección de la misma, esta función la lleva a cabo una cámara mediante las vías ópticas, para luego poder transferirla a un sistema electrónico. Es muy importante realizar una adecuada selección de la cámara que se va a utilizar, porque existe una para cada aplicación como: color, monocromo, alta definición, alta sensibilidad, alta velocidad, infrarrojas, etc.

El problema al capturar una imagen es la iluminación de la escena, esto puede generar un sinnúmero de anomalías difíciles de eliminar como son las sombras, altos y bajos niveles de contraste, contornos no muy definidos y es por esta razón que este punto se debe tomar con la mayor atención posible.

#### **1.6.1.2. Digitalización de las imágenes.**

Es almacenar la señal que proviene de un módulo de digitalización que se encuentra posterior a la captura de imagen, este módulo convierte la señal eléctrica a código binario que puede ser interpretado por el computador para conformar la imagen que es objeto de estudio.

#### **1.6.1.3. Procesado.**

El procesado se realiza mediante el computador para poder extraer información contenida en una imagen, los procesamientos fundamentales de una imagen son:

- Mejora de la imagen: conjunto de procesos que se utilizan para mejorar la imagen digital, las mejoras pueden ser eliminación de ruido aplicando diversos filtros, este proceso se realiza para que la imagen pueda ser utilizada a futuro y no arroje problemas la información obtenida.
- Segmentación de la imagen: es el proceso de dividir una imagen en múltiples partes, usado típicamente para identificar objetos y otra información relevante dentro de una imagen.
- Registro de imágenes: proceso que permite una comparación cuantitativa de un conjunto o configuración de datos, elimina cambios debido al movimiento de la cámara. En este punto se puede hacer el seguimiento de un objeto con determinadas características.

#### 1.6.1.4. Interpretación de los resultados.

Terminado el procesado la información obtenida se puede usar para ejecutar acciones con el robot, en este caso en particular se obtendrán los centros de los objetos colocados en el espacio de trabajo para poder ubicar el efecto final del prototipo.

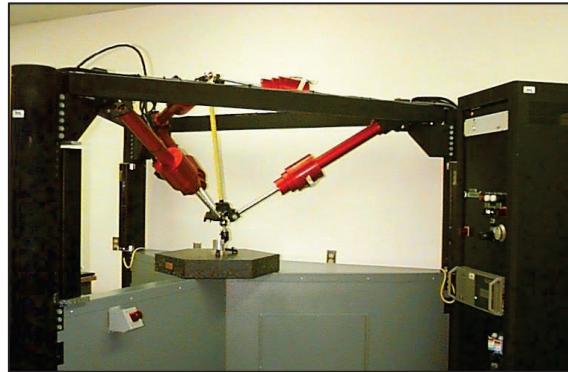
### 1.7. APLICACIONES DEL ROBOT DELTA

Este manipulador debido a su gran velocidad es comúnmente utilizado en la sujeción y empacado de objetos delicados como es el caso de la industria alimenticia. En la Figura 1.15, se observa la instalación de varios robots tipo delta para el envasado de galletas en una panadería industrial.



Figura 1.15. Robot delta utilizado en el proceso de empaquetamiento de galletas. [8]

La alta precisión de la estructura en forma de tetraedro le confiere una gran rigidez y las convierten en buenas candidatas para máquinas de medición como el de la Figura 1.16, con una precisión de medida que va en el orden de los 0.05 milímetros.



**Figura 1.16.** Máquina de medición HCCM desarrollado por la universidad de Florida y Perry automation. [13]

Este tipo de robot en los últimos tiempos ha ido incursionando lentamente en el área médica facilitando el posicionamiento de microscopios e incluso en la cirugía de pacientes, como se observa en la Figura 1.17.



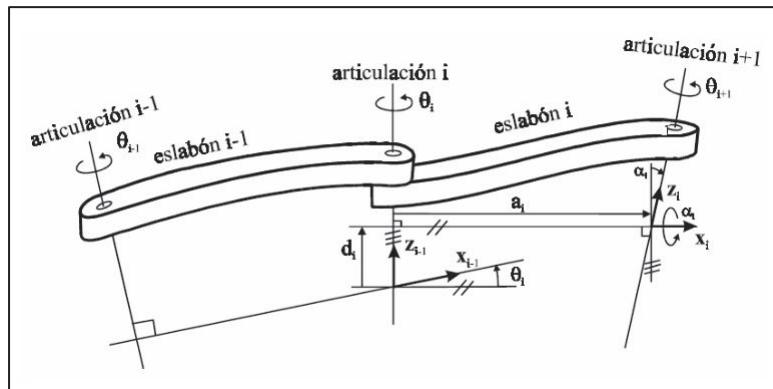
**Figura 1.17.** Robot delta para cirugías. [8]

## CAPITULO 2

### DISEÑO E IMPLEMENTACIÓN DEL PROTOTIPO

#### 2.1. ANÁLISIS CINEMÁTICO

Los análisis más comunes empleados para la resolución de la cinemática directa e inversa son la representación de Denavit-Hartenberg (D-H) y la solución geométrica. La representación D-H es un método sistemático para la selección de marcos de referencia en cada eslabón del robot de tal forma que la matriz de transformación homogénea entre ellos dependa de cuatro parámetros que serán descritos en base a la Figura 2.1:



**Figura 2.1.** Convención de parámetros para la representación D-H. [14]

$\alpha_i$ : Corresponde al ángulo de rotación sobre el eje  $x_i$ , es decir el ángulo existente entre los ejes  $z_{i-1}$  y  $z_i$ .

$a_i$ : Distancia de translación a lo largo del eje  $x_i$  desde el eje  $z_{i-1}$  hasta el origen del sistema i-esimo.

$\theta_i$ : Ángulo de rotación sobre el eje  $z_{i-1}$ , es decir, el ángulo existente entre los ejes  $x_{i-1}$  y  $x_i$ .

$d_i$ : Distancia de translación a lo largo del eje  $z_{i-1}$  desde el origen del sistema de coordenadas (i-1)-esimo hasta el eje  $x_i$  [14].

El análisis geométrico se basa en la descomposición de la geometría espacial del robot en varios problemas geométricos planos, para los cuales se usan herramientas geométricas y trigonométricas para encontrar el valor de las variables articulares o la posición y orientación del elemento terminal.

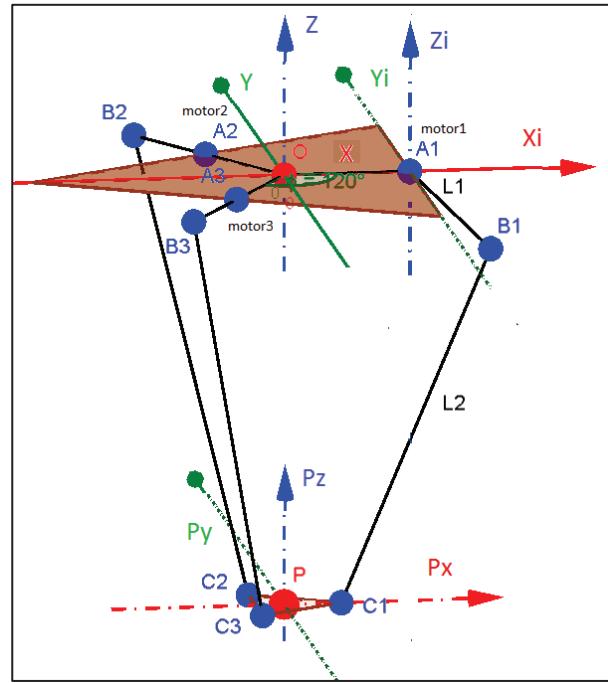
Para el robot delta la existencia de los tres lazos cerrados dificulta la aplicación de la representación D-H. Además, la existencia de articulaciones con más de un grado de libertad como el caso de las articulaciones esféricas imposibilitan usar esta representación tal y como está definida, pues más de cuatro parámetros por eslabón serían necesarios para definir completamente una determinada configuración del robot. Es por ello que para este tipo de robots se prefiere el enfoque geométrico, definiéndose ecuaciones vectoriales para cada brazo en donde las variables articulares pasivas pueden ser eliminadas del análisis [15].

### **2.1.1 GEOMETRÍA DEL ROBOT DELTA [15] [16]**

El análisis geométrico se desarrolla en base a la Figura 2.2, en la base fija se ubican los tres motores a una misma distancia del centro de la base y separados  $120^\circ$  entre sí. Se coloca un sistema de referencia global  $OXYZ$  en el centro de la base fija, con el eje  $Z$  perpendicular a la misma y el eje  $X$  perpendicular al eje del motor 1, la base móvil está conectada a la base fija por medio de tres cadenas cinemáticas, cada uno compuesta por dos eslabones, cuyas dimensiones son  $L_1$  y  $L_2$  respectivamente.

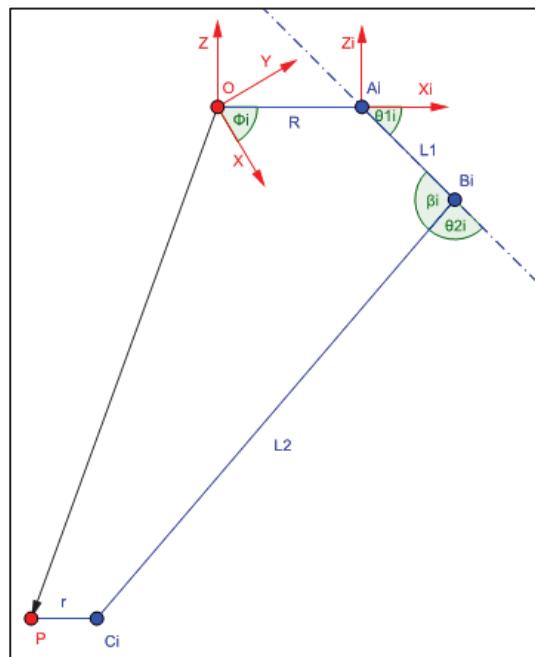
El eslabón  $L_1$  se une a la base fija por medio de la articulación  $A_i$  ( $i = 1,2,3$ ) y al eslabón  $L_2$  por la articulación  $B_i$ . El eslabón  $L_2$  se une a la base móvil por medio de la articulación  $C_i$ , el punto  $P(P_x, P_y, P_z)$  colocado en el centro de la plataforma móvil, representa la posición operacional del robot con respecto al sistema referencial global  $OXYZ$ .

Sobre cada brazo se selecciona un sistema de coordenadas  $X_iY_iZ_i$  localizados a una distancia  $R$  del sistema de referencia global  $OXYZ$ , y rotado un ángulo  $\phi_i$  ( $i = 1,2,3$ ) igual a  $0^\circ$ ,  $120^\circ$  y  $240^\circ$  respectivamente.



**Figura 2.2.** Modelo geométrico del robot delta

Considerando la simetría del robot, cada cadena cinemática puede ser tratada por separado. En la Figura 2.3 se define los ángulos de las articulaciones asociadas a la  $i$ -ésima cadena cinemática.



**Figura 2.3** Descripción de los ángulos  $\theta_{1i}, \theta_{2i}, \beta_i$  y  $\phi_i$ .

$\overrightarrow{OP}$ : Vector de posición del centro de la base móvil.

$\theta_{1i}$ : Es el ángulo que se forma entre  $X_i$  y el vector  $\overrightarrow{A_iB_i}$ .

$\theta_{2i}$ : Es el ángulo que se forma entre la proyección de  $\overrightarrow{A_iB_i}$  y la intersección del plano del paralelogramo y el plano  $X_iZ_i$ .

$\beta_i$ : Es el ángulo formado entre los eslabones  $L_1$  y  $L_2$ .

Una matriz de transformación  $T$  representa la orientación y la posición de un sistema  $O'X'Y'Z'$  rotado y trasladado con respecto a un sistema de referencia  $OXYZ$ . Esta matriz es útil para conocer las coordenadas del vector  $\vec{r} = [r_x, r_y, r_z]^T$  en el sistema de referencia, a partir del vector  $\vec{r}' = [r'_x, r'_y, r'_z]^T$  en el sistema de referencia  $O'X'Y'Z'$  [2].

$$\begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} = T \begin{bmatrix} r'_x \\ r'_y \\ r'_z \end{bmatrix} \quad (2.1)$$

Donde la matriz  $T$  es:

$$T = \begin{bmatrix} R_{3x3} & P_{3x1} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \text{Rotación} & \text{Traslación} \\ 0 & 1 \end{bmatrix} \quad (2.2)$$

El sistema  $A_iX_iY_iZ_i$  está trasladado  $R$  a lo largo del eje  $OX$  y rotado  $\phi_i$  alrededor de  $OZ$ , con respecto al sistema de referencia  $OXYZ$ . Para conocer los vectores  $\overrightarrow{OA_i}$ ,  $\overrightarrow{OB_i}$  y  $\overrightarrow{OC_i}$ , conociendo los vectores  $\overrightarrow{A_iB_i}$  y  $\overrightarrow{A_iC_i}$ , se utiliza una traslación seguida de una rotación. Si se traslada un vector con coordenadas  $XYZ$ , seguidas de una rotación sobre el eje  $Z$ , la matriz  $T$  es:

$$T(P, (Z, \alpha)) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 & Px\cos(\alpha) - Py\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) & 0 & Px\sin(\alpha) + Py\cos(\alpha) \\ 0 & 0 & 1 & Pz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

Empleando la ecuación (2.1) y (2.3) se obtienen los vectores  $\overrightarrow{OA_i}$ ,  $\overrightarrow{OB_i}$  y  $\overrightarrow{OC_i}$ .

Por facilidad de expresión se denota el coseno del ángulo  $\text{Cos}(x)$  como  $c(x)$  y  $\text{Sen}(x)$  como  $s(x)$ .

$$\overrightarrow{OA}_I = \begin{bmatrix} c\phi_i & -s\phi_i & 0 & R c\phi_i \\ s\phi_i & c\phi_i & 0 & R s\phi_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} R c\phi_i \\ R s\phi_i \\ 0 \\ 1 \end{bmatrix} \quad (2.4)$$

$$\overrightarrow{OB}_I = \begin{bmatrix} c\phi_i & -s\phi_i & 0 & R c\phi_i \\ s\phi_i & c\phi_i & 0 & R s\phi_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} L_1 c\theta_{1i} \\ 0 \\ -L_1 s\theta_{1i} \\ 1 \end{bmatrix} = \begin{bmatrix} c\phi_i(L_1 c\theta_{1i} + R) \\ s\phi_i(L_1 c\theta_{1i} + R) \\ -L_1 s\theta_{1i} \\ 1 \end{bmatrix} \quad (2.5)$$

$$\overrightarrow{OC}_I = \begin{bmatrix} c\phi_i & -s\phi_i & 0 & R c\phi_i \\ s\phi_i & c\phi_i & 0 & R s\phi_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_x c\phi_i + P_y s\phi_i + r - R \\ P_y c\phi_i - P_x s\phi_i \\ P_z \\ 1 \end{bmatrix} = \begin{bmatrix} P_x + r c\phi_i \\ P_y + r s\phi_i \\ P_z \\ 1 \end{bmatrix} \quad (2.6)$$

Con las ecuaciones (2.4) a (2.6) se obtienen los vectores  $\overrightarrow{BA}_I$ ,  $\overrightarrow{BC}_I$  y  $\overrightarrow{AC}_I$ .

$$\overrightarrow{BA}_I = \overrightarrow{OA}_I - \overrightarrow{OB}_I = \begin{bmatrix} -c\phi_i L_1 c\theta_{1i} \\ -s\phi_i L_1 c\theta_{1i} \\ L_1 s\theta_{1i} \end{bmatrix} \quad (2.7)$$

$$\overrightarrow{BC}_I = \overrightarrow{OC}_I - \overrightarrow{OB}_I = \begin{bmatrix} P_x - c\phi_i(R + L_1 c\theta_{1i} - r) \\ P_y - s\phi_i(R + L_1 c\theta_{1i} - r) \\ P_z + L_1 s\theta_{1i} \end{bmatrix} \quad (2.8)$$

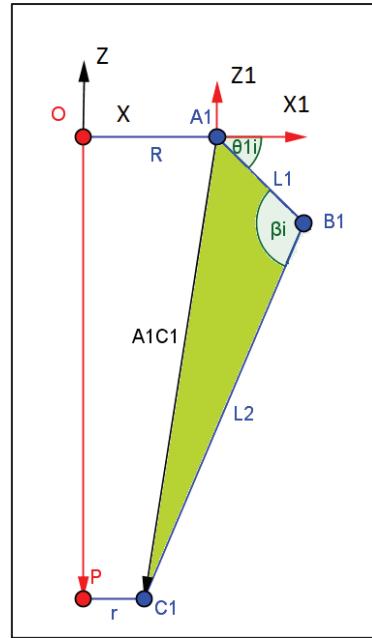
$$\overrightarrow{AC}_I = \overrightarrow{OC}_I - \overrightarrow{OA}_I = \begin{bmatrix} P_x - c\phi_i(r - R) \\ P_y - s\phi_i(r - R) \\ P_z \end{bmatrix} \quad (2.9)$$

### 2.1.2. DESARROLLO DE LA CINEMÁTICA INVERSA. [16]

La solución del modelo cinemático inverso permite encontrar los ángulos de entrada al robot  $\theta_{1i}$  para una posición dada del punto  $P (P_x, P_y, P_z)$ .

Con los vértices  $A_iB_iC_i$  se forma un triángulo que se aprecia en la Figura 2.4. Aplicando la ley de coseno se puede determinar la magnitud de  $A_iC_i$ . Entonces se tiene que:

$$A_iC_i^2 = L_1^2 + L_2^2 - 2L_1L_2 \cos(\beta_i) \quad (2.10)$$



**Figura 2.4.** Representación vectorial de la primera cadena cinemática.

El  $\cos(\beta_i)$  se calcula aplicando la definición de producto escalar:

$$\overrightarrow{B_iA_i} * \overrightarrow{B_iC_i} = \|B_iA_i\| \|B_iC_i\| \cos(\beta_i) \quad (2.11)$$

Como  $\|B_iA_i\| = L_1$  y  $\|B_iC_i\| = L_2$ , se tiene que:

$$\cos(\beta_i) = \frac{\overrightarrow{B_iA_i} * \overrightarrow{B_iC_i}}{L_1L_2} \quad (2.12)$$

Ahora, remplazando (2.12) en (2.10) se obtiene:

$$A_iC_i^2 = L_1^2 + L_2^2 - 2(\overrightarrow{B_iA_i} * \overrightarrow{B_iC_i}) \quad (2.13)$$

Multiplicando (2.7) y (2.8), se obtiene:

$$\overrightarrow{B_l A_l} * \overrightarrow{B_l C_l} = L_1^2 - L_1 c \theta_{1i} (P_x c \phi_i + P_y s \phi_i + r - R) - L_1 s \theta_{1i} P_z \quad (2.14)$$

Remplazando (2.14) en (2.13) se obtiene:

$$A_i C_i^2 = L_2^2 - L_1^2 + 2L_1 c \theta_{1i} (P_x c \phi_i + P_y s \phi_i + r - R) - 2L_1 s \theta_{1i} P_z \quad (2.15)$$

La magnitud del vector  $\overrightarrow{A_l C_l}$  de la ecuación (2.9) es:

$$\|\overrightarrow{A_l C_l}\| = A_i C_i^2 = (P_x + c \phi_i (r - R))^2 + (P_y + s \phi_i (r - R))^2 + P_z^2 \quad (2.16)$$

Igualando (2.16) y (2.15) se obtiene una ecuación de la forma:

$$a_i \sin \theta_{1i} + b_i \cos \theta_{1i} = c_i \quad (2.17)$$

Donde:

$$a_i = -2P_z L_1$$

$$b_i = 2L_1 (P_x c \phi_i + P_y s \phi_i + r - R)$$

$$c_i = (P_x + c \phi_i (r - R))^2 + (P_y + s \phi_i (r - R))^2 + P_z^2 + L_1^2 - L_2^2$$

Tomando en cuenta las identidades trigonométricas de suma de senos y cosenos se obtiene:

$$a_i \sin \theta_{1i} + b_i \cos \theta_{1i} = R \sin(\theta_{1i} - \alpha) \quad (2.18)$$

Donde:

$$R = \sqrt{a_i^2 + b_i^2}$$

$$\alpha = \tan^{-1} \left( \frac{b_i}{a_i} \right)$$

Reemplazando la ecuación (2.18) en (2.17) se tiene:

$$\sqrt{a_i^2 + b_i^2} \sin \left( \theta_{1i} + \tan^{-1} \left( \frac{b_i}{a_i} \right) \right) = c_i \quad (2.19)$$

Resolviendo la ecuación (2.19) se obtiene la solución:

$$\theta_{1i} = \sin^{-1} \left( \frac{c_i}{\sqrt{a_i^2 + b_i^2}} \right) - \tan^{-1} \left( \frac{b_i}{a_i} \right) \quad (2.20)$$

Por lo tanto se puede determinar los ángulos de las articulaciones:

$\theta_{11}$ , considerando el valor de  $\phi_1$  igual a  $0^\circ$  se tiene que:

$$a_1 = -2P_z L_1$$

$$b_1 = 2L_1(P_x + r - R)$$

$$c_1 = (P_x + (r - R))^2 + P_y^2 + P_z^2 + L_1^2 - L_2^2$$

$$\theta_{11} = \sin^{-1} \left( \frac{c_1}{\sqrt{a_1^2 + b_1^2}} \right) - \tan^{-1} \left( \frac{b_1}{a_1} \right) \quad (2.21)$$

$\theta_{12}$ , considerando el valor de  $\phi_2$  igual a  $120^\circ$  se tiene que:

$$a_2 = -2P_z L_1$$

$$b_2 = L_1(-P_x + \sqrt{3}P_y + 2r - 2R)$$

$$c_2 = (P_x - \frac{1}{2}(r - R))^2 + (P_y + \frac{\sqrt{3}}{2}(r - R))^2 + P_z^2 + L_1^2 - L_2^2$$

$$\theta_{12} = \sin^{-1} \left( \frac{c_2}{\sqrt{a_2^2 + b_2^2}} \right) - \tan^{-1} \left( \frac{b_2}{a_2} \right) \quad (2.22)$$

$\theta_{13}$ , considerando el valor de  $\emptyset_3$  igual a  $240^\circ$  se tiene que:

$$\begin{aligned} a_2 &= -2P_z L_1 \\ b_2 &= L_1(-2P_x - \sqrt{3}P_y + 2r - 2R) \\ c_2 &= (P_x - \frac{1}{2}(r - R))^2 + (P_y - \frac{\sqrt{3}}{2}(r - R))^2 + P_z^2 + L_1^2 - L_2^2 \end{aligned}$$

$$\theta_{13} = \sin^{-1} \left( \frac{c_3}{\sqrt{a_3^2 + b_3^2}} \right) - \tan^{-1} \left( \frac{b_3}{a_3} \right) \quad (2.23)$$

### 2.1.3. DESARROLLO DE LA CINEMÁTICA DIRECTA. [16]

Para este caso los ángulos  $\theta_{1i}$  ( $i = 1, 2, 3$ ) son datos y se debe determinar la posición del punto  $P$  ( $P_x, P_y, P_z$ ).

Partiendo de la ecuación (2.8) donde se define el vector  $\overrightarrow{BC}_l$  y cuya magnitud es  $L_2$  se tiene:

$$\begin{aligned} \|\overrightarrow{BC}_l\| &= (P_x - c\emptyset_i(R + L_1 c\theta_{1i} - r))^2 + (P_y - s\emptyset_i(R + L_1 c\theta_{1i} - r))^2 + (P_z + L_1 s\theta_{1i})^2 = \\ &= L_2^2 \quad (2.24) \end{aligned}$$

Con la ecuación (2.24) se obtiene un sistema de tres ecuaciones no lineales que relacionan las variables de las articulaciones  $\theta_{1i}$  ( $i = 1, 2, 3$ ) con la posición del efecto móvil.

Para  $\emptyset_1$  igual a  $0^\circ$ :

$$(P_x - R - L_1 c\theta_{11} + r)^2 + (P_y)^2 + (P_z + L_1 s\theta_{11})^2 = L_2^2 \quad (2.25)$$

La ecuación (2.25) se puede escribir de la siguiente forma:

$$P_x^2 + 2a_1 P_x + P_y^2 + P_z^2 - 2a_2 P_z = a_3 \quad (2.26)$$

Donde:

$$a_1 = r - R - L_1 c\theta_{11}$$

$$a_2 = L_1 s\theta_{11}$$

$$a_3 = L_2^2 - a_1^2 - a_2^2$$

Para  $\phi_2$  igual a  $120^\circ$ :

$$(P_x + \frac{1}{2}(R + L_1 c\theta_{12} - r))^2 + (P_y - \frac{\sqrt{3}}{2}(R + L_1 c\theta_{12} - r))^2 + (P_z + L_1 s\theta_{12})^2 = L_2^2 \quad (2.27)$$

La ecuación (2.27) se puede escribir de la siguiente forma:

$$P_x^2 + b_1 P_x + P_y^2 - \sqrt{3}b_1 P_y + P_z^2 - 2b_2 P_z = b_3 \quad (2.28)$$

Donde:

$$b_1 = R - r + L_1 c\theta_{12}$$

$$b_2 = L_1 s\theta_{12}$$

$$b_3 = L_2^2 - b_1^2 - b_2^2$$

Para  $\phi_3$  igual a  $240^\circ$ :

$$(P_x + \frac{1}{2}(R + L_1 c\theta_{13} - r))^2 + (P_y + \frac{\sqrt{3}}{2}(R + L_1 c\theta_{13} - r))^2 + (P_z + L_1 s\theta_{13})^2 = L_2^2 \quad (2.29)$$

La ecuación (2.29) se puede escribir de la siguiente forma:

$$P_x^2 + c_1 P_x + P_y^2 + \sqrt{3}c_1 P_y + P_z^2 - 2c_2 P_z = c_3 \quad (2.30)$$

Donde:

$$c_1 = R - r + L_1 c \theta_{13}$$

$$c_2 = L_1 s \theta_{13}$$

$$c_3 = L_2^2 - c_1^2 - c_2^2$$

Se restan las ecuaciones (2.28) y (2.30) con la ecuación (2.26) respectivamente, se obtienen los siguientes resultados:

$$b_1 P_x - 2a_1 P_x - \sqrt{3} b_1 P_y - 2b_2 P_z + 2a_2 P_z = b_3 - a_3 \quad (2.31)$$

$$c_1 P_x - 2a_1 P_x + \sqrt{3} c_1 P_y - 2c_2 P_z - 2a_2 P_z = c_3 - a_3 \quad (2.32)$$

Despejando  $P_x$  de las ecuaciones (2.31) y (2.32):

$$P_x = \frac{b_3 - a_3 + \sqrt{3} b_1 P_y + 2b_2 P_z - 2a_2 P_z}{b_1 - 2a_1} \quad (2.33)$$

$$P_x = \frac{c_3 - a_3 - \sqrt{3} c_1 P_y + 2c_2 P_z + 2a_2 P_z}{c_1 - 2a_1} \quad (2.34)$$

Igualando las ecuaciones (2.33) y (2.34) se determina  $P_y$ :

$$P_y = \frac{A + B P_z}{C} \quad (2.35)$$

Donde:

$$A = \frac{a_3 - b_3}{2a_1 - b_1} - \frac{a_3 - c_3}{2a_1 - c_1}$$

$$B = \frac{2(c_2 - a_2)}{2a_1 - c_1} - \frac{2(b_2 - a_2)}{2a_1 - b_1}$$

$$C = \frac{\sqrt{3} b_1}{2a_1 - b_1} + \frac{\sqrt{3} c_1}{2a_1 - c_1}$$

$$P_x = D - E P_z \quad (2.36)$$

Donde:

$$D = \frac{a_3 - b_3}{2a_1 - b_1} - \frac{\sqrt{3}b_1A}{C(2a_1 - b_1)}$$

$$E = \frac{2(b_2 - a_2)}{2a_1 - b_1} + \frac{\sqrt{3}b_1A}{C(2a_1 - b_1)}$$

Sustituyendo las ecuaciones (2.35), (2.36) en (2.28) se obtiene:

$$aP_z^2 + bPz + c = 0 \quad (2.37)$$

Donde:

$$a = E^2 + \left(\frac{B}{C}\right)^2 + 1$$

$$b = 2\left(\frac{AB}{C^2} - DE - a_1E - a_2\right)$$

$$c = D^2 + \left(\frac{A}{C}\right)^2 + 2a_1D - a_3$$

La ecuación (2.37) es un sistema de segundo orden el cual tiene dos soluciones, pero si se considera el sistema de referencia del robot Delta,  $P_z$  no puede tener valores positivos, finalmente se obtiene:

$$P_z = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \quad (2.38)$$

#### 2.1.4. ANALISIS DE SINGULARIDADES [2] [4] [17]

Una configuración singular es aquella en la que el efecto final adopta una posición de tal manera que hace que el robot paralelo pierda su rigidez, o este forzando el movimiento de una juntura, pudiendo de esta manera desestabilizar al robot.

Si se considera un robot paralelo de  $n$  grados de libertad, donde el movimiento de su efecto final está dado por una configuración de parámetros  $X$ , mientras que el movimiento de las juntas tanto pasivas como activas son descritas por  $\theta$ . El cálculo de las singularidades se realiza mediante el análisis de una matriz denominada Jacobiana, la cual relaciona las derivadas de los parámetros  $X$  y  $\theta$ .

#### 2.1.4.1. Matriz Jacobiana

En el análisis de singularidad de una cadena cinemática cerrada se plantea que estas poseen un vector de entrada  $\theta$  y un vector de salida  $X$ , que corresponden a las articulaciones activas y a las coordenadas cartesianas del efecto final respectivamente, donde la relación de estos vectores es:

$$F(\theta, x) = 0 \quad (2.39)$$

Donde  $F$  es una función implícita  $n$ -dimensional de  $\theta$  y  $x$ , por otra parte  $0$  es el vector cero  $n$ -dimensional. Derivando la ecuación (2.39) con respecto al tiempo se tiene la relación entre las velocidades de entrada y salida de la siguiente forma:

$$\frac{\partial F}{\partial \theta} \dot{\theta} + \frac{\partial F}{\partial x} \dot{x} = J_\theta \dot{\theta} + J_x \dot{x} = 0 \quad (2.40)$$

Donde la matriz Jacobiana permite establecer una relación entre las velocidades de entrada y salida en un manipulador paralelo como se muestra a continuación:

$$\dot{\theta} = J \dot{x} \quad (2.41)$$

Donde:

$$J = (J_\theta)^{-1} J_x \quad (2.42)$$

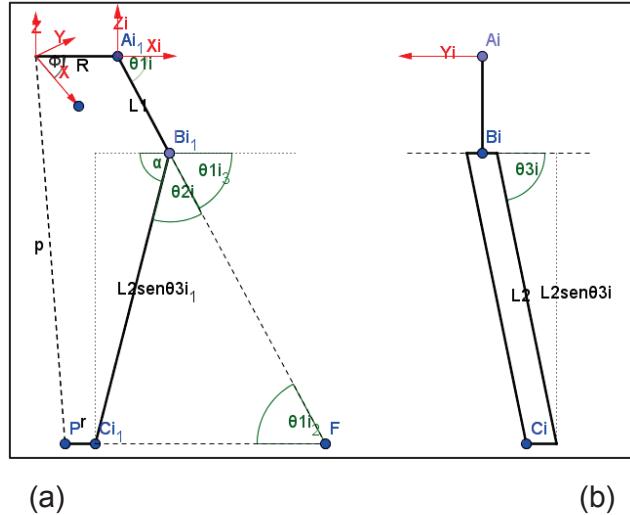
Entonces se obtiene una matriz Jacobiana que queda separada en una matriz asociada a la cinemática directa  $J_x$  y otra matriz asociada a la cinemática inversa  $J_\theta$ , donde las dos matrices son de orden  $nxn$ .

De esta forma la ecuación (2.41) se puede escribir de la siguiente manera:

$$\begin{bmatrix} \dot{\theta}_{11} \\ \dot{\theta}_{12} \\ \dot{\theta}_{13} \end{bmatrix} = (J_\theta)^{-1} J_x \begin{bmatrix} V_{px} \\ V_{py} \\ V_{pz} \end{bmatrix} \quad (2.43)$$

Basado en la Figura 2.5 (a), se establece una  $i$ -ésima cadena cinemática cerrada descrita mediante la siguiente ecuación:

$$\overrightarrow{OP} + \overrightarrow{PC_i} = \overrightarrow{OA_i} + \overrightarrow{A_iB_i} + \overrightarrow{B_iC_i} \quad (2.44)$$



**Figura 2.5.** (a) Proyección del robot delta en el plano  $x_i z_i$ , (b) Proyección de  $L_2$  en el plano  $x_i z_i$ .

Reescribiendo la ecuación (2.44).

$$T * \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} + \begin{bmatrix} r \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} R \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} L_1 c \theta_{1i} \\ 0 \\ L_1 s \theta_{1i} \end{bmatrix} + \begin{bmatrix} L_2 s \theta_{3i} c(180 - (\theta_{1i} + \theta_{2i})) \\ L_2 c \theta_{3i} \\ L_2 s \theta_{3i} s(180 - (\theta_{1i} + \theta_{2i})) \end{bmatrix} \quad (2.45)$$

La derivada de la ecuación (2.45) con respecto al tiempo conduce hacia la matriz Jacobiana, se toma en cuenta que el vector  $\overrightarrow{OA_l}$  es un vector que caracteriza a la plataforma fija y se obtiene la siguiente ecuación:

$$\overrightarrow{V_p} = \dot{\overrightarrow{A_l B_l}} + \dot{\overrightarrow{B_l C_l}} \quad (2.46)$$

Esta ecuación indica que cada punto de la plataforma móvil se mueve a la misma velocidad, por lo tanto las velocidades lineales del lado derecho se pueden convertir fácilmente en velocidades angulares mediante el uso de identidades.

$$\overrightarrow{V_p} = \overrightarrow{\omega_{1l}} \times \overrightarrow{A_l B_l} + \overrightarrow{\omega_{2l}} \times \overrightarrow{B_l C_l} \quad (2.47)$$

Donde  $\omega_{ji}$  es la velocidad angular del j-ésimo eslabón de la i-ésima cadena cinemática, sin embargo la velocidad angular  $\overrightarrow{\omega_{2l}}$  muestra dependencia de las variables  $\dot{\theta_{2l}}$  y  $\dot{\theta_{3l}}$ , que pertenecen a las articulaciones pasivas y no son de interés por lo tanto se puede multiplicar escalarmente el vector  $\overrightarrow{B_l C_l}$  por toda la ecuación (2.47) de tal manera que el segundo término de la derecha se elimine por existir triple producto con dos vectores idénticos (Propiedad de ortogonalidad del producto cruz).

$$\begin{aligned} \overrightarrow{B_l C_l} * \overrightarrow{V_p} &= \overrightarrow{B_l C_l} * (\overrightarrow{\omega_{1l}} \times \overrightarrow{A_l B_l}) + \overrightarrow{B_l C_l} * (\overrightarrow{\omega_{2l}} \times \overrightarrow{B_l C_l}) \\ \overrightarrow{B_l C_l} * \overrightarrow{V_p} &= \overrightarrow{B_l C_l} * (\overrightarrow{\omega_{1l}} \times \overrightarrow{A_l B_l}) \end{aligned} \quad (2.48)$$

El lado izquierdo de la ecuación al ser resuelta queda escrita de la siguiente manera:

$$\begin{aligned} \overrightarrow{B_l C_l} * \overrightarrow{V_p} &= \\ \begin{bmatrix} L_2 S \theta_{3i} C(\theta_{1i} + \theta_{2i}) \\ L_2 C \theta_{3i} \\ L_2 S \theta_{3i} S(\theta_{1i} + \theta_{2i}) \end{bmatrix} * \begin{bmatrix} V_{px} C \phi_i + V_{py} S \phi_i \\ V_{py} C \phi_i - V_{px} S \phi_i \\ V_{pz} \end{bmatrix} \end{aligned}$$

$$\begin{aligned}\overrightarrow{B_i C_i} * \overrightarrow{V_p} &= L_2 S \theta_{3i} C(\theta_{1i} + \theta_{2i}) * V_{pX} C\phi_i + L_2 S \theta_{3i} C(\theta_{1i} + \theta_{2i}) * V_{pY} S\phi_i + L_2 C \theta_{3i} * V_{pY} C\phi_i \\ &\quad - L_2 C \theta_{3i} * V_{pX} S\phi_i + L_2 S \theta_{3i} S(\theta_{1i} + \theta_{2i}) * V_{pZ}\end{aligned}$$

$$\begin{aligned}\overrightarrow{B_i C_i} * \overrightarrow{V_p} &= [L_2 S \theta_{3i} C(\theta_{1i} + \theta_{2i}) C\phi_i - L_2 C \theta_{3i} S\phi_i] V_{pX} + [L_2 S \theta_{3i} C(\theta_{1i} + \theta_{2i}) S\phi_i \\ &\quad + L_2 C \theta_{3i} C\phi_i] V_{pY} + L_2 S \theta_{3i} S(\theta_{1i} + \theta_{2i}) * V_{pZ} \\ \overrightarrow{B_i C_i} * \overrightarrow{V_p} &= j_{xi1} V_{pX} + j_{xi2} V_{pY} + j_{xi3} V_{pZ}\end{aligned}\tag{2.49}$$

Donde:

$$j_{xi1} = S \theta_{3i} C(\theta_{1i} + \theta_{2i}) C\phi_i - C \theta_{3i} S\phi_i$$

$$j_{xi2} = S \theta_{3i} C(\theta_{1i} + \theta_{2i}) S\phi_i + C \theta_{3i} C\phi_i$$

$$j_{xi3} = S \theta_{3i} S(\theta_{1i} + \theta_{2i})$$

En el lado derecho de la ecuación (2.48) el movimiento del eslabón  $L_1$  ó  $\overrightarrow{A_i B_i}$  sucede solo en el plano  $X_i Z_i$ , por lo tanto la velocidad angular que tiene en ese plano estará en el eje  $Y_i$ .

$$\overrightarrow{\omega_{1i}} = \begin{bmatrix} 0 \\ -\dot{\theta}_{1i} \\ 0 \end{bmatrix}$$

$$\overrightarrow{\omega_{1i}} x \overrightarrow{A_i B_i} = \begin{vmatrix} i & j & k \\ 0 & -\dot{\theta}_{1i} & 0 \\ C \theta_{1i} & 0 & S \theta_{1i} \end{vmatrix} = -(\dot{\theta}_{1i} S \theta_{1i}) i - (0) j + (\dot{\theta}_{1i} C \theta_{1i}) k$$

$$\overrightarrow{B_i C_i} * (\overrightarrow{\omega_{1i}} x \overrightarrow{A_i B_i}) = -S \theta_{3i} C(\theta_{1i} + \theta_{2i}) \dot{\theta}_{1i} S \theta_{1i} + S \theta_{3i} S(\theta_{1i} + \theta_{2i}) \dot{\theta}_{1i} C \theta_{1i}$$

$$\overrightarrow{B_i C_i} * (\overrightarrow{\omega_{1i}} x \overrightarrow{A_i B_i}) = \dot{\theta}_{1i} [-S \theta_{3i} C(\theta_{1i} + \theta_{2i}) S \theta_{1i} + S \theta_{3i} S(\theta_{1i} + \theta_{2i}) C \theta_{1i}]$$

$$\overrightarrow{B_i C_i} * (\overrightarrow{\omega_{1i}} x \overrightarrow{A_i B_i}) = L_1 S \theta_{3i} S \theta_{2i} \dot{\theta}_{1i}$$

$$\overrightarrow{B_l C_l} * (\overrightarrow{\omega_{1l}} x \overrightarrow{A_l B_l}) = j_{\theta i} \dot{\theta}_{1l} \quad (2.50)$$

Donde:

$$j_{\theta i} = L_1 S \theta_{3i} S \theta_{2i}$$

Ahora se reemplazan las ecuaciones (2.49) y (2.50) en la ecuación (2.48) y se obtiene la representación de la matriz Jacobiana con sus asociaciones inversa y directa.

$$\overrightarrow{B_l C_l} * \overrightarrow{V_p} = \overrightarrow{B_l C_l} * (\overrightarrow{\omega_{1l}} x \overrightarrow{A_l B_l})$$

$$j_{xi1} V_{pX} + j_{xi2} V_{pY} + j_{xi3} V_{pZ} = j_{\theta 1} \dot{\theta}_{1l}$$

$$J_x \dot{x} = J_\theta \dot{\theta}$$

$$\begin{bmatrix} j_{x11} & j_{x12} & j_{x13} \\ j_{x21} & j_{x22} & j_{x23} \\ j_{x31} & j_{x32} & j_{x33} \end{bmatrix} \begin{bmatrix} V_{pX} \\ V_{pY} \\ V_{pZ} \end{bmatrix} = \begin{bmatrix} j_{\theta 1} & 0 & 0 \\ 0 & j_{\theta 2} & 0 \\ 0 & 0 & j_{\theta 3} \end{bmatrix} \begin{bmatrix} \dot{\theta}_{11} \\ \dot{\theta}_{12} \\ \dot{\theta}_{13} \end{bmatrix} \quad (2.51)$$

$$J = (J_\theta)^{-1} J_x = \left( \begin{bmatrix} j_{\theta 1} & 0 & 0 \\ 0 & j_{\theta 2} & 0 \\ 0 & 0 & j_{\theta 3} \end{bmatrix} \right)^{-1} \begin{bmatrix} j_{x11} & j_{x12} & j_{x13} \\ j_{x21} & j_{x22} & j_{x23} \\ j_{x31} & j_{x32} & j_{x33} \end{bmatrix} \quad (2.52)$$

Como se menciona en 2.1.4, para obtener las singularidades se hace uso de una matriz denominada Jacobiana, y obtenida esta matriz en la ecuación (2.52) se analiza las condiciones para que dicha matriz sea singular, en este caso se analizan las dos partes que componen esta matriz y de esta manera poder clasificar las singularidades.

#### 2.1.4.2. Clasificación de singularidades [17] [18]

Una matriz de orden n se denomina singular si no posee una matriz inversa.

Dado la matriz  $A_{n \times n}$ , su inversa queda definida por:

$$A^{-1} = \frac{1}{|A|} * Adj(A) \quad (2.53)$$

Por lo tanto una matriz es singular si su determinante es cero.

De la ecuación (2.52) se puede decir que  $J$  es singular si  $J_\theta$  o  $J_x$  son singulares, de esta manera dependiendo de cuál matriz es singular, un mecanismo de lazo cerrado puede tener una singularidad de cinemática directa, una singularidad de cinemática inversa o ambas.

#### 2.1.4.2.1. Singularidades de la cinemática inversa

Este tipo de singularidades están asociadas con la Jacobiana inversa  $J_\theta$  y tiene lugar cuando el  $\det(J_\theta) = 0$ . Esta configuración es en la cual la cadena alcanza un límite de su espacio de trabajo, también sucede cuando alcanza un límite interno de la cadena, que limita diferentes subregiones del área de trabajo.

Esto implica que pequeños movimientos del efecto final en ciertas direcciones no pueden ser realizados.

$$\det \begin{pmatrix} j_{\theta_1} & 0 & 0 \\ 0 & j_{\theta_2} & 0 \\ 0 & 0 & j_{\theta_3} \end{pmatrix} = j_{\theta_1} * (j_{\theta_2} * j_{\theta_3}) \quad (2.54)$$

Reemplazando  $j_{\theta_i} = L_1 S\theta_{3i} S\theta_{2i}$  en (2.54) se tiene:

$$\det(J_\theta) = L_1 S\theta_{31} S\theta_{21} * L_1 S\theta_{32} S\theta_{22} * L_1 S\theta_{33} S\theta_{23} \quad (2.55)$$

Estableciendo la condición de singularidad se tiene que.

$$\det(J_\theta) = 0$$

$$(S\theta_{31}S\theta_{21}) * (S\theta_{32}S\theta_{22}) * (S\theta_{33}S\theta_{23}) = 0 \quad (2.56)$$

Para que cumpla esta condición se tiene que:

$$\sin\theta_{3i} = 0 \quad \text{ó} \quad \sin\theta_{2i} = 0$$

Entonces se obtienen las siguientes condiciones:

$$\theta_{2i} = 0^\circ \quad \text{o} \quad 180^\circ$$

$$\theta_{3i} = 0^\circ \quad \text{o} \quad 180^\circ$$

#### *2.1.4.2.2 Singularidades de la cinemática directa*

Este tipo de singularidades están asociadas con la Jacobiana directa  $J_x$  y tiene lugar cuando  $\det(J_x) = 0$ . Esto corresponde a configuraciones en las que el efecto final está localmente móvil aún cuando sus articulaciones de accionamiento están bloqueadas; es decir que la velocidad de la plataforma es diferente de cero aun cuando la velocidad de los actuadores es cero.

#### **2.1.4.3. Destreza del robot delta [4] [19]**

La destreza de un robot paralelo puede definirse como la precisión del posicionamiento de su efecto final dentro de su espacio de trabajo y para definirla se debe hacer uso de la matriz Jacobiana  $J = (J_\theta)^{-1}J_x$  representada mediante el número de condición  $k(J)$  como se indica a continuación:

$$k(J) = \|J\| \|J^{-1}\| \quad (2.57)$$

Los robots manipuladores en donde su número de condición  $k(J)$  es 1, se denominan

isotrópicos, en el cual el sistema es capaz de desarrollar la misma cantidad de fuerza y velocidad en todas las direcciones del efecto final.

Para valores altos de  $k(J)$  existen instrucciones del efecto final donde el manipulador desarrollará más fuerza y velocidad que en otras direcciones, en cambio si el valor de  $k(J)$  es infinito el manipulador entra en una configuración singular en donde el efecto final no desarrollara ningún movimiento.

Entonces la destreza local  $CI$  para un manipulador queda definida como el inverso del número de condición:

$$CI = \frac{1}{k(J)}; \quad 0 \leq CI \leq 1 \quad (2.58)$$

Este índice depende generalmente de la posición en la que se encuentre el efecto final, sin embargo se establece un índice que permite calcular la destreza de un robot en todo su espacio de trabajo, este se denomina índice de destreza global  $GCI$ .

$$GCI = \frac{\int_W CI dW}{\int_W dW} \quad (2.59)$$

$$GCI = \frac{\sum_{i=1}^W CI}{W} \quad (2.60)$$

Donde  $\sum_{i=1}^W CI$  es la sumatoria de las destrezas locales del robot y  $W$  es el espacio de trabajo del robot.

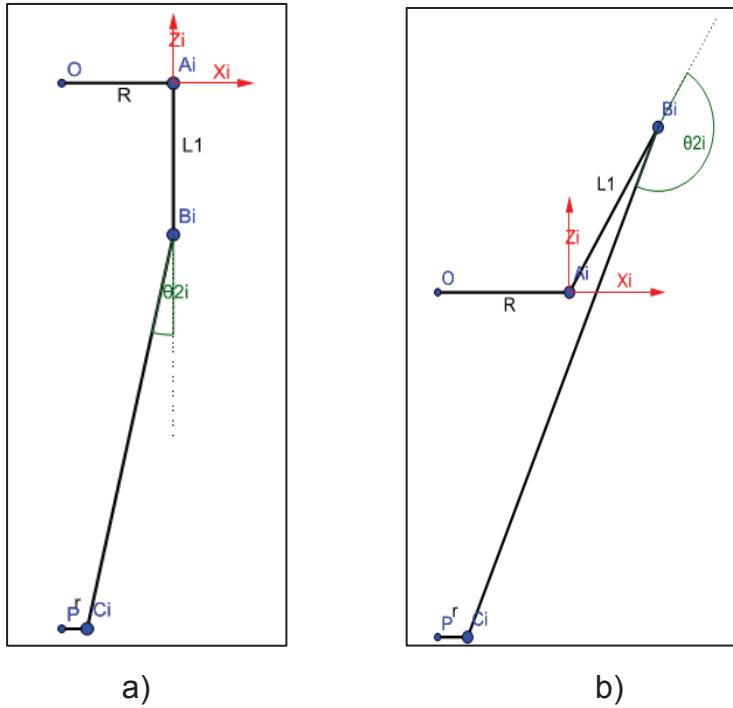
### 2.1.5. ANÁLISIS DEL ESPACIO DE TRABAJO [16]

Para determinar el área de trabajo es necesario detectar algunas colisiones que pueden darse debido a los componentes mecánicos del robot, para lo cual es

necesario conocer los ángulos  $\theta_{2i}, \theta_{3i}$  y  $\beta_i$  para de esta manera poder restringir su valor.

La restricción del ángulo  $\theta_{3i}$  a pesar de estar condicionado con los valores de  $0^\circ$  y  $180^\circ$  para que no entre en configuración singular, también nos indica mediante la Figura 2.5 (b) que pertenece a una juntura pasiva donde el ángulo de operación  $\epsilon[-45,45]$  aproximadamente.

El ángulo  $\theta_{2i}$  se encuentra en el plano  $X_iZ_i$  formado por la prolongación del eslabón  $L_1$  y la proyección del eslabón  $L_2$  y tiene como restricción no adoptar los valores de  $0^\circ$  y  $180^\circ$  para no entrar en configuración singular. En la Figura 2.6 se puede observar la representación del ángulo  $\theta_{2i}$  y además se puede estimar el conjunto de valores que podría tomar este ángulo. Por lo tanto  $\theta_{2i} \in [10^\circ, 180^\circ]$ .



**Figura 2.6.** a) Mínimo valor del ángulo  $\theta_{2i}$ , b) Máximo valor del ángulo  $\theta_{2i}$ .

La representación del ángulo  $\beta_i$  se encuentra en la Figura 2.4 y está formado por el eslabón  $L_1$  y  $L_2$ , igual que en el caso anterior se puede estimar los límites de operación

para un correcto funcionamiento. Por lo tanto  $\beta_i \in [10^\circ, 180^\circ]$ .

De la Figura 2.5 se obtiene la siguiente expresión:

$$\overrightarrow{A_l C_l} = \overrightarrow{A_l B_l} + \overrightarrow{B_l C_l} \quad (2.61)$$

Donde de acuerdo a la ecuación (2.6) y la ecuación (2.45) esta igualdad puede quedar expresado de la siguiente manera:

$$\begin{bmatrix} P_X C \phi_i + P_Y S \phi_i + r - R \\ P_Y C \phi_i - P_X S \phi_i \\ P_Z \end{bmatrix} = \begin{bmatrix} L_1 c \theta_{1i} \\ 0 \\ L_1 s \theta_{1i} \end{bmatrix} + \begin{bmatrix} L_2 s \theta_{3i} c (180^\circ - \theta_{1i} - \theta_{2i}) \\ L_2 c \theta_{3i} \\ L_2 s \theta_{3i} s (180^\circ - \theta_{1i} - \theta_{2i}) \end{bmatrix}$$

De la expresión anterior se obtiene la siguiente igualdad para el componente en  $X$ .

$$\begin{aligned} P_Y C \phi_i - P_X S \phi_i &= L_2 c \theta_{3i} \\ \theta_{3i} &= \cos^{-1} \left( \frac{P_Y C \phi_i - P_X S \phi_i}{L_2} \right) \end{aligned} \quad (2.62)$$

Para el componente en  $Z$  se tiene:

$$\begin{aligned} P_Z &= L_1 s \theta_{1i} + L_2 s \theta_{3i} s (180^\circ - \theta_{1i} - \theta_{2i}) \\ \theta_{2i} &= 180 - \sin^{-1} \left( \frac{P_Z - L_1 s \theta_{1i}}{L_2 s \theta_{3i}} \right) - \theta_{1i} \end{aligned} \quad (2.63)$$

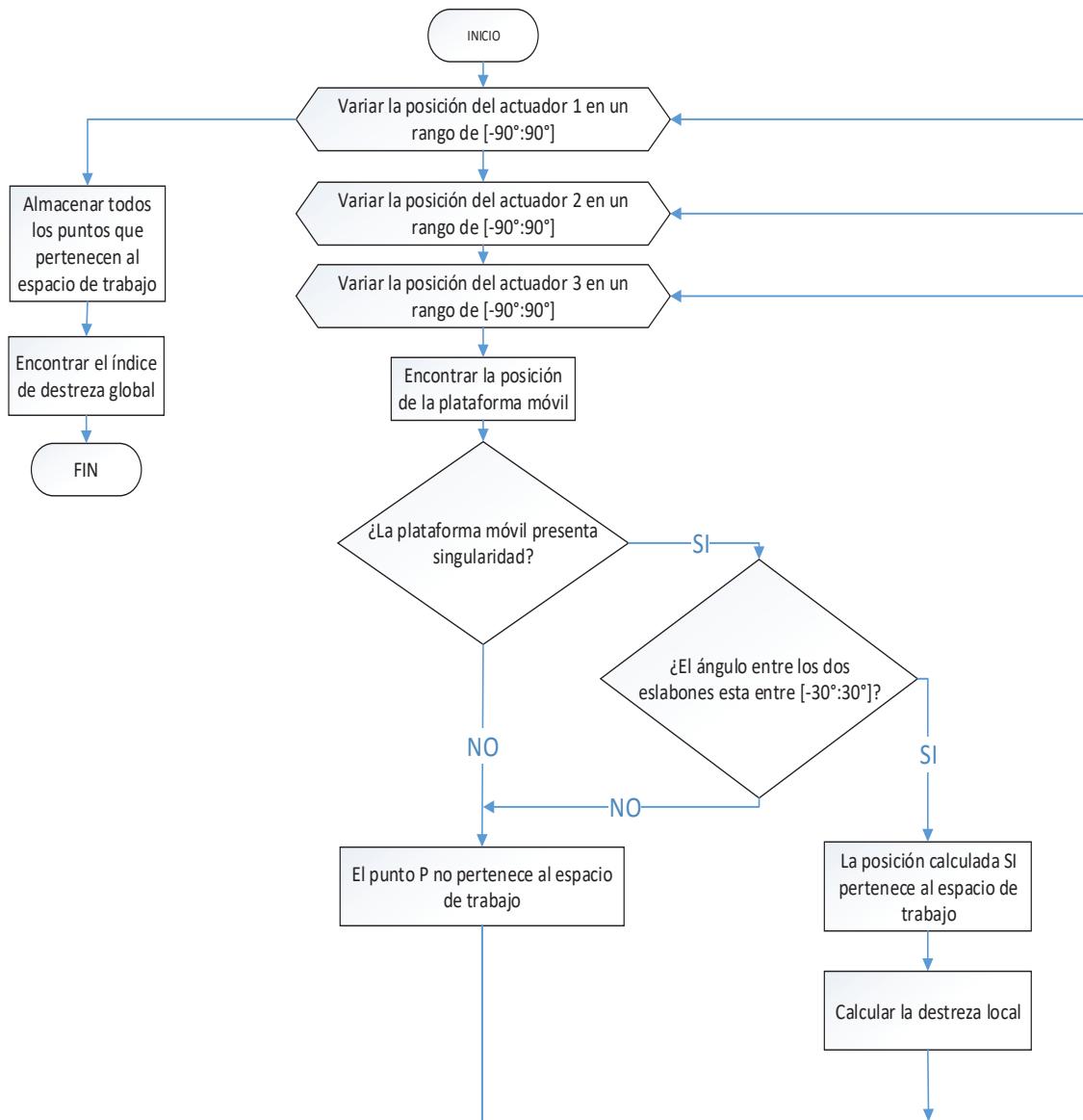
Para encontrar la expresión del ángulo  $\beta_i$ , se despeja la variable de la ecuación (2.10).

$$\beta_i = \cos^{-1} \left( \frac{L_1^2 + L_2^2 - A_i C_i^2}{2 L_1 L_2} \right) \quad (2.64)$$

Establecidas las expresiones para las restricciones se selecciona el método de discretización establecido en el punto 1.4.3.2 para determinar el espacio de trabajo.

En el método de discretización se seleccionan puntos aleatorios dentro de un volumen para determinar si pertenece o no al espacio de trabajo, pero para determinar la cantidad exacta de puntos dentro el espacio de trabajo se hace variar los ángulos de los actuadores dentro de un límite y con la implementación de la cinemática directa poder determinar la posición del efecto final, además se incluyen las restricciones y de esta manera se obtiene los puntos pertenecientes al espacio de trabajo.

El algoritmo utilizado se visualiza en el diagrama de flujo de la Figura 2.7.



**Figura 2.7.** Diagrama de flujo cálculo del espacio de trabajo y destreza global.

## 2.2. DISEÑO MECÁNICO

### 2.2.1. PARÁMETROS DE DISEÑO.

Para la construcción del prototipo se debe decidir las dimensiones y el tipo de material que se va a utilizar, estos dos parámetros influyen de manera directa en la selección del actuador, el parámetro de dimensiones es importante para definir el área de trabajo del manipulador.

#### 2.2.1.1. Dimensiones del prototipo

##### 2.2.1.1.1. Algoritmo para obtener las dimensiones.

Para definir las dimensiones del prototipo se toma como referencia las relaciones existentes en el robot IRB-360 de ABB mostrado en la Tabla 2.1.

**Tabla 2.1.** Parámetros robot IRB-360-ABB

Parámetro	Valor
Radio base Fija (R)	200 mm
Radio base móvil (r)	50 mm
Eslabon1 (L1)	350 mm
Eslabon2 (L2)	800 mm
Carga máxima a mover	2 Kg

En publicaciones respecto a robots paralelos [20][21][22][23] y con base a la Tabla 2.1 se pueden establecer las siguientes relaciones de construcción de un robot paralelo tipo delta.

$$R > r \quad (2.65)$$

$$L_1 < L_2 \quad (2.66)$$

$$L_2 \approx 2L_1 \quad (2.67)$$

La expresión (2.67) corresponde a un trabajo de optimización dimensional para un robot delta basado en el menor consumo de energía. [20]

En este proyecto se plantea un espacio de trabajo específico que este contenido dentro del espacio de trabajo del robot delta, y para encontrar las medidas del prototipo que cumplan este objetivo se hace uso de la técnica de optimización combinatoria bajo restricción, que encuentra los valores mínimos factibles de un conjunto de soluciones globales.[21]

Los pasos a seguir son los siguientes:

- a) Establecer una figura geométrica que contenga el espacio de trabajo deseado con  $k$  vértices  $P_k = [Px_k \ Py_k \ Pz_k]$ .
- b) Generar todas las posibles combinaciones de los parámetros dimensionales  $L_1, L_2, R, r$  y  $h$  haciendo un barrido en un rango definido y además tomando en cuenta las condiciones de (2.65), (2.66) y (2.67).
- c) Aplicar la cinemática inversa (2.20) y calcular los ángulos  $\theta_{11} \ \theta_{12} \ \theta_{13}$  y verificar si están dentro de los límites de correcto funcionamiento.
- d) Si los ángulos calculados anteriormente son correctos entonces el grupo de valores utilizados pasan a formar parte de una matriz  $S_{m \times n}$ , la cual acumula todas las soluciones que satisfacen el problema.

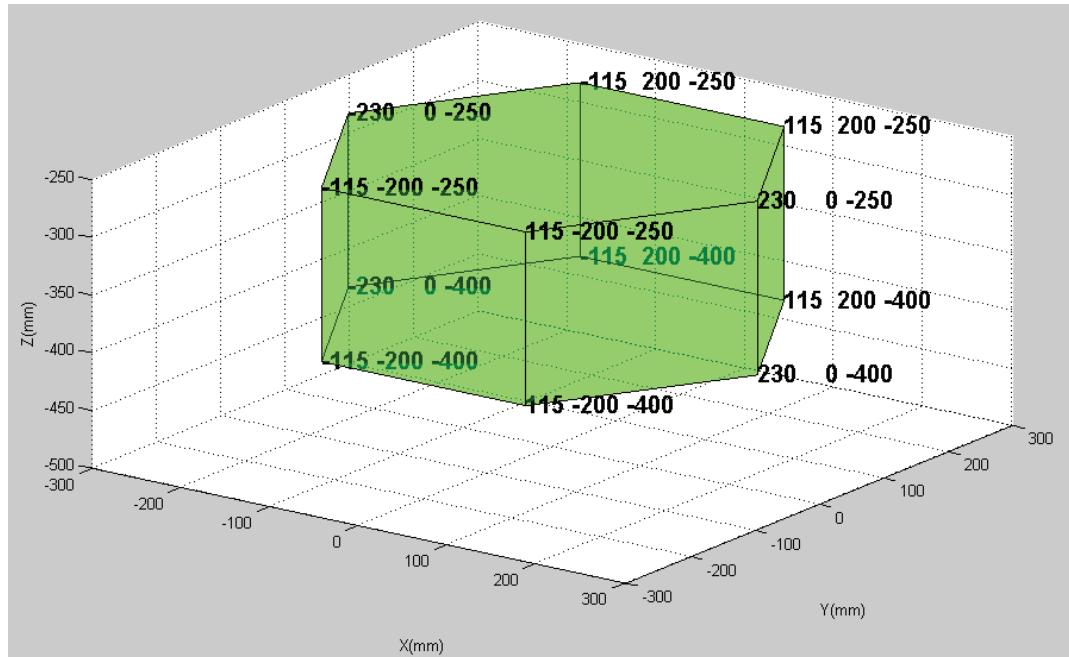
$$S = \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{bmatrix} \quad (2.68)$$

- e) A partir de la matriz  $s$  se procede a obtener los valores mínimos mediante la optimización combinatoria por técnica de búsqueda local.

De la matriz  $s$  se escogen las combinaciones con los valores mínimos para la primera variable, de esta manera se genera la matriz  $M_1$  a la cual se escogen las combinaciones con los valores mínimos de la segunda variable obteniendo la matriz  $M_2$ , de igual manera se produce la matriz  $M_3$  y así sucesivamente hasta obtener  $M_{(n-1)}$ , donde  $n$  es el numero de elementos del vector  $X$ .

#### 2.2.1.1.2 Aplicación del algoritmo.

En la Figura 2.8 se puede apreciar el volumen de trabajo determinado para el prototipo, estas medidas se escogen considerando que el robot pueda tener buena movilidad, los  $k$  puntos que lo conforman se muestran en la Tabla 2.2.



**Figura 2.8.** Espacio de trabajo deseado.

**Tabla 2.2.** Coordenadas del espacio de trabajo deseado.

$P_k$	$Px_k \text{ mm}$	$Py_k \text{ mm}$	$Pz_k \text{ mm}$
$P_1$	115	200	-h
$P_2$	-115	200	-h
$P_3$	-230	0	-h
$P_4$	-115	-200	-h
$P_5$	115	-200	-h
$P_6$	230	0	-h
$P_7$	115	200	-(h+150)
$P_8$	-115	200	-(h+150)
$P_9$	-230	0	-(h+150)
$P_{10}$	-115	-200	-(h+150)
$P_{11}$	115	-200	-(h+150)
$P_{12}$	230	0	-(h+150)

La variable  $h$  es la distancia que existe entre la base fija y el espacio de trabajo deseado, para los cálculos se asume que el valor de  $h = 200 \text{ mm}$ .

Posteriormente se plantea el problema que será resuelto por el software MATLAB 2014A.

Encontrar el vector:

$$X = [L_1 \ L_2 \ R \ r \ h] \quad (2.69)$$

Sujeto a:

$$R > r$$

$$L_1 < L_2$$

$$P_k \in \text{Espacio de trabajo deseado} \quad (2.70)$$

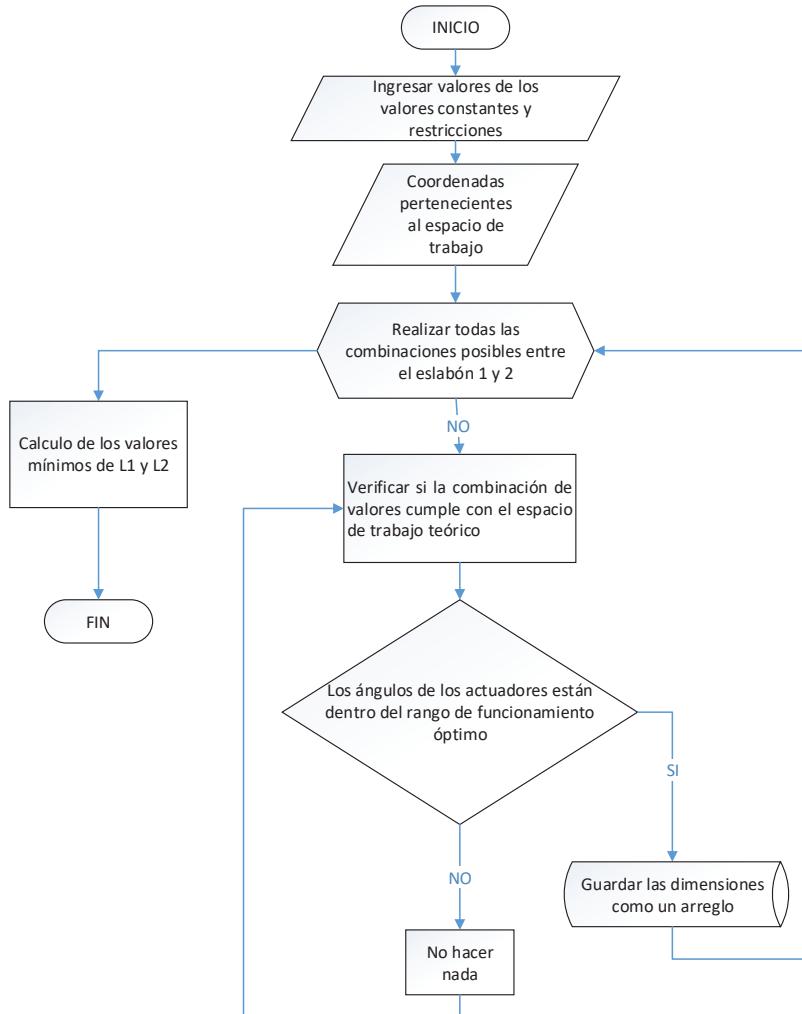
Donde:

$$150 \text{ mm} < L_1(\Delta L_1 = 1) < 250 \text{ mm} \quad (2.71)$$

$$300 \text{ mm} < L_2(\Delta L_2 = 1) < 500 \text{ mm} \quad (2.72)$$

Se asumen el valor del radio de la base fija  $R=110[\text{mm}]$  con el objetivo de tener un espacio suficiente entre los actuadores, el radio de la parte móvil se lo asume de tal manera que pueda ser colocado un actuador por lo cual  $r=30\text{mm}$ .

Para aplicar la técnica de optimización combinatoria bajo restricción se hace uso del software MATLAB 2014A y se procede como muestra el diagrama de flujo de la Figura 2.9.



**Figura 2.9.** Diagrama de flujo de la optimización combinatoria bajo restricciones

Luego de haber aplicado el algoritmo de la figura anterior los resultados son:

$$X' = [167 \quad 400 \quad 110 \quad 30 \quad 200] \quad (2.73)$$

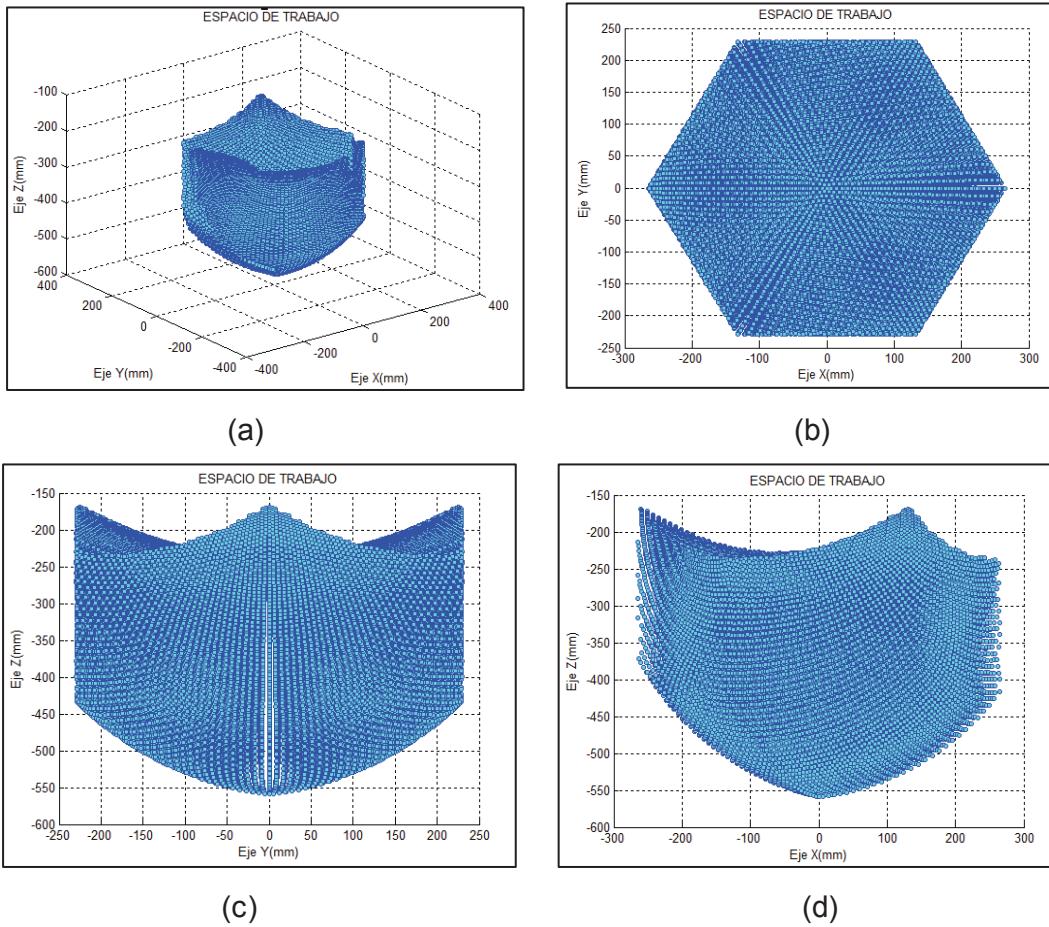
$$L_1 = 167 \text{ mm}, \quad L_2 = 400 \text{ mm}, \quad R = 110 \text{ mm}, \quad r = 30 \text{ mm}$$

#### 2.2.1.1.3. Verificación del espacio de trabajo

Con los datos calculados en el punto anterior, y el algoritmo presentado en la Figura 2.7, se obtiene el espacio de trabajo que se muestra en la Figura 2.10.

Datos  $L_1 = 167 \text{ mm}$ ,  $L_2 = 400 \text{ mm}$ ,  $R = 110 \text{ mm}$ ,  $r = 30 \text{ mm}$ ,

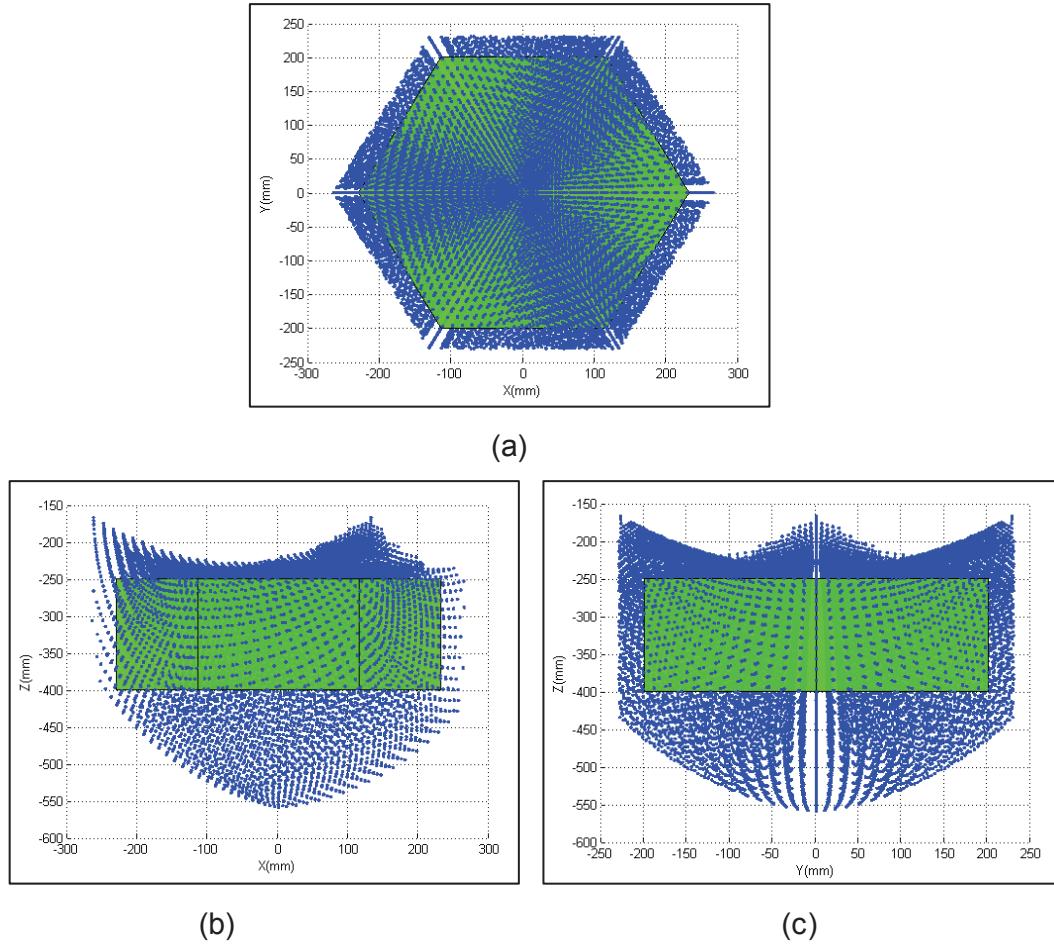
$$-90 < \theta_{1i} (\Delta\theta_{1i} = 3^\circ) < 90$$



**Figura 2.10.** (a)Espacio de trabajo robot delta, (b) Vista Superior, (c) Vista Plano YZ, (d) Vista Plano XZ.

Para verificar si el espacio de trabajo obtenido con estas medidas satisface el espacio prescrito se muestran las imágenes montadas en la Figura 2.11.

$$\text{Datos } L_1 = 167 \text{ mm}, L_2 = 400 \text{ mm}, R = 110 \text{ mm}, r = 30 \text{ mm}, \\ -90 < \theta_{1i} (\Delta\theta_{1i} = 5^\circ) < 90$$



**Figura 2.11.** (a)Plano XY espacio deseado (verde), espacio de trabajo robot delta (azul), (b) Vista Plano XZ, (c) Vista Plano YZ.

La Figura 2.11 muestra que el espacio alcanzado por el robot delta color azul es mayor que el espacio requerido inicialmente color verde, por lo tanto el siguiente punto a analizar es la destreza global del manipulador que se menciona en la sección 2.1.4.2

y se obtiene en base al diagrama de flujo de la Figura 2.7, mediante el software MATLAB. Además tomando en cuenta la expresión (2.67) la cual indica que  $L_2 \approx 2L_1$ , y con el objetivo de mejorar la destreza se manipulan las variables de  $L_1$  y  $L_2$  y los resultados se muestran en la Tabla 2.3.

**Tabla 2.3.** Destreza global para diferentes valores de los eslabones  $L_1$  y  $L_2$

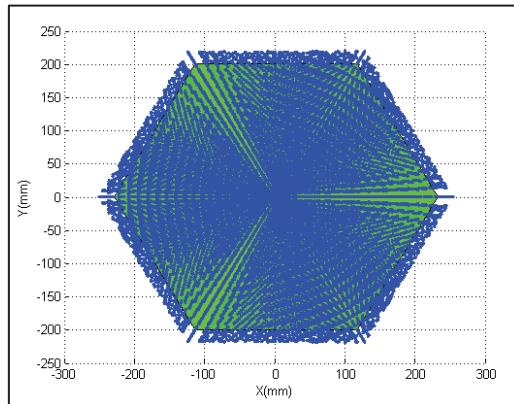
R mm	r mm	$L_1$ mm	$L_2$ mm	GCI
110	30	167	400	0.3685
110	30	200	400	0.4342
110	30	167	334	0.4529
110	30	180	380	0.4179

De los valores obtenidos en la Tabla 2.3 se puede apreciar que la destreza global en el tercer caso es más alta, pero con las dimensiones de los eslabones el prototipo no logra cubrir el espacio de trabajo establecido.

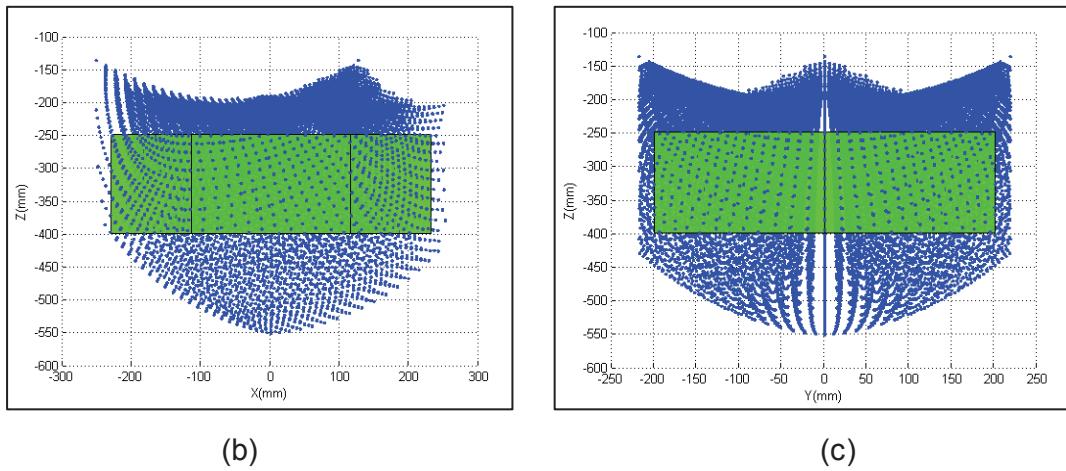
Las opciones restantes cumplen con el espacio de trabajo requerido y su destreza global es más alta que en el primer caso y para la construcción se tomó el caso 4 donde las dimensiones de los eslabones tienen los siguientes valores:

Datos  $L_1 = 180$  mm,  $L_2 = 380$  mm,  $R = 110$  mm,  $r = 30$  mm,

$$-90 < \theta_{1i} (\Delta\theta_{1i} = 5^\circ) < 90$$

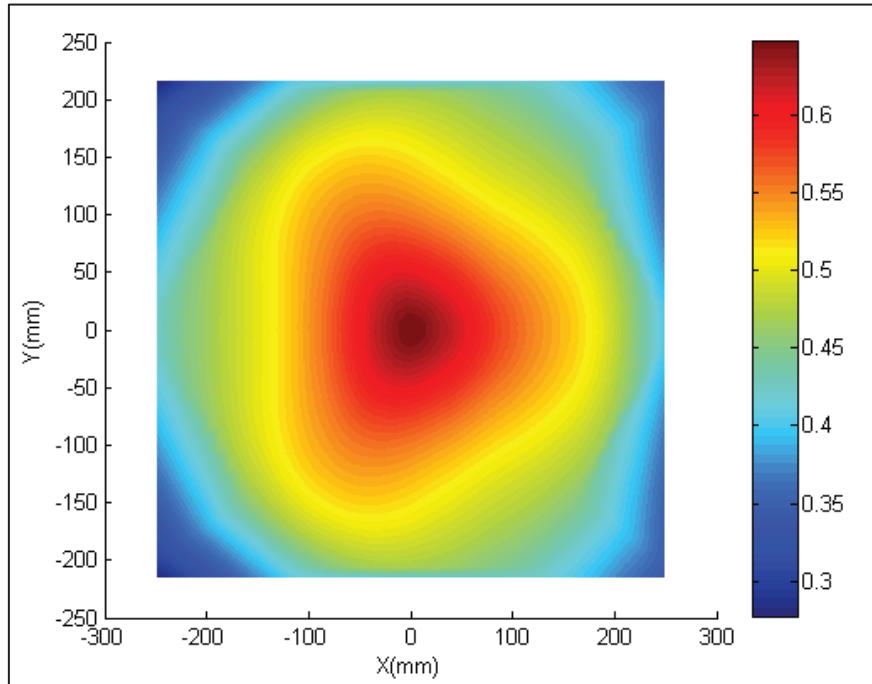


(a)

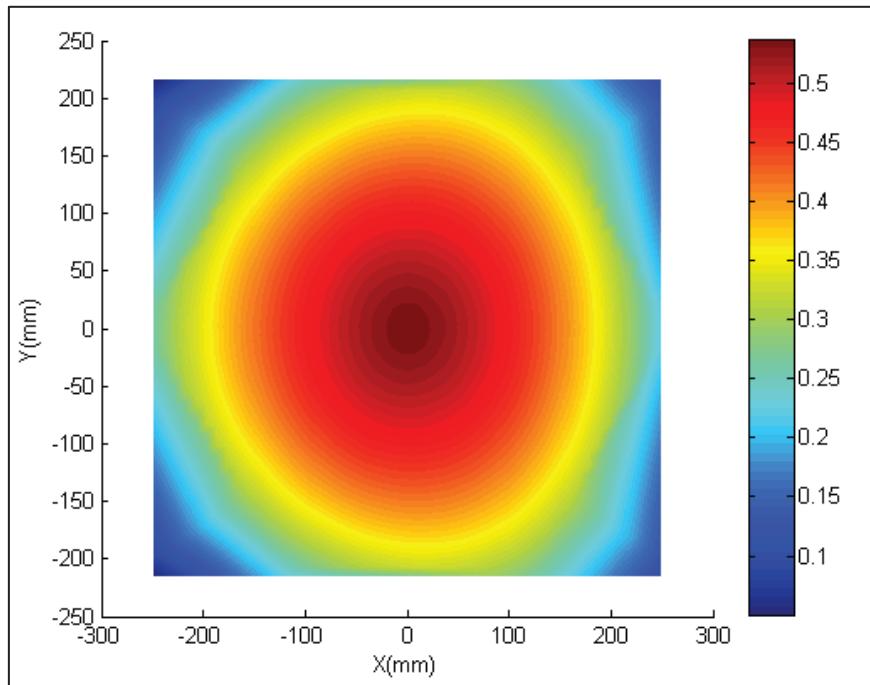


**Figura 2.12.** Espacio de trabajo recalculado. (a)Plano XY espacio deseado (verde), espacio de trabajo robot delta (azul), (b) Vista Plano XZ, (c) Vista Plano YZ.

Para finalizar con el análisis se procede a calcular la destreza local en el plano  $z=-250$  mm] y  $z=-400$  mm, como se muestran en las Figuras 2.13 y 2.14 respectivamente.



**Figura 2.13.** Destreza local plano  $z=-250$  mm.



**Figura 2.14.** Destreza local plano  $z=-400$  mm.

En conclusión se tiene las siguientes dimensiones para la construcción del prototipo y para el alcance del espacio de trabajo.

- Dimensiones de construcción:

$$L_1 = 180\text{mm}, L_2 = 380\text{mm}, R = 110\text{mm}, r = 30\text{mm}.$$

- Alcance eje x:  $-251\text{ mm} < X < 251\text{ mm}$
- Alcance eje y:  $-217\text{ mm} < Y < 217\text{ mm}$
- Alcance eje z:  $-200\text{ mm} < Z < 551\text{ mm}$

#### 2.2.1.3. Tipo de material

Para construir manipuladores existen materiales como el plástico, el metal o una combinación de ambos, sin embargo los dos tienen ventajas y desventajas.

Por una parte los plásticos tienen baja densidad, son fáciles de manejar y moldear pero poca resistencia al esfuerzo físico, en cambio el metal posee alta densidad, son más difíciles de moldear, pero resisten mejor a los esfuerzos.

Se puede deducir que las partes del robot pueden ser de metal si va a realizar esfuerzo constante o va a levantar algún tipo de carga, sin embargo se debe destacar que el metal es pesado y para el movimiento de dicha parte el actuador deberá tener un alto torque.

$$\tau = (m * r^2) * \alpha \quad (2.73)$$

Donde:

$m$  = Masa del cuerpo.

$r$  = Distancia desde del sistema de referencia hasta el punto de giro.

$\alpha$  = Aceleración angular.

De la ecuación (2.73) la cual da información de la relación de masa y distancia con el torque se puede sacar las siguientes conclusiones:

- Si la masa de la pieza del robot es muy pesada, entonces se necesitará un actuador con el torque muy alto.
- Si la pieza es demasiada larga, entonces de manera similar se necesitará un actuador con torque alto para poder moverla.

Por lo tanto se necesita un material que sea fuerte pero de baja densidad para las partes móviles del robot y un material pesado para las bases que permitirán que el manipulador se mantenga fijo.

Las características de los materiales del prototipo se muestran en la Tabla 2.4.

**Tabla 2.4.** Propiedades de los materiales del prototipo

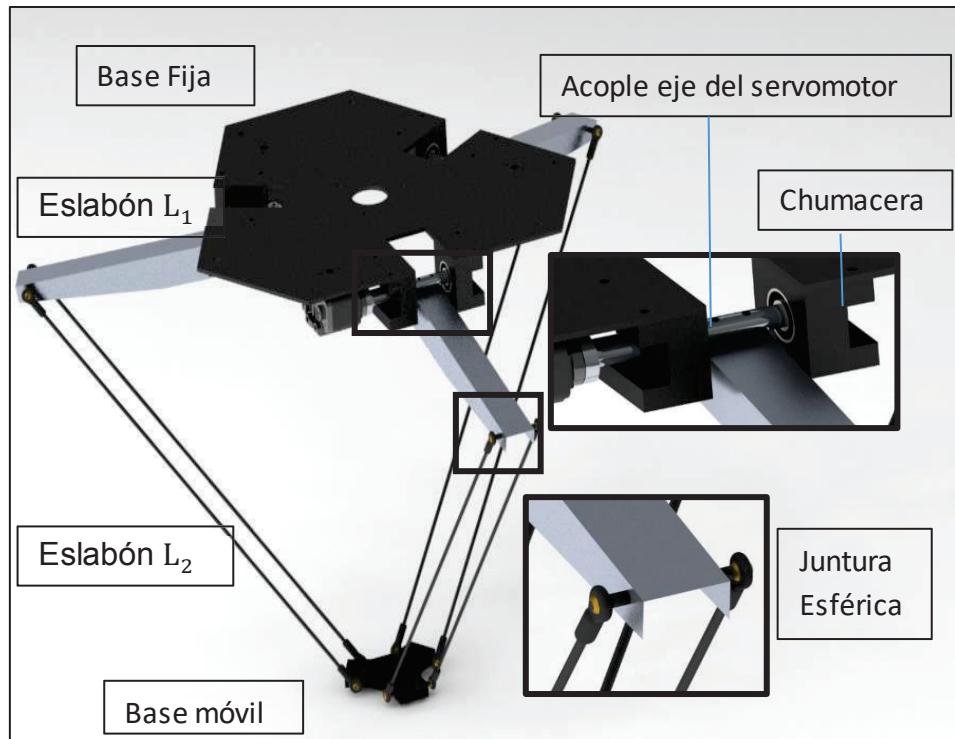
Material	Densidad [ $\frac{g}{cm^3}$ ]	Límite de fluencia [ $\frac{N}{m^2}$ ]
Aluminio	2.700	27574200
Hierro/acero	7.100	551485000
Plástico ABS	1.02	-----

- **Aluminio:** metal ligero no ferromagnético, es un material blando y maleable que abunda en la naturaleza siendo muy usado en la industria solo superado por el hierro/acero. El aluminio se utiliza para la construcción del acople de eje y el Eslabón  $L_1$ .
- **Hierro/acero:** es una aleación formada por hierro y carbono, este último en un porcentaje bajo de 1- 2%. Se utiliza en la construcción del soporte y el Eslabón  $L_2$ .
- **Plástico ABS:** este nombre es dado a una familia de termoplásticos, llamado también plástico de ingeniería, es el más utilizado en la impresión de objetos en tres dimensiones. En este proyecto se utiliza en la construcción del efecto final y los acoplos para los rodamientos.

### 2.2.2. DISEÑO DE LAS PARTES DEL PROTOTIPO.

El diseño de las partes mecánicas del prototipo es realizado en el programa SOLIDWORK, el cual es un tipo de software CAD (Diseño Asistido por computadora). Este programa permite modelar, ensamblar y extraer de ellos planos técnicos como otro tipo de información necesaria para la construcción de las piezas mecánicas.

A continuación, en la Figura 2.15 se puede apreciar el diseño del robot tipo delta, donde se muestran claramente sus partes que posteriormente serán descritas.



**Figura 2.15.** Prototipo diseñado y partes principales.

Finalmente se presenta el prototipo de robot delta junto con el soporte, la figura 2.16 muestra el prototipo terminado.



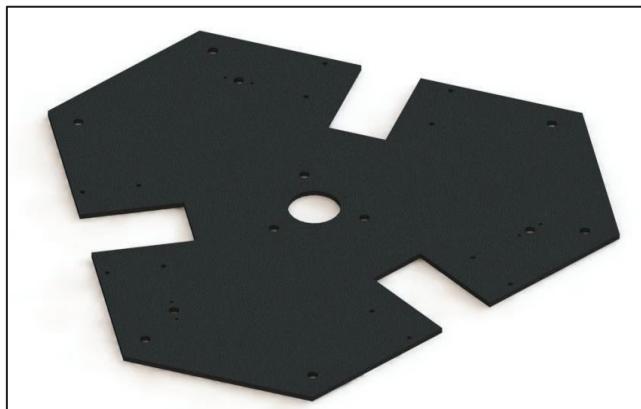
**Figura 2.16.** Prototipo robot delta diseño final.

### 2.2.2.1. Descripción de los elementos diseñados

#### 2.2.2.1.1. Base Fija

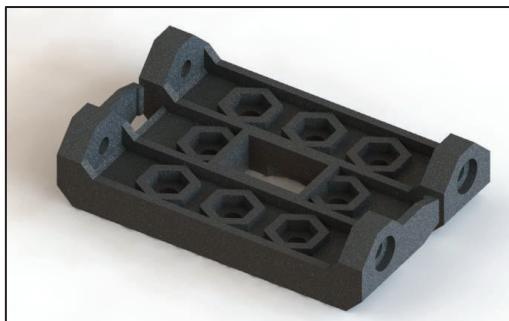
El diseño de la base fija se lo realiza en función de un polígono de seis lados (Figura 2.17.), tal que se pueda colocar los tres servomotores separados  $120^{\circ}$  entre sí y a una distancia de 110 mm del centro de la base. Adicionalmente este elemento sirve como acople con la estructura que sostiene a todo el sistema.

Para su construcción se considera que debe ser de un material rígido como el hierro con un espesor de 3 mm de tal manera que no se deforme.



**Figura 2.17.** Diseño CAD de la base fija.

Para montar el actuador en la base fija se utiliza el soporte que se puede observar en la Figura 2.18, la cual es parte del kit del servomotor Dynamixel AX-12 A y está construido en plástico ABS.

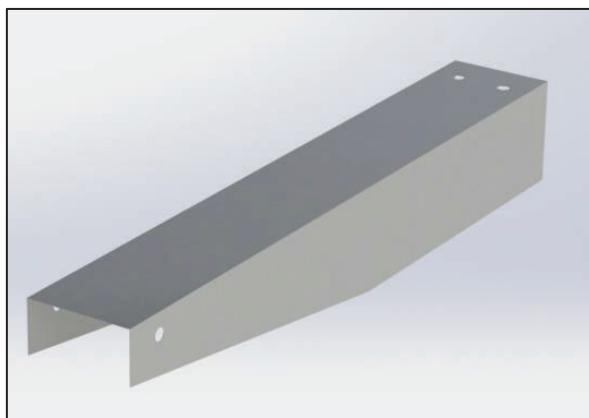


**Figura 2.18.** Diseño CAD del soporte del Dynamixel AX-12 A.

#### 2.2.2.1.2. Eslabón $L_1$

El eslabón  $L_1$  está diseñado en forma de un tubo rectangular con una longitud de 180 mm de largo por 25.4 mm de ancho. En uno de sus extremos se realiza las perforaciones para acoplar el eje del motor, mientras en el otro extremo se realiza las perforaciones para poder colocar las juntas esféricas, como se puede apreciar en la Figura 2.19.

Para su construcción se utiliza un tubo de aluminio debido a que es de peso ligero y resistente a la deformación.



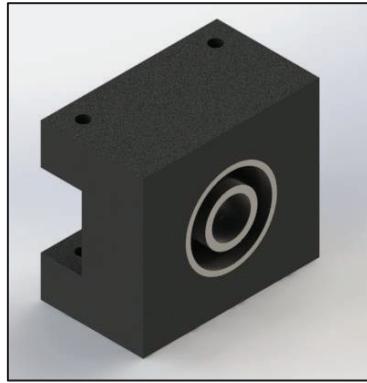
**Figura 2.19.** Diseño CAD del eslabón  $L_1$ .

Para transmitir el movimiento rotacional del eje del servomotor y acoplarlo con el eslabón  $L_1$  se utiliza el elemento de la Figura 2.20.



**Figura 2.20.** Acople eje del servomotor – eslabón  $L_1$ .

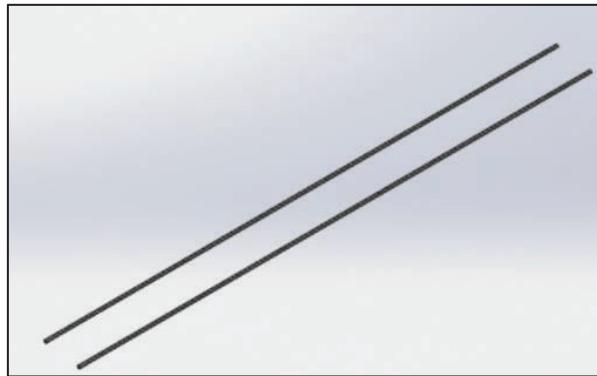
Para ayudar al movimiento del acople se añade rodamientos que se colocan en las chumaceras de plástico ABS (Figura 2.21.), y van sujetas a la plataforma fija.



**Figura 2.21** Rodamiento y chumacera.

#### 2.2.2.1.3. Eslabón $L_2$

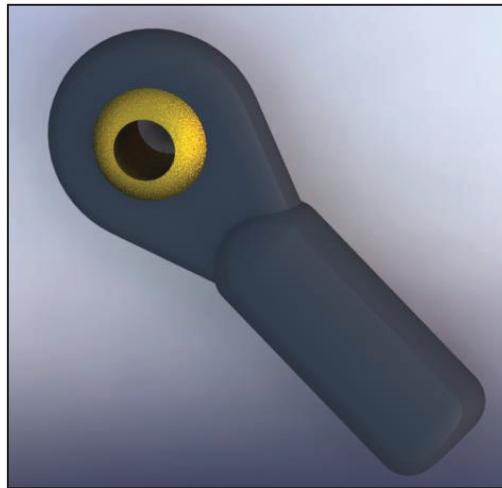
El eslabón  $L_2$  está constituido por dos barras roscadas, paralelas entre sí con diámetro de 3 mm cada uno y una longitud de 380 mm, (Figura 2.22)



**Figura 2.22.** Diseño CAD del eslabón  $L_2$ .

#### 2.2.2.1.4. Junturas esféricas

Debido a su difícil fabricación, esta pieza se adquiere en el mercado, sin embargo para poder ver los efectos de transmisión de movimiento de los motores hacia la base móvil se realizó su diseño CAD, que se puede observar en la Figura 2.23.

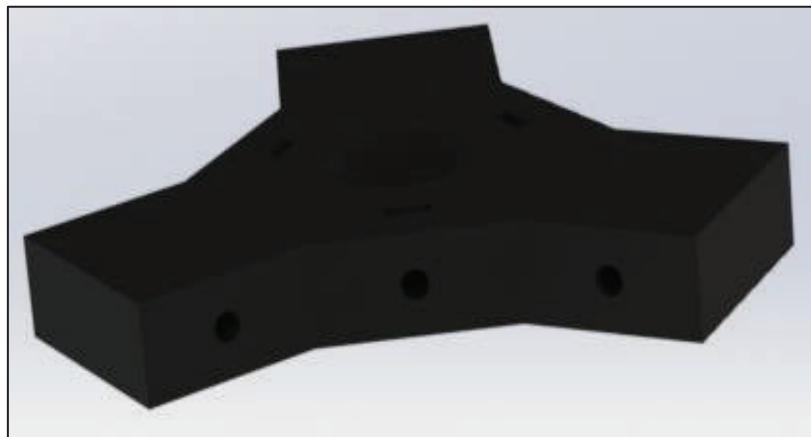


**Figura 2.23.** Diseño CAD de la juntura esférica.

#### 2.2.2.1.5. Base móvil

La base móvil se diseña tomando en cuenta que es el elemento donde se cierran las tres cadenas seriales y su centro es el punto de operación del prototipo donde se ubica el efecto final, por lo que su forma quedó establecido como se muestra en la Figura 2.24.

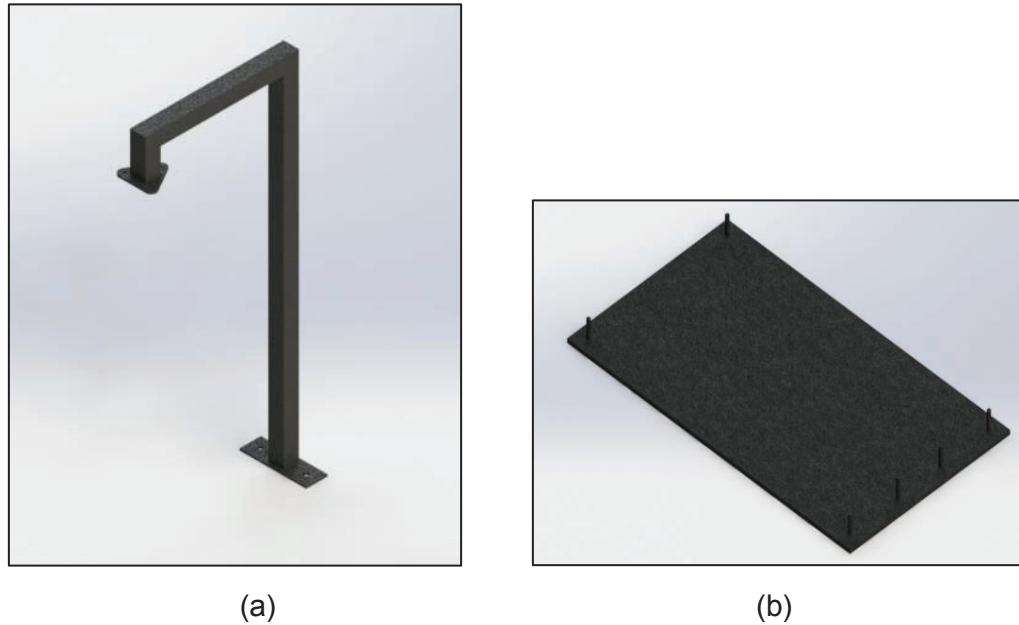
Para su construcción se realiza una impresión en 3D utilizando material plástico ABS para tener un elemento de peso ligero.



**Figura 2.24.** Diseño CAD del efecto final.

#### 2.2.2.1.6. Estructura de soporte

La estructura de soporte se la divide en dos piezas: Un tubo rectangular en forma de L como el de la Figura 2.25 (a), cuya función es la de un brazo de sujeción y una plancha de 300 mm\*500 mm y un espesor de 6 mm que actúa como un contrapeso para sostener el brazo que se muestra en la Figura 2.25 (b).



**Figura 2.25.** (a) Brazo de sujeción, (b) plancha inferior.

#### 2.2.2.2. Análisis estructural. [9][16][29]

SolidWorks integra herramientas de análisis fáciles de utilizar, para comprobar la operación y el rendimiento durante el desarrollo del producto. El análisis de diseño de SolidWorks aumenta la innovación en el producto al reducir los riesgos del diseño, ya que reduce significativamente el número de prototipos físicos necesarios y ayuda a reducir los costes de material, entre otros.

El software utiliza el método de elemento finito (FEM) que es una técnica numérica para analizar diseños de ingeniería. El FEM está aceptado como el método de análisis estándar debido a su generalidad y compatibilidad para ser implementado en computadoras. El FEM divide el modelo en numerosas piezas pequeñas de formas simples llamadas "elementos", que reemplazan eficazmente un problema complejo por muchos problemas simples que deben ser resueltos de manera simultánea.

El proceso de división del modelo en pequeñas piezas se denomina mallado, con este método se pueden resolver problemas unidimensionales, bidimensionales y tridimensionales. El programa asigna automáticamente el tipo de mallado de acuerdo a la geometría de la pieza; existe el mallado sólido tetraédrico que es asignado a figuras sólidas, las chapas metálicas o planchas por el contrario utilizan la malla de vaciado, como se verán en las figuras siguientes en donde se realizó el análisis la técnica de mallado que se asignó es la tetraédrica sólida en su versión parabólica que brinda mejores resultados que los elementos lineales porque:

- representan contornos curvos con mayor precisión
- producen mejores aproximaciones matemáticas que los elementos lineales.

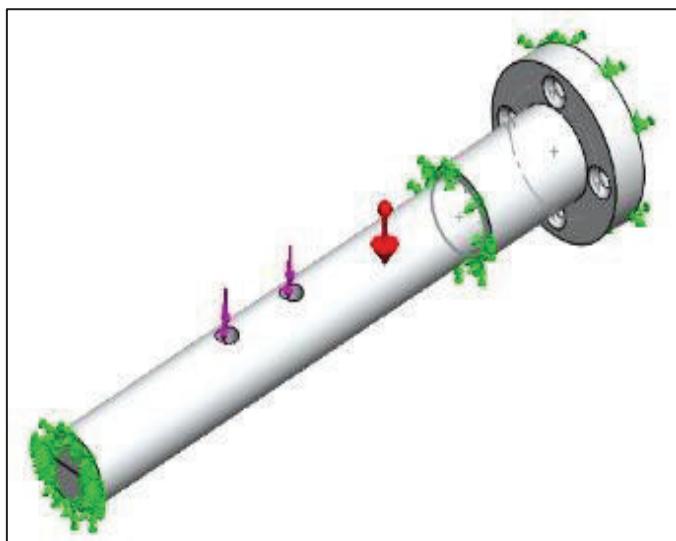
El análisis que se realiza a las piezas del prototipo es basado en el estudio estático, el cual calcula desplazamientos, fuerzas de reacción, deformaciones unitarias, tensiones y la distribución del factor de seguridad, estos estudios ayudan a evitar fallos ocasionados por altas tensiones.

El objetivo de este análisis es validar las dimensiones y material del prototipo de robot delta para su posterior construcción, por lo tanto no se realiza un análisis exhaustivo y no se pretende llevar a un diseño óptimo de los elementos del robot. El alcance de este proyecto es la construcción del robot delta que permita realizar tareas de apilamiento y colocación de objetos en puntos determinados.

El robot está diseñado para manejar cargas livianas, principalmente para ubicarse en puntos deseados comprobando de esta manera la cinemática del prototipo, por esta razón el análisis estructural se realiza en las piezas en donde se considera que soportan la mayor carga.

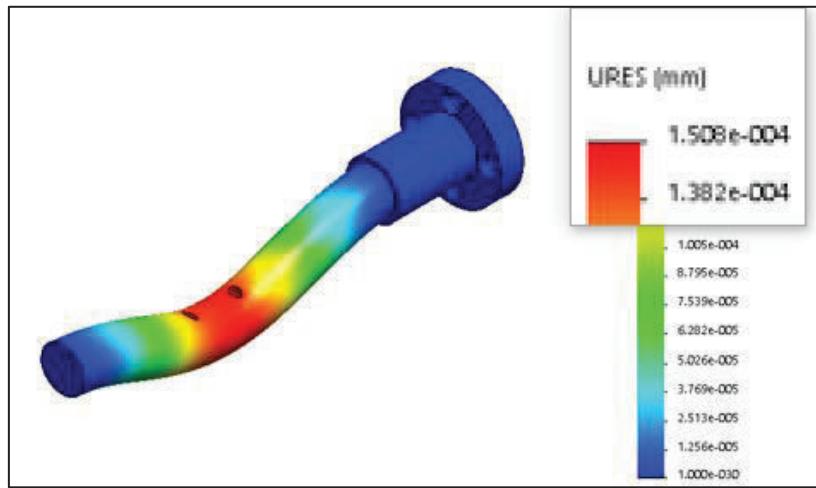
El primer estudio estático se realizó en el acople del eje del actuador con el eslabón  $L_1$ , cuyas longitud es de 85 mm y un radio de 5 mm. La sujeción del eje, son las caras cilíndricas en las cuales el eje se soporta sobre dos rodamientos, uno en cada extremo aplicando una fuerza de 10 N, en la misma dirección de la gravedad.

Como se observa en la Figura 2.26, las flechas verdes representan la sujeción, la flecha roja representa la gravedad y la flecha morada representa la fuerza aplicada.



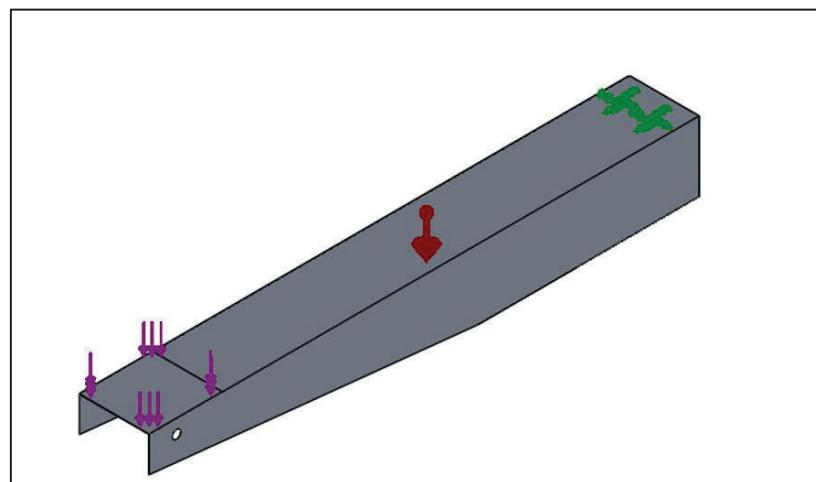
**Figura 2.26.** Análisis de Fuerzas sobre el eje de acople.

A continuación se muestran el análisis por elementos finitos que muestra la máxima deformación que se tiene en la pieza representada de color rojo, cuyo valor de desplazamiento es  $1.508e-4\text{mm}$ , este valor es muy pequeño y nos indica que la pieza trabajará correctamente con cargas menores a los 10 N (Figura 2.27)



**Figura 2.27.** Desplazamientos en el eje de acople.

El segundo estudio estático se realiza en el eslabón  $L_1$ , cuya función es la de transmitir el movimiento del actuador al eslabón  $L_2$  y a la base móvil. El eslabón  $L_1$  es un tubo de aluminio hueco de 25.4 mm\*25.4 mm y una longitud de 190 mm, a la cual se le aplicara una fuerza de 10 N identificado con el color morado y la sujeción se aplica en un extremo como se puede apreciar en la Figura 2.28.



**Figura 2.28.** Análisis de fuerzas sobre el eslabón  $L_1$ .

La Figura 2.29 muestra el análisis de desplazamiento máximo en la zona de color rojo y al igual que en la Figura 2.27 el valor es muy pequeño, lo que indica que puede trabajar correctamente con cargas menores a 10N.

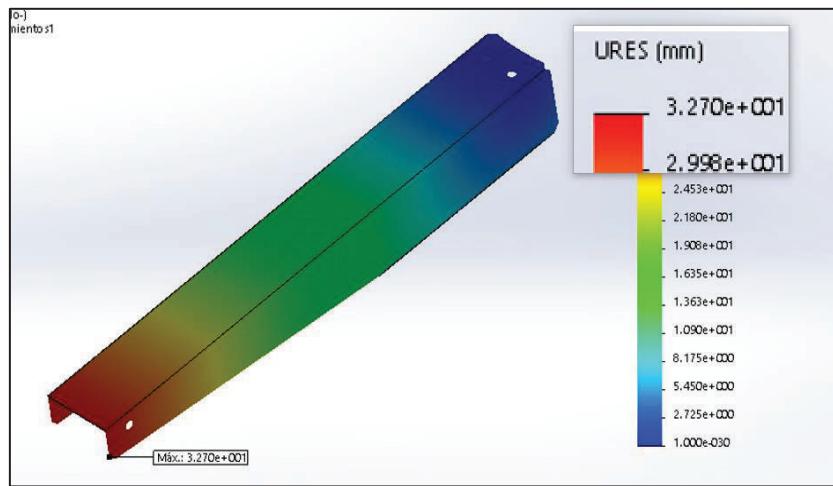


Figura 2.29. Desplazamientos del eslabón  $L_1$ .

Finalmente se hace el análisis sobre el brazo de sujeción estática el cual está construida con un tubo de hierro de 30 mm x 30 mm y cuyo resultado se pueden apreciar en las Figuras 2.30 y 2.31, en este caso la carga aplicada es de 50N.

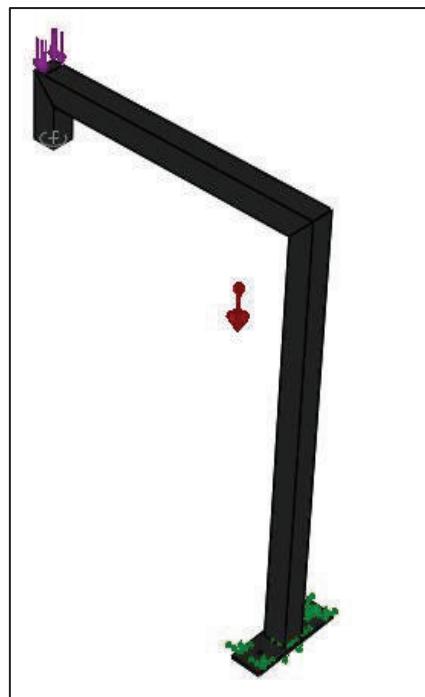
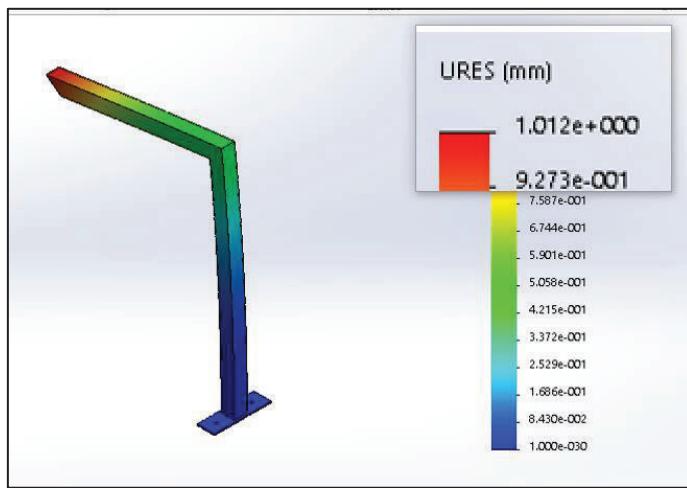


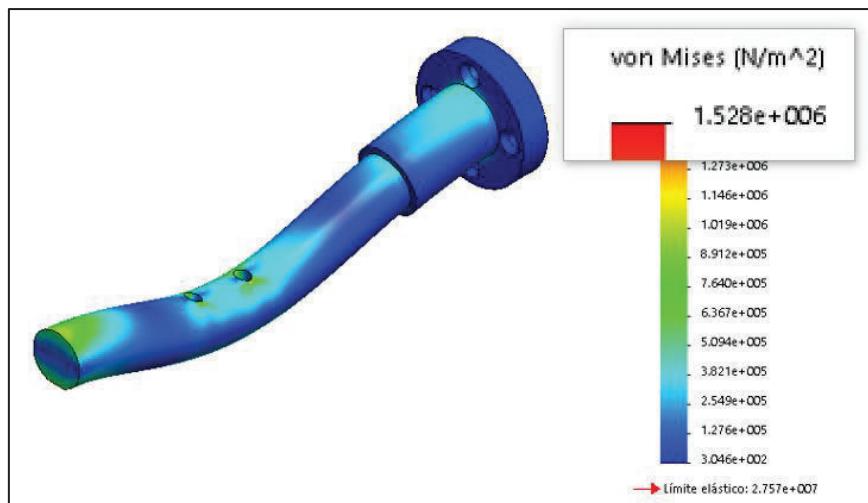
Figura 2.30. Análisis de fuerzas del brazo de sujeción estática.



**Figura 2.31.** Desplazamientos del brazo de sujeción estática.

Además se obtiene información sobre la tensión de Von-Mises o también denominada teoría de energía de distorsión, esta teoría se presenta en materiales dúctiles que son capaces de absorber una cierta cantidad de energía antes de sufrir una falla o de romperse.

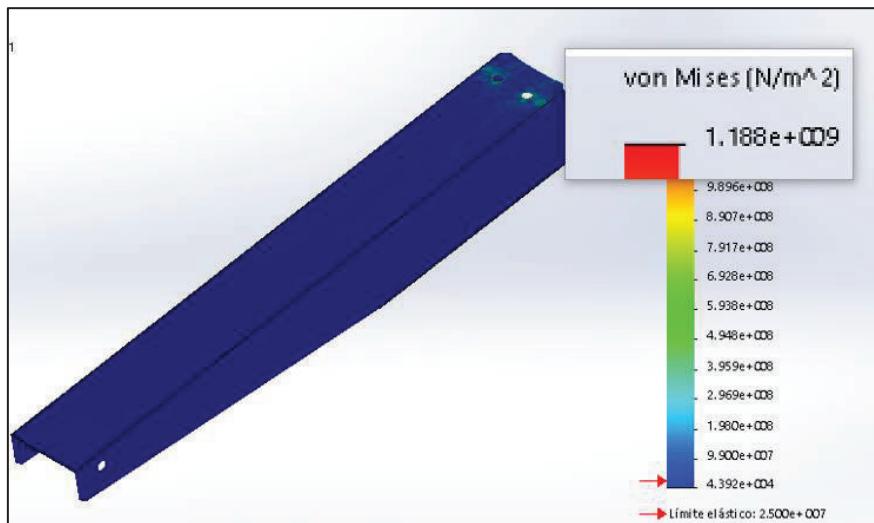
La Figura 2.32 muestra que el valor máximo de tensión resultante al aplicar una fuerza de 10N es de  $1.528\text{e}6 \text{ N/m}^2$  que es mucho menor al límite de fluencia que tiene el aluminio cuyo valor es  $2.757\text{e}7 \text{ N/m}^2$ , este valor indica que el material no se deformará de manera considerable.



**Figura 2.32.** Tensión de Von-Mises en el eje de acople.

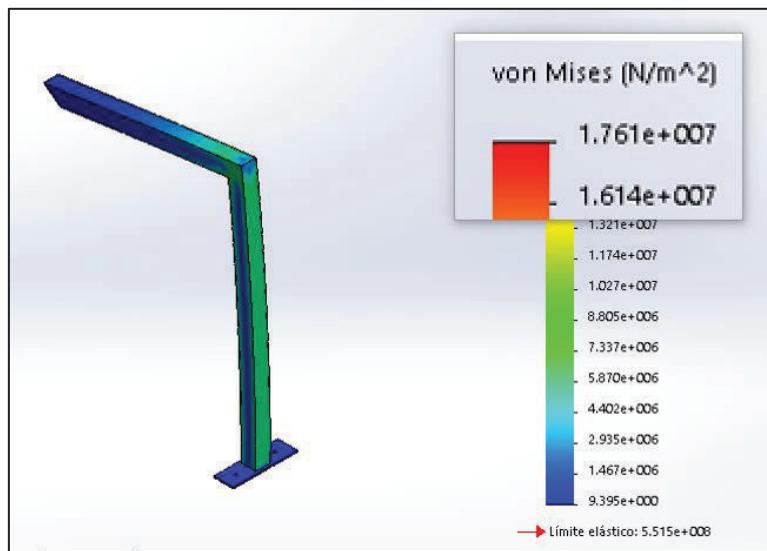
El análisis para la Figura 2.33 y la Figura 2.34 es similar al descrito en la Figura 2.32, si el valor máximo de tensión es menor al valor límite del material la pieza diseñada es adecuada para el correcto funcionamiento.

En la Figura 2.33 el valor de tensión obtenido es  $1.188e9 N/m^2$  que es mayor al límite de fluencia del aluminio, en este caso cabe destacar que la zona más afectada se presenta de color rojo lo cual es imperceptible en la figura, esto quiere decir que la deformación ocurre a un grado muy pequeño y ocurrirá en los orificios por el cual pasara un tornillo y se puede solucionar agregando una rodela para que el área de presión sea más grande y no solo en los filos de dicho orificio.



**Figura 2.33.** Tensión de Von-Mises del eslabón  $L_1$ .

La Figura 2.34 muestra los resultados de tensión de Von-Mises de una de las partes más importantes del prototipo, esta pieza diseñada se encargara de sujetar al robot el cual está constituido por los tres actuadores, tres eslabones  $L_1$ , 6 eslabones  $L_2$ , la base fija, la base móvil, por lo tanto debe ser la pieza de más resistencia y de una tensión de fluencia alta, por lo tanto el material utilizado es el hierro que presenta un límite de fluencia con un valor de  $5.515e8 N/m^2$ , el resultado de la simulación arroja un valor de tensión máximo de  $1.761e7 N/m^2$  que es mucho menor al límite, por lo tanto este brazo podrá soportar el peso del robot.



**Figura 2.34.** Tensión de Von-Mises del brazo de sujeción estática.

## 2.3. COMPONENTES ELECTRÓNICOS.

### 2.3.1. ACTUADORES

Los actuadores son los que proporcionan la fuerza o par necesario para generar el movimiento de los elementos del robot. Los actuadores pueden ser alimentados con energía neumática, hidráulica o eléctrica. [2]

Dentro del desarrollo de prototipos robóticos se emplea en su gran mayoría motores eléctricos y en particular motores de corriente continua, motores a pasos y servomotores, estos últimos pueden ser clasificarlos en dos tipos analógicos y digitales.

Los servomotores digitales difieren de los análogos en la placa de control donde se ha añadido un microprocesador que se hace cargo de analizar la señal, procesarla y controlar el motor, estos son capaces de memorizar parámetros de programación, que varían de acuerdo a cada fabricante pero en general son:

- Se puede programar el sentido de giro como "normal" o "inverso".
- Se puede variar la velocidad de respuesta del servo.
- Se puede programar una posición central (o posición neutra) diferente, sin afectar los radios de giro.
- Se pueden determinar diferentes topes de recorrido para cada lado.
- Es posible programar qué debe hacer el servo en caso de sufrir una pérdida de señal.
- Es posible programar la resolución.

Otra de las grandes diferencias está en la velocidad a la que reacciona el servo a un cambio en la señal. En un mismo lapso, el servo digital puede recibir cinco o seis veces más pulsos de control que un analógico. Como resultado la respuesta del servo a un cambio en la orden de posición es mucho más veloz.

### 2.3.1.1 Dynamixel AX-12 A. [24][25]

En el prototipo se utiliza el servomotor digital DYNAMIXEL AX-12A creado por ROBOTIS el cual trae incorporado un microcontrolador ATMEGA\_8 capaz de entender 50 comandos de lectura y escritura que definen su comportamiento, como la capacidad de controlar su velocidad, temperatura, posición, tensión, etc.. En la Tabla 2.5 se puede ver las características generales dadas por el fabricante.



**Figura 2.35.** Servomotor Dynamixel AX-12A [25]

**Tabla 2.5.** Características técnicas del servo Dynamixel AX-12A

Peso	54.6 g
Dimensiones	32x50x40 mm
Resolución	0.29°
Relación de engranes	254:1
Torque	15.2 Kgf*cm (a 12.0 V, 1.5 A)
Velocidad	59 rpm (a 12.0 V) sin carga
Grado de ejecución	0° a 300°
Rango de Temperatura	-5°C a 70°C
Voltaje	9 a 12 V (Voltaje recomendado 11.1 V)
Señal de Control	Paquete Digital
Tipo de Protocolo	Comunicación Serial Asíncrono Half Duplex (8 bit, 1 bit parada, No paridad)
Enlace (Física)	TTL
ID	254 ID (0 a 253)
Velocidad de Comunicación	7343 bps a 1 Mbps
Realimentación	Posición, Temperatura, Carga, Voltaje de entrada

#### 2.3.1.1.1. Conexión de los Dynamixel

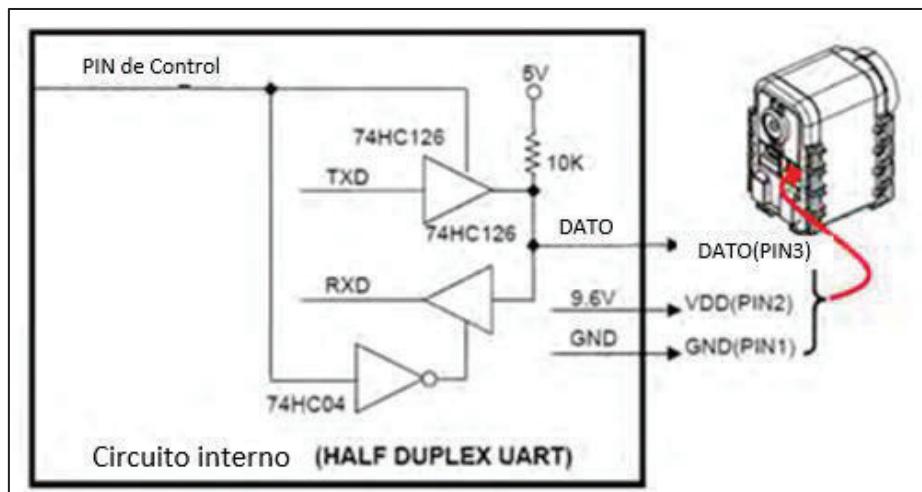
Los Dynamixel AX-12A utilizan una topología denominada cadena margarita la cual consiste en una sucesión de enlaces tal que un dispositivo A es conectado a un dispositivo B, el mismo dispositivo B a un dispositivo C, este dispositivo C a un dispositivo D, y así sucesivamente.

Las conexiones no forman redes es decir el ultimo dispositivo conectado no puede ser conectado directamente al primer dispositivo, como se puede observar en la Figura 2.36.



**Figura 2.36.** Conexión en cadena margarita de los Dynamixel AX-12A [25]

Para el control del servomotor se puede utilizar la placa Arduino directamente si la comunicación es en un solo sentido, pero para el caso de una comunicación bidireccional half duplex se requiere un buffer de tres estados como el C.I. 74HC241, que controla el sentido de flujo de la comunicación. En la Figura 2.37 se observa el circuito interno de un C.I. 74HC241.



**Figura 2.37.** Conexión UART-Dynamixel [25]

#### 2.3.1.1.2 Protocolo de Comunicación

Los actuadores emplean un protocolo basado en paquetes, dentro del cual existen dos posibles tipos:

- Los paquetes de instrucción: Estos paquetes son recibidos por los actuadores y contienen comandos que deben ser ejecutados.
- Los paquetes de estatus, enviados por los actuadores, contienen la respuesta a un paquete de instrucción.

En la Figura 2.38, se muestra el protocolo para los dos tipos de paquetes, los cuales están precedidos por dos caracteres de inicio, un encabezado el cual contiene la dirección o identificador del actuador al que corresponde el paquete, la longitud del paquete y un último campo que corresponde a la instrucción que debe ser ejecutado (en el caso de un paquete de instrucción), o un código de lectura (cuando el paquete es de estatus).

El paquete puede contener también información adicional, necesaria para la ejecución de una instrucción o que corresponde a la información que le fue solicitada al actuador.

Finalmente se incluye una suma de comprobación para permitir la detección de errores en el paquete.



**Figura 2.38.** Estructura de los paquetes utilizados por el actuador Dynamixel

### 2.3.2. ARDUINO

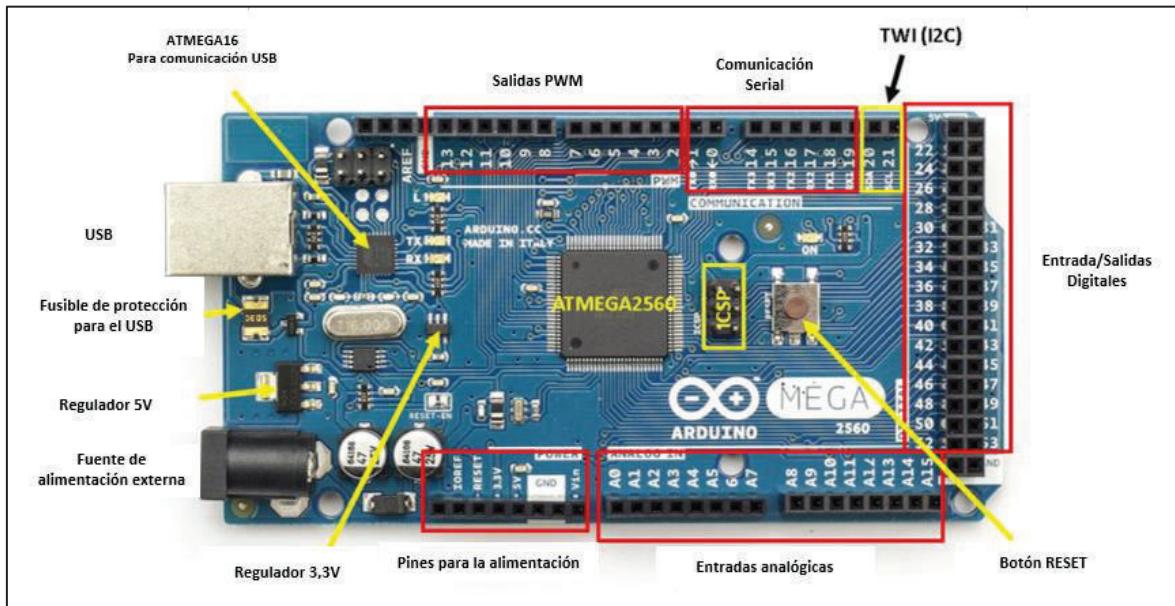
Arduino es una plataforma open-hardware basada en el diseño de una placa sencilla de entradas y salidas analógicas y digitales.

Dentro del proyecto se utiliza la placa Arduino Mega, cuyas características se puede ver en la Tabla 2.6.

**Tabla 2.6.** Especificaciones técnicas de Arduino Mega.

Microcontrolador	ATmega2560
Voltaje de operación	5 V
Voltaje de entrada recomendado	7-12 V
Voltajes de entrada límite	6-20 V
Pines digitales I/O	54 (de los cuales 15 proveen salida PWM)
Pines de entrada analógica	16
Corriente DC por Pin I/O	20 mA
Flash Memory	256 KB de los cuales 8 KB son usados por el gestor de arranque
SRAM	8 KB
EEPROM	4 KB
Oscilador	16 MHz
Puerto Serie de Hardware	4

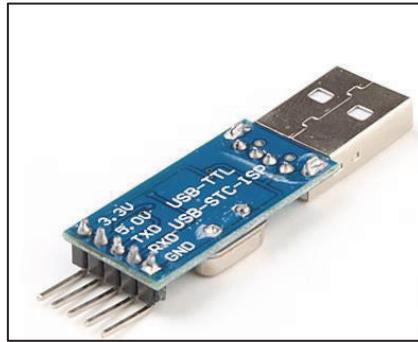
La Figura 2.39 permite visualizar la distribución de los pines de la placa Arduino.



**Figura 2.39.** Distribución de pines de la placa Arduino

### 2.3.3. CONVERTIDOR DE SEÑAL TTL A USB

Este módulo es utilizado para conectar los dos pines de un puerto serial del Arduino al puerto USB de cualquier computador, (Figura 2.40.).



**Figura 2.40.** Conversor de TTL a USB

### 2.3.4 FUENTE DE ALIMENTACIÓN

Para alimentar el sistema del prototipo se consideran tres fuentes de voltaje, dividiendo al sistema de la siguiente manera:

- Sistema de control: Esta comprendido por la placa Arduino, el dispositivo convertidor de señal TTL a USB y el circuito integrado 74HC241, los cuales requieren un voltaje de 5 voltios.
- Servomotores: El voltaje adecuado para el funcionamiento de los actuadores es de 12 V, con un considerable consumo de energía
- Efector final: El cual puede ser un electroimán que requiere una alimentación de 12 V.

Considerando los puntos expuestos y debido al alto consumo de corriente se decidió utilizar una fuente conmutada, también conocido como Swiching Power Supply, cuyas características podemos observar en la Tabla 2.7.

Las fuentes de alimentación conmutada se caracterizan por un aumento de la frecuencia de la corriente, que pasa de oscilar 60 Hz a más de 100KHz, reduciendo de esta forma las pérdidas en el transformador consecuentemente se reduce el tamaño, volumen y peso.

**Tabla 2.7.** Características de la Fuente de Alimentación

Entrada	Voltaje (AC) V	Corriente A	Frecuencia Hz
115	12	60/50	
	6		
Salida (600w)	Voltaje (DC) V	Corriente A	Color del cable
	+3.3	32	Naranja
	+5	36	Rojo
	+12	18	Amarillo
	+12	20	Amarillo/Negro
	5 VSB	2	Morado
	-12	1	Azul
	PS-ON		Verde
	P.G		Gris

## 2.4. MONTAJE DEL PROTOTIPO.

Una vez diseñado todas las partes constitutivas del prototipo se procede a realizar el ensamblaje como se describe a continuación:

1. Acoplar el brazo de sujeción estático a la plancha, colocando el brazo en los pernos de 8 mm de diámetros que se encuentran soldados a la plancha y utilizando sus respectivas tuercas se debe realizar su ajuste (Figura 2.41).



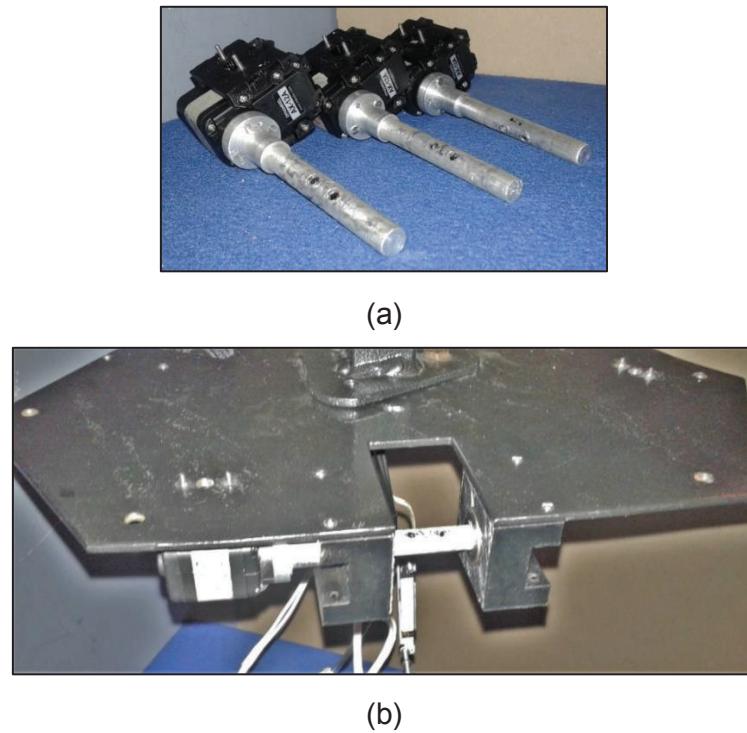
**Figura 2.41.** Soporte del prototipo con cables de funcionamiento.

2. Acoplar la plataforma fija al brazo de sujeción, colocando los pernos de 6 mm de diámetro (Figura 2.42).



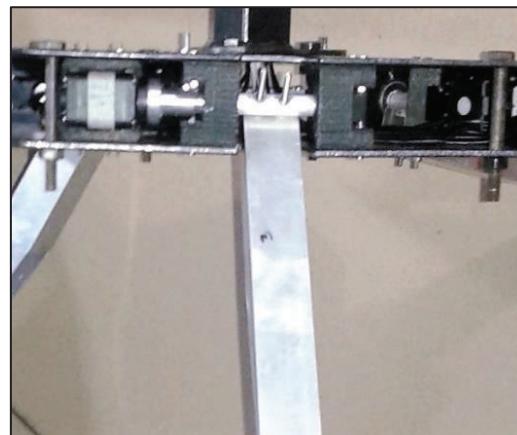
**Figura 2.42.** Plataforma fija acoplada a la base.

3. Colocar el acople en el eje del actuador y también se debe colocar los frames F3 los cuales son utilizado para poder sujetar el actuador en la plataforma fija. El acople del eje va asentado sobre dos rodamientos tal como se muestra en la Figura 2.43 los cuales están colocados en unas chumaceras, las que van sujetas a la plataforma fija.



**Figura 2.43.** (a) Acople eje-actuador, (b) Acople actuador-plataforma fija.

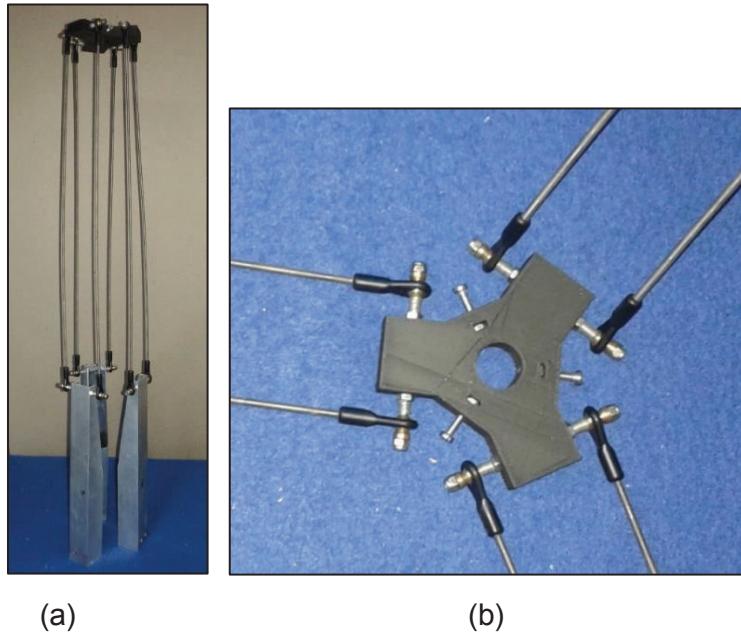
4. Colocar el eslabón  $L_1$  al acople del eje del actuador utilizando unos tornillos de diámetro 3 mm y los sujetamos con dos tuercas en cada extremo como se aprecia en la Figura 2.44.



**Figura 2.44.** Acople eslabón  $L_1$  al eje del actuador.

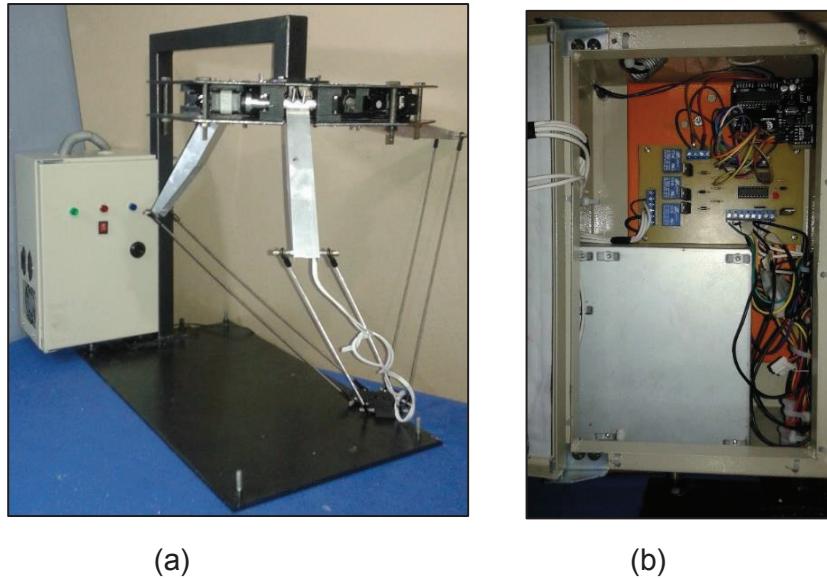
5. Acoplar el eslabón  $L_1$  al eslabón  $L_2$ . En uno de los extremos del eslabón  $L_1$  se pasa una varilla de 50 mm de largo con un diámetro de 3 mm sobre los

extremos de dicha varilla colgara las juntas esféricas formando un paralelogramo con las varillas que son parte del eslabón  $L_2$  y son cerradas con la plataforma móvil (Figura 2.45).



**Figura 2.45.** (a) Acople eslabón  $L_1$ -eslabón  $L_2$ , (b) Acople eslabón  $L_2$ -plataforma móvil.

6. Finalmente se debe montar el gabinete que contiene las placas electrónicas y la fuente de alimentación del mecanismo.



**Figura 2.46.** (a) Prototipo terminado, (b) Gabinete de placas electrónicas.

## 2.5. COSTO DEL PROYECTO.

Para hacer una estimación del costo total del proyecto se realiza la Tabla 4.5, que contiene los precios de los materiales utilizados desde la fase inicial del proyecto, comenzando por la estructura inicial y la cantidad de motores utilizados hasta conseguir el prototipo expuesto en este trabajo.

**Tabla 2.8.** Detalles del costo del proyecto.

Estructura inicial					
Elemento	Cantidad	Precio unitario	Precio total		
Aluminio (60x40x6)[mm]	1	32	32		
Rotula RS acero M6	12	5	60		
Varilla de aluminio M6	1	18	18		
Fabricación piezas mecánicas	1	224	224		
Soporte brazo	1	40	40		
Servomotores Dynamixel AX-12 A	3	70	210		
Arduino UNO	1	20	20		
Cables y tornillería	1	7	7		
	Subtotoal_1		611		
Estructura Final					
Componentes mecánicos					
Elemento	Cantidad	Precio unitario	Precio total		
Varilla roscada M3	6	5	30		
Efector final (Impresión 3D)	1	10	10		
Chumaceras (Impresión 3D)	6	3	18		
Rodamientos (3/8")	6	3,5	21		
Eslabón L1	3	10	30		
Base fija	2	30	60		
Acople motor-eslabón L1	3	10	30		
Tornillos M2,M3,M6,M8	1	10	10		
Tabla de base(590x460x12)[mm]	1	6	6		
Servomotores Dynamixel AX-12 A(extras)	3	70	210		
Rotulas de plástico M3	12	1,7	20,4		
	Subtotoal_2		445,4		

Componentes electrónicos			
Elemento	Cantidad	Precio unitario	Precio total
Arduino MEGA	1	20	20
lamina de transferencia	1	0,89	0,89
Fundas de 20 gramos de Cloruro	2	0,31	0,62
Baquelita Fibra 1 Lado 10x20	1	2,42	2,42
Resistencias de 1/4 W	9	0,02	0,18
4N35 ECG NPN Transistor Output	1	0,79	0,79
TIP 122 Transistor Darlington	4	0,47	1,88
Conector Header Tipo Hembra 40 Pin	1	1,52	1,52
Zócalo 20 pines	1	1,45	1,45
Zócalo 6 pines	1	0,77	0,77
CI 74LS241	1	2,12	2,12
Relé de 5Vdc	3	1,27	3,81
Luz Piloto	2	2	4
Interruptor	1	2	2
Pulsador	1	0,33	0,33
borneras de 3 pines	5	0,29	1,45
Diodos 1N4004	4	0,2	0,8
Piñones para Dynamixel AX-12 A	4	17,25	69
Cable para servo 20[mm]	10	2,7	27
Cable negro hilado #18	10	0,44	4,4
Cable USB macho-hembra	1	3	3
USB-TTL	1	5	5
Cámara para detección de objetos	1	12	12
Fuente conmutada	1	20	20
Electroimán(zumbador)	1	4	4
	Subtotal_3		189,43
Componentes extras			
Elemento	Cantidad	Precio unitario	Precio total
Caja metálica (300x200x150)[mm]	1	22,37	22,37
Armando de objetos	10	0,7	7
Maguera corrugada (1/2")	1	0,8	0,8
	Subtotal_4		30,17
Valor horas de trabajo			
Horas de trabajo	800	7	5600
	Subtotal_5		5600
Valor total del proyecto			6875.87

## CAPÍTULO 3

### DESARROLLO DEL PROGRAMA DE CONTROL

En el presente capítulo se muestra el desarrollo de los programas computacionales que permiten el control del robot delta, el cual se ha dividido en dos partes, siendo la primera aquella que corresponde al enlace entre el prototipo y la computadora, esta parte estará compuesta de un sistema microprocesado que se encarga de interpretar los datos recibidos por el computador y los coloca en una trama que permite el funcionamiento de los servomotores Dynamixel que se explicaron en el capítulo anterior.

La segunda parte consta de un programa desarrollado en la computadora mediante el software MATLAB 2014A, que calcula los valores de los ángulos que se envían a los actuadores, que permiten el movimiento del prototipo y de esta manera poder realizar el seguimiento de trayectorias continuas y la ejecución de trayectorias punto a punto.

Se crea una interfaz gráfica para que el usuario por medio de botones y valores ingresados pueda operar el mecanismo, esta interfaz tiene partes que detallan el cálculo del espacio de trabajo, seguimiento de trayectorias continuas y una parte referente a tareas de colocación de objetos.

Para la creación de la interfaz de usuario se utilizó el GUI (Interfaz gráfica de usuario) de MATLAB, esta interfaz permite variar los parámetros del robot y de esta manera observar los efectos dentro del prototipo mediante animación, además se puede observar las tareas ejecutadas y el cálculo de la posición de objetos a recoger.

En el desarrollo de este capítulo se detallan las consideraciones tomadas para la elaboración de los programas.

### 3.1. PROGRAMA DE CONTROL PARA EL PROTOTIPO

Los programas del prototipo son los que se encargan de dar los valores a los ángulos a los actuadores, por una parte se hace uso de las cinemática inversa donde, como dato se ingresa los valores de la trayectoria calculada o establecida y como resultado se obtiene los ángulos de los tres servomotores, antes de que estos valores sean enviados a la tarjeta Arduino donde se ejecuta el programa de enlace deben ser transformados a valores que sean interpretados por los servomotores, tomando en cuenta la posición de  $0^{\circ}$  que tienen estos servos, luego de pasar por la posición de acondicionamiento los valores son enviados por comunicación serial a la parte de enlace donde se procesa y se envía a los servomotores, como se muestra en la Figura 3.1.



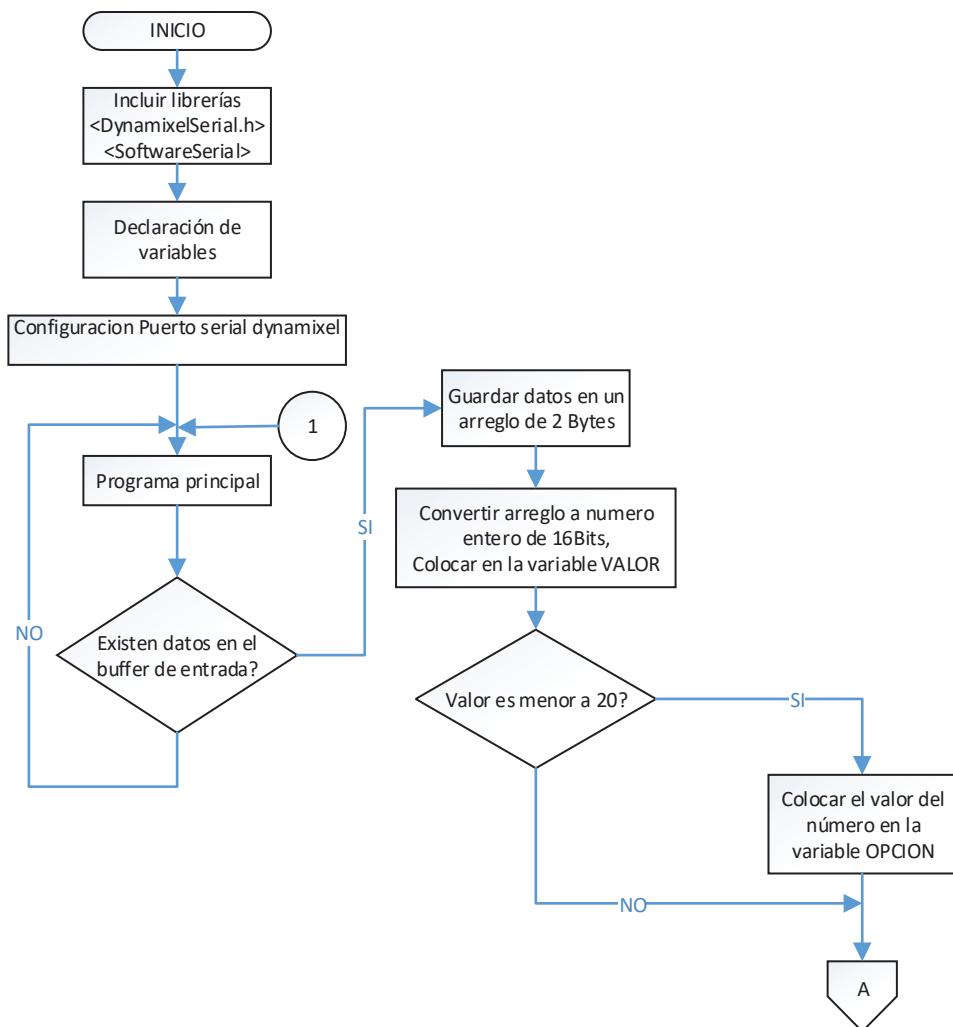
**Figura 3.1.** Secuencia programa de control del prototipo

#### 3.1.1. PROGRAMA DE ENLACE PC-ROBOT, ROBOT-PC.

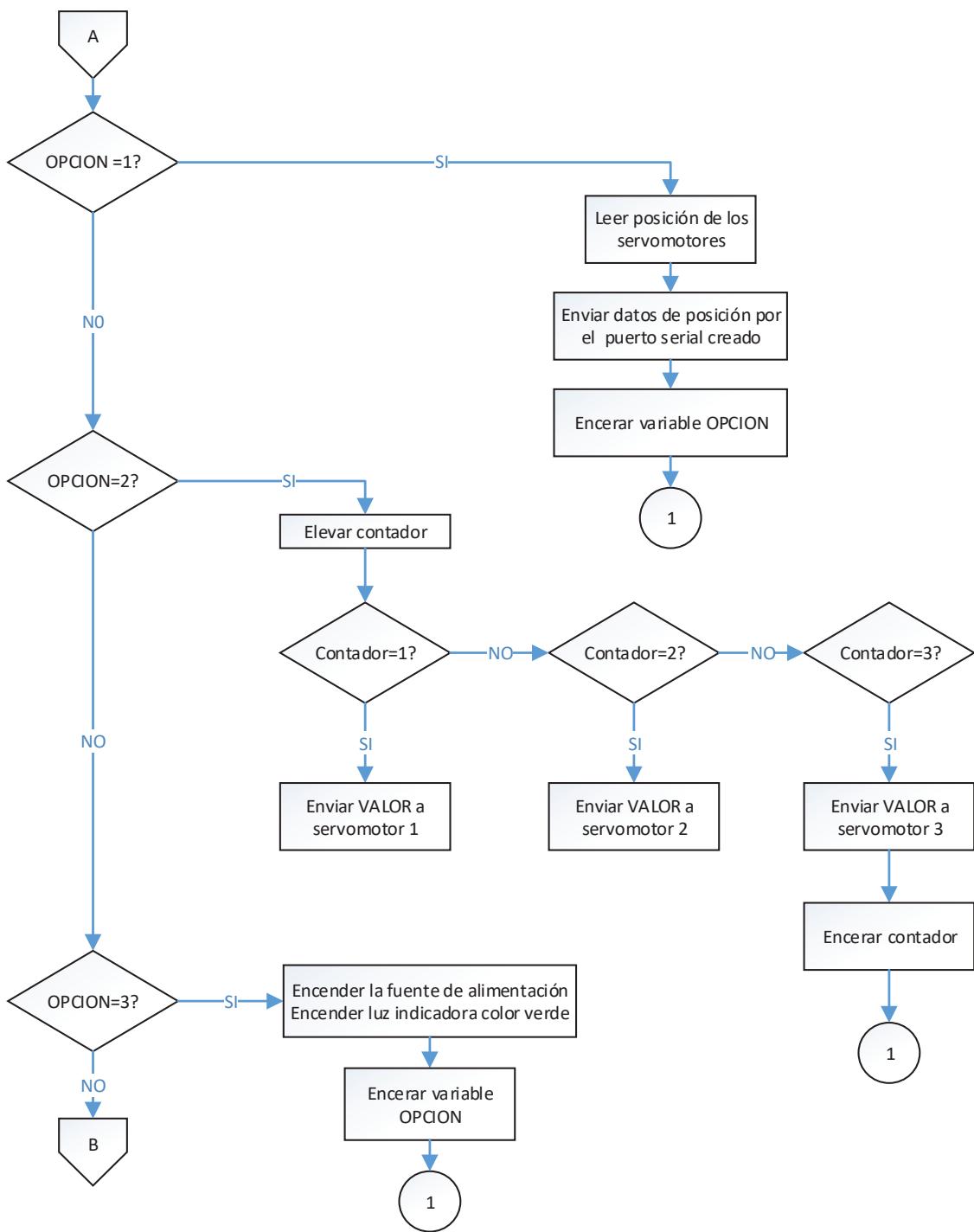
El programa de enlace está desarrollado en la tarjeta Arduino UNO, esta tarjeta se encarga de enviar y recibir datos de los actuadores, posteriormente los datos adquiridos de los servomotores se envían al computador si es que estos son requeridos, la tarjeta también recibe los datos codificados desde el programa principal que se encuentra en la computadora, estos datos codificados son las posiciones angulares que deben adoptar los actuadores, para realizar estas tareas se utilizan dos puertos seriales.

El puerto serial principal se encarga de hacer la comunicación tipo half dúplex que necesitan los servomotores, para esta función es necesario un CI 74HC241 que es un buffer de tres estados, el otro puerto serial necesario se crea mediante software para que pueda comunicarse con la computadora y se ocupe del tránsito de los datos codificados.

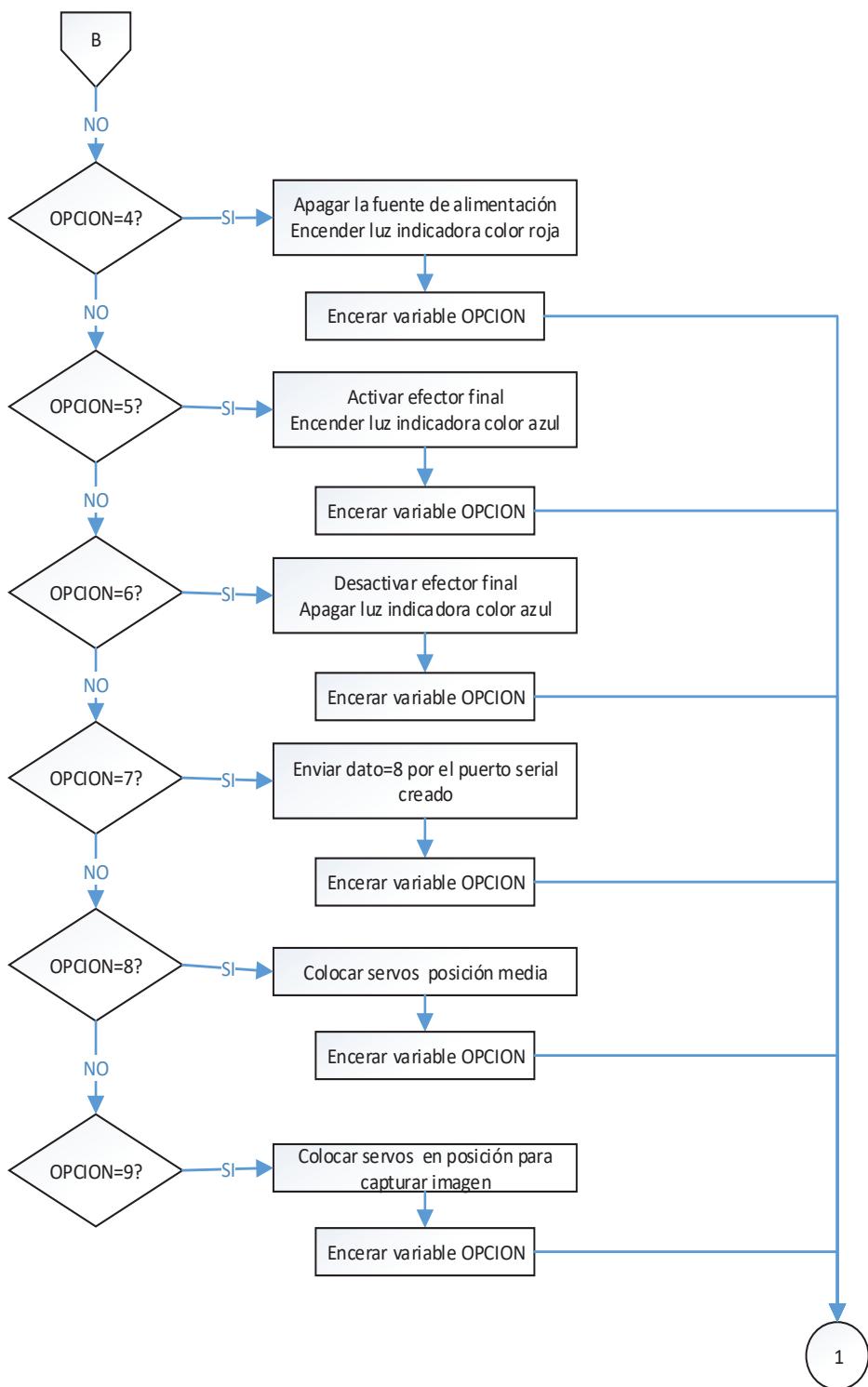
Al momento de encender la fuente se alimenta la tarjeta Arduino y comienza a configurar los puertos seriales y pines de salida para poder operar, la tarjeta queda en modo pasivo hasta recibir instrucciones desde el computador. La Figura 3.2 muestra el diagrama de flujo del programa subido en la tarjeta.



**Figura 3.2.** Diagrama de flujo del programa de enlace.



**Figura 3.2.** Diagrama de flujo del programa de enlace (continuación).



**Figura 3.2.** Diagrama de flujo del programa de enlace (continuación).

### **3.1.2. PROGRAMA DE CONTROL PRINCIPAL.**

El programa para el control del sistema se lo realiza en el software MATLAB. Este software permite crear archivos con extensión .m los cuales contienen los algoritmos matemáticos que gobiernan los movimientos del mecanismo. Además este software permite establecer la relación PC-mundo exterior a través de la configuración de la comunicación serial la cual activa el puerto del ordenador donde se encuentre conectado algún dispositivo.

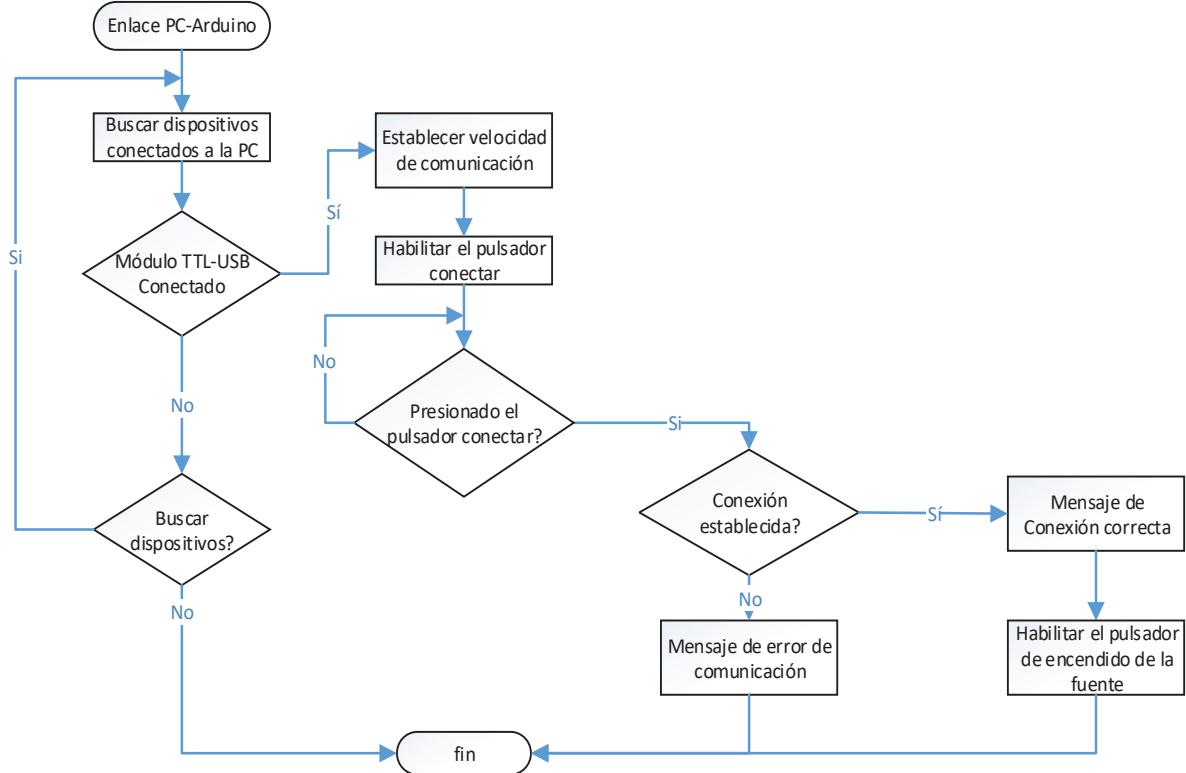
MATLAB ofrece una gran variedad de aplicaciones tales como procesamiento de señales e imágenes, diseños de control, modelado y puede mostrar los resultados gráficamente. [27]

#### **3.1.2.1. Comunicación Matlab-Arduino**

Una vez alimentado todo el sistema, para poder controlar el mecanismo se debe establecer una comunicación serial entre la PC y la placa Arduino, procediendo de la siguiente forma: Se debe verificar los dispositivos conectados a la PC, buscando el puerto serial en el cual se encuentre conectado el módulo TTL-USB, luego se configura los parámetros de comunicación como la velocidad de trasmisión, los bits de datos, bits de parada, paridad.

Una vez establecida la comunicación en la PC se debe esperar a un comando de confirmación, para luego proceder con el encendido de la fuente de alimentación de los actuadores, quedando de esta forma el sistema listo para ejecutar todos los comandos e instrucciones que se le envíe desde el computador.

En la Figura 3.3, se muestra el seudocódigo para establecer la comunicación entre MATLAB y la placa Arduino.



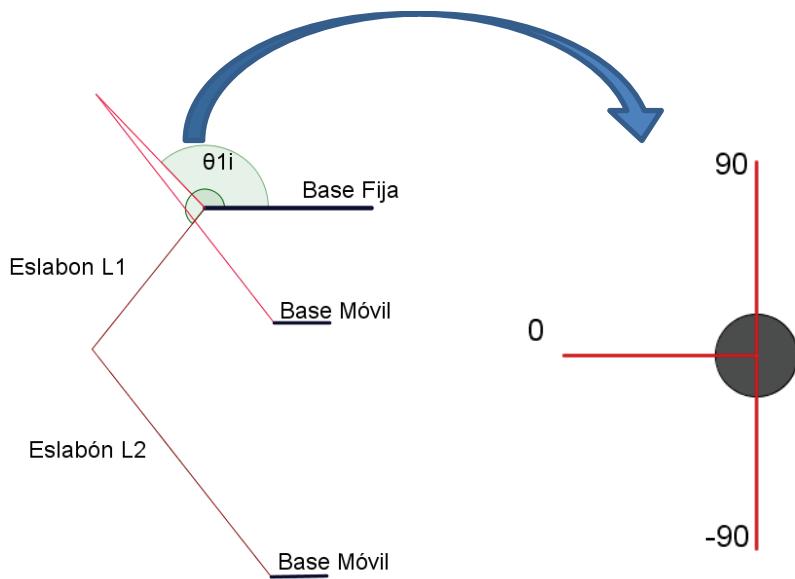
**Figura 3.3.** Diagrama de flujo para establecer la conexión PC-Arduino.

### 3.1.2.2. Acondicionamiento de datos para los actuadores

Para saber qué posición deben adquirir los actuadores se hace uso de la cinemática inversa, en donde los datos de entrada serán los puntos de la trayectoria deseada  $P(x_{(n)}, y_{(n)}, z_{(n)})$ , y los datos de salida serán los ángulos o posiciones de los servomotores  $(\theta_{11}, \theta_{12}, \theta_{13})$ .

$$(\theta_{11}, \theta_{12}, \theta_{13}) = \text{CinemáticaInversa}(x_{(n)}, y_{(n)}, z_{(n)}) \quad (3.1)$$

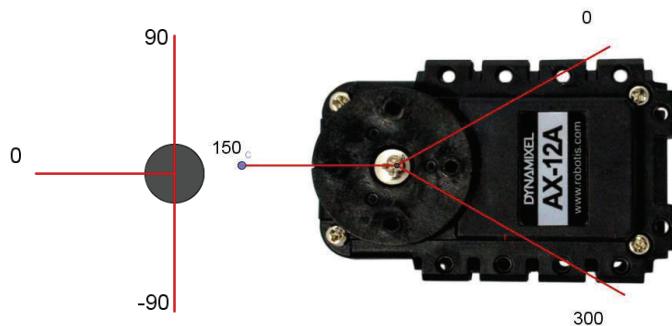
Los valores de los ángulos calculados  $\theta_{1i}$  ( $i = 1, 2, 3$ ), van desde  $(-90^\circ$  hasta  $90^\circ)$  como se muestra en la Figura 3.4.



**Figura 3.4.** Interpretación de los ángulos calculados en la cinemática inversa.

Los actuadores Dynamixel presentan una configuración diferente en cuanto a la ubicación de  $0^\circ$ , además el acondicionamiento también dependerá de la forma en la que el servomotor es ubicado en el prototipo.

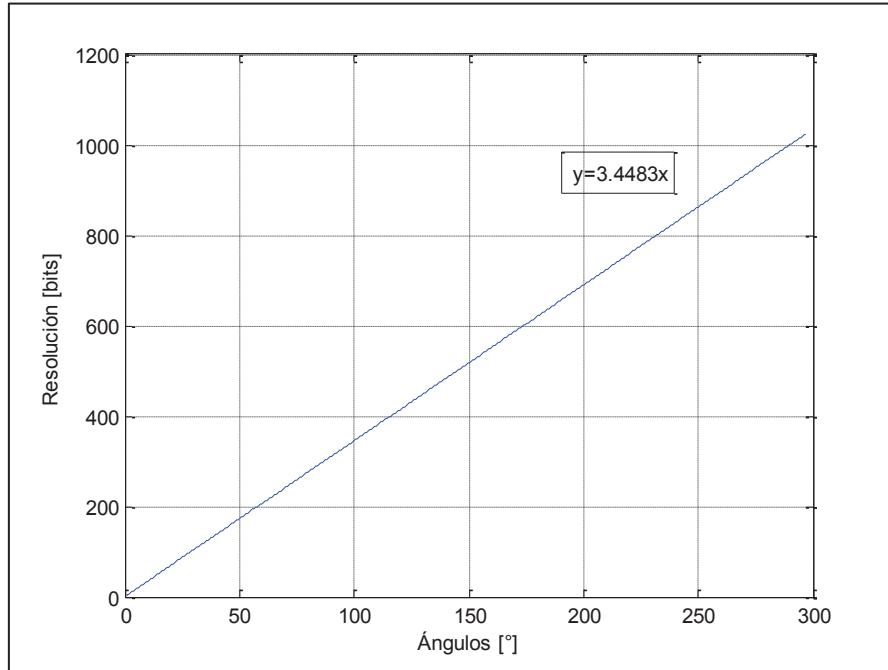
En la Figura 3.5 se puede observar la diferencia entre los ángulos calculados por la cinemática inversa y los ángulos de los actuadores en el mecanismo.



**Figura 3.5** Ángulos de la cinemática inversa y ángulos de los servomotores Dynamixel.

La Figura 3.5 muestra que el desplazamiento en cuanto al cero es de  $150^\circ$ , por lo tanto los valores que se calculan en la cinemática inversa deben ser corregidos aplicando este desplazamiento, además los servomotores Dynamixel reciben el valor de su posicionamiento en bits esto es, para  $0^\circ(0)$ , para  $150^\circ(512)$  y para  $300^\circ(1024)$ .

Los valores de los ángulos que serán enviados a la etapa de enlace estarán en su valor equivalente a bits tomando en cuenta la resolución en grados por bits que tiene el servomotor que en este caso es  $0.29^\circ$ . La Figura 3.6 muestra la curva de la resolución en grados vs su equivalente en bits.



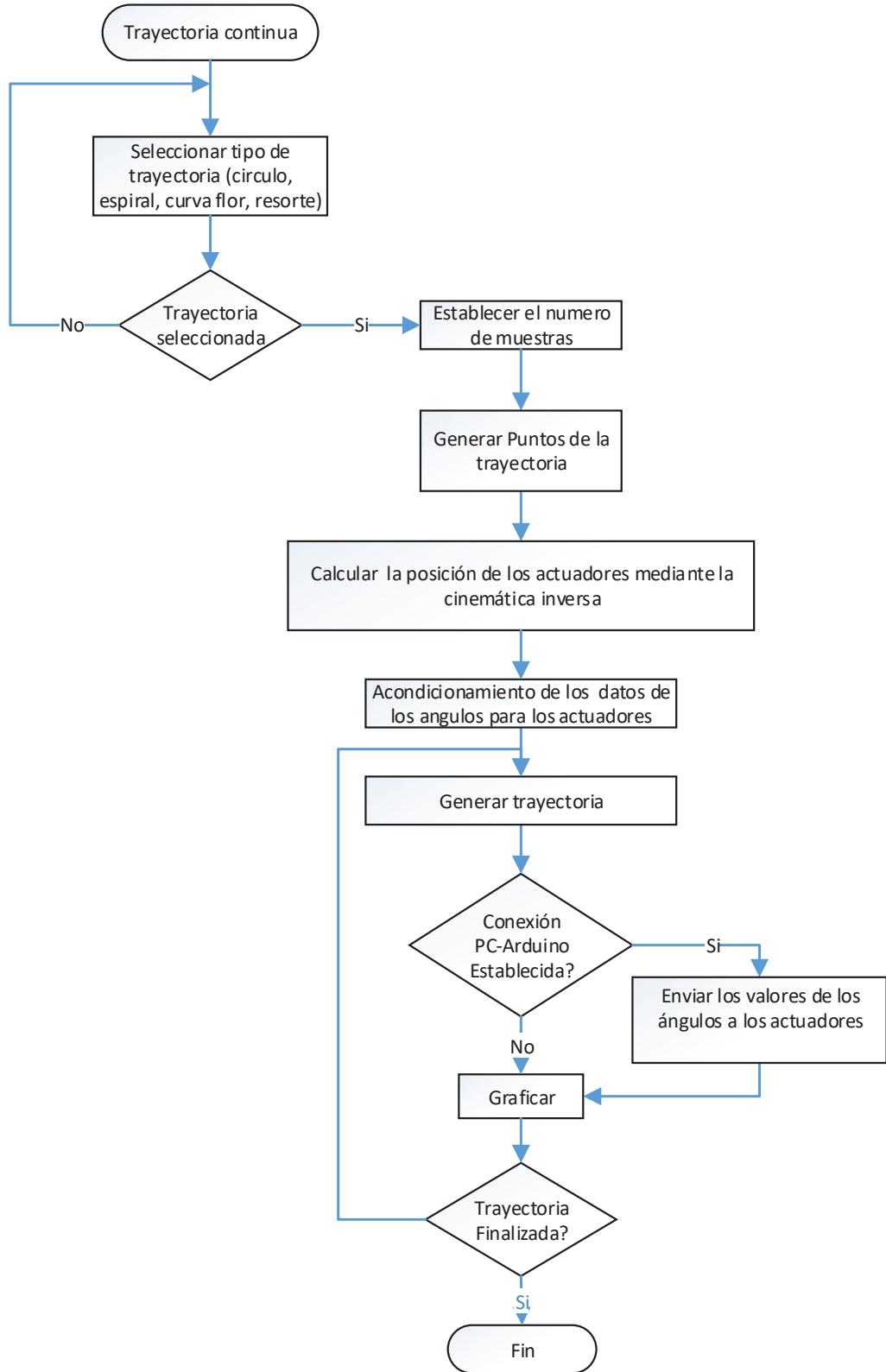
**Figura 3.6.** Curva de la resolución del servomotor Dynamixel.

Finalmente los datos calculados en la cinemática inversa serán enviados a la tarjeta Arduino de la siguiente manera:

$$\theta_{1i(\text{enviado})} = (150^\circ - \theta_{1i(\text{calculado})}) * 3.4483 \quad (3.2)$$

### 3.1.2.3. Ejecución de trayectorias continuas.

Una forma básica de especificar el movimiento que debe ejecutar el robot es especificando el camino que debe unir los puntos mediante una determinada trayectoria, tal como una línea recta o un círculo. En este caso, el sistema de control debe hacer que el robot reproduzca lo más fielmente posible la trayectoria especificada. [1]



**Figura 3.7.** Diagrama de flujo para ejecución de trayectorias continuas

### 3.1.2.4. Ejecución de trayectorias punto a punto. [1]

El control punto a punto se aplica cuando la posición inicial y la posición final están suficientemente separados, caso contrario las especificaciones serían muy laboriosas.

Con el fin de tener un control de movimiento se debe realizar una interpolación entre los puntos especificados, para lo cual se emplea una definición paramétrica de la curva que permita contemplar fácilmente la generación de movimientos tanto en el espacio de trabajo como en el espacio de las variables articulares.

Para este trabajo con el fin de que los movimientos sean suaves se utiliza un polinomio de quinto grado.

$$P(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \quad (3.3)$$

Obteniendo la primera derivada

$$\dot{P}(t) = a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4 \quad (3.4)$$

Para la segunda derivada

$$\ddot{P}(t) = 2a_2 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3 \quad (3.5)$$

Para  $t = 0$  se tiene:

$$P(0) = a_0 \quad (3.6)$$

$$\dot{P}(0) = a_1 \quad (3.7)$$

$$\ddot{P}(0) = 2a_2 \quad (3.8)$$

Considerando que la primera y segunda derivada representan la velocidad y aceleración respectivamente, para  $t = 0$ , se tiene que:

$$a_1 = 0 \quad (3.9)$$

$$2a_2 = 0 \rightarrow a_2 = 0 \quad (3.10)$$

Ahora considerando  $t = t_f$ , donde  $t_f$  es el tiempo final del recorrido, se tiene:

$$P(t_f) = a_3 t_f^3 + a_4 t_f^4 + a_5 t_f^5 = 0 \quad (3.11)$$

$$\dot{P}(t_f) = 3a_3 t_f^2 + 4a_4 t_f^3 + 5a_5 t_f^4 = 0 \quad (3.12)$$

$$\ddot{P}(t_f) = 6a_3 t_f + 12a_4 t_f^2 + 20a_5 t_f^3 = 0 \quad (3.13)$$

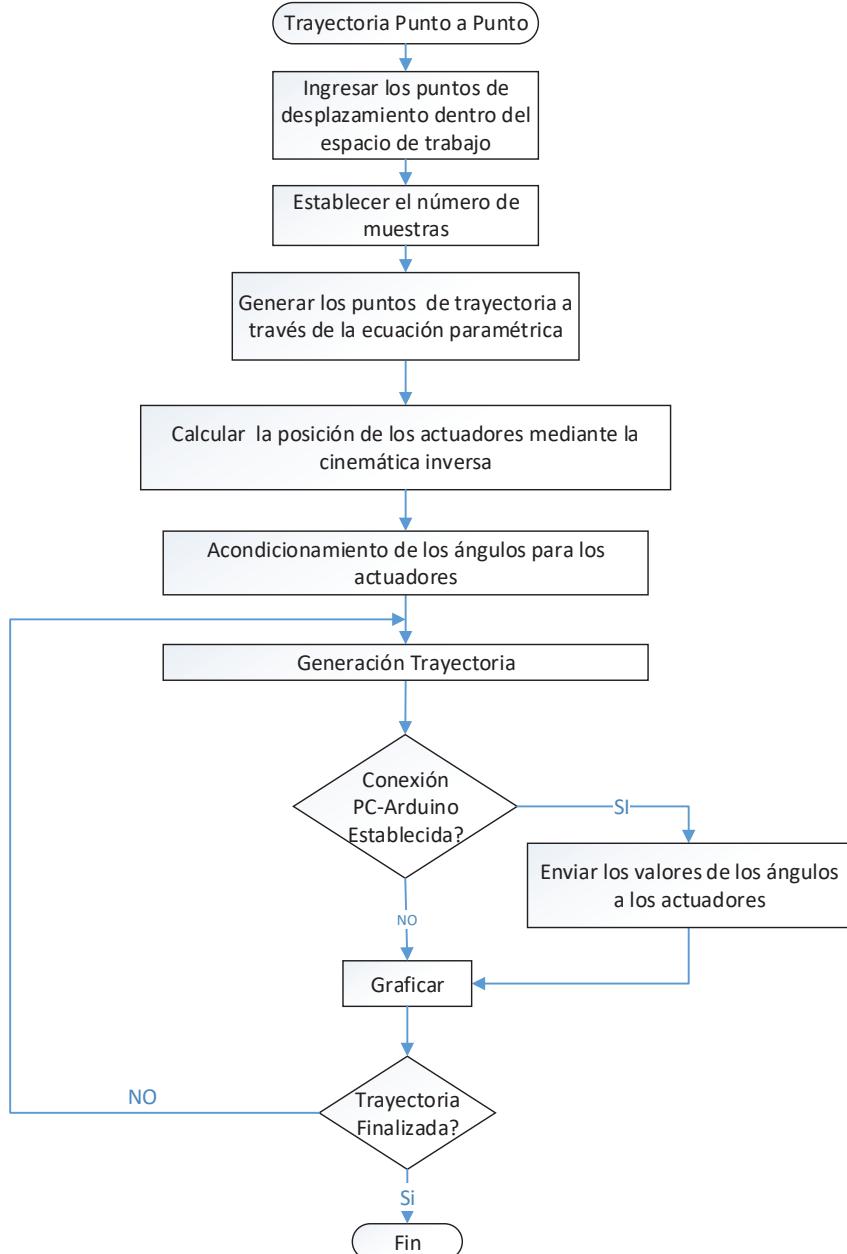
Resolviendo el sistema de ecuaciones se llega a:

$$a_3 = \frac{10}{t_f^3}; \quad a_4 = \frac{15}{t_f^4}; \quad a_5 = \frac{6}{t_f^5} \quad (3.14)$$

Finalmente se obtiene:

$$P(t) = 10 \left( \frac{t}{t_f} \right)^3 + 15 \left( \frac{t}{t_f} \right)^4 + 6 \left( \frac{t}{t_f} \right)^5 \quad (3.15)$$

En la Figura 3.8 se tiene el diagrama de flujo aplicando el control punto a punto, utilizando el polinomio de quinto grado para que el manipulador siga una trayectoria suave.



**Figura 3.8.** Diagrama de flujo para ejecución de trayectorias punto a punto

### 3.1.2.5. Adquisición de puntos mediante visión artificial

Para localizar objetos dentro del espacio de trabajo se hace uso de la visión artificial llevado a cabo con una cámara de computadora portátil HP-G62.

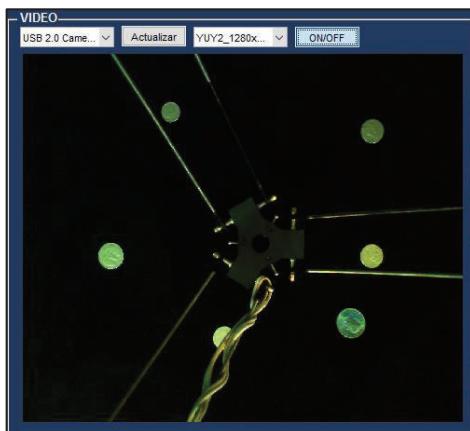


**Figura 3.9.** Cámara VGA de una laptop HP-G62.

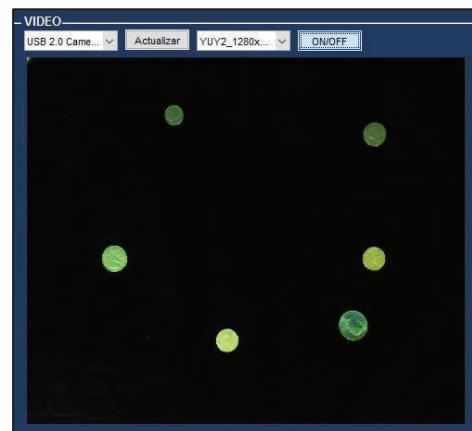
Este tipo de cámaras tienen un gran espacio de enfoque que satisfacen los requerimientos del espacio de trabajo del prototipo.

Para adquirir una imagen y poder procesarla primero se debe detectar la cámara desde MATLAB, posteriormente se escoge la resolución con la que se va a trabajar; la resolución es el número de pixeles que pueden ser mostrados en la pantalla, siendo el pixel la menor unidad en color que forma parte de una imagen digital.

La imagen se captura posicionando los actuadores en una ubicación específica, en donde los eslabones no aparecen en la imagen y de esta manera capturar únicamente los objetos en el espacio de trabajo. (Figura 3.10)



(a)



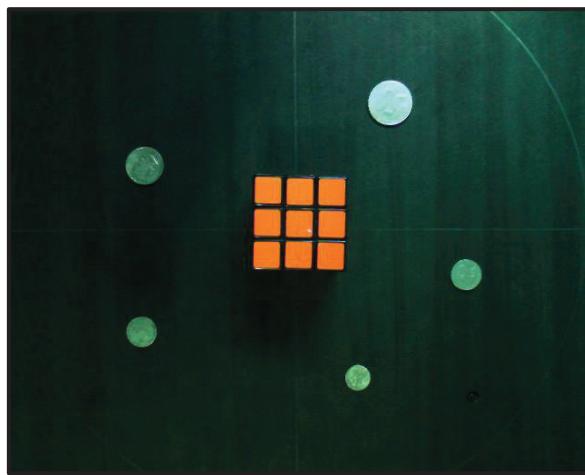
(b)

**Figura 3.10.** Enfoque de la cámara (a) Efecto final sin posicionarse, (b) Efecto final posicionado.

Una vez capturada la imagen se procede a convertirla a escala de grises para poder procesarla, es importante que la diferencia de los colores de los objetos con el fondo sean totalmente opuestos para no tener muchos problemas con la luz.

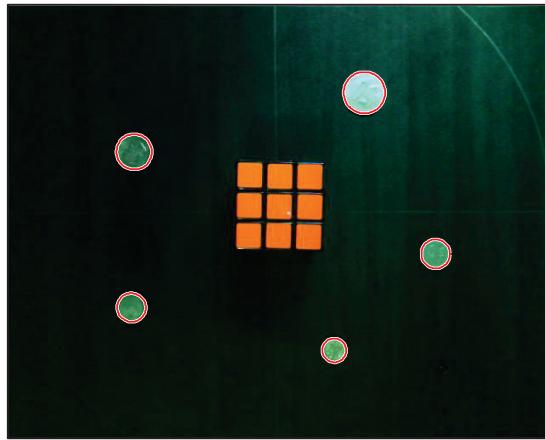
Con la imagen en escala de grises se obtiene su dimensión que será una matriz de orden  $m \times n$  que será utilizada posteriormente. Se utiliza las herramientas de procesamiento de imagen que posee MATLAB, específicamente la medición de las características de una imagen que permiten detectar figuras geométricas y distancias.

El primer paso para utilizar esta herramienta es establecer un radio de búsqueda dentro de la imagen capturada, se puede configurar la función para encontrar figuras brillantes o figuras oscuras dependiendo del fondo de la imagen, además cuenta con un valor de sensibilidad de búsqueda que oscila entre 0 y 1, finalmente se incluye dentro de la función un rango donde estarán las medidas de la figura, en este caso en particular se buscan figuras circulares (Figura 3.11) )y el resultado que se obtiene luego de analizar toda la imagen son dos matrices que contienen los centros y los radios de los círculos encontrados.



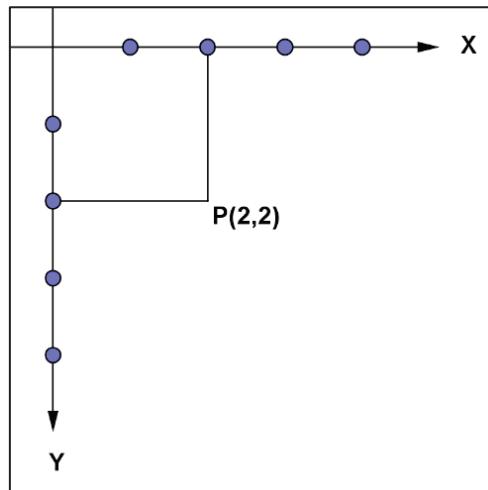
**Figura 3.11.** Imagen previa a la detección de círculos.

En la Figura 3.12 se muestra el resultado del procesamiento de la imagen de la Figura 3.11, la cual consiste en encontrar figuras circulares que las enmarca en color rojo.



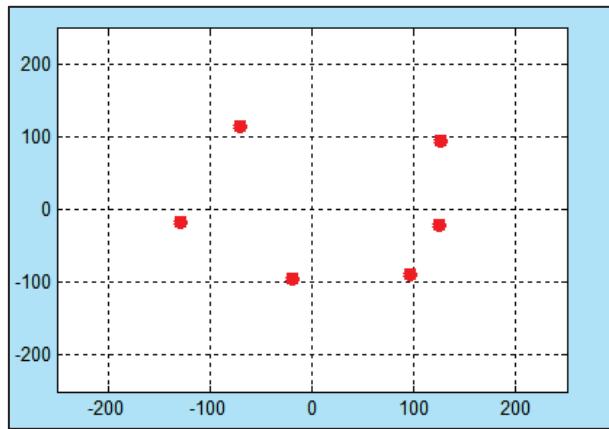
**Figura 3.12.** Imagen que muestra figuras circulares encontradas

Con los objetos que se quiere localizar bien identificados se calcula la posición de cada uno dentro de la imagen, se debe tener en cuenta que la posición que se calcula es en base a pixeles, donde la posición del eje x(ancho) va de izquierda a derecha, pero el eje y(altura) va creciendo hacia abajo, Figura 3.13.



**Figura 3.13.** Representación de un pixel

Para representar un punto de una imagen capturada en el espacio de trabajo del prototipo es necesario hacer una trasformación de pixeles a milímetros y de coordenadas de pixeles a coordenadas cartesianas para finalmente obtener el resultado de la Figura 3.14.

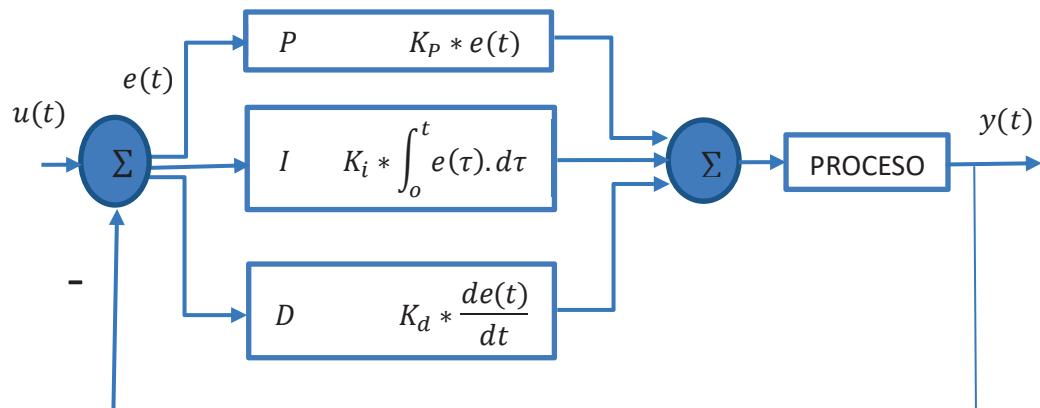


**Figura 3.14.** Representación de los puntos localizados en el plano cartesiano.

### 3.1.2.6. Controlador proporcional, integral, derivativo (PID). [26]

Este tipo de control es el más utilizado ya que en la mayoría de los casos suele dar buenos resultados. Para utilizar este controlador se parte de una base que es el error entre el valor deseado y el valor medido, este último debe ser lo más preciso posible para que el resultado de la variable controlada sea mejor.

En la Figura 3.15 se puede apreciar el diagrama de bloques de un controlador PID que indica que es un sistema en lazo cerrado donde la variable medida  $y(t)$  es realimentada y restada del punto de referencia  $u(t)$ , para obtener el error  $e(t)$  luego se evalúa en los bloques PID y de esta manera se obtiene una salida que será la encargada de controlar el proceso que se esté realizando.



**Figura 3.15.** Diagrama de bloques controlador PID.

Los bloques representados con las letras PID corresponden a la acción proporcional, integral y derivativa que basan su comportamiento de acuerdo al valor de las constantes  $K_p$ ,  $K_i$  y  $K_d$ , estas acciones se detallan a continuación:

- Acción proporcional (P): genera una salida directamente proporcional al error existente, es decir responde rápidamente cuando el error es grande, pero esto puede ocasionar que la señal se pase del punto de referencia teniendo oscilaciones.
- Acción integral (I): esta acción permite corregir el error en estado estable.
- Acción derivativa (D): esta acción estabiliza el proceso de una manera rápida teniendo un aspecto predictivo.

En la mayoría de los casos la acción derivativa no es usada y es puesta en off, esta acción es más usada cuando las constantes de tiempo difieren en magnitud como por ejemplo el control de temperatura o para tener un control más fino en un sistema de alto orden.

El controlador en este proyecto es aplicado independientemente a cada actuador con el fin de evitar las oscilaciones producidas cuando el efecto final alcanza su posición final, en este contexto la variable a controlar es la posición angular del actuador, esta posición es una variable de 16Bits que es enviada por el motor cuando la tarjeta Arduino la requiera, por lo tanto el control se realiza en la tarjeta y es necesario tener un sistema de control discreto, es por eso que se usa la versión digital del PID. [26]

$$U_{[n]} = U_{[n-1]} + E_{[n]} * \frac{2K_p T_m + 2K_d + K_i T_m^2}{2T_m} + E_{[n-1]} * \frac{K_i T_m^2 - 2K_p T_m - 4K_d}{2T_m} + E_{[n-2]} * \frac{K_d}{T_m} \quad (3.16)$$

Donde:

$U$  = señal de control o punto de referencia.

$E$  = error (señal de control-señal realimentada).

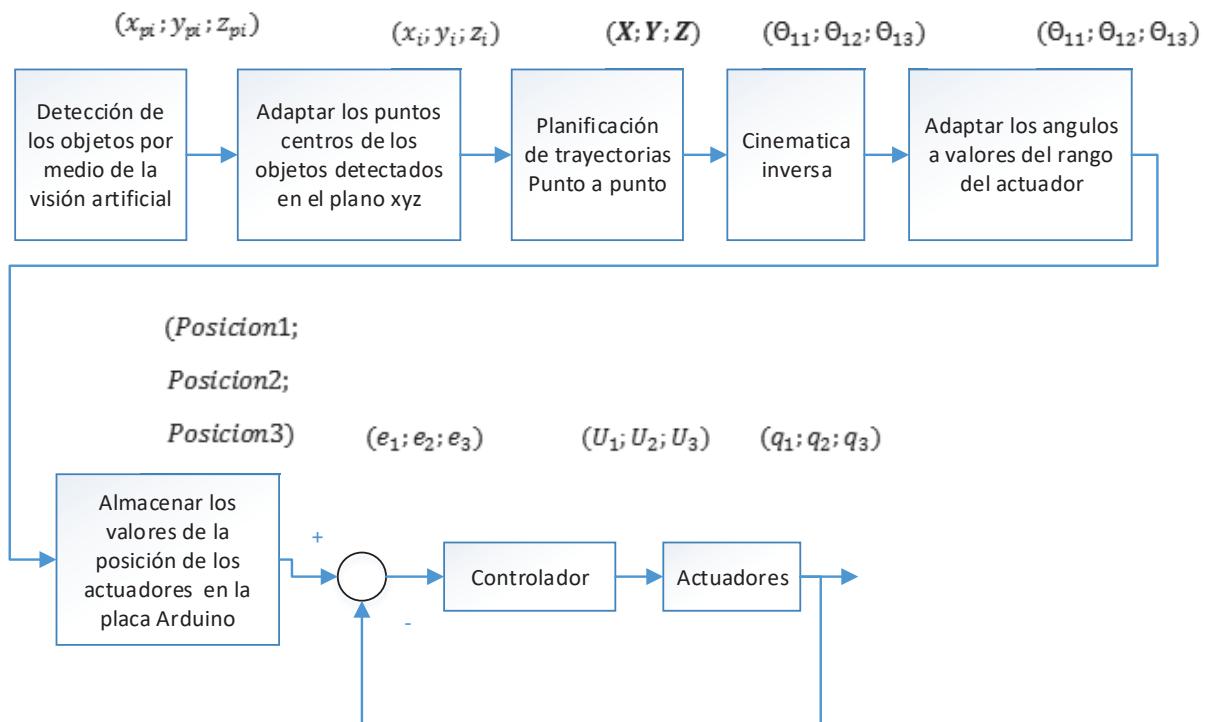
$T_m$  = tiempo de muestreo.

$K_p$  = constante proporcional.

$K_i$  = constante integral.

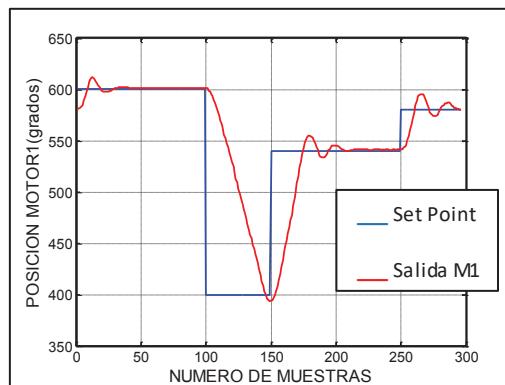
$K_d$  = constante derivativa.

El diagrama de bloques del sistema de control se muestra en la Figura 3.16.

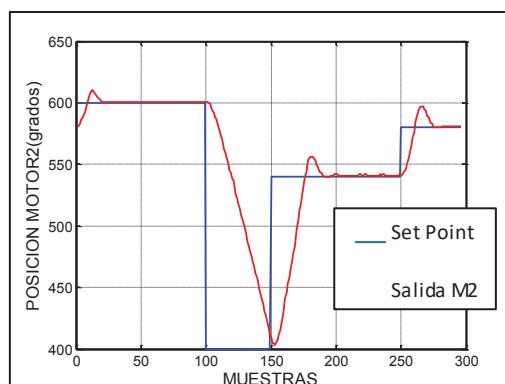


**Figura 3.16.** Diagrama de bloques del sistema de control para los actuadores.

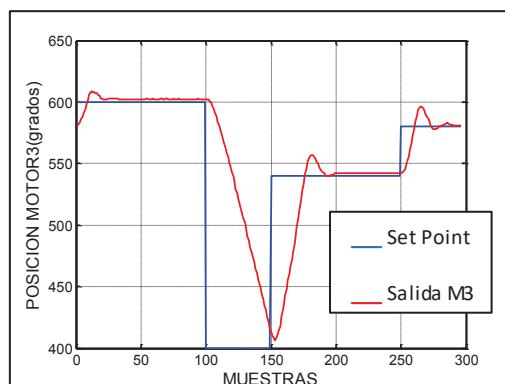
El procedimiento para calibrar el controlador PID comienza con la manipulación de la constante proporcional hasta obtener una respuesta rápida pero sin que existan oscilaciones, posterior a este paso se manipula la constante derivativa hasta su máximo valor sin generar oscilaciones y por último se manipula el controlador con la constante integral hasta que el error en estado estable sea lo más pequeño posible. A continuación en las Figuras 3.17, 3.18 y 3.19, se muestra la respuesta de los tres actuadores sin acción de control.



**Figura 3.17.** Respuesta actuador 1 sin acción de control.



**Figura 3.18.** Respuesta actuador 2 sin acción de control.



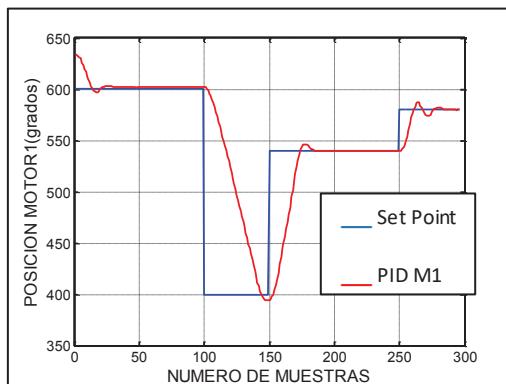
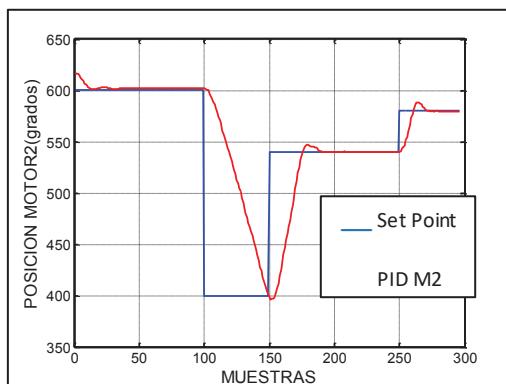
**Figura 3.19.** Respuesta actuador 3 sin acción de control.

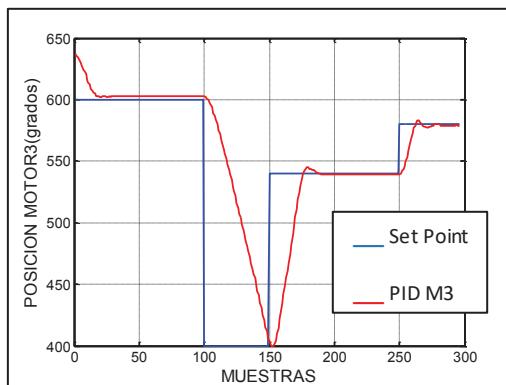
La Tabla 3.1 muestra los valores de las constantes  $K_p$ ,  $K_i$  y  $K_d$  que se aplican al controlador de los actuadores y que a su vez permiten un movimiento más suave del efecto final.

**Tabla 3.1.** Constantes de los controladores PID del sistema.

PID actuadores	$K_p$	$K_i$	$K_d$
Actuador 1	0.58	0.05	0
Actuador 2	0.68	0.07	0
Actuador 3	0.58	0.09	0

Los datos obtenidos en la Tabla 3.1 muestran que la acción de control únicamente será un proporcional-integral (PI), las respuestas que se obtienen con estos datos se observan en las Figuras 3.20, 3.21 y 3.22, donde se aprecia que las oscilaciones disminuyen lo que garantizara que el movimiento del efecto final sea suave.

**Figura 3.20.** Respuesta del controlador en el actuador 1.**Figura 3.21.** Respuesta del controlador en el actuador 2.



**Figura 3.22.** Respuesta del controlador en el actuador 3.

### 3.2. DISEÑO DE LA INTERFAZ

La creación de una interfaz gráfica permite la interacción entre el usuario y el programa que controla el mecanismo. Una interfaz gráfica de usuario (GUI) contiene diferentes elementos gráficos tales como: botones, campos de texto, menús, gráficos, etc.

Una de las herramientas con las que cuenta MATLAB, es la creación de GUIs. La forma de implementar las GUI, es crear los objetos y definir las acciones que cada uno va a realizar, al usar GUIDE de MATLAB se crean dos archivos:

- Un archivo .fig donde se describen los componentes gráficos que contiene la interfaz.
- Un archivo .m que contiene las funciones y los controles del GUI así como el callback.

Un callback se define como la acción que llevará a cabo un objeto de la GUI cuando el usuario la active.

Con el objetivo, que la interfaz sea lo más interactiva se ha añadido indicadores de funcionamiento, se activado mensajes de alerta, indicadores de valores de los ángulos de cada articulación y además un sistema de controles para la configuración de

funcionamiento y control del mecanismo. A continuación se describe cada una de las ventanas creadas.

### 3.2.1. VENTANA ESPACIO DE TRABAJO

El objetivo de esta ventana es mostrar los cambios que sufre el espacio de trabajo al ir variando las dimensiones de los diferentes elementos del robot delta a continuación se describe la función de cada panel de la Figura 3.23

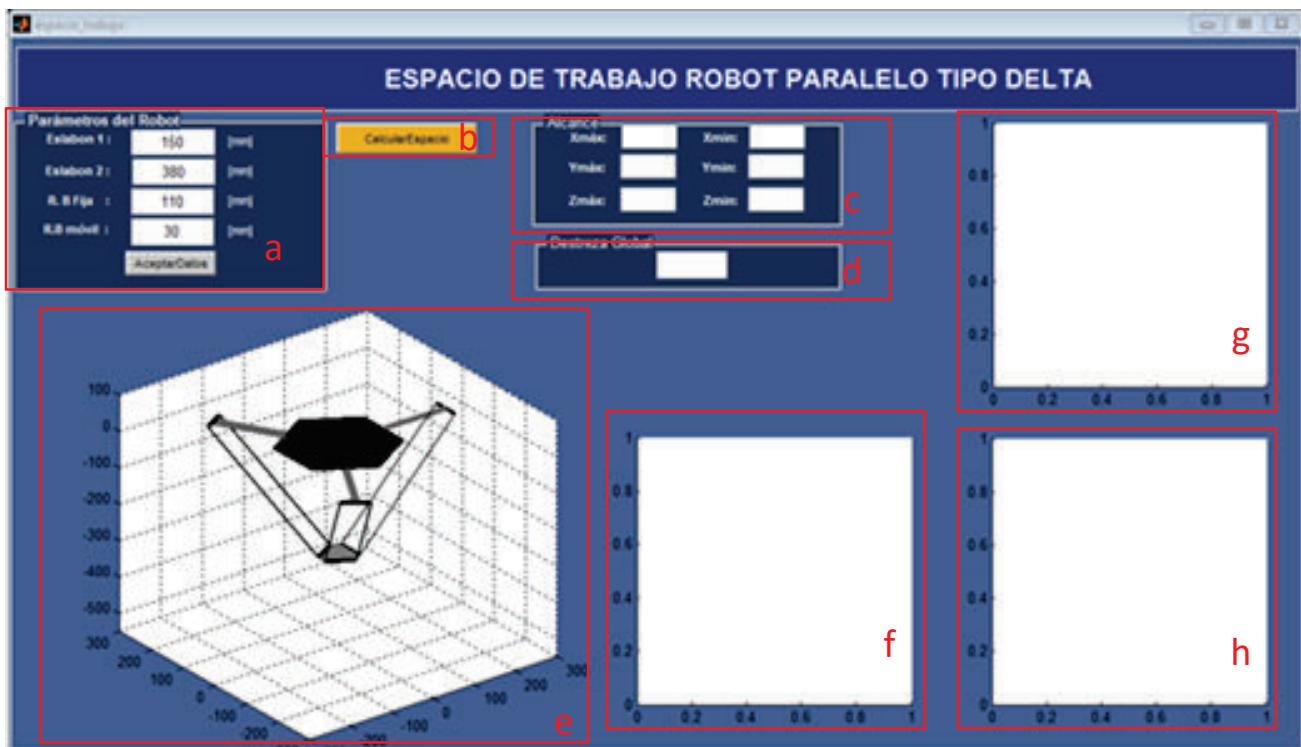


Figura 3.23. Ventana del Espacio de trabajo del robot delta

En el bloque (a) se encuentran los texto que permiten ir modificando los valores del eslabón  $L_1$  y  $L_2$ , el radio de la base fija y el radio de la base móvil; la variación se debe realizar teniendo en cuenta las condiciones establecidas por las ecuaciones (2.6), (2.7), (2.8). Para que los datos ingresados sean válidos se debe presionar el pulsador “Aceptar Datos” e inmediatamente la gráfica del bloque (e) cambia de acuerdo a los valores ingresados.

El bloque (b) contiene el botón “Calcular Espacio”, que permite ejecutar el algoritmo que determina el máximo y mínimo alcance cuyos valores se mostraran en los textos del bloque (c), la destreza global se mostrara en el bloque (d) y en los bloques (f), (g) y (h) se va mostrando las gráficas del espacio de trabajo del robot en los diferentes planos

### 3.2.2. VENTANA DE LA SECCIÓN DEMOSTRATIVA

La ventana de la Figura 3.24, tiene como objetivo mostrar al usuario el funcionamiento del prototipo ejecutando trayectorias continuas tales como un círculo, una espiral, curvas que permiten dibujar una flor dentro del plano xy. Además se pueden realizar movimientos en los ejes xyz, activar y desactivar el efecto final presionando los botones del teclado de la PC, cumpliendo con el objetivo de recoger y apilar los objetos de forma manual.



Figura 3.24. Ventana de la sección demostrativa

Esta ventana contiene los siguientes paneles de control:

a) Comunicación.

Esta sección permite el enlace de interacción entre la interfaz y el prototipo a través del puerto en el que se encuentre conectado el módulo TTL-USB, que se mostrara en el selector de objetos.

En los bloques de texto se mostrará el nombre del puerto seleccionado, así como la velocidad de comunicación. El botón “Conectar” habilita y deshabilita la comunicación, mientras el botón “ON” permite encender y apagar la fuente de corriente continua.

Para que la ventana sea más interactiva se han colocado cuadros de texto que muestran los valores enviados a los actuadores (motor1, motor2, motor3).

b) Manejo de las Coordenadas del efecto final.

Esta ventana permite mover el prototipo, según se presionen los botones del teclado que controla cada uno de los ejes. También se tiene un botón llamado “D Efecto Final (ON)” el cual activa y desactiva el efecto final lo que permite al mecanismo coger y soltar objetos para ubicarlos en el espacio de trabajo según el criterio del usuario.

c) Trayectorias continuas.

Este panel contiene las diferentes trayectorias que puede escoger el usuario para que puedan ser ejecutadas por el prototipo. Dependiendo de la trayectoria elegida se puede modificar las siguientes características: radio, altura de ejecución en el eje z y el número de repeticiones.

El botón “Iniciar Simulación”, permite enviar los datos de los ángulos para posicionar los actuadores. Además se tiene un botón que permite realizar la ejecución de trayectorias con o sin la simulación. Con el propósito que el usuario pueda ver los resultados de la trayectoria ejecutada se puede ir calibrando el controlador e ir observando su respuesta al presionar los botones:

- “Graficas Coordenadas XYZ”. Con la ayuda de la cinemática inversa se muestra la trayectoria ejecutada por el prototipo.
- “graficar actuadores”. Aquí se muestra la respuesta de cambio de posición angular de los actuadores para cumplir con la trayectoria establecida.

d) Ventana 3D Prototipo.

Dentro de esta ventana se añadido un axes donde se pueden observar la simulación de los movimiento realizados por el prototipo con la gráfica del robot.

e) Cámara.

Aquí se puede activar la cámara instalada en el sistema y a través de ella ir observando los movimientos ejecutados por el prototipo.

### **3.2.3. VENTANA DE TAREAS.**

La ventana de tareas muestra una interfaz que permite hacer uso de la visión artificial para procesar una imagen y obtener la posición de ciertos objetos en el espacio de trabajo.

La localización de objetos tiene como finalidad automatizar la colocación del efecto final para realizar tareas de apilamiento y ordenamiento de objetos, de esta manera no es necesario colocar los puntos manualmente para demostrar los objetivos planteados en este proyecto. En la Figura 3.17 se muestra la ventana de tareas con los diferentes paneles que se detallaran a continuación.



**Figura 3.25.** Ventana de tareas.

Comenzando de izquierda a derecha se tiene el bloque (a) donde está el panel de comunicación que funciona de manera similar al de la ventana de la sección demostrativa, establece la comunicación entre la PC y la tarjeta Arduino, además muestra cuadros de texto donde se visualiza los datos enviados a la tarjeta de enlace.

El bloque (b) permite escoger la cámara de video que está instalada en el prototipo, así como su resolución con el botón “ON/OFF” que activa o desactiva la cámara.

Dentro del bloque (c) se tiene los botones para capturar las imágenes enfocadas por la cámara, así como los botones que permite seleccionar el modo de manejo manual y automático.

Finalmente se tiene el bloque (d), este panel contiene un botón que permite ubicar los objetos detectados en el plano cartesiano, para su posterior operación.

Las operaciones a realizarse en esta sección son apilamiento de los objetos detectados en una ubicación determinada por el usuario y la operación de ordenamiento que va recogiendo los objetos detectados y los pondrá en un arreglo geométrico como línea recta o en forma de cuadrado.

## CAPITULO 4

### PRUEBAS Y RESULTADOS

En el desarrollo de este capítulo se muestra el funcionamiento del prototipo, verificando paso a paso las etapas que permitan cumplir con los objetivos y alcances planteados para este proyecto. Se realizan las pruebas del prototipo conjuntamente con la interfaz diseñada, con el fin de detectar posibles fallas en ambos y de esta manera encontrar las mejores soluciones para que el sistema funcione correctamente.

#### 4.1. PRUEBAS ELECTRÓNICAS.

Este tipo de pruebas permitirán que posteriormente el sistema opere sin ninguna novedad, para esto se plantea realizar pruebas de comunicación entre los diferentes bloques, además se verifica las conexiones de todo el sistema tanto de alimentación como de comunicación para descartar problemas de cableado.

El gabinete que contiene las tarjetas electrónicas y la fuente de alimentación debe estar conectado a la red eléctrica 110 VAC; para que el sistema sea encendido se ha colocado un interruptor en la parte frontal que deberá ser presionado, una luz indicadora de color roja se encenderá señalando que el sistema está activo, también se debe verificar la conexión del cable USB, en el gabinete y en un puerto del computador.

#### 4.2. FUNCIONAMIENTO DE LA CINEMÁTICA DEL ROBOT

##### 4.2.1. LIMITES DE MOVIMIENTO DE LA ESTRUCTURA

En esta sección se pone en funcionamiento al prototipo con el fin de obtener los alcances máximos y mínimos que tienen conjuntamente los ángulos de los actuadores

y los ejes coordenados, todo esto con el objetivo de comprobar el espacio de trabajo analizado en el capítulo 2.

#### 4.2.1.1. Posición de los actuadores y alcance en los ejes coordinados.

Empleando la interfaz denominada sección demostrativa se hace uso del panel de control manual el cual contiene pulsadores que permiten mover el prototipo de manera discreta en los ejes xyz, estas coordenadas son tratadas por la cinemática inversa que calcula el valor de los ángulos que debe tener cada actuador para adquirir la posición deseada, estos valores se envían al prototipo y se visualizan en el panel de comunicación. Primeramente se coloca un transportador en el eje de los actuadores para observar el ángulo en el que está ubicado el eslabón  $L_1$ , con la ayuda de la interfaz diseñada anteriormente se elabora la Tabla 4.1 donde se muestran los valores teóricos y medidos en el eje z y las medidas de los ángulos de los actuadores.

**Tabla 4.1.** Comparación de los valores teóricos y reales de los actuadores.

Valores eje Z mm		Posición de los actuadores[grados sexagesimales]					
Zmedido	Zteórico	MOTOR 1		MOTOR 2		MOTOR 3	
		Medido	Teórico	Medido	Teórico	Medido	Teórico
-187	-185	51	56	47	56	47	56
-204	-195	37	41	32	41	32	41
-219	-205	28	32	22	32	23	32
-232	-215	20	26	18	26	17	26
-244	-225	16	20	12	20	12	20
-259	-235	11	16	8	16	8	16
-270	-245	8	11	4	11	4	11
-282	-255	2	8	0	8	0	8
-294	-265	0	4	-5	4	-5	4
-307	-275	-3	-1	-8	-1	-7	-1
-320	-285	-8	-2	-11	-2	-11	-2
-332	-295	-10	-5	-15	-5	-14	-5
-340	-305	-13	-8	-18	-8	-17	-8
-352	-315	-17	-11	-20	-11	-20	-11
-360	-325	-20	-14	-22	-14	-22	-14
-372	-335	-21	-16	-25	-16	-25	-16
-378	-345	-25	-19	-27	-19	-27	-19

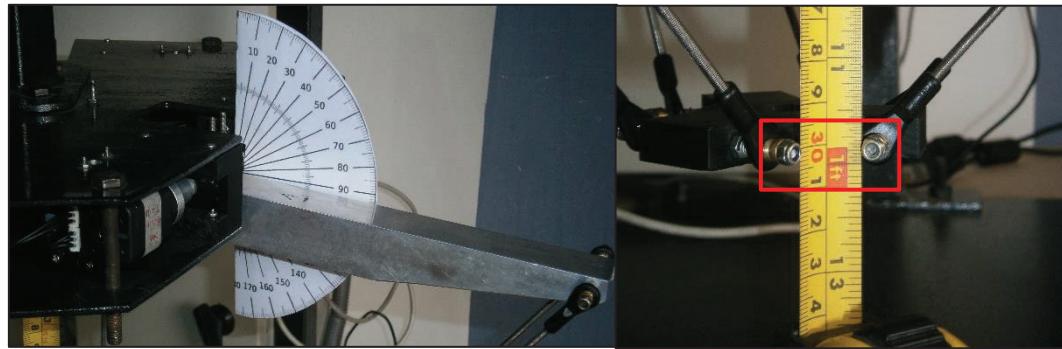
Se comparan los valores reales y los teóricos de la Tabla 4.1 y se observa que existen errores apreciables los cuales se deben a fallas en la construcción mecánica como por ejemplo juego mecánico entre el acople motor-eje-eslabón  $L_1$ , la Tabla 4.2 muestra los errores que se tiene en los motores y se aprecia que el error tiende a ser constante según va variando el eje z.

Este error puede ser corregido mediante software ya que la diferencia entre los valores reales y medidos es aproximadamente la misma.

**Tabla 4.2.** Errores en los actuadores.

Errores[grados sexagesimales]		
MOTOR 1	MOTOR 2	MOTOR 3
5	9	9
4	9	9
4	10	9
6	8	9
4	8	8
5	8	8
3	7	7
6	8	8
4	9	9
2	7	6
6	9	9
5	10	9
5	10	9
6	9	9
6	8	8
5	9	9
6	8	8
<b>P=4.8</b>	<b>P=8.6</b>	<b>P=8.4</b>

Para apreciar la corrección de los errores se muestran en la Figura 4.1 el valor del ángulo del actuador junto con la altura del efecto final, de igual manera se presenta en la Figura 4.2 las mismas medidas pero con el error corregido mediante software.



**Figura 4.1.** Medida del actuador uno y altura del efector final.



**Figura 4.2.** Medida del actuador uno y altura del efector final corregidos por software.

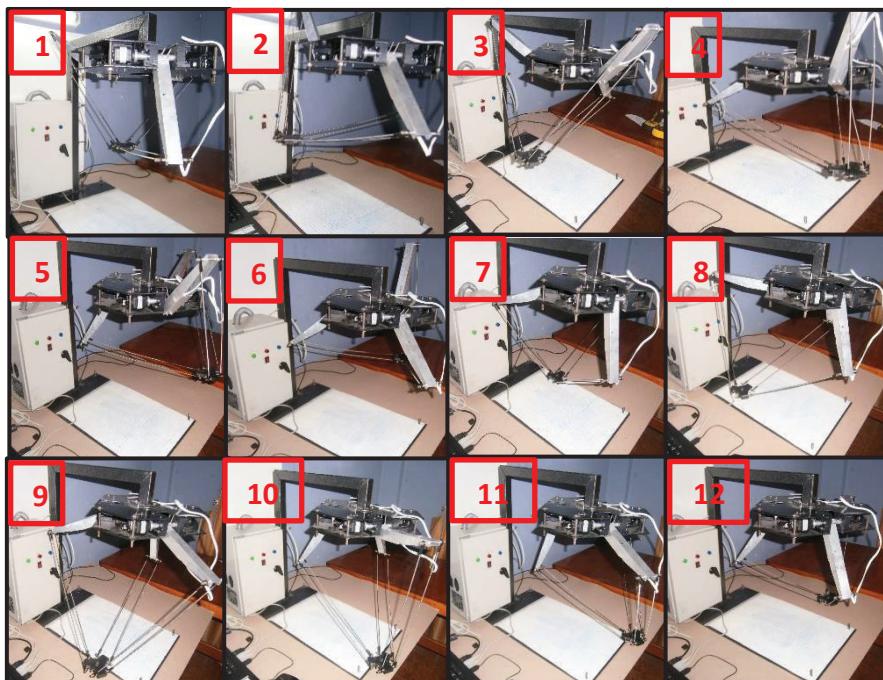
Finalmente cuando el error por factores mecánicos ha sido corregido se procede a encontrar el rango de los ángulos que pueden ser adoptados por los actuadores del prototipo. Estos ángulos dependen de la posición en la que se encuentre el efector final por tal motivo para restringir el movimiento del mecanismo se lo hace a través de la posición del efector final como se muestra en la Tabla 4.3.

**Tabla 4.3.** Representación de los puntos extremos del prototipo.

Posición del efector final			Posición de los actuadores		
Px mm	Py mm	Pz mm	M1 °	M2 °	M3 °
0	0	-190	-47	-47	-47
0	0	-400	-33	-33	-33
210	0	-400	-7	-71	-71
-210	0	-400	-82	-30	-30
0	210	-400	-53	-14	-81
0	-210	-400	-53	-81	-14

#### 4.2.1.2. Verificación del espacio de trabajo

En el capítulo 2 se habla sobre las dimensiones del prototipo y el espacio de trabajo que se requiere para demostrar los objetivos de este proyecto, en esta sección se compara el espacio de trabajo alcanzado por el prototipo de robot delta construido y el que se planteó inicialmente en la Tabla 2.2. (Figura 4.3)



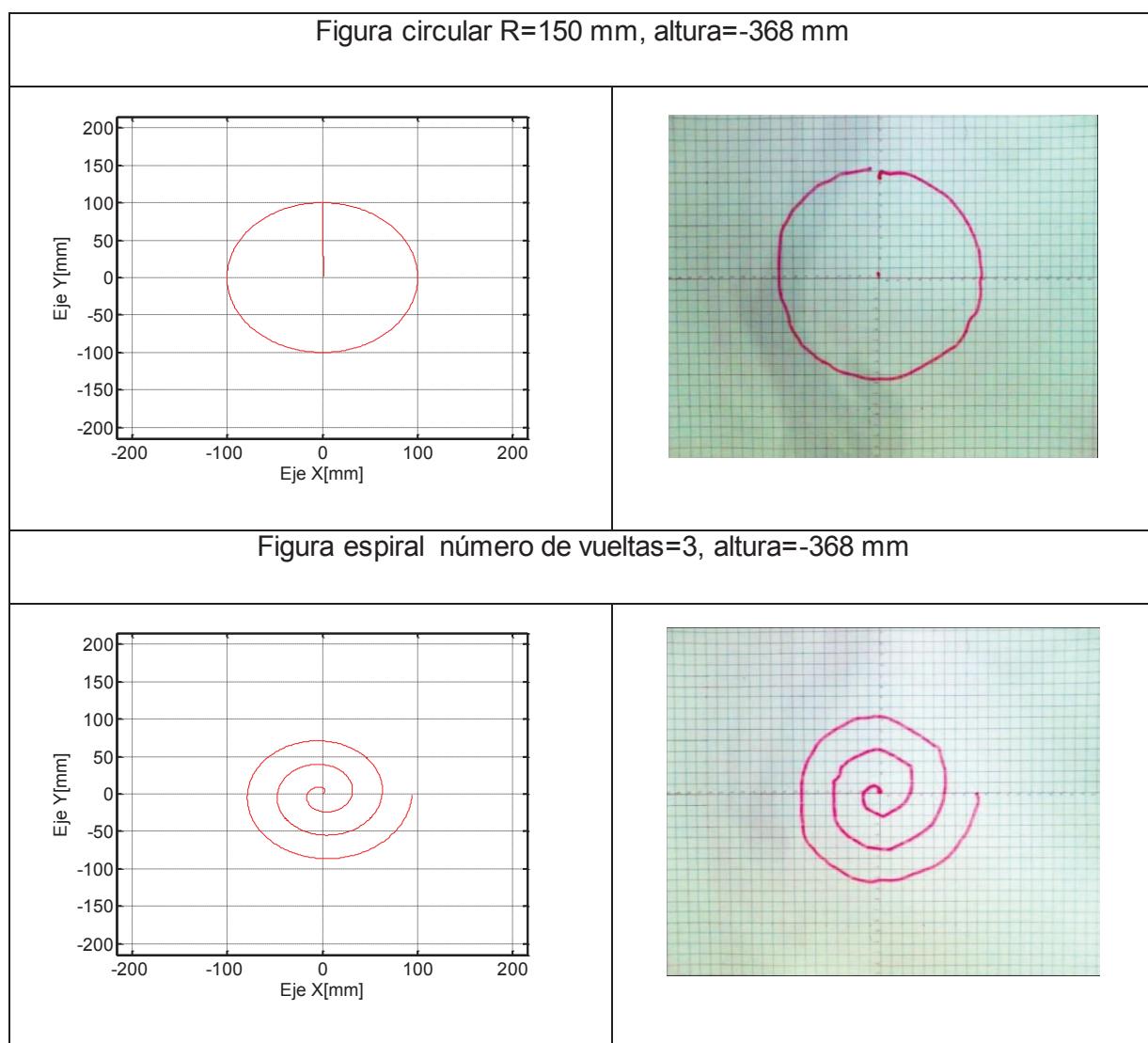
**Figura 4.3.** Prototipo de robot delta recorriendo el espacio de trabajo.

En la Figura 4.3 se observa al prototipo alcanzando todos los puntos del espacio de trabajo establecidos, los puntos en donde el prototipo tiene dificultades es cuando alcanza las posturas de los recuadros 3, 6, 9 y 12 que corresponden a los extremos del espacio. Por lo tanto el robot tendrá un espacio de trabajo igual a un cilindro de 200 mm de radio y de 158 mm de altura.

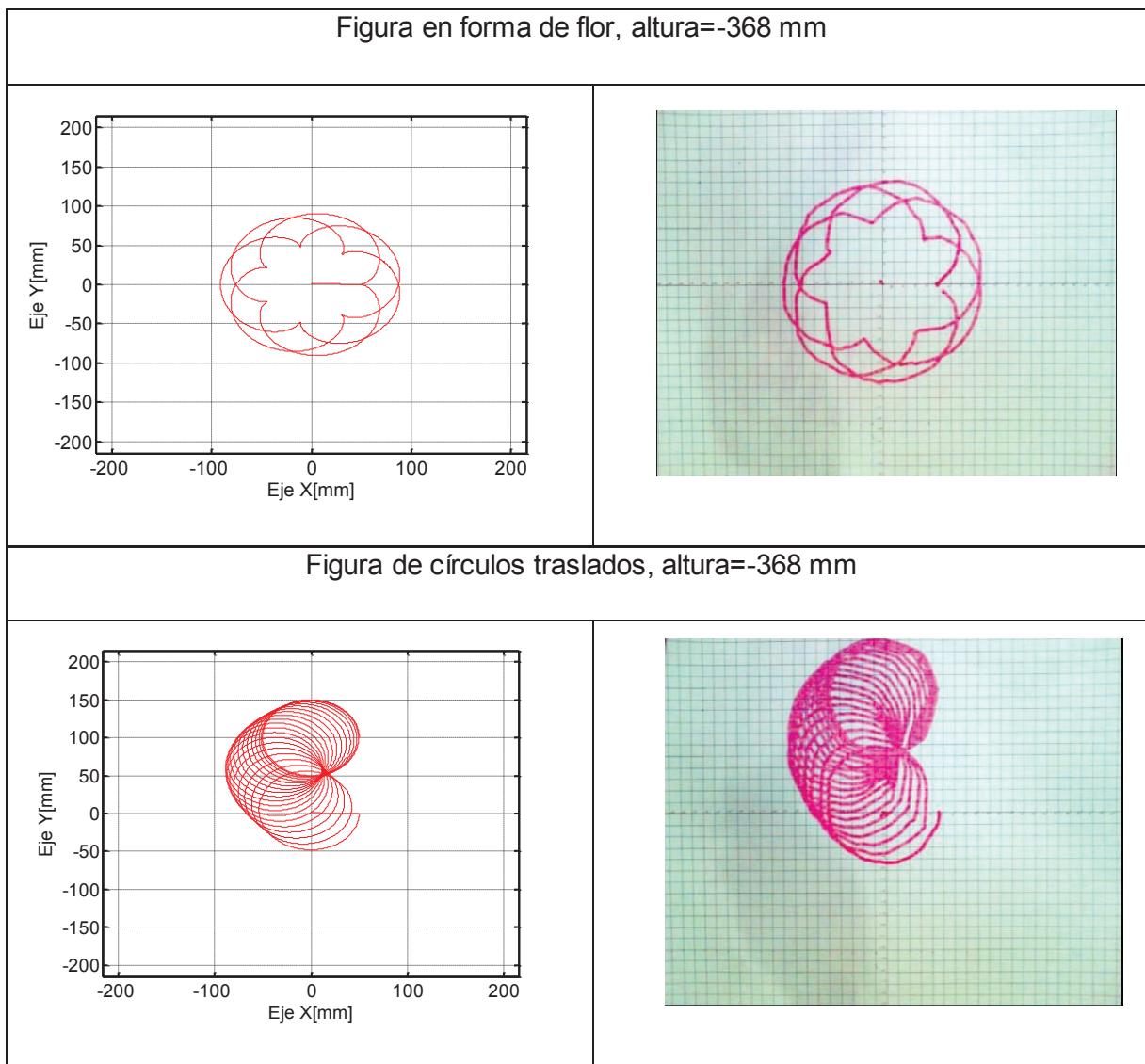
#### 4.2.2. EJECUCIÓN DE TRAYECTORIAS

En este punto con el objetivo de comprobar la cinemática, el robot ejecuta trayectorias continuas como círculos, espirales, etc. Para verificar su funcionamiento se coloca un marcador en la parte móvil que dibuja las trayectorias mencionadas y se observa en la

Figura 4.4., en el lado izquierdo se encuentra la trayectoria deseada y en el lado derecho el rastro dejado por el marcador sobre una cartulina cuadriculada donde el valor de cada cuadrado es de 5 mm x 5 mm. En la interfaz denominada ventana demostrativa se selecciona el tipo de trayectoria que se desea realizar, posteriormente se puede variar características de dicha trayectoria como altura, número de repeticiones y radio en el caso de la circunferencia, cuando se establece todos los parámetros se procede a ejecutar la trayectoria.



**Figura 4.4.** Ejecución de trayectorias en el plano xy



**Figura 4.4.** Ejecución de trayectorias en el plano xy.

### 4.3. PRUEBAS DE TAREAS PICK & PLACE

Esta prueba se lleva a cabo para dar cumplimiento con los alcances del proyecto el cual es desplazar y colocar objetos en un espacio de trabajo establecido. Los métodos para ejecutar las tareas se lo realiza manual y automáticamente, en este último se utiliza visión artificial para detectar los objetos.

### 4.3.1. MODO AUTOMÁTICO

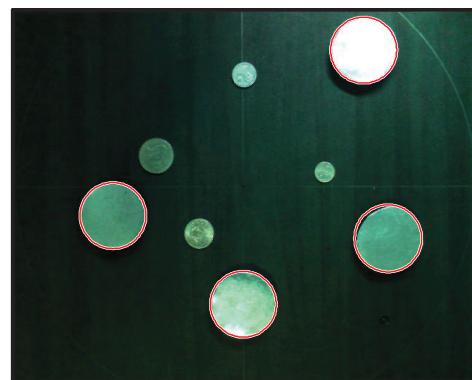
#### 4.3.1.1. Prueba de detección de objetos.

Para la detección de objetos se coloca una cámara en la base fija del prototipo y a través de unos pernos y tuercas que la sujetan se calibra hasta tener un enfoque adecuado del espacio de trabajo, como se muestra en la Figura 4.5. Con el propósito de obtener una imagen limpia del espacio de trabajo los eslabones se colocan de tal manera que la base móvil se ubica en un punto fuera del enfoque de la cámara.



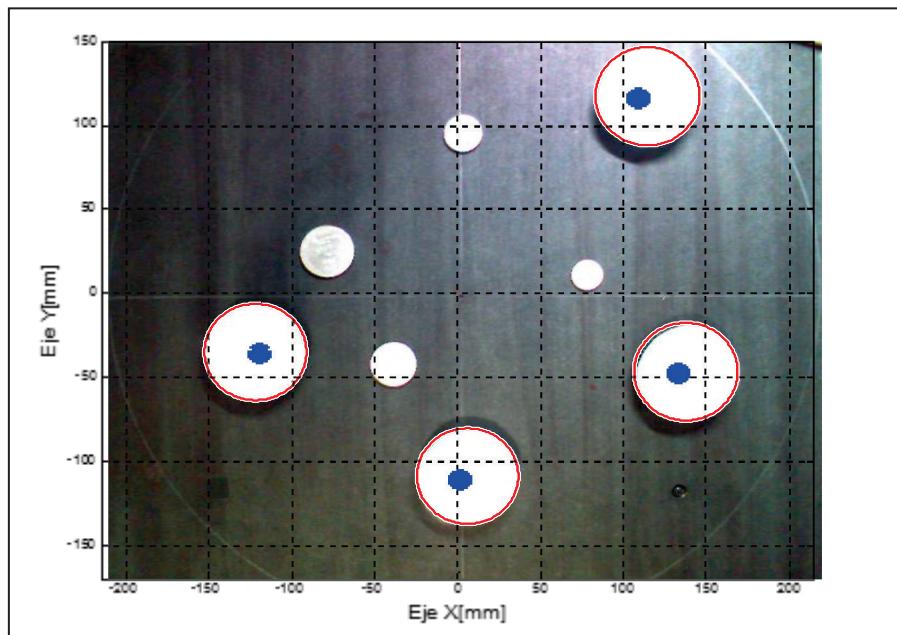
**Figura 4.5.** Enfoque del espacio de trabajo

En el presente proyecto las figuras con las que se va a trabajar son de forma circular con un diámetro de 60 mm y una altura de 10 mm, por lo tanto en la función de MATLAB que procesa imágenes el rango para la búsqueda de círculos se establece entre 55 mm y 65 mm, fuera de este rango los círculos presentes en la imagen no serán tomados en cuenta.



**Figura 4.6.** Detección de círculos con un diámetro determinado.

La Figura 4.6 es el resultado de la operación realizada por MATLAB que además genera matrices que contienen los centro de los círculos y sus diámetros, los valores que se van a utilizar se encuentran en la matriz de centros que vienen dados en coordenadas xy pero en pixeles. El siguiente paso es transformar estos valores a las coordenadas xy del espacio de trabajo que posteriormente serán interpretadas por el prototipo para colocarse sobre cada una de estas y poder realizar las tareas de apilamiento y ordenamiento.



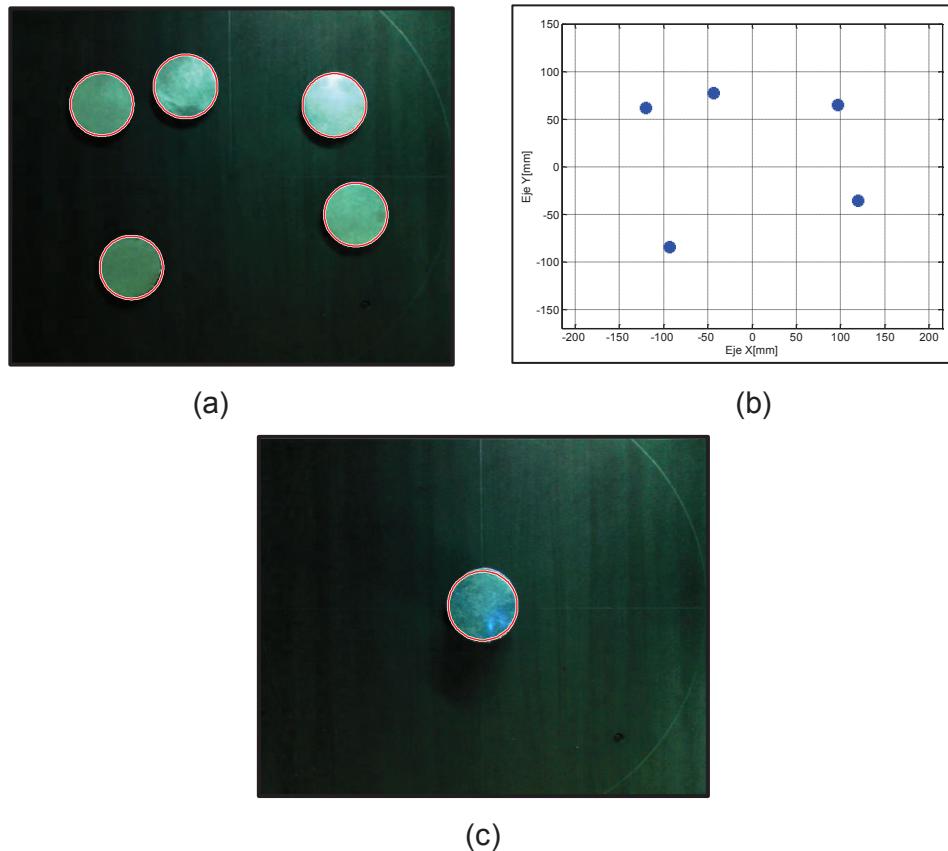
**Figura 4.7.** Ubicación de los centros en el espacio de trabajo

#### 4.3.1.2. Ejecución de tareas

##### 4.3.1.2.1. Apilamiento

Una de las tareas más comunes que se pueden ejecutar con un manipulador, es ir montando un objeto sobre otro de las mismas características dimensionales dentro de un punto del espacio de trabajo definido. Para llevar a cabo este proceso de manera automática el usuario desde la interfaz procederá a detectar y ubicar los objetos con la ayuda de la cámara y podrá ingresar el punto de consigna de ubicación de los

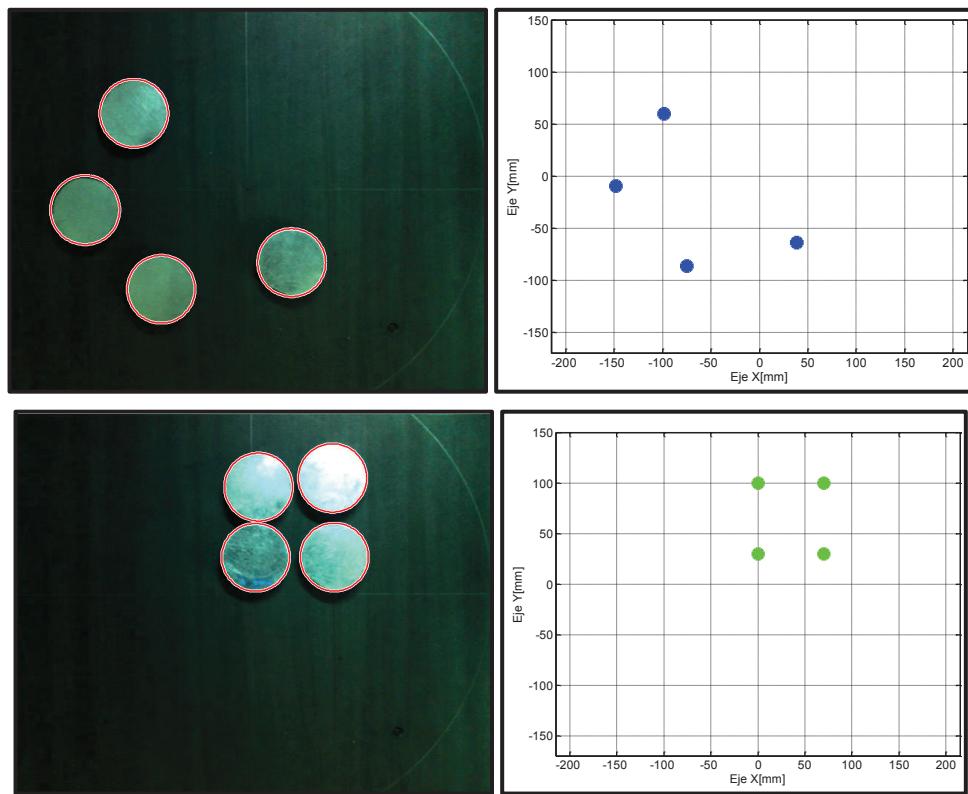
objetos. En la Figura 4.8 se muestra los pasos y la ejecución del prototipo para ubicar a los objetos en el punto central del área de trabajo.



**Figura 4.8.** (a) Objetos detectados, (b) Ubicación de los objetos en el plano cartesiano, (c) Objetos apilados por el prototipo en el punto (0,0) mm, visto desde la cámara.

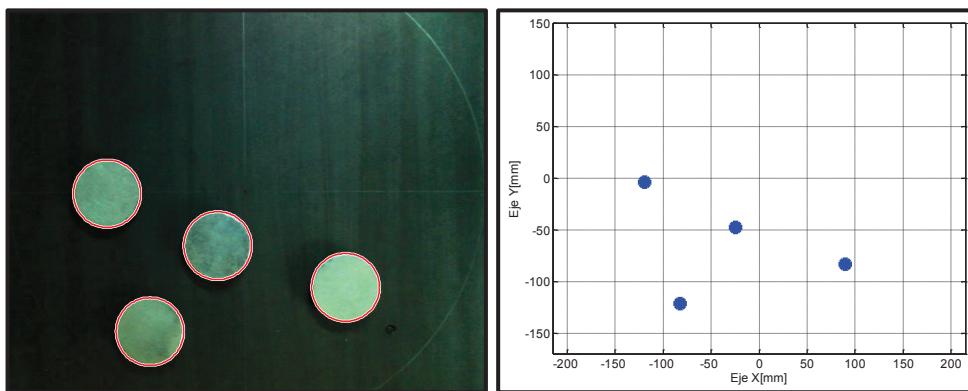
#### 4.3.1.2.2. Ordenamiento

Estas tareas se ejecutan con el objetivo de mostrar la capacidad del prototipo para colocar un objeto en un lugar dentro del espacio de trabajo que le corresponde según un criterio determinado. En la Figura 4.9, se puede apreciar un ordenamiento desde el punto (0; 100) mm, asignado por el usuario dentro de la interfaz, formando finalmente un cuadrado. Si los objetos detectados son más de cuatro entonces el prototipo seguirá apilando e irá formando el cuadrado

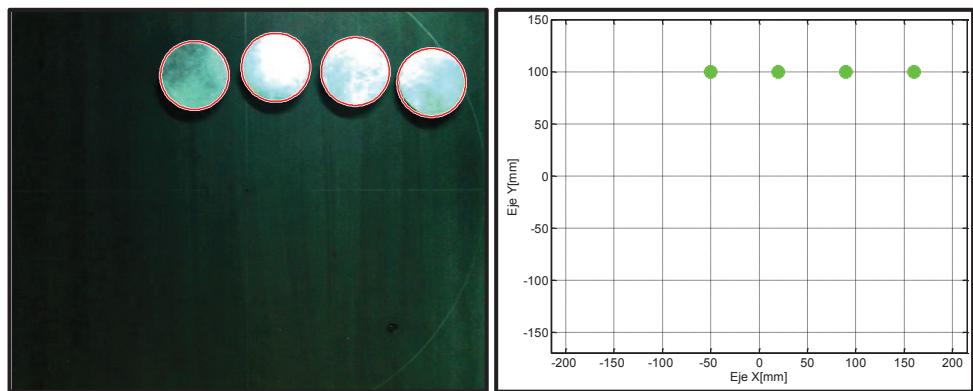


**Figura 4.9.** Ordenamiento formando un cuadrado desde el punto (0; 100) mm.

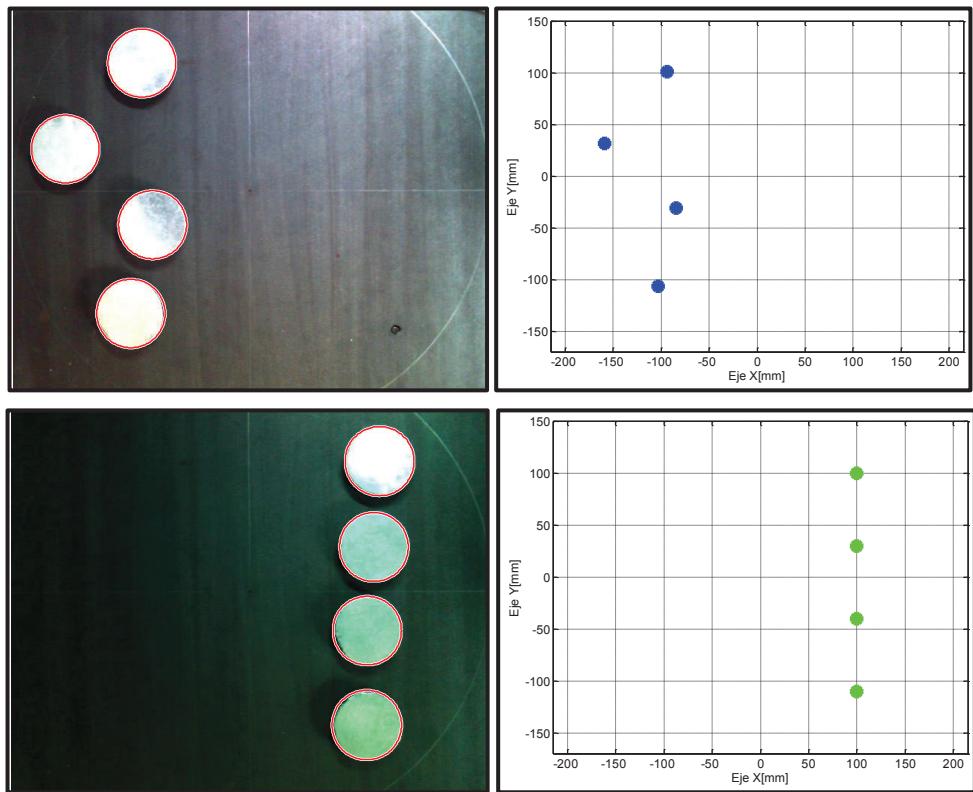
En la Figura 4.10 y la Figura 4.11 se puede ver el ordenamiento horizontal y vertical respectivamente ejecutado por el prototipo desde un punto dentro del espacio de trabajo designado por el usuario en la interfaz, el prototipo tiene restricciones para no ubicar los objetos fuera del área de trabajo volviendo a ubicar los objetos desde el punto asignado.



**Figura 4.10.** Ordenamiento horizontal desde el punto (-50,100) mm.



**Figura 4.10.** Ordenamiento horizontal desde el punto (-50,100) mm.



**Figura 4.11.** Ordenamiento vertical desde el punto (100,100) mm.

Las Figuras 4.8, 4.9, 4.10, 4.11 muestran que el robot delta construido cumple con los alcances establecidos, sin embargo existe un problema en el instante en el que el efecto final llega al objeto y son pequeñas oscilaciones en los actuadores; para disminuir estos problemas se implementa un controlador PID en los actuadores.

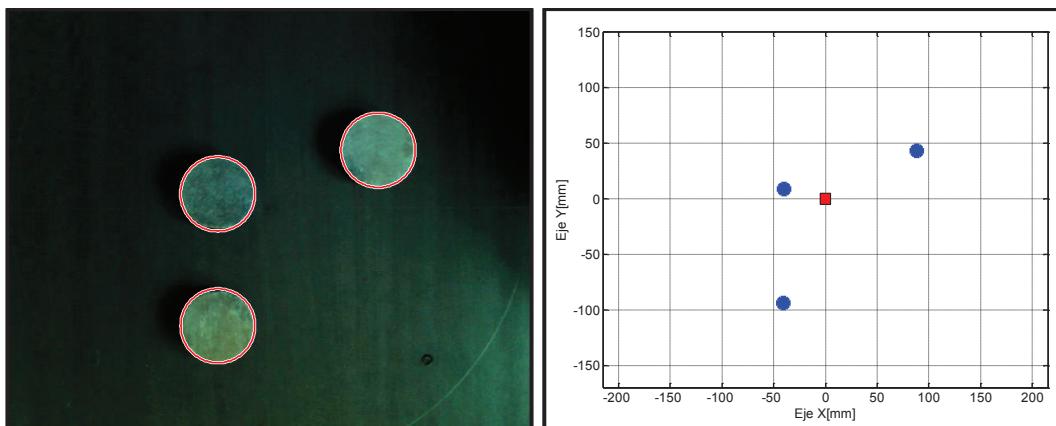
Es importante recalcar que este tipo de controladores no sacan el mayor provecho a los robots de esta categoría debido a la alta dinámica que poseen, por esta razón los controladores utilizados suelen ser los basados en el modelo, específicamente por par calculado o los controladores de tipo predictivo.[5]

#### 4.3.2. MODO MANUAL

El funcionamiento del prototipo en forma manual se efectúa con la ayuda de un dispositivo externo que servirá como un joystick de una consola de juego de video, el prototipo seguirá los movimientos del controlador alrededor de un espacio que es limitado por un eje coordenado que se presenta en la ventana de tareas.

El controlador o joystick para este caso está representado por los movimientos del mouse o el touchpad de una computadora portátil, este se encarga de efectuar los movimientos sobre el plano xy, para los movimientos del prototipo a lo largo del eje z se utilizan las teclas "X" y "Z".

El proceso comienza de manera similar al control automático, primero se captura la imagen y se localiza los centros de los objetos circulares y se muestran en un eje coordenado dentro de la interfaz, posteriormente se escoge el control manual y el prototipo se coloca en una posición central ubicada en los puntos (0, 0, -277) mm y está representado con un cuadrado de color rojo dentro del eje coordenado como se muestra en la Figura 4.12.



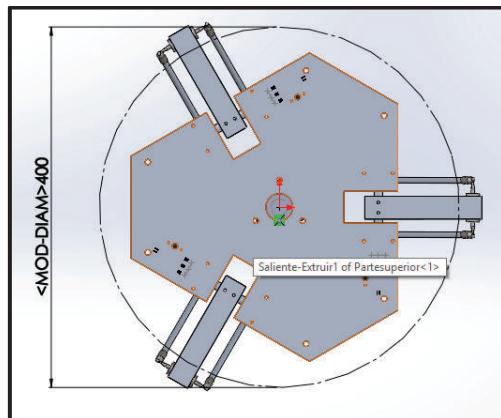
**Figura 4.12.** Imagen con centros adaptados y con la posición inicial del mouse.

Se coloca el puntero del mouse sobre el cuadrado de color rojo y presionando el botón izquierdo se arrastra hasta la posición de cada objeto y se lo baja hasta la posición adecuada para ser recogido, una vez ubicado en la posición deseada se activa el efecto final con la tecla “D”. Este proceso también puede ser efectuado sin la necesidad de hacer una captura de imagen y guiándose simplemente por la vista como en un juego de video.

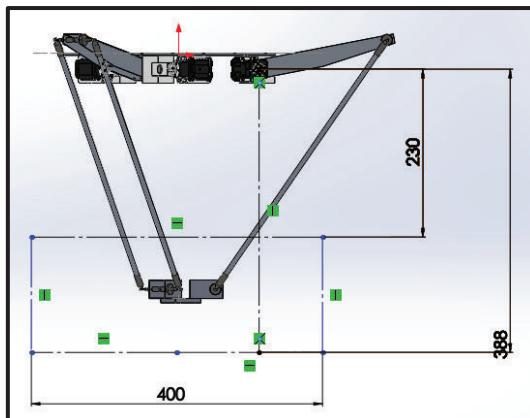
#### 4.5. CARACTERÍSTICAS TÉCNICAS DEL PROTOTIPO.

Una vez concluidas las pruebas con resultados satisfactorios se presenta en la Tabla 4.4, las características técnicas del prototipo como son datos de voltaje, corriente, peso y espacio de trabajo.

En las Figuras 4.13 y 4.14, se muestra el área de trabajo y las dimensiones físicas.



**Figura 4.13.** Vista superior del área de trabajo radio=200 mm.



**Figura 4.14.** Vista frontal del área de trabajo cilindro de radio=200 mm y altura=158 mm

**Tabla 4.4.** Especificaciones técnicas.

Características Físicas	
Peso	2.5 Kg
Eslabón L1	180 mm
Eslabón L2	380 mm
Radio base fija	110 mm
radio base móvil	30 mm
Conexiones eléctricas	
Tensión de Alimentación	110 V, 60 Hz
Potencia de la fuente	600 W
Consumo de actuadores a la máxima carga	
Voltaje administrado	12 VDC
Corriente	700 mA
Movimiento Pick and Place	8.4 W
Consumo de electroimán	
Voltaje administrado	12 VDC
Corriente	90 mA
Capacidad de carga	
Capacidad de los actuadores	150 g
Capacidad del electroimán	60 g
Posicionamiento efector final	
Error	5 mm
Velocidad	~100 mm/s

## CAPITULO 5

### CONCLUSIONES Y RECOMENDACIONES

#### 5.1. CONCLUSIONES.

- La configuración mecánica del robot delta construido presenta problemas de singularidades en las posiciones extremas de la plataforma móvil, por lo cual es necesario realizar restricciones en el posicionamiento de los actuadores mediante el software, y evitar de esta manera daños en las piezas mecánicas.
- La visión artificial es usada para dar autonomía a los manipuladores generando información como color, forma, ubicación y dimensiones. Estos datos son procesados para generar órdenes que son enviadas a los actuadores del manipulador, por lo que se debe tener datos veraces sobre el medio en el que se desenvuelve el robot, siendo necesario tener un sistema de iluminación controlado y a su vez un sistema de adquisición de imágenes con una tecnología avanzada lo que permitirá rapidez y precisión al momento de obtener la información.
- Mediante el software de diseño y simulación CAD se realizó el estudio estático a las piezas en donde se consideró que el esfuerzo era mayor, obteniendo deformaciones con valores muy pequeños, con lo cual se puede afirmar que el robot es resistente siempre que se trabaje con el máximo peso descrito en las características del robot.
- MATLAB es un software utilizado para el análisis y simulación de sistemas robóticos, que cuenta con herramientas que permiten la interacción entre el

computador y el sistema a ser controlado; sin embargo, al simular los movimientos en la interfaz se generan retardos que afectan al envío de datos. La solución planteada en este proyecto es almacenar los valores de los actuadores que generen una trayectoria, tanto en Arduino como en MATLAB y operarlos de forma simultánea.

- Debido a la alta dinámica de los robots paralelos el uso de controladores clásicos tipo PID no es suficiente para regular su funcionamiento; sin embargo, dentro del proyecto se hace uso de este controlador con el objetivo de atenuar la vibraciones en la plataforma móvil cuando este llega a un punto final deseado, la calibración y sintonización se hace usando un método heurístico, obteniendo un controlador PI que permite disminuir las oscilaciones en los actuadores.
- Durante el funcionamiento del prototipo en lazo cerrado, este alcanzó posturas no deseadas generando movimientos violentos que condujo a investigar la calidad de los datos realimentados y se constató que algunos datos llegaban erróneos y fuera de rango, por lo que para solucionar este problema se tuvo que cerrar la cadena de conexión de los actuadores con la referencia a tierra, aunque en los manuales no se indica esto.
- El método geométrico para la obtención del modelo cinemático es el más factible en este tipo de robots debido a la simetría que presentan, esto se puede afirmar porque el análisis se realiza a una sola cadena cinemática y luego mediante matrices de transformación homogénea se obtiene la solución de las otras cadenas.
- En la construcción de robots manipuladores es primordial saber el espacio de trabajo que se va a necesitar para cumplir una tarea específica, a partir de esta primera información se pueden plantear posibles dimensiones que tendrá el robot para posteriormente optimizarlas.

## 5.1. RECOMENDACIONES.

- El estudio cinemático en este tipo de robot está basado en la simetría, lo que hace fácil el análisis geométrico, por lo tanto se recomienda que la construcción de las piezas sean lo más exacta posibles, esto hará que la posición de la plataforma móvil en el mundo físico sea lo más aproximado al calculado mediante la cinemática
- En el prototipo de robot delta construido se utilizó materiales de bajo peso lo que ocasiona que el movimiento sea inestable y oscile antes de colocarse en su posición final deseada, a pesar de contar con un controlador que ayuda a disminuir estas oscilaciones, por lo tanto se recomienda que el movimiento entre distancias grandes relativas al robot siempre sean por medio de una trayectoria generada lo que ayuda a suavizar el movimiento y además ayuda para que el controlador funcione de manera correcta.
- Debido a la restricción de memoria para almacenar los datos de las trayectorias en la tarjeta Arduino, se recomienda que los elementos de cada motor no supere las 400 muestras entre punto y punto, además si se tiene valores muy seguidos entre las distancias el movimiento del actuador será demasiado lento.
- Antes de iniciar con el diseño mecánico de un robot es recomendable tener conocimiento de los materiales existentes en el mercado debido a que los diferentes elementos mecánicos están estandarizados por lo que no existen en cualquier dimensión.
- Los movimientos de las cadenas cinemáticas del robot delta dependen una de la otra por lo que con un análisis cinemático es difícil obtener un buen control de la velocidad de movimiento para mejorar este aspecto se recomienda realizar el estudio dinámico del robot.

- Para realizar cualquier movimiento del robot dentro del espacio de trabajo deben estar colocados solo los objetos a ser desplazados puesto que no se cuenta con un sensor o un control para la evasión de obstáculos, lo que daría como consecuencia una mala ejecución de las tareas asignadas.
- Los objetos a ser ordenados, deben estar colocados de tal forma que no interfiera en el camino de los puntos de ubicación. Para realizar el proceso de apilamiento los objetos deben estar a un determinado radio dado que objetos muy cercanos a la posición final deseada los eslabones podrían desestabilizar los objetos.
- Para tener un mayor dinamismo entre el robot y los objetos ubicados en el espacio de trabajo se recomienda un control servo visual lo que permitirá colocar a los objetos en una banda transportadora, permitiendo de esta manera ubicar el efecto final a objetos en movimiento.
- Con el propósito de mejorar el funcionamiento del prototipo, se recomienda trabajar en la parte de la visión artificial, permitiendo detectar objetos de cualquier dimensión y forma, para que la plataforma móvil se pueda posicionar en los objetos detectados. Además con el objetivo de manipular los mismos también se debe cambiar el efecto final por una garra mecánica (gripper).

## REFERENCIAS BIBLIOGRÁFICAS

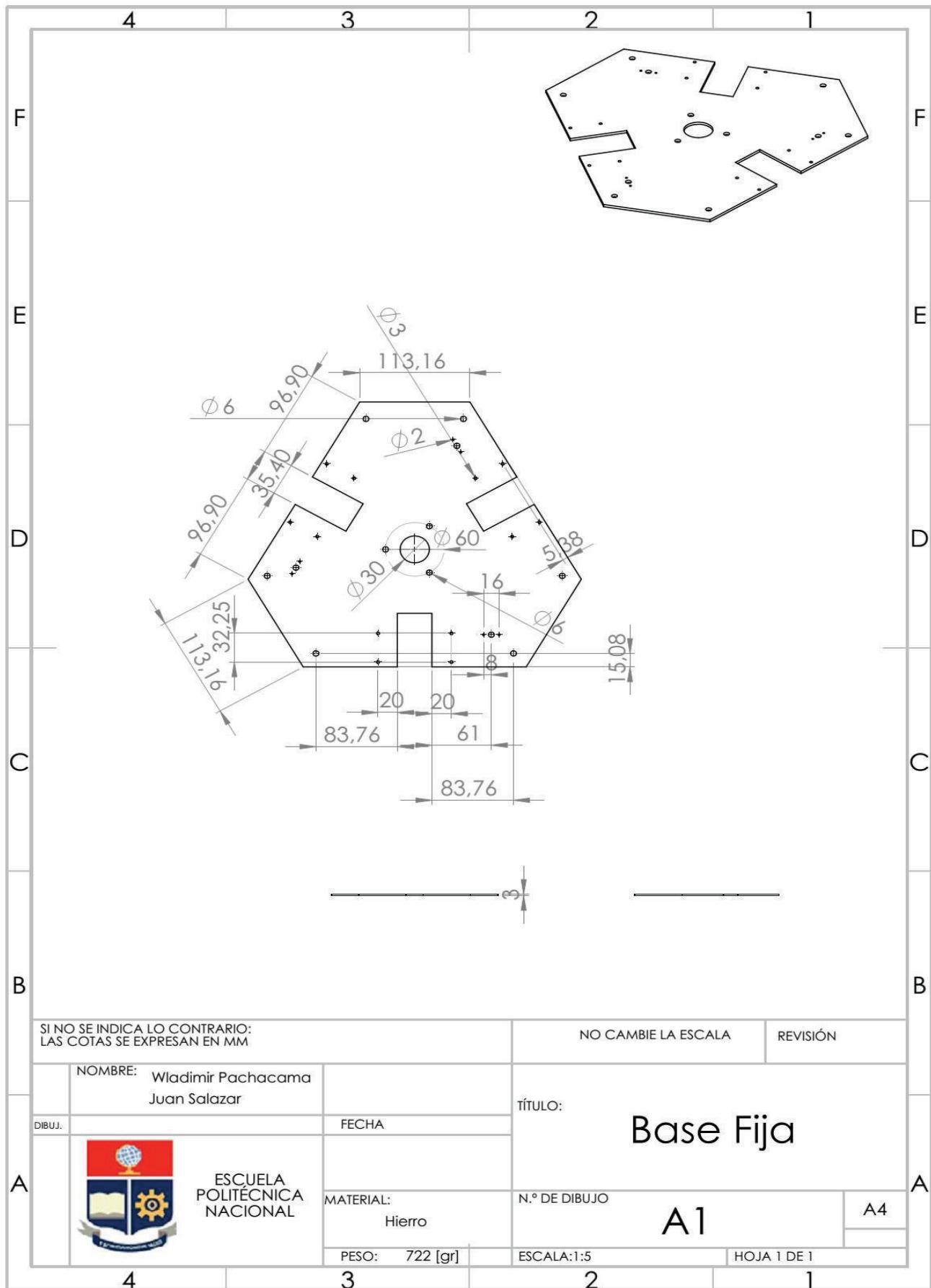
- [1] Ollero, A. (2001), Robótica: Manipuladores y robots móviles, Editorial Marcombo S.A., Barcelona.
- [2] Barrientos, A., Peñin, L., Balaguer, C., & Aracil, R, (2007), *Fundamentos de Robótica*, Editorial McGraw-Hill/Interamericana, Madrid.
- [3] Ingeniare. Revista Chilena de ingeniería, 2011, [En línea]. Disponible en: [http://www.scielo.cl/scielo.php?pid=S071833052011000300002&script=sci\\_arttext](http://www.scielo.cl/scielo.php?pid=S071833052011000300002&script=sci_arttext)
- [4] Merlet, J.P., (2006), Parallel Robots. Segunda edición, Editorial Springer, The Netherlands, Dordrecht.
- [5] Vivas, A. Robótica Paralela: Aplicaciones Industriales, Modelado y Control. Artículo Universidad del Cauca, Departamento de Electrónica, Instrumentación y Control.
- [6] Aracil, R., (2006), Robot Paralelo: Máquinas con un Pasado para una Robótica del Futuro, Artículo Universidad Politécnica de Madrid.
- [7] Briones, J., (2009), Diseño, Análisis y Construcción de un Robot Paralelo Traslacional (Tesis de Maestría), Instituto Politécnico Nacional; México, Querétaro. Recuperado de:  
<http://itzamna.bnct.ipn.mx:8080/dspace/bitstream/123456789/4900/1/JABL.PDF>
- [8] Bonev, I, (2001), Affordable Academic Parallel Robots, [En línea]. Disponible en: <http://www.parallemic.org/Reviews/Review002.html>
- [9] Álvarez, J., & Duque, G., & Flores, G., & Galindo, L., (2010), Modelado y Manufactura de un Robot Delta para aplicaciones Pick and Place (Tesis de pregrado), Instituto Politécnico Nacional, México.
- [10] Cañadas, P., & Haro, G., (2011), Prototipo de un sistema de adquisición de imágenes de vehículos, detección y reconocimiento automático de los caracteres de la placa en tiempo real por medio de visión artificial, aplicado al control vehicular (Tesis de pregrado), Escuela Politécnica Nacional, Quito
- [11] (2016) <http://www.mathworks.com/products/image/>
- [12] (2016) Visión Artificial. [En linea]. Disponible en:  
<http://www.etitudela.com/celula/downloads/visionartificial.pdf>

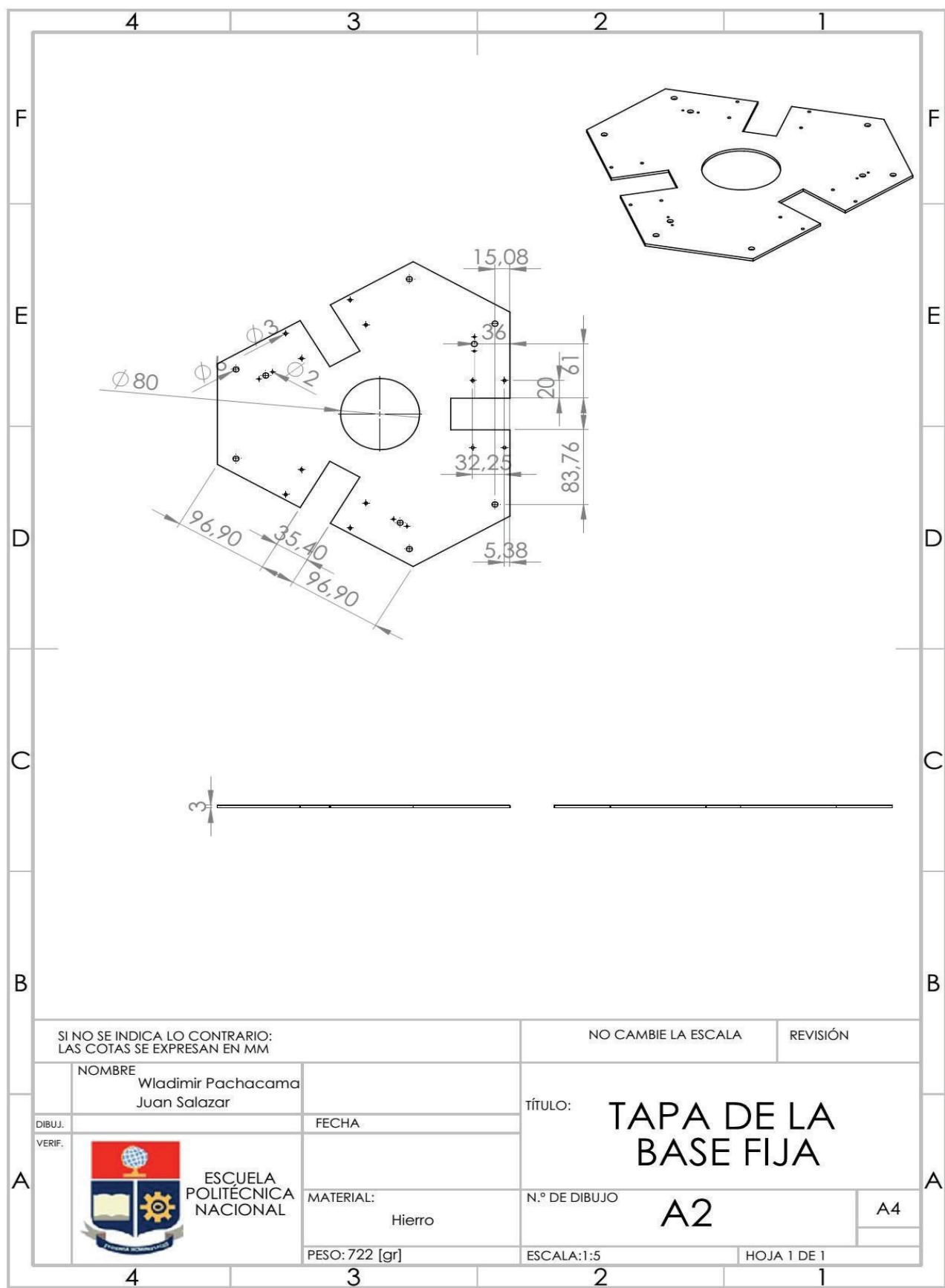
- [13] (2012) Control System Design for Precision Engineering Applications, [En línea]. Disponible en: <http://www.perryautomation.com/Index.htm>
- [14] Spong, W., & Hutchinson, S., & Vidyasagar, M., (Segunda Edición), (2004) Robot Dynamics and Control.
- [15] Silva, L., (2005), Control visual de robots paralelos. Análisis, desarrollo y aplicación a la plataforma Robotenis (Tesis doctoral), Universidad Politécnica de Madrid.
- [16] Urrea, L., & Medina, S., (2012), Diseño e implementación de una plataforma robótica tipo delta (Tesis de pregrado), Universidad Militar Nueva Granada, Bogotá.
- [17] Gosselin & Angeles, (1990), Singularity Analysis of Closed-Loop Kinematic Chain, IEEE Transactions on Robotics and Automation, DOI: 10.1109/70.56660 · Source: IEEE Xplore.
- [18] López, M., & Castillo, E., & García, G., & Bashir, A., (2005) Delta robot: inverse, direct, and intermediate Jacobians, DOI: 10.1243/095440606X78263.
- [19] Badescu, M., (2002), Workspace optimization of 3-UPU parallel platforms with joint constraints, IEEE International Conference on Robotics & Automation, Washington DC. 0-7803-7272-71021\$17.0002002 IEEE.
- [20] Peña, C., & Martínez, E., & Cárdenas, P., (2011), Optimización dimensional de un robot paralelo tipo delta basado en el menor consumo de energía, Ciencia e Ingeniería Neogranadina, Vol. 21-1, pp, 73-88.
- [21] Balmaceda, A., (2011), Metodología de rediseño de un robot paralelo tipo delta de 3 grados de libertad en función de un espacio de trabajo prescrito (Tesis de maestría), Instituto Politécnico Nacional, Querétaro.
- [22] Pedraza, M., & Cárdenas, P., & Rodriguez, F., & Yime, E., (2015), Aproximación al diseño de robots paralelos, análisis del caso del robot delta, IV congreso internacional de ingeniería mecatrónica y automatización.
- [23] Cortés, J., & Cortés, H., & Garcia, J., (2013), Diseño óptimo dimensional de una máquina robótica de arquitectura paralela para aplicaciones pick & place, VI congreso internacional de ingeniería mecánica y IV de ingeniería mechatrónica.
- [24] Manual de usuario Dynamixel AX-12 A , 2006.
- [25] Aceves. A., PHd, Manual Bioloid de Robotis.

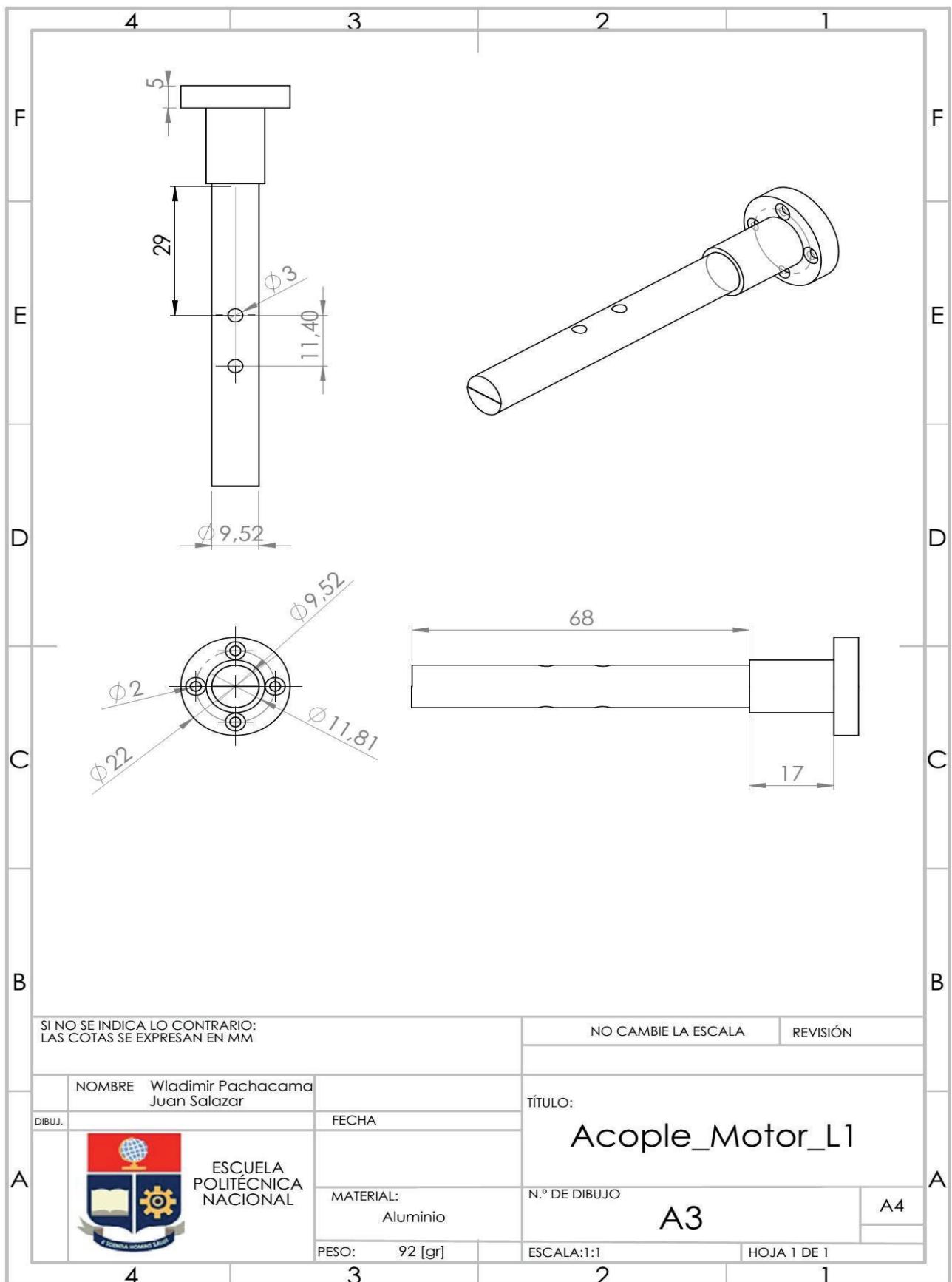
- [26] Sotomayor, N., Msc, (2009), Control con microprocesadores, Escuela Politécnica Nacional, Quito.
- [27] (2016) <http://www.mathworks.com/products/matlab/>
- [28] (2011) <http://robotparaleloupb.blogspot.com/>
- [29] (2016) <http://help.solidworks.com>

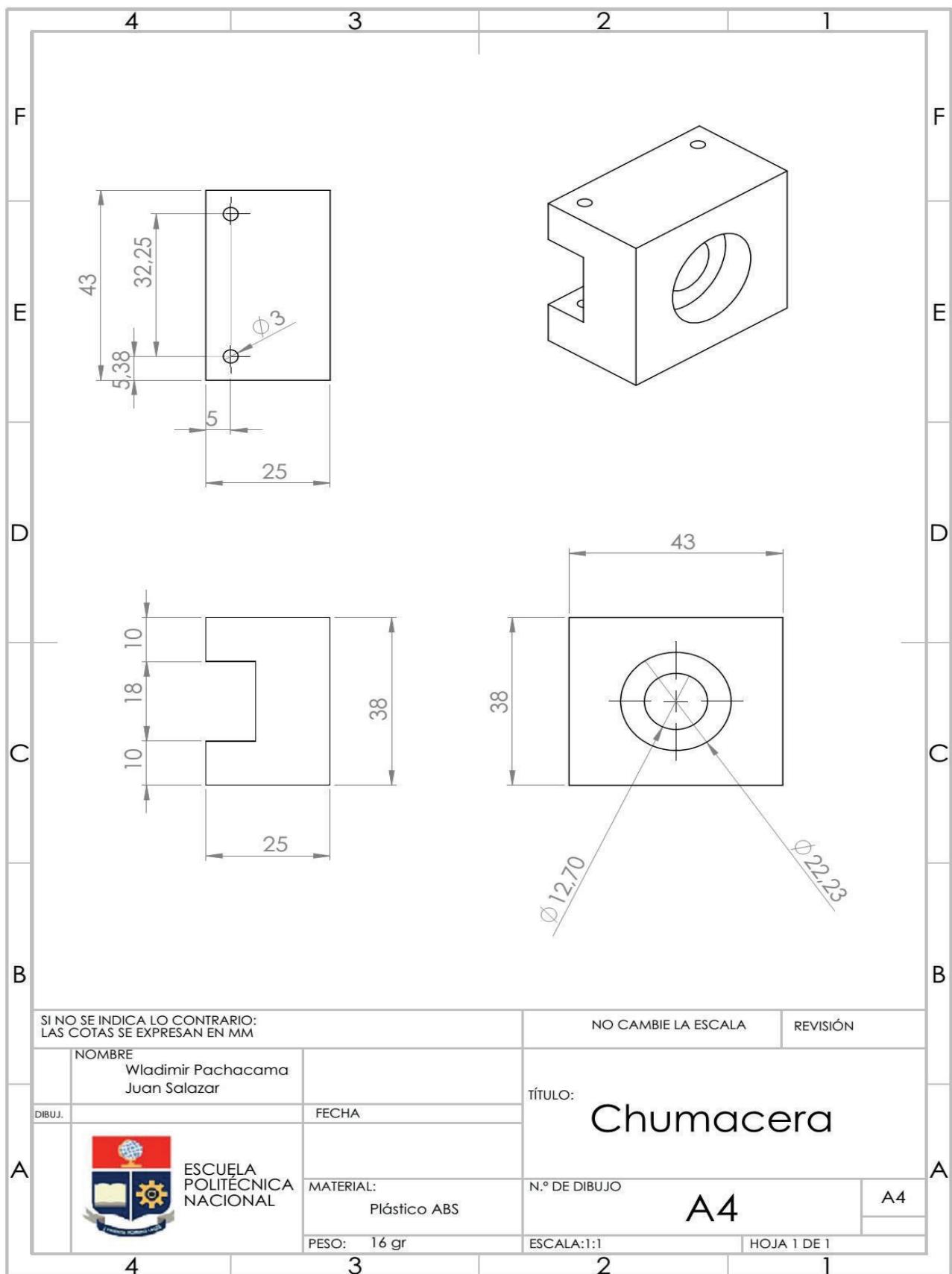
**ANEXO A**

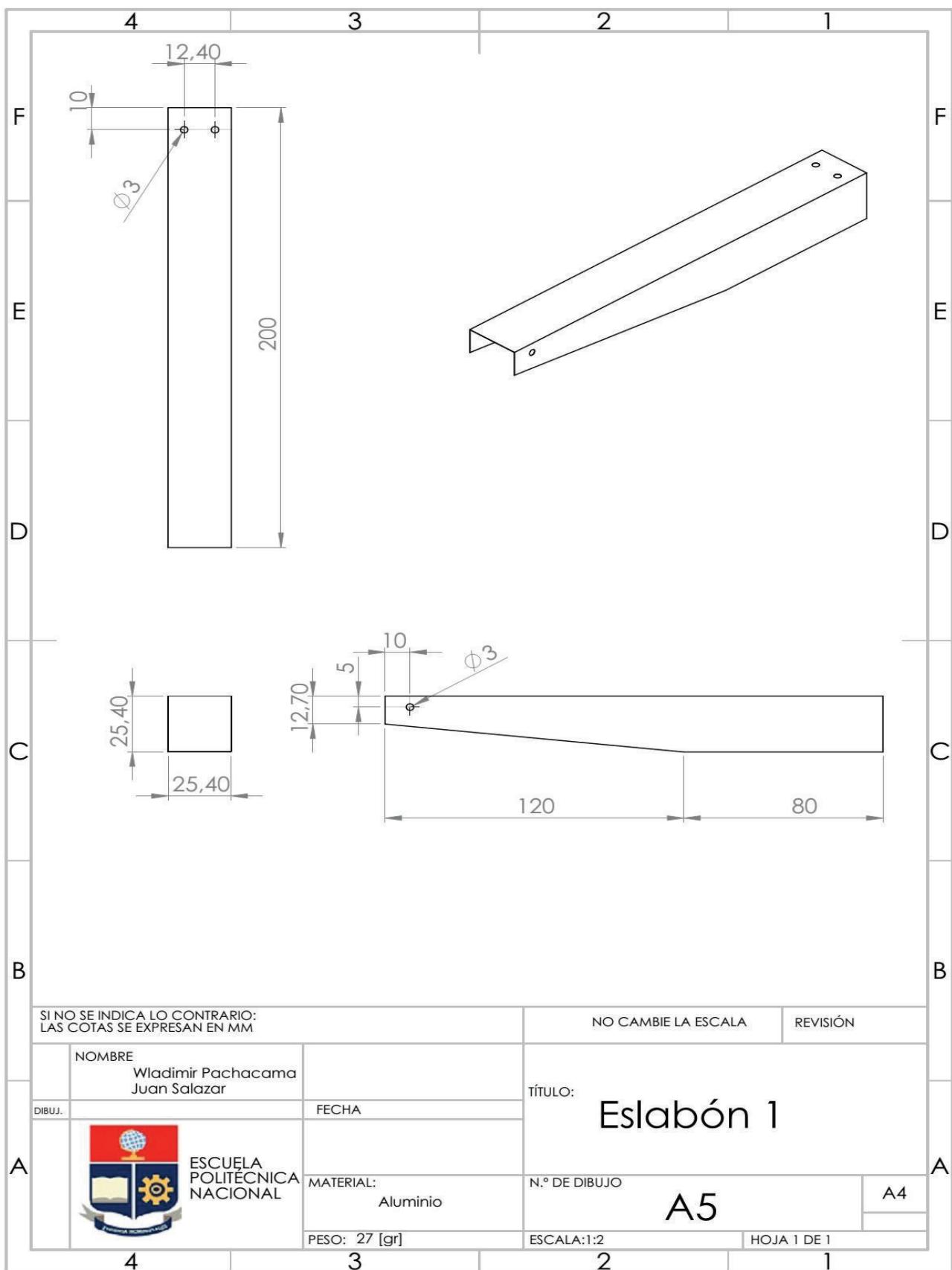
**PLANOS MECÁNICOS**

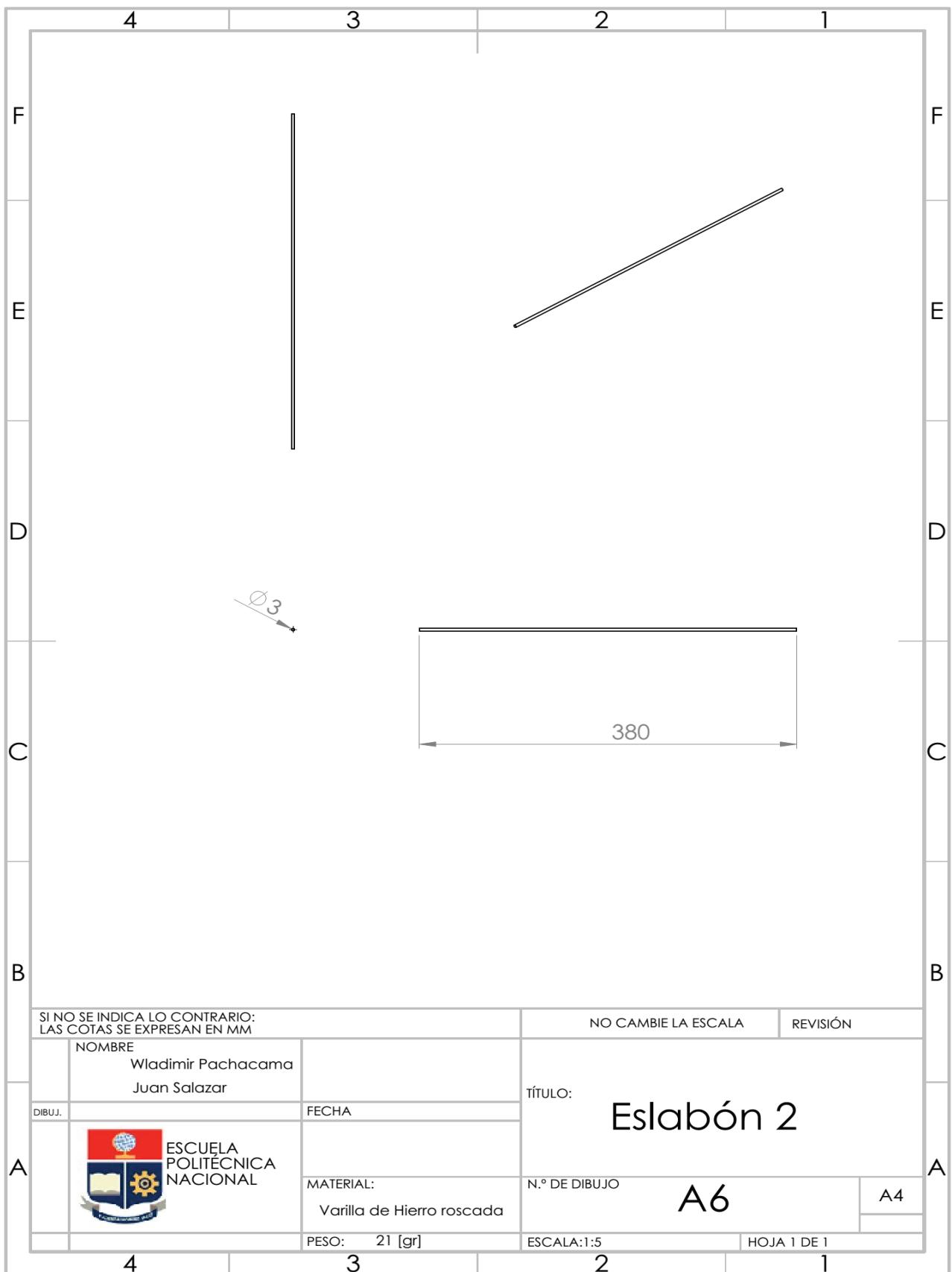


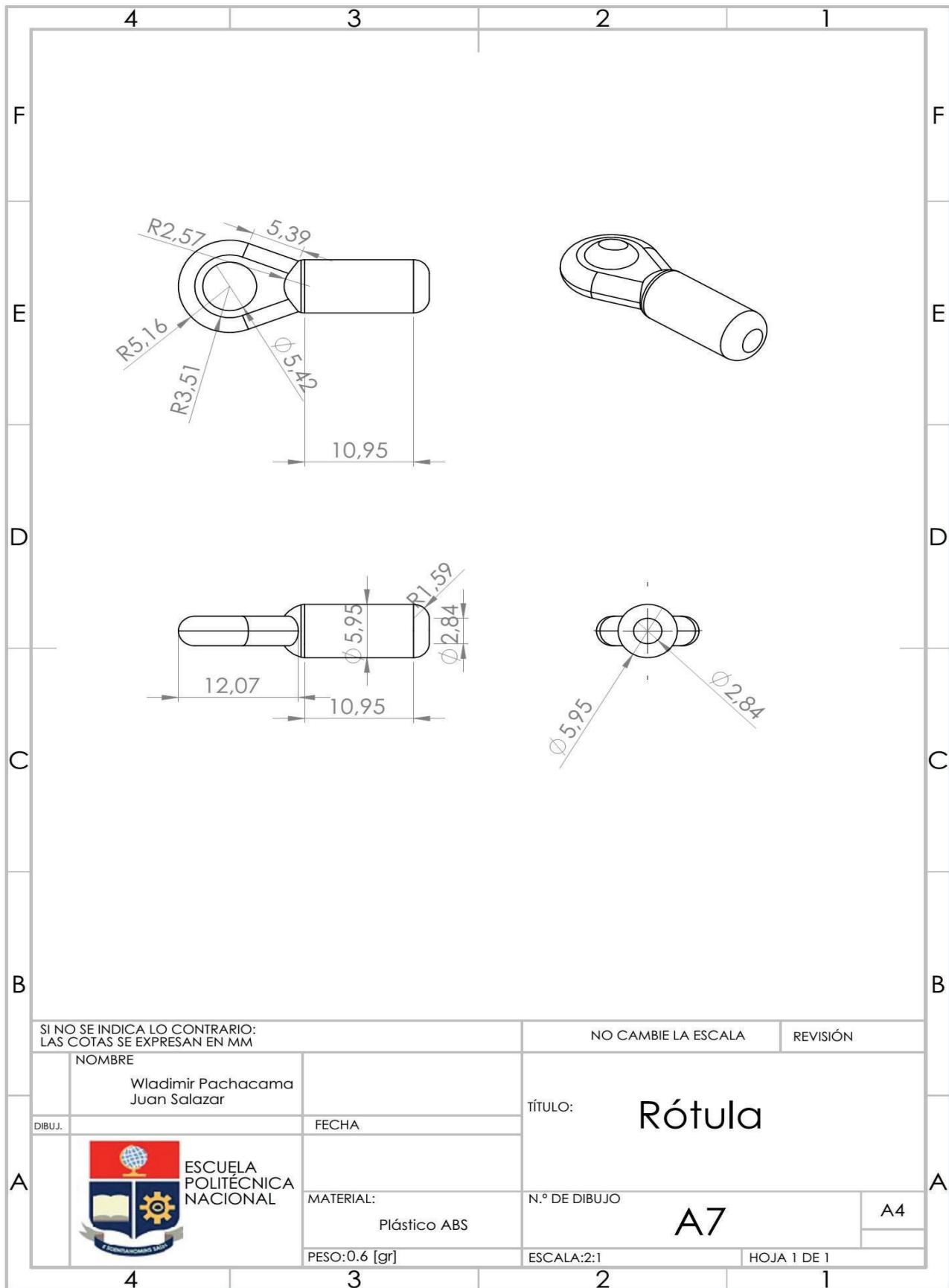


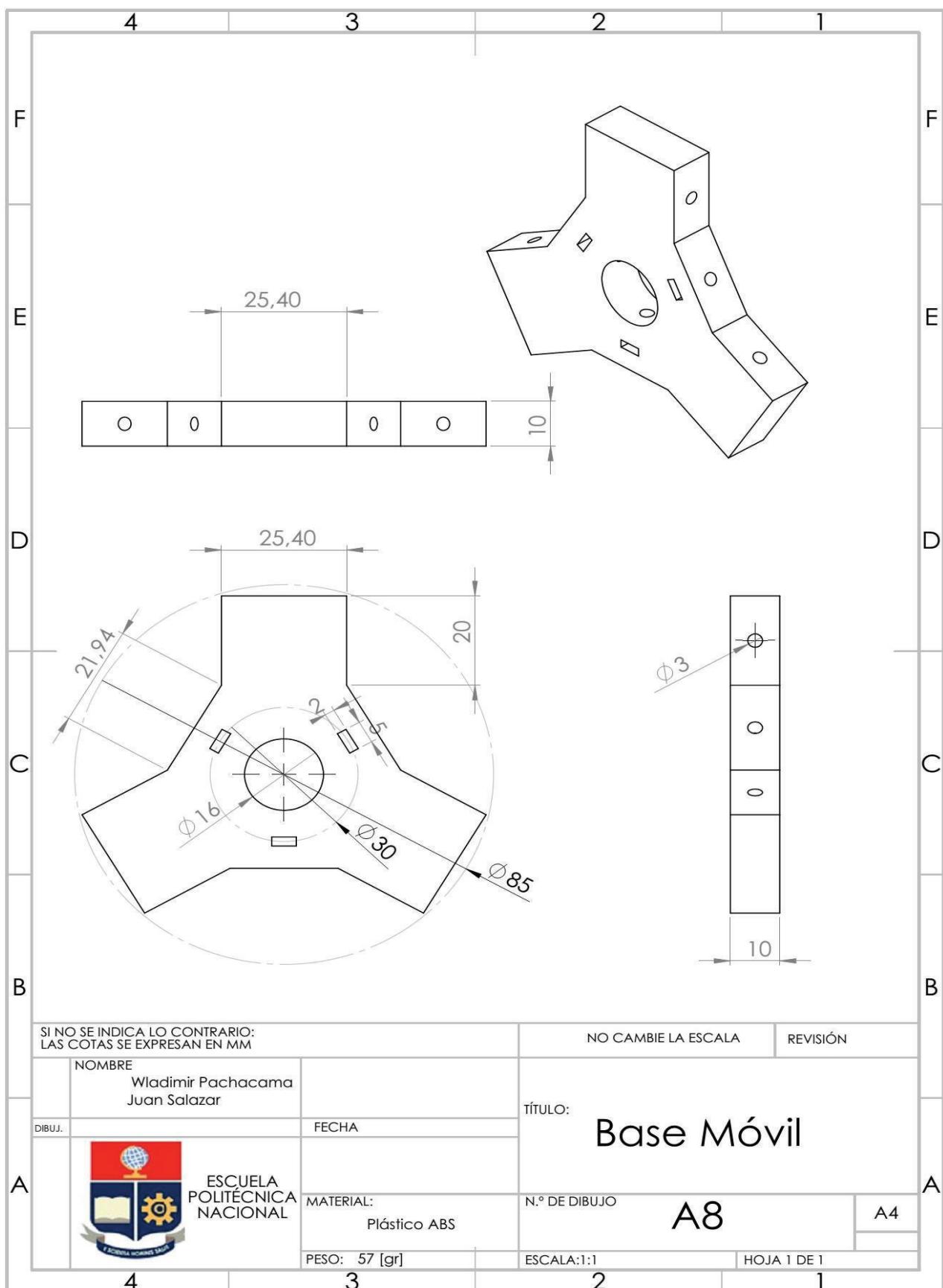


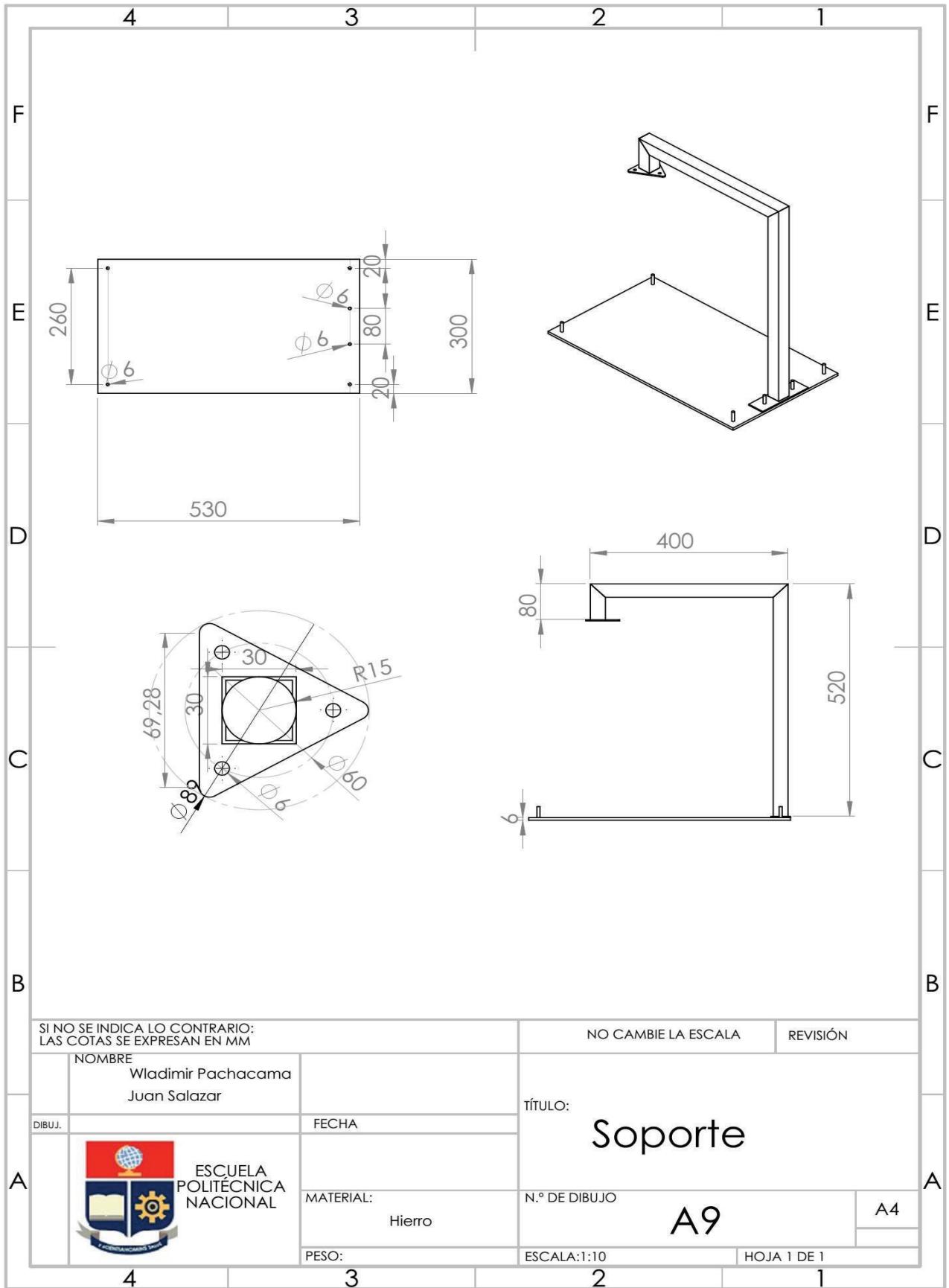




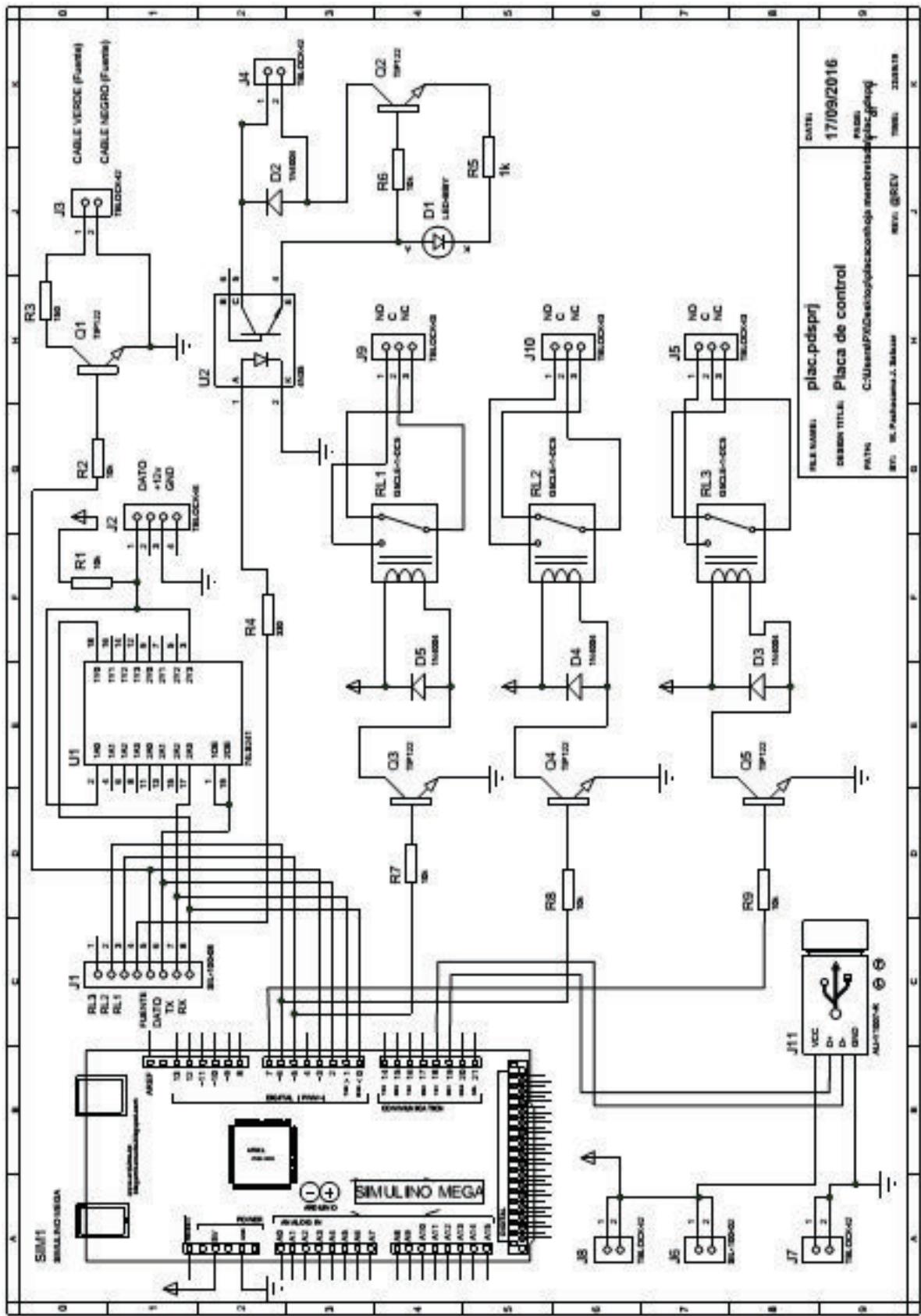








**ANEXO B**  
**PLANO ELÉCTRICO**



**ANEXO C**

**MANUAL DE USUARIO**

# **MANUAL DE USUARIO**

## **1. INTRODUCCIÓN.**

Este manual tiene el propósito de enseñar el manejo del sistema del robot delta, exponiendo la forma de conexión y el manejo del software.

Con la ayuda de la interfaz el robot puede ser programado para apilar y ordenar objetos de forma automática y manual, además se puede realizar movimientos de diferentes trayectorias como círculos, espiras, etc.

## **2. INSTALACIÓN.**

### **2.1. Requerimiento de Hardware.**

- Computador.
- Dos Cables USB macho hembra.
- Cable USB de tipo A/B.

### **2.2 Requerimiento de Software.**

- MATLAB 2014A o superior.

### **2.3 Procedimiento**

- Se debe conectar el gabinete a una toma de corriente a 110 VAC, 60 Hz.
- Conectar el cable USB macho hembra en el puerto de la PC y en el gabinete.
- Conectar el cable USB entre la PC y la cámara VGA.

## **3. FUNCIONAMIENTO.**

### **3.1. Inicialización.**

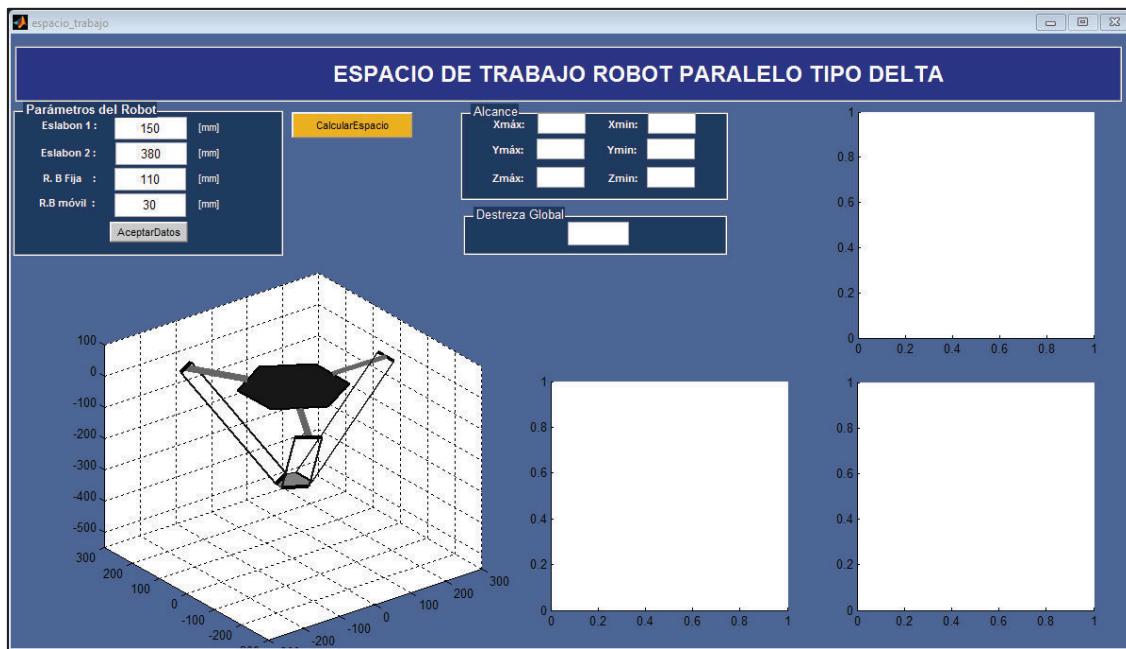
- a. En el gabinete se debe accionar el interruptor ON/OFF y dejarlo en la posición ON, la luz piloto color rojo debe encenderse, si no se debe verificar la conexión a la red, probablemente el cable de alimentación no está enchufado.

- b. Abrir la carpeta “PROGRAMAS MATLAB”, y ejecutar el archivo “PRESENTACION.m” donde se presenta una pantalla de bienvenida y se puede seleccionar la ventana espacio de trabajo, Demostración y aplicación como se puede observar en la Figura C1.



**Figura C1.** Pantalla principal

- c. En caso de seleccionar “Espacio de trabajo”, se abre una nueva ventana mostrada en la Figura C2, donde se puede manipular las dimensiones del robot y se obtiene como resultado las gráficas del espacio de trabajo además se muestra la destreza del robot y los límites en el plano xyz.



**Figura C2.** Ventana espacio de Trabajo.

d. Si se selecciona la opción “Demostración” se abre la ventana de la Figura C3, desde aquí se puede ejecutar un control manual del movimiento del robot con el teclado del computador, además se puede seleccionar las trayectorias que puede realizar el robot (círculo, espiral, círculos desplazados y curva en forma de flor) y se puede modificar los parámetros que caracterizan a la curva como por ejemplo el radio.

Dentro de esta ventana también se tiene dos botones que permiten abrir las ventanas donde se puede observar la trayectoria ejecutada en el plano xyz y el cambio de la posición angular de los tres actuadores.



Figura C3. Ventana de la Sección Demostrativa.

- e. Si selecciona “Aplicación” se abre la ventana de la Figura C4, desde donde puede seleccionar el control manual o el control automático para la ejecución de tareas de ordenamiento y apilamiento.

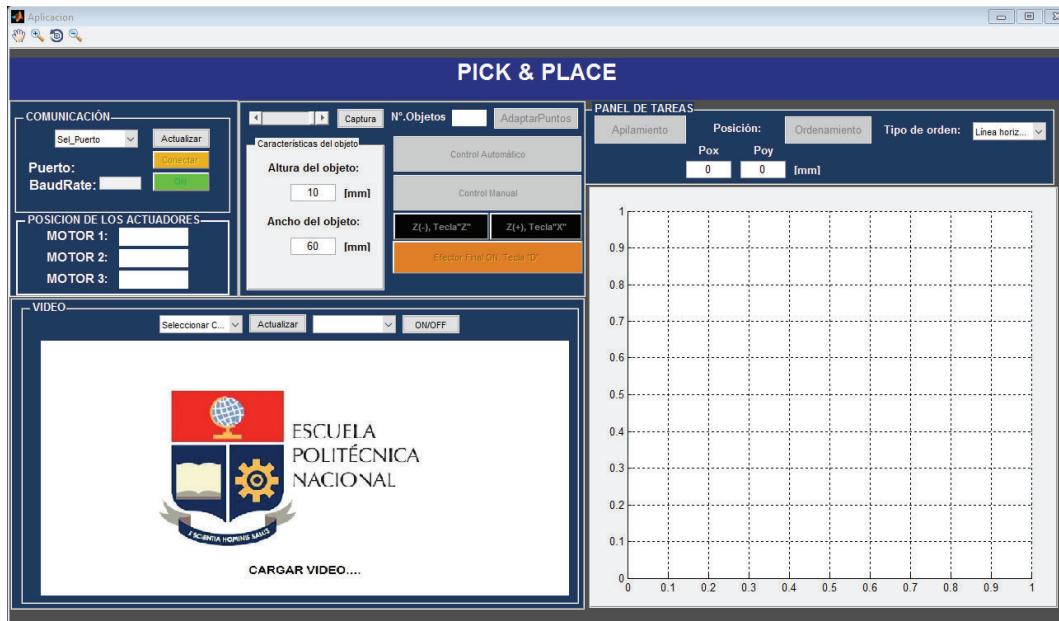


Figura C4. Ventana de tareas

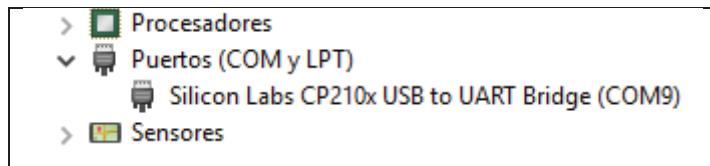
## 2.4 Comunicación.



**Figura C5.** Panel de comunicación

Este panel está disponible en la ventana de “Demostración” y “Aplicación”, permitiendo el enlace PC-Arduino, para lo cual se procede de la siguiente forma:

- Se debe buscar en la ventana de administrador de dispositivos del computador el puerto correspondiente al módulo TTL-USB y verificar en que COM se encuentra conectado como se observa en la Figura C6.



**Figura C6.** Detección del Módulo TTL-USB.

- Desde la interfaz se selecciona el COM y de inmediato en los bloques de texto aparece el nombre del puerto y la velocidad de comunicación que es de 19200 bps, además se activa el botón “Conectar”.



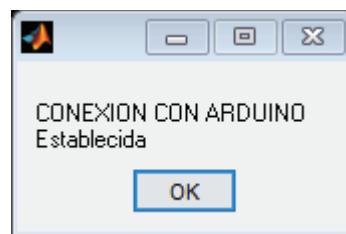
**Figura C7.** Selección y conexión del módulo TTL-USB.

- Si la conexión resulta exitosa la tarjeta Arduino, envía un dato de confirmación y la interfaz abre un cuadro de dialogo indicando que la comunicación con Arduino está establecida y se habilitara el botón “ON”.



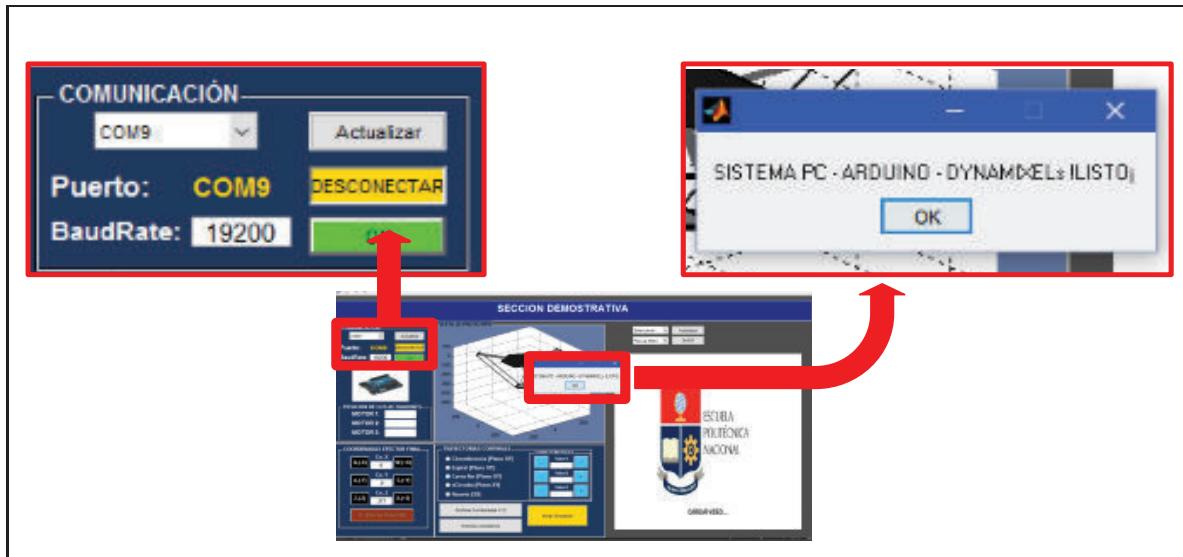
**Figura C8.** Mensaje sobre el estado de la comunicación

- En el caso donde la comunicación sea errónea el cuadro de dialogo mostrará que la comunicación no ha sido establecida y se debe verificar si el cable USB está conectado de manera correcta en los dos extremos y si el puerto seleccionado corresponde al módulo.



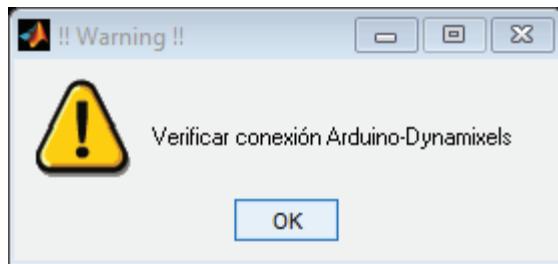
**Figura C9.** Mensaje de error de conexión de la interfaz.

- Al presionar el botón ON se envía un comando a la placa Arduino, cuya orden activa una salida digital que conecta la fuente mediante un transistor TIP-122, transcurrido tres segundos la interfaz envía una segunda orden la que permite al Arduino leer el nivel de voltaje existente en los servomotores Dynamixel si el valor del voltaje recibido desde los actuadores es mayor a 9[V], la tarjeta Arduino enciende la luz piloto de color verde y envía un comando hacia la interfaz donde se abre un cuadro de dialogo indicando que el sistema PC-Arduino-Dynamixel está listo para ser usado e inmediatamente la interfaz envía los valores a los actuadores para que se coloquen en una posición inicial, de esta manera el sistema queda listo para comprobar su funcionamiento.



**Figura C10.** Mensaje sobre el estado de la comunicación.

- En el caso, que la comunicación entre Arduino y Dynamixel no este establecida, la tarjeta Arduino envía un comando diferente que genera en la interfaz un mensaje de alerta indicando que la etapa Arduino-Dynamixel debe ser verificada; las posibles causas para que se de este fallo son: Mala conexión en el extremo de alguno de los actuadores, cables sueltos en la tarjeta electrónica ubicado en el gabinete o los servomotores están dañados.

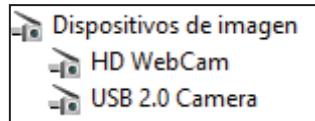


**Figura C11.** Mensaje de fallo en la conexión Arduino-Dynamixel.

## 2.5 Detección de objetos.

El sistema cuenta con una cámara web que permite la aplicación de la visión artificial para detectar objetos circulares de un diámetro de 60 mm. El proceso para la detección de los objetos se procede de la siguiente forma:

- Verificar en el computador la conexión de la cámara, para esto se debe ir al administrador de dispositivos como se muestra en la Figura C12.



**Figura C12.** Detección de la cámara VGA.

- Encender la cámara desde el panel de video ubicado en la ventana de aplicación. Además se debe escoger la resolución más adecuada de la cámara.



**Figura C13.** Encendido de la cámara a través de la interfaz

- Presionar el botón “Capturar”, e inmediatamente en la pantalla aparece la foto captada por la cámara, donde se resalta los objetos circulares, junto al botón capturar se encuentra un slider que permite ajustar la sensibilidad con el que se encuentran los de objetos circulares.

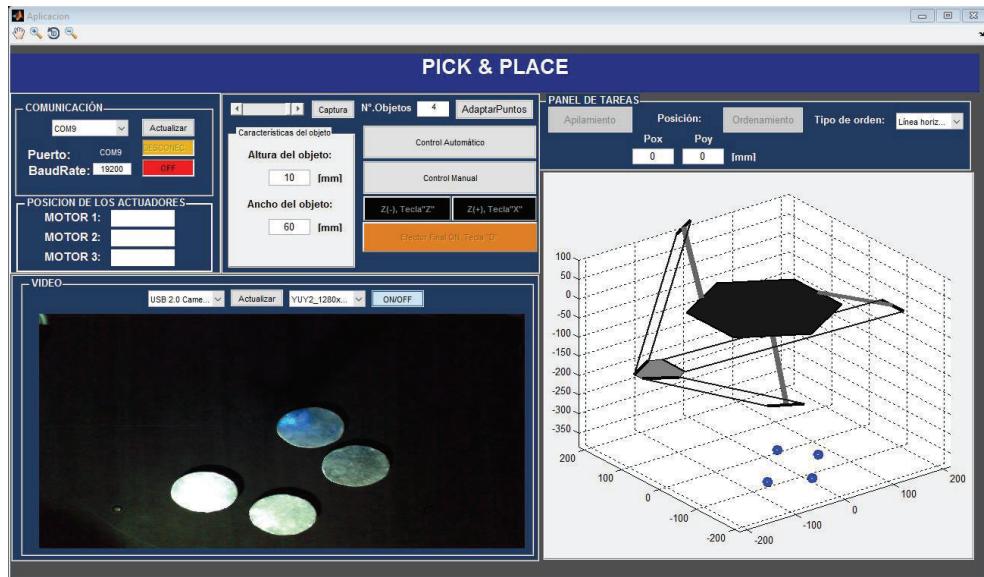


**Figura C14.** Figura Capturada



**Figura C15.** Imagen ajustada la sensibilidad

- Presionar el botón “Adaptar puntos”. La imagen captada es remplazado por la ubicación en el plano xy de los puntos centros de los objetos circulares captados.



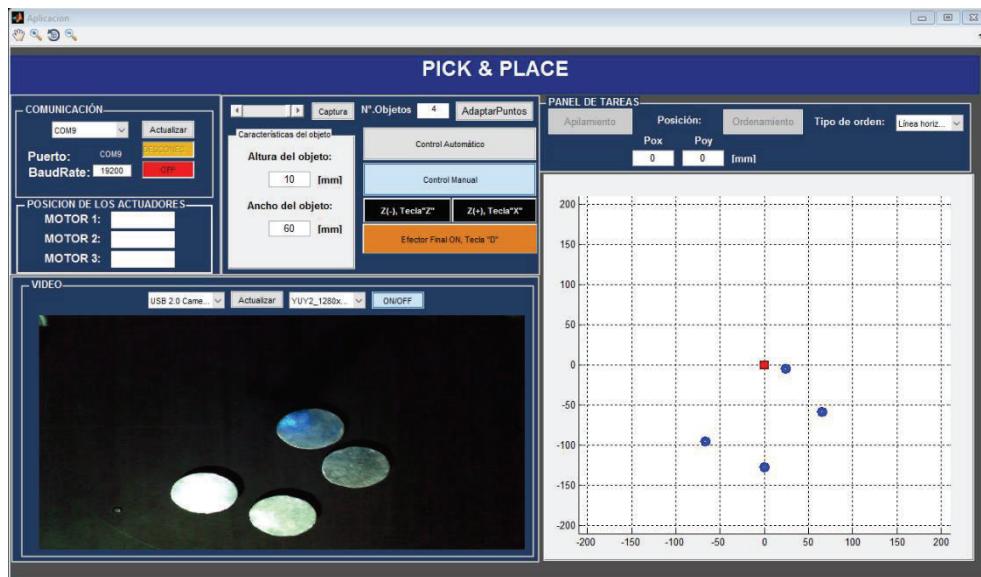
**Figura C16.** Adaptación de los puntos centros de los objetos detectados en el plano xy

- Finalmente en la ventana se puede escoger el modo de trabajo “MANUAL” o “AUTOMATICO”.

## 2.6 Modo manual.

Escogida esta opción se puede ir moviendo el robot según el criterio del operario. En la pantalla donde se encuentra los puntos centros de los objetos captados, aparece un punto rojo que se mueve de acuerdo a los movimientos efectuado por el mouse en el plano xy.

Con las teclas “X” y “Z”, se modifica el movimiento del robot en el eje z. Para activar y desactivar el electroimán se presiona la tecla “D”.(Figura C17)



**Figura C17.** Pantalla de aplicación en modo manual

## 2.7 Modo automático.

Dentro de esta opción se tiene varias tareas que pueden ser seleccionadas como el apilar objetos u ordenar de forma horizontal, vertical o formar un cuadrado, dentro del espacio de trabajo.

Antes de ejecutar cualquiera de estas tareas se debe ingresar en la interfaz el punto de referencia dentro del espacio de trabajo donde el robot apilara los objetos o de ser el caso el punto de inicio del ordenamiento.