



**ESPE**

**UNIVERSIDAD DE LAS FUERZAS ARMADAS**  
**INNOVACIÓN PARA LA EXCELENCIA**

**DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA  
Y MECÁNICA**

**CARRERA DE INGENIERÍA MECATRÓNICA**

**TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL TÍTULO  
DE INGENIERO EN MECATRÓNICA**

**TEMA:**

**DISEÑO Y CONSTRUCCIÓN DE UN ROBOT PUMA CON UN SISTEMA  
DE SIMULACIÓN EN TIEMPO REAL**

**AUTORES:**

**ALLAUCA FAJARDO, SANTIAGO ANDRÉS**  
**ESTÉVEZ CAMPOVERDE, MARCO ANTONIO**

**DIRECTOR:**

**ING. LOZA MATOVELLE, DAVID CÉSAR**  
**SANGOLQUÍ**

**2019**



# ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA  
CARRERA DE INGENIERÍA MECATRÓNICA**

**CERTIFICACIÓN**

Certifico que el trabajo de titulación, **“DISEÑO Y CONSTRUCCIÓN DE UN ROBOT PUMA CON UN SISTEMA DE SIMULACIÓN EN TIEMPO REAL”** fue realizado por los señores **ALLAUCA FAJARDO, SANTIAGO ANDRÉS** y **ESTÉVEZ CAMPOVERDE, MARCO ANTONIO** el mismo que ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 29 de julio de 2019

A handwritten signature in blue ink, appearing to read 'David Loza', is written over a horizontal line. Below the line is a dotted line.

ING. DAVID CÉSAR LOZA MATOVELLE, MSc.

DIRECTOR

C. C.: 1708661549



# ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA  
CARRERA DE INGENIERÍA MECATRÓNICA**

**AUTORÍA DE RESPONSABILIDAD**

Nosotros, **ALLAUCA FAJARDO, SANTIAGO ANDRÉS**, con cédula de identidad N° 1725044588 y **ESTÉVEZ CAMPOVERDE, MARCO ANTONIO**, con cédula de identidad N° 1723295588, declaramos que el contenido, ideas y criterios del trabajo de titulación: **“DISEÑO Y CONSTRUCCIÓN DE UN ROBOT PUMA CON UN SISTEMA DE SIMULACIÓN EN TIEMPO REAL”** es de nuestra autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Consecuentemente el contenido de la investigación mencionada es veraz.

Sangolquí, 29 de julio de 2019

ALLAUCA FAJARDO  
SANTIAGO ANDRÉS

ESTÉVEZ CAMPOVERDE  
MARCO ANTONIO



# ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA  
CARRERA DE INGENIERÍA MECATRÓNICA**

**AUTORIZACIÓN**

Nosotros, **ALLAUCA FAJARDO, SANTIAGO ANDRÉS y ESTÉVEZ CAMPOVERDE, MARCO ANTONIO** autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **“DISEÑO Y CONSTRUCCIÓN DE UN ROBOT PUMA CON UN SISTEMA DE SIMULACIÓN EN TIEMPO REAL”** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

Sangolquí, 29 de julio de 2019

ALLAUCA FAJARDO  
SANTIAGO ANDRÉS

ESTÉVEZ CAMPOVERDE  
MARCO ANTONIO

## **DEDICATORIA**

A mis padres por apoyarme en este camino de conseguir mis metas, por el apoyo económico y por el apoyo emocional, que como padres inculcaron en mi valores como responsabilidad, honestidad y respeto.

A mis hermanos para que se motiven en conseguir sus metas y para las demás personas que me ayudaron indirectamente en este camino.

Santiago Allauca

## **DEDICATORIA**

A mi familia, quienes siempre estuvieron a mi lado brindándome su apoyo incondicional.

Marco Estévez

## **AGRADECIMIENTO**

Agradezco a mi madre por el apoyo de estar conmigo en todo momento, y ser mi guía y soporte en esta trayectoria. A mi padre por ser parte de este logro y que con su conocimiento guiarme. A mis hermanos por ayudarme y estar conmigo, aparte de ser un motivo más para superarme.

A todas las personas que formaron parte de esta trayectoria a conseguir una de mis metas, amigos y tutor.

Santiago Allauca

## **AGRADECIMIENTO**

A mi familia, por ser los pilares de mi vida y estar a mi lado en todo momento, con quienes he reído y llorado, superado adversidades y quienes, a pesar de todo problema, siempre han sabido demostrarme su comprensión, su apoyo incondicional y su amor.

A mis amigos, con quienes viví las mejores experiencias en mi vida académica y fuera de ella, y que siempre me supieron brindar ayuda en los momentos de mayor preocupación.

Marco Estévez



## ÍNDICE DE CONTENIDOS

<b>CERTIFICACIÓN</b> .....	i
<b>AUTORÍA DE RESPONSABILIDAD</b> .....	ii
<b>AUTORIZACIÓN</b> .....	iii
<b>DEDICATORIA</b> .....	iv
<b>DEDICATORIA</b> .....	v
<b>AGRADECIMIENTO</b> .....	vi
<b>AGRADECIMIENTO</b> .....	vii
<b>ÍNDICE DE CONTENIDOS</b> .....	viii
<b>ÍNDICE DE TABLAS</b> .....	xiii
<b>ÍNDICE DE FIGURAS</b> .....	xv
<b>RESUMEN</b> .....	xix
<b>ABSTRACT</b> .....	xx
<b>CAPÍTULO 1</b> .....	1
<b>GENERALIDADES</b> .....	1
1.1 Antecedentes.....	1
1.1.1 Robótica industrial .....	1
1.1.2 Robot tipo PUMA .....	1
1.1.3 Robot con sistema de teleoperación .....	3

1.1.4 Realidad virtual .....	3
1.2 Justificación e importancia.....	4
1.3 Alcance.....	5
1.3.1 Componentes mecánicos .....	5
1.3.2 Componentes electrónicos y eléctricos .....	6
1.3.3 Componentes de software .....	7
1.4 Objetivos .....	7
1.4.1 Objetivo general .....	7
1.4.2 Objetivos específicos.....	7
1.5 Estructura del documento .....	8
<b>CAPÍTULO 2</b> .....	<b>9</b>
<b>ESTADO DEL ARTE</b> .....	<b>9</b>
2.1 Robot tipo PUMA .....	9
2.1.1 Definición.....	9
2.1.2 Aplicaciones del robot PUMA .....	13
2.2 Telerobótica.....	15
2.2.1 Historia de la telerobótica.....	16
2.2.2 Tipos de control de la telerobótica .....	19
2.2.3 Aplicaciones de la telerobótica.....	23

2.3 Software de simulación .....	25
2.3.1 Simuladores de brazos robóticos .....	25
2.4 Resumen .....	34
<b>CAPÍTULO 3</b> .....	<b>35</b>
<b>DISEÑO E IMPLEMENTACIÓN</b> .....	<b>35</b>
3.1 Requerimientos del proyecto .....	35
3.2 Subsistema Robot PUMA .....	37
3.2.1 Diseño del Robot PUMA .....	37
3.2.2 Desarrollo de conceptos del subsistema Robot PUMA.....	53
3.2.2.1 Categoría ‘Material del brazo’ del subsistema Robot PUMA.....	53
3.2.2.2 Categoría ‘Material de la muñeca’ del subsistema Robot PUMA .....	55
3.2.2.3 Categoría ‘Material de la base’ del subsistema Robot PUMA.....	55
3.2.2.4 Categoría ‘Aplicación del efector final’ del subsistema Robot PUMA .....	56
3.2.2.5 Categoría ‘Actuadores’ del subsistema Robot PUMA.....	56
3.1.2 Selección de conceptos del subsistema Robot PUMA .....	57
3.3 Subsistema Teleoperación .....	57
3.3.1 Diseño de la Teleoperación .....	57
3.3.1.1 Diseño de la HMI .....	57
3.3.1.2 Diseño de los nodos de comunicación y control .....	65

3.3.1.3 Comunicación y control desde la HMI hacia los servomotores .....	75
3.3.2 Desarrollo de conceptos del subsistema Teleoperación .....	77
3.3.2.1 Categoría ‘Fase de control’ del subsistema Teleoperación .....	78
3.3.2.2 Categoría ‘Fase de potencia’ del subsistema Teleoperación.....	78
3.3.2.3 Categoría ‘HMI’ del subsistema Teleoperación.....	79
3.3.3 Selección de conceptos del subsistema Teleoperación .....	79
3.4 Subsistema Simulación .....	80
3.4.1 Diseño del subsistema Simulación.....	80
3.4.1.1 Diseño del simulador del Robot PUMA .....	80
3.4.1.2 Diseño del control del simulador mediante ROS .....	83
3.4.2 Desarrollo de conceptos del subsistema Simulación.....	84
3.4.2.1 Categoría ‘Software de simulación’ del subsistema Simulación .....	85
3.4.2.2 Categoría ‘Tipo de simulación’ del subsistema Simulación .....	86
3.4.3 Selección de conceptos del subsistema Simulación.....	86
3.5 Resumen .....	86
<b>CAPÍTULO 4.....</b>	<b>88</b>
<b>PRUEBAS Y RESULTADOS .....</b>	<b>88</b>
4.1 Robot PUMA.....	88
4.2 Simulador .....	93

4.3 Comunicación.....	94
4.4 Resumen .....	95
<b>CAPÍTULO 5.....</b>	<b>96</b>
<b>CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS.....</b>	<b>96</b>
5.1 Conclusiones .....	96
5.2 Recomendaciones.....	98
5.3 Trabajos futuros.....	99
<b>BIBLIOGRAFÍA.....</b>	<b>100</b>

## ÍNDICE DE TABLAS

<b>Tabla 1</b> <i>Parámetros de Denavit-Hartenberg para el robot PUMA</i> .....	12
<b>Tabla 2</b> <i>Ventajas y desventajas de los tipos de control de la telerobótica</i> .....	22
<b>Tabla 3</b> <i>Comparación de diferentes softwares de simulación para brazos robóticos</i> .....	33
<b>Tabla 4</b> <i>Matriz QFD para el Proyecto</i> .....	36
<b>Tabla 5</b> <i>Parámetros de Denavit-Hartenberg para el robot PUMA</i> .....	42
<b>Tabla 6</b> <i>Cálculo del error porcentual para las ecuaciones de la cinemática inversa</i> .....	46
<b>Tabla 7</b> <i>Selección de servomotores en función de los torques dinámicos</i> .....	53
<b>Tabla 8</b> <i>Matriz morfológica de los conceptos para el subsistema Robot PUMA</i> .....	53
<b>Tabla 9</b> <i>Matriz de selección de criterios para la categoría ‘Material del brazo’ del subsistema Robot PUMA</i> .....	54
<b>Tabla 10</b> <i>Calificación de las opciones de la categoría ‘Material del brazo’</i> .....	54
<b>Tabla 11</b> <i>Matriz de Holmes de la categoría ‘Material del brazo’ para el subsistema Robot PUMA</i> .....	55
<b>Tabla 12</b> <i>Matriz de Holmes de la categoría ‘Material de la muñeca’ para el subsistema Robot PUMA</i> .....	55
<b>Tabla 13</b> <i>Matriz de Holmes de la categoría ‘Material de la base’ para el subsistema Robot PUMA</i> .....	56
<b>Tabla 14</b> <i>Matriz de Holmes de la categoría ‘Aplicación del efector final’ para el subsistema Robot PUMA</i> .....	56
<b>Tabla 15</b> <i>Matriz de Holmes de la categoría ‘Actuadores’ para el subsistema Robot PUMA</i> .....	57
<b>Tabla 16</b> <i>Conceptos seleccionados para cada categoría del subsistema Robot PUMA</i> .....	57

<b>Tabla 17</b> <i>Matriz morfológica de los conceptos para el subsistema Teleoperación</i> .....	78
<b>Tabla 18</b> <i>Matriz de Holmes de la categoría ‘Fase de control’ para el subsistema Teleoperación</i> .....	78
<b>Tabla 19</b> <i>Matriz de Holmes de la categoría ‘Fase de potencia’ para el subsistema Teleoperación</i> .....	79
<b>Tabla 20</b> <i>Matriz de Holmes de la categoría ‘HMI’ para el subsistema Teleoperación</i> .....	79
<b>Tabla 21</b> <i>Conceptos seleccionados para cada categoría del subsistema Teleoperación</i> .....	79
<b>Tabla 22</b> <i>Matriz morfológica de los conceptos para el subsistema Teleoperación</i> .....	85
<b>Tabla 23</b> <i>Matriz de Holmes de la categoría ‘Software de simulación’ para el subsistema Simulación</i> .....	85
<b>Tabla 24</b> <i>Matriz de Holmes de la categoría ‘Tipo de simulación’ para el subsistema Simulación</i> .....	86
<b>Tabla 25</b> <i>Conceptos seleccionados para cada categoría del subsistema Teleoperación</i> .....	86
<b>Tabla 26</b> <i>Velocidades nominales de los servomotores</i> .....	89
<b>Tabla 27</b> <i>Velocidades de los servomotores para la segunda prueba</i> .....	89
<b>Tabla 28</b> <i>Velocidades de los servomotores para la tercera prueba</i> .....	90
<b>Tabla 29</b> <i>Rango de valores de rotación para los servomotores del Robot PUMA</i> .....	91
<b>Tabla 30</b> <i>Tiempo de respuesta del simulador y del robot PUMA</i> .....	95

## ÍNDICE DE FIGURAS

<b>Figura 1.</b> Robot PUMA de seis grados de libertad.....	2
<b>Figura 2.</b> Estructura general del subsistema mecánico .....	6
<b>Figura 3.</b> Estructura general del subsistema electrónico y eléctrico .....	6
<b>Figura 4.</b> Estructura general del subsistema de software .....	7
<b>Figura 5.</b> Robot PUMA en la línea de ensamblaje de la empresa General Motors, tomada en el año de 1979.....	10
<b>Figura 6.</b> Relación existente entre la cinemática directa y la cinemática inversa de un robot .....	11
<b>Figura 7.</b> Representación de los sistemas de referencia para el robot PUMA .....	12
<b>Figura 8.</b> Prostatectomía asistida por tres brazos robóticos .....	14
<b>Figura 9.</b> Proceso de soldadura TIG realizado por un brazo robótico.....	15
<b>Figura 10.</b> Robot da Vinci controlado remotamente por un cirujano.....	16
<b>Figura 11.</b> Sistema teleoperado desarrollado por Raymond Goertz, que permite la manipulación de materiales radioactivos .....	17
<b>Figura 12.</b> Sistema telerobótico CRL Model 2 para verificar las estructuras espaciales de armadura .....	18
<b>Figura 13.</b> Primer sistema telerobótico enviado al espacio, ROTEX.....	19
<b>Figura 14.</b> Robot controlado por el astronauta alemán Alexander Gerst .....	20
<b>Figura 15.</b> Robot aéreo controlado para la detección y detonación de minas personales .....	22
<b>Figura 16.</b> PackBot diseñado por iRobot .....	24
<b>Figura 17.</b> Robot Rover, situado en Marte, es un telerobot que permite estudiar el planeta Marte desde la Tierra .....	24



<b>Figura 18.</b> Simulador Brazo Robot, donde se muestra un brazo robótico virtual .....	26
<b>Figura 19.</b> Variación del simulador Brazo Robot, conocida como DiezCubos, que simula aplicaciones de pick and place .....	27
<b>Figura 20.</b> Interfaz del software de simulación de un brazo robótico de cuatro grados de libertad .....	28
<b>Figura 21.</b> Interfaz del software de simulación Webots, desarrollado por Cyberbotics Ltd. ....	29
<b>Figura 22.</b> Simulación robótica 2D realizada en Player Stage .....	30
<b>Figura 23.</b> Simulación realizada en Microsoft Robotics Studio.....	31
<b>Figura 24.</b> Simulación de un robot en Marilou Robotics .....	32
<b>Figura 25.</b> Brazo robótico simulado en Roboworks.....	33
<b>Figura 26.</b> Primer diseño CAD del prototipo del robot PUMA .....	38
<b>Figura 27.</b> Segundo diseño CAD del prototipo del robot PUMA .....	39
<b>Figura 28.</b> Representación de las variables articulares del robot PUMA .....	40
<b>Figura 29.</b> Representación de los sistemas de referencia para el robot PUMA .....	41
<b>Figura 30.</b> Torque del primer actuador en función de la dinámica .....	49
<b>Figura 31.</b> Torque del segundo actuador en función de la dinámica.....	50
<b>Figura 32.</b> Torque del tercer actuador en función de la dinámica.....	51
<b>Figura 33.</b> Torque del cuarto actuador en función de la dinámica.....	51
<b>Figura 34.</b> Torque del quinto actuador en función de la dinámica.....	52
<b>Figura 35.</b> Ventana de inicio de la HMI de control.....	58
<b>Figura 36.</b> Ventana principal de la HMI de control .....	59
<b>Figura 37.</b> Ventana de cinemática directa del Robot PUMA de la HMI de control .....	60
<b>Figura 38.</b> Ventana de cinemática inversa del Robot PUMA de la HMI de control.....	61

<b>Figura 39.</b> Ventana de control por Joystick de la HMI de control .....	62
<b>Figura 40.</b> Ventana de Dibujo de la HMI de control.....	63
<b>Figura 41.</b> Ventana de carga de archivos de la HMI de control.....	64
<b>Figura 42.</b> Arquitectura de los nodos de control del subsistema Teleoperación.....	65
<b>Figura 43.</b> Diagrama de flujo del nodo de comunicación entre ROS y Blender .....	66
<b>Figura 44.</b> Diagrama de flujo de la cinemática directa del Robot PUMA .....	67
<b>Figura 45.</b> Diagrama de flujo de la cinemática inversa del Robot PUMA.....	68
<b>Figura 46.</b> Diagrama de flujo del control por joystick del Robot PUMA (primera parte).....	69
<b>Figura 47.</b> Diagrama de flujo del control por joystick del Robot PUMA (segunda parte) .....	71
<b>Figura 48.</b> Diagrama de flujo del nodo de dibujo (primera parte) .....	72
<b>Figura 49.</b> Diagrama de flujo del nodo de dibujo (segunda parte).....	73
<b>Figura 50.</b> Diagrama de flujo del nodo exportación de archivos .....	74
<b>Figura 51.</b> Diagrama de flujo del nodo de comunicación entre la HMI y la placa Raspberry /simulador.....	76
<b>Figura 52.</b> Diagrama de flujo del nodo receptor de la placa Raspberry.....	77
<b>Figura 53.</b> Modelo del simulador realizado en Blender v2.79 .....	80
<b>Figura 54.</b> Modelo renderizado del simulador realizado en Blender v2.79 .....	81
<b>Figura 55.</b> Modelo del simulador con armadura formada por Bones.....	82
<b>Figura 56.</b> Bloqueo de rotaciones para los Bones del simulador .....	83
<b>Figura 57.</b> Diagrama de flujo del nodo de control del simulador .....	84
<b>Figura 58.</b> Robot PUMA diseñado.....	88
<b>Figura 59.</b> Espacio de trabajo para el robot PUMA .....	91
<b>Figura 60.</b> Dibujo realizado por el Robot PUMA .....	92

**Figura 61.** Segunda prueba de dibujo realizada con el robot PUMA..... 93

**Figura 62.** Renderizado final del simulador del Robot PUMA..... 94

## RESUMEN

El uso de robots a nivel industrial requiere de un alto nivel de conocimientos y experiencia para evitar accidentes causados por negligencias de parte del operario. Sin embargo, en muchas ocasiones, estudiantes universitarios que siguen carreras afines a la robótica no tienen el tiempo necesario de preparación en el uso y manejo de robots a nivel industrial, por lo cual no consiguen la experticia necesaria para desenvolverse de la manera adecuada. La presente investigación tuvo como principal objetivo desarrollar un simulador que permita replicar los movimientos de un robot PUMA en tiempo real. La elección del robot PUMA fue debido a su alta versatilidad y aplicabilidad en diferentes ramas, principalmente la industria. El primer paso de la investigación fue el diseño y la construcción de un robot PUMA con propósitos educativos, de dimensiones y características menores a los modelos industriales, pero con una forma y funciones similares. Posteriormente, se desarrolló el simulador en Blender y una HMI de control en PyQt Designer. Para lograr la comunicación y el control entre el sistema real y el sistema virtual se realizó algoritmos basados en la cinemática directa y la cinemática inversa del robot PUMA diseñado, utilizados en nodos que se desarrollaron en la plataforma ROS.

### **PALABRAS CLAVE:**

- **ROBOT PUMA**
- **TELEOPERACIÓN**
- **SIMULADOR**
- **TIEMPO REAL**

## **ABSTRACT**

The use of robots at industrial level requires a high level of knowledge and experience to avoid accidents caused by negligence on the part of the operator. However, on many occasions, university students who follow robotics-related careers do not have the necessary time to prepare and use robots at the industrial level, so they do not get the necessary expertise to function properly. The main objective of the present investigation was to develop a simulator that allows replicating the movements of a PUMA robot in real time. The choice of the PUMA robot was due to its high versatility and applicability in different branches, mainly the industry. The first step of the research was the design and construction of a PUMA robot for educational purposes, of smaller dimensions and characteristics than the industrial models, but with a similar shape and functions. Subsequently, the simulator was developed in Blender and a control HMI in PyQt Designer. To achieve communication and control between the real system and the virtual system, algorithms based on the direct kinematics and the inverse kinematics of the designed PUMA robot were applied, used in nodes that were developed in the ROS platform.

### **KEYWORDS:**

- **ROBOT PUMA**
- **TELEOPERATION**
- **SIMULATOR**
- **REAL TIME**

## **CAPÍTULO 1**

### **GENERALIDADES**

El capítulo expone los antecedentes que dieron origen al problema de investigación, a través de una revisión bibliográfica del tema de estudio. También se detalla la importancia y la justificación sobre la cual se asienta el estudio de la presente investigación. Además, se delimita el alcance, se muestran los objetivos a cumplir y se presenta una estructura general del documento.

#### **1.1 Antecedentes**

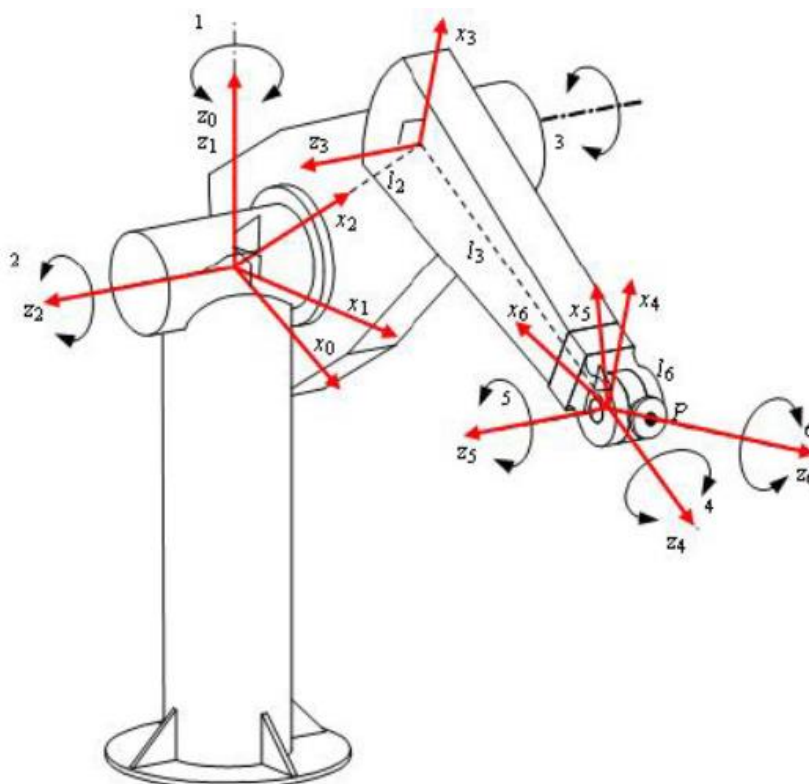
##### **1.1.1 Robótica industrial**

Según Avilés et al. (1999) un robot industrial es un manipulador reprogramable y multifuncional que realiza funciones de movilidad de cargas, herramientas o dispositivos especiales. Esta definición excluye a aquellos manipuladores que no tienen la flexibilidad necesaria para cumplir con diferentes tareas sin cambiar su estructura mecánica. Un robot industrial tiene como principal característica la reprogramación, lo que brinda una gran versatilidad en ambientes industriales.

##### **1.1.2 Robot tipo PUMA**

El robot tipo PUMA (Programmable Universal Machine for Assembly o Programmable Universal Manipulation Arm) es, según Cortés (2008), un brazo articulado que consta de seis articulaciones rotatorias, las cuales permiten establecer la pose al efector final. En la Figura 1 se muestra un ejemplo de robot PUMA, donde se ven los seis giros que dan la movilidad al manipulador. La estructura del robot se compone por el sistema Hombro – Codo – Muñeca, que son las tres primeras articulaciones rotacionales, el cual se encarga de posicionar al efector final en

la posición deseada, además que estas articulaciones son las más importantes al definir la trayectoria deseada. El otro componente del brazo robótico es el efector final, el cual consiste en las tres últimas articulaciones rotatorias, las cuales permiten establecer la orientación para alcanzar el punto de interés final.



**Figura 1.** Robot PUMA de seis grados de libertad

Fuente: (Harib et al., 2012)

La gran versatilidad de estos manipuladores ha permitido su implementación en diferentes procesos industriales, como en líneas de ensamblaje, procesos de soldadura, aplicaciones pick and place. La presente investigación se centrará en esta última aplicación, la cual consiste en el transporte de material de una banda a otra, con la característica que el producto se mantenga en movimiento constante.

### **1.1.3 Robot con sistema de teleoperación**

Los robots con sistemas de teleoperación, según Cerón et al. (2005), son aquellos controlados a distancia desde una estación remota. La principal ventaja que presentan estos sistemas de control es el asegurar la integridad física del teleoperador, pues estos sistemas se utilizan principalmente en áreas que atentan contra la seguridad de las personas. Un sistema teleoperado está compuesto por una estación de teleoperación, un sistema de comunicación y por un esclavo, que en este caso es el manipulador robótico. La estación permite controlar los movimientos del robot, siendo el sistema de comunicación el responsable de enviar la información de control hacia el autómata.

Otro punto importante que considerar dentro de los sistemas de teleoperación es el control que se utiliza. Según Erfani et al. (2018), debido a la dinámica compleja que presentan los sistemas de teleoperación, el objetivo principal del controlador es alcanzar un rendimiento óptimo y la estabilidad en presencia de factores. Para el caso de un manipulador tipo PUMA, se utilizan sistemas de control en lazo cerrado, con una retroalimentación de la señal de posición para controlar el estado de los diferentes componentes del robot, con el uso de controladores PID para cada variable articular del mecanismo.

### **1.1.4 Realidad virtual**

Los sistemas virtuales han sido un gran avance en la colaboración entre humanos y robots, y han presentado ventajas en procesos industriales complejos, según Rückert et al. (2018), pues estos sistemas permiten simular los procesos industriales, haciendo a estos más seguros para los operadores. La realidad virtual permite entender los procesos y determinar los parámetros y variables implicadas en estos, a través de simulaciones. Otra de las ventajas que presentan estos



sistemas es que permiten determinar posibles errores, y estos se pueden prevenir al momento de pasar al proceso real.

Dentro del campo de la robótica industrial, los sistemas virtuales más comunes son los sistemas desktop de realidad virtual y la telepresencia, según BBVA (2015). La telepresencia es una aplicación de la realidad virtual que se combina principalmente con los campos de la medicina y la robótica, donde periféricos externos, junto a un sistema de control remoto permite manipular robots a distancia. Por otro lado, los sistemas desktop de realidad virtual muestran una imagen en 2D o 3D en una pantalla de computador y el usuario interactúa directamente con esta.

## **1.2 Justificación e importancia**

La robótica industrial es una materia cuyo objetivo es analizar el comportamiento de los robots industriales, sus características, ventajas, aplicaciones, entre otros. Una forma de familiarizar a los estudiantes con estos autómatas es el trabajo práctico, donde se desarrolla habilidades en el manejo de los robots a nivel industrial. La principal desventaja de los robots industriales es su baja versatilidad en cuanto al desplazamiento del robot en sí, debido a que estos tienen dimensiones grandes y un peso elevado; esto hace que su espacio de trabajo se vea muy limitado al lugar donde se lo instaló para su uso.

El presente trabajo se enfocará en la construcción de un robot Puma, cuyas dimensiones permitan su desplazamiento con facilidad, para utilizarlo como apoyo de un robot industrial. Adicionalmente, se desarrollará un simulador que permita su teleoperación, de tal manera que se disponga de un sistema de simulación para realizar trabajos previos de programación sobre el robot, para verificar el desempeño del código realizado, esto previo a su implementación en el robot físico.

La ventaja del simulador esto es evitar accidentes que podrían ocurrir al trabajar con el robot industrial sin experiencia previa, lo que podría ocasionar daño al operador o al autómatas

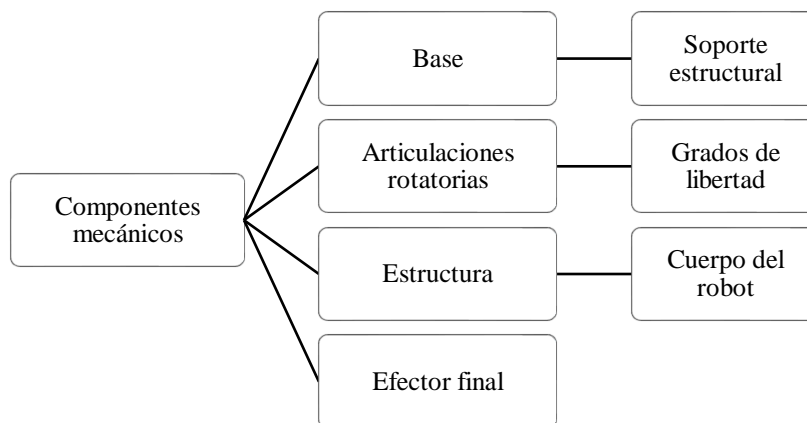
Dentro del campo de la robótica industrial, los robots Puma presentan una gran versatilidad al momento de realizar trabajos industriales, pues se pueden utilizar en aplicaciones como montaje y desmontaje de equipos, procesos de soldadura, procesos pick and place, entre otros. Esta es la principal razón por la cual en el presente trabajo se optó por construir un robot de este tipo. Igualmente, la información existente para los manipuladores Puma es muy basta, lo que facilitará el trabajo de diseño y construcción, así como de seleccionar los materiales adecuados y la instrumentación necesaria para su operación correcta.

### **1.3 Alcance**

#### **1.3.1 Componentes mecánicos**

En la Figura 2 se muestra el esquema general de los componentes mecánicos del presente proyecto, el cual se compone de las siguientes estructuras básicas:

- Base o tierra: se realizará de metal para garantizar una sujeción de todo el sistema.
- Articulaciones rotatorias (Hombro – Codo – Muñeca): las articulaciones se las realizará en impresión 3D.
- Estructura general: se realizará en impresión 3D y constituye el brazo robótico en general.
- Efecto final: se realizará en impresión 3D y se orientará a aplicación de pick and place

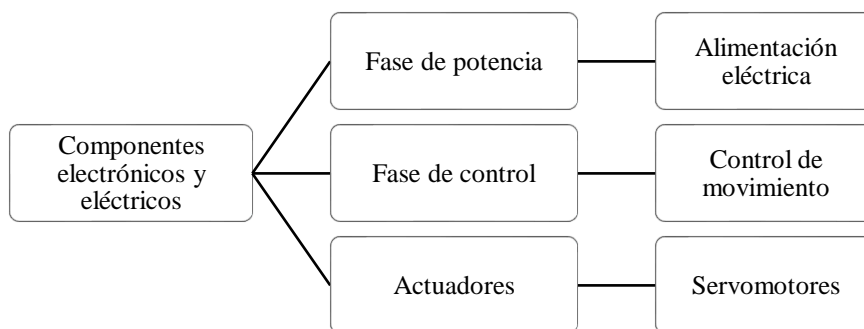


**Figura 2.** Estructura general del subsistema mecánico

### 1.3.2 Componentes electrónicos y eléctricos

En la Figura 3 se muestra el esquema general del sistema electrónico y eléctrico. Los componentes de este sistema son los siguientes:

- Fase de Potencia: subsistema que alimenta a los componentes del robot.
- Fase de control: subsistema que controla las acciones del robot.
- Actuadores: servomotores que permiten la movilidad del robot.

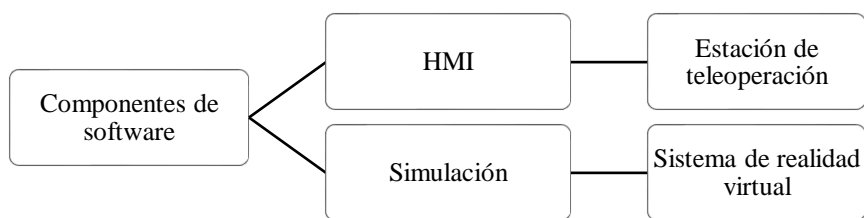


**Figura 3.** Estructura general del subsistema electrónico y eléctrico

### 1.3.3 Componentes de software

En la Figura 4 se muestran los componentes del sistema de software:

- HMI: constituye la estación de teleoperación. Esta interfaz permitirá realizar el control del robot y el simulador, o únicamente del simulador.
- Sistema de simulación: simulador del robot, donde se observará la trayectoria realizada por el robot. La simulación será cinemática, debido a que la posición del robot es la variable más importante para determinar a lo largo de su trayectoria, el análisis dinámico se lo realizará para el cálculo de los actuadores necesarios para garantizar la estabilidad cinemática del sistema.



*Figura 4.* Estructura general del subsistema de software

## 1.4 Objetivos

### 1.4.1 Objetivo general

Diseñar y construir un robot puma teleoperado con un sistema de simulación en tiempo real

### 1.4.2 Objetivos específicos

- Analizar la obtención del modelo estático y dinámico de un robot puma de 6 articulaciones rotatorias actuado por servomotores.

- Estudiar e implementar un sistema de control basado en algoritmos de movimiento en base a la cinemática directa e inversa del robot puma.
- Desarrollar un sistema de simulación para el robot puma que permita realizar operaciones fuera de línea y su teleoperación.

### **1.5 Estructura del documento**

El presente documento recopila la información utilizada para la realización de la investigación, y lo resume en 5 capítulos. El capítulo 1 muestra las generalidades sobre las cuales se asienta la investigación, como investigaciones previas sobre simuladores robóticos expuestos en los antecedentes. También se justifica la importancia de desarrollar el proyecto dentro del alcance y los objetivos especificados.

El capítulo 2 corresponde a la fundamentación teórica sobre los conceptos fundamentales del tema de investigación del presente proyecto, como lo es la robótica industrial, el robot PUMA, la teleoperación y los sistemas simulados.

El capítulo 3 presenta el diseño realizado para el desarrollo del Robot PUMA, la teleoperación y la simulación. Además, se muestra la implementación a través de software para el control de los sistemas, así como la comunicación de los mismos.

El capítulo 4 expone las pruebas realizadas y los resultados de las mismas, en función al diseño implementado. Aquí también se exponen los valores utilizados y permisibles para el control óptimo de los sistemas.

Por último, en el capítulo 5 se detallan las conclusiones y las recomendaciones que se logró obtener después de desarrollar el proyecto de investigación.

## **CAPÍTULO 2**

### **ESTADO DEL ARTE**

El capítulo expone las bases científicas y tecnológicas sobre las cuales se desarrolló el proyecto. Se presenta definiciones de temas claves dentro de los robots PUMA en la robótica industrial, la telerobótica y los sistemas de simulación para robots. Se incluyen temas que abarcan la historia, desarrollo, aplicaciones y usos dentro de los campos industrial y educativo, para los tres grupos mencionados.

#### **2.1 Robot tipo PUMA**

##### **2.1.1 Definición**

Según Avilés et al. (1999) un robot industrial es un manipulador reprogramable y multifuncional que realiza funciones de movilidad de cargas, herramientas o dispositivos especiales. La definición excluye a aquellos manipuladores que no tienen la flexibilidad necesaria para cumplir con diferentes tareas sin cambiar su estructura mecánica. Un robot industrial tiene como principal característica la reprogramación, lo que brinda una gran versatilidad en ambientes industriales. Uno de los primeros robots industriales producidos fue el robot PUMA, desarrollado por la empresa Unimation en el año de 1979, el cual se muestra en la Figura 5, modelo que fue desarrollado en conjunto y vendido a la empresa General Motors para la línea de ensamblaje.

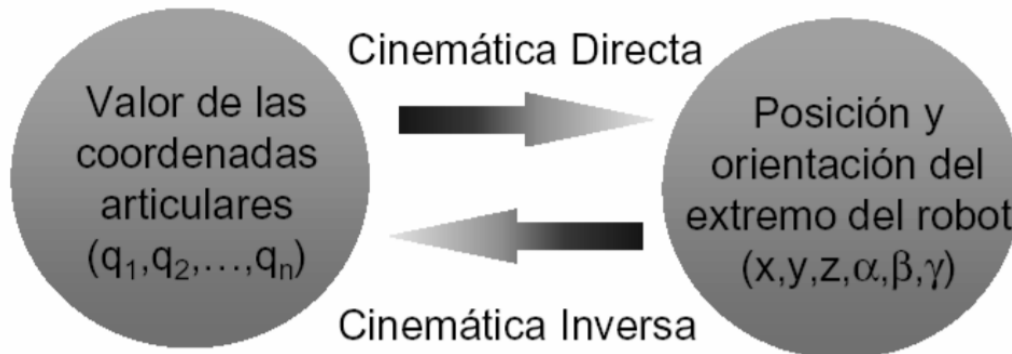


**Figura 5.** Robot PUMA en la línea de ensamblaje de la empresa General Motors, tomada en el año de 1979  
Fuente: (The Historic Images Outlet, 2009)

Según Cortés (2008), un robot PUMA (Programmable Universal Machine for Assembly) es un brazo articulado con seis articulaciones rotatorias, de las cuales sus tres primeras rotaciones permiten establecer la posición y las últimas tres rotaciones le dan la orientación a un efector final. Los robots PUMA se fabricaron en primera instancia para asemejar el movimiento de un brazo humano (articulaciones de hombro, codo y muñeca), y tienen una gran variedad de aplicaciones en la actualidad, aunque originalmente se los desarrolló para el ensamblaje en líneas de producción. El robot tipo PUMA ha sido de los más populares en la historia de la robótica industrial, debido a su versatilidad y facilidad de uso.

El análisis de los robots permite determinar sus características estáticas y dinámicas, lo cual se utiliza en todos los robots industriales, incluyendo al robot PUMA. El objetivo del análisis es determinar la cinemática directa, cinemática inversa, torques requeridos, dimensionamiento de

sensores y actuadores. La cinemática directa, según Jaramillo (2005), es el paso desde las variables articulares de un robot, hacia su posición final, mientras que la cinemática inversa es el camino opuesto, en donde se va desde una posición deseada para encontrar el valor de las variables articulares que lo satisfaga. En la Figura 6 se muestra la relación existente entre ambas.

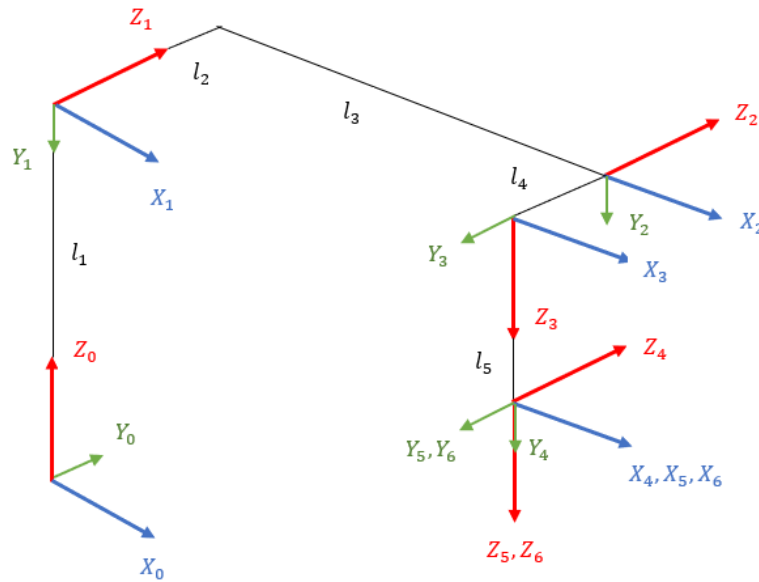


**Figura 6.** Relación existente entre la cinemática directa y la cinemática inversa de un robot

Fuente: (Universidad de Sevilla, 2018)

Una de las formas en que se realiza el análisis de la cinemática directa es a través del método de Denavit-Hartenberg (1955). El método fue propuesto para modelar la cinemática de robots de forma general, considerando ciertos parámetros dentro del mecanismo. En la Figura 7 se muestra la configuración de eslabones para el robot PUMA. Se puede ver los eslabones, que representan la estructura del autómatas, y los sistemas de referencia, los cuales representan la orientación de giro de las variables articulares.





**Figura 7.** Representación de los sistemas de referencia para el robot PUMA

En la Tabla 1 se muestran los parámetros de Denavit-Hartenberg para el robot PUMA de la Figura 7. Las variables articulares son representadas por  $\theta_i$ , las cuales corresponden a las seis rotaciones del autómat. Las longitudes de los diferentes componentes son  $l_i$ , que corresponden a los desplazamientos entre los sistemas de referencia. El sistema cuenta con un cambio de ejes de referencia desde el  $H_0$  hasta  $H_6$ , debido a los seis grados de libertad rotacionales presentes.

**Tabla 1**

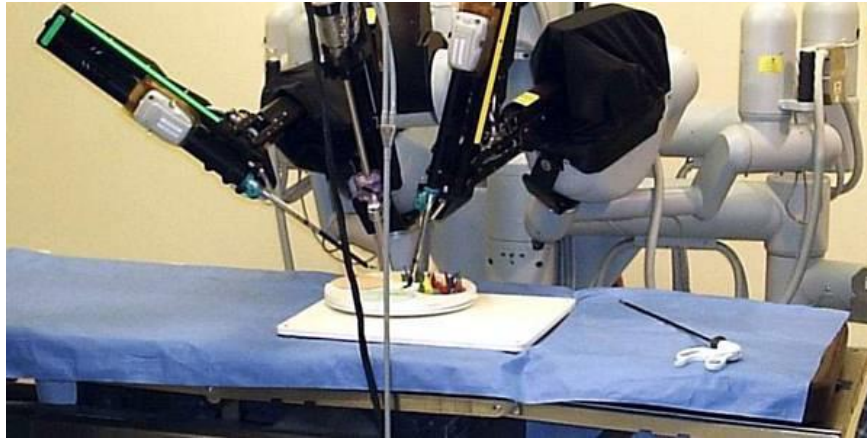
*Parámetros de Denavit-Hartenberg para el robot PUMA*

	$\theta$	$d$	$a$	$\alpha$
$H_0^1$	$\theta_1$	$l_1$	0	-90
$H_1^2$	$\theta_2$	$l_2$	$l_3$	0
$H_2^3$	$\theta_3$	$-l_4$	0	-90
$H_3^4$	$\theta_4$	$l_5$	0	90
$H_4^5$	$\theta_5$	0	0	-90
$H_5^6$	$\theta_6$	0	0	0

El robot PUMA presenta una ventaja mecánica, debido a su configuración, lo que permite realizar el análisis de la cinemática y la dinámica de una manera más fácil y eficiente. La complejidad del análisis de la cinemática inversa se reduce al utilizar un desacople cinemático, matemáticamente hablando, de este manipulador. El robot PUMA presenta 6 grados de libertad, siendo los 3 primeros para posicionar al efector final, y los 3 últimos para orientarlo, lo que permite alcanzar la pose del efector final. Ramírez (2012) muestra la ventaja de la característica de los robots de 6 grados de libertad, al separar el análisis de todos los grados de libertad, para analizar por separado la posición y la orientación. Para lograr este fin, se encuentran los 3 primeros valores de las variables articulares (posición). Encontrados los valores, se evalúa la matriz de transformación correspondiente a la posición del robot, y por último se determinan las últimas 3 variables articulares con la información previa.

### **2.1.2 Aplicaciones del robot PUMA**

Aunque originalmente el robot PUMA fue diseñado para realizar aplicaciones de ensamblaje, las aplicaciones del autómatas han crecido a lo largo de los años. Dentro del campo de la medicina, han sido varias las aplicaciones encontradas para el autómatas, como lo describen Davis et al (1991), donde se desarrolló un robot para realizar prostatectomías, la cual es una cirugía para remover tejido prostático, y la opción seleccionada fue un robot PUMA como punto de partida, gracias a sus seis grados de libertad que garantiza la posición y orientación del efector final. En la Figura 8 se muestra el avance que tuvo el procedimiento quirúrgico asistido por robots, en la actualidad se utilizan tres brazos robóticos para lograr este objetivo.



**Figura 8.** Prostatectomía asistida por tres brazos robóticos  
Fuente: (2 Minute Medicine, 2013)

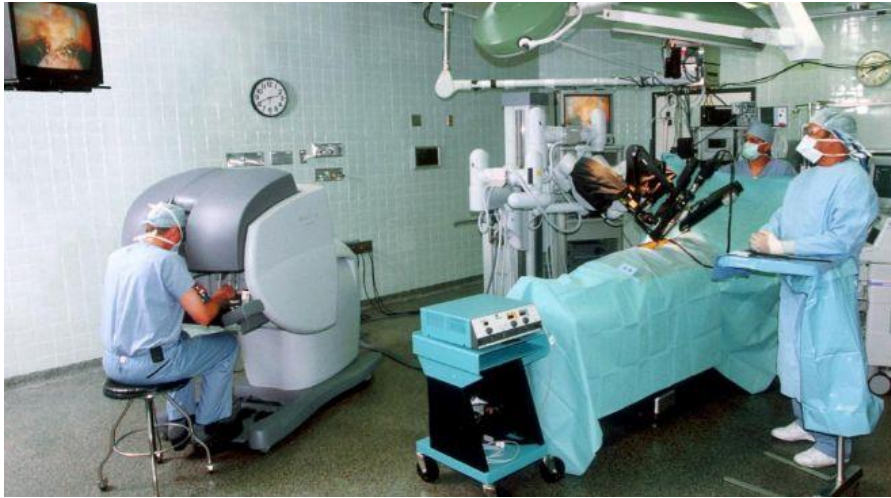
Dentro del campo industrial, el robot PUMA cumple con varias funciones dentro de los procesos de producción. La soldadura es uno de los campos en los cuales el robot PUMA se desenvuelve, como en la investigación realizada por Yang et al (2016), en la cual se desarrolló un control para la trayectoria del efector final del robot, debido a que, en el área de la soldadura, una de las variables más importantes es el recorrido del electrodo. En la Figura 9 se muestra un robot realizando un proceso de soldadura TIG, en donde el efector final es el electrodo que se encarga de realizar el arco de soldadura, y el resto del equipo soldador se encuentra montado sobre la estructura del robot.



**Figura 9.** Proceso de soldadura TIG realizado por un brazo robótico  
Fuente: (Serecon, 2016)

## 2.2 Telerobótica

La telerobótica, según Siciliano y Khatib (2016), es literalmente robótica a distancia, lo que se puede interpretar como el control de un autómatas por parte de un operario. La definición se puede complementar con la dada por Basañez (2015), que agrega el monitoreo y la reprogramación del robot a distancia, a parte del control del mismo. En resumen, la telerobótica es la operación de un robot a robot en un ambiente a distancia, donde el operario tiene la capacidad de manipular, reprogramar y monitorear las acciones que realiza el autómatas. En la Figura 10 se muestra un médico cirujano controlando un robot da Vinci en una interfaz que permite controlar los movimientos del robot.



**Figura 10.** Robot da Vinci controlado remotamente por un cirujano  
Fuente: (Expansión tecnología, 2017)

Los objetivos de la telerobótica son separar el espacio de trabajo del operador, conocido como el sitio local, y el espacio de trabajo del robot o sitio remoto. Las ventajas brindadas por esta separación son la seguridad del operario, la precisión del trabajo realizado por el autómata, aprovechar ciertas características del robot como fuerza, repetitividad, versatilidad bajo la supervisión de un humano.

### **2.2.1 Historia de la telerobótica**

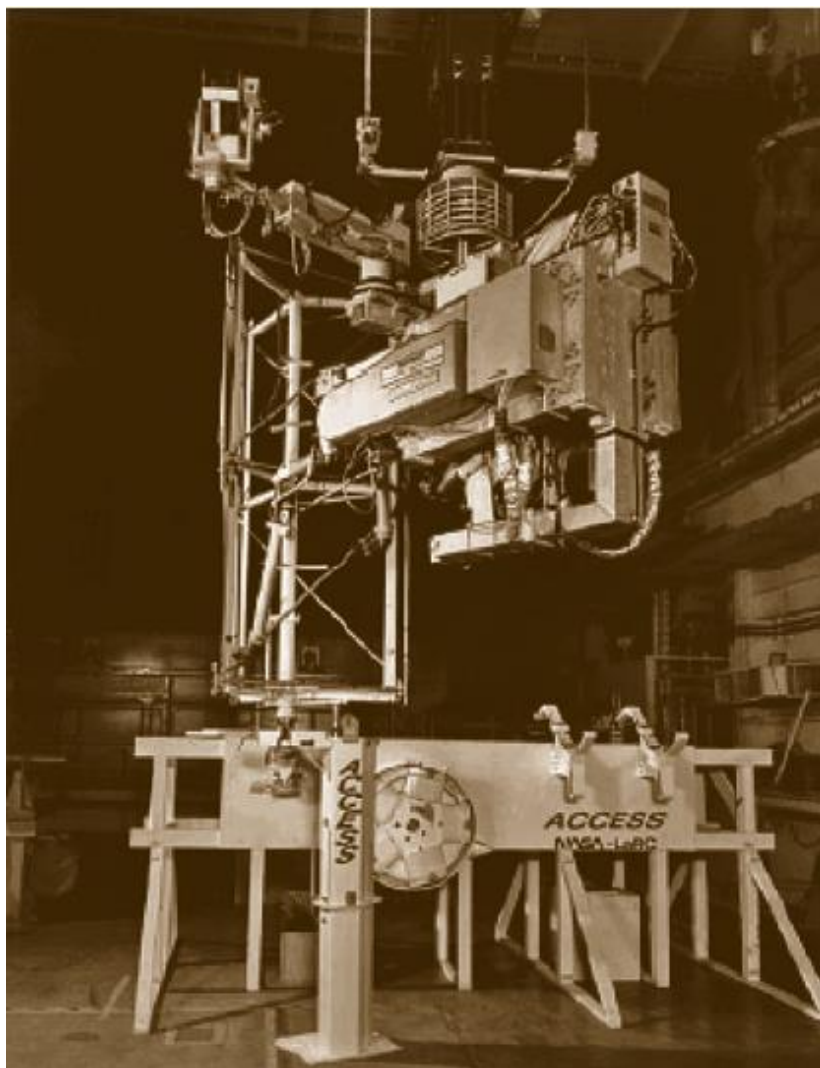
La telerobótica tuvo sus orígenes en la década de los 40's y 50's con la investigación de Raymond Goertz (1954), quien creó el primer robot teleoperado. El sistema permitía manipular material radioactivo, separando al robot del operario por una pared blindada. El autómata, como se muestra en la Figura 11, tiene un sistema de control eléctrico, compuesto por un conjunto de activadores on-off que controlan una serie de motores, los cuales transmiten los movimientos del operario hacia el robot.



**Figura 11.** Sistema teleoperado desarrollado por Raymond Goertz, que permite la manipulación de materiales radioactivos

Fuente: (Cyberneticzoo, 2014)

Los problemas existentes en el primer sistema teleoperado fue la lentitud de respuesta del robot frente a las acciones del operario. Otro problema fue el hecho que el sistema no era fácil de operar. Los dos puntos se fueron desarrollando con el paso de los años. En la obra de Ghosh, Xi y Tarn (1999) se menciona el sistema CRL Model M2, creado en el año de 1982 y el cual se muestra en la Figura 12, desarrollado en conjunto por el Laboratorio Nacional de Oak Ridge y la NASA. El sistema permite verificar el ensamblaje de las estructuras de armadura espacial, el modelo fue el primero en implementar un sistema de control teleoperado con un lazo cerrado forzado.



**Figura 12.** Sistema telerobótico CRL Model 2 para verificar las estructuras espaciales de armadura  
Fuente: (Corredor, 2011)

El campo aeroespacial tuvo un gran interés en la telerobótica y sus beneficios, por lo que esta tuvo un gran avance por parte de centros aeroespaciales en todo el mundo. Hirzinger et al (1994) publicaron un artículo sobre el robot ROTEX, desarrollado en el Centro Aeroespacial de Alemania. El robot fue el primero en ser enviado al espacio, y ser controlado remotamente, gracias a la implementación de lazos cerrados sensoriales, pantallas que permitían monitorear el estado del

robot y sistemas de teleoperación. En la Figura 13 se muestra el sistema de monitoreo y control de ROTEX siendo operado.



**Figura 13.** Primer sistema telerobótico enviado al espacio, ROTEX  
Fuente: (Artigas et al, 2012)

### **2.2.2 Tipos de control de la telerobótica**

La telerobótica se puede clasificar de acuerdo con la autonomía que presente el robot al momento de cumplir con las órdenes del operario. Bajo la definición existen tres tipos de control dentro de un sistema teleoperado: control manual o directo, control compartido y control supervisado. Cada tipo de control presenta su diferentes ventajas y aplicaciones en diferentes campos.



El control supervisado, según Ferrell y Sheridan (1967), es un sistema donde el robot presenta un nivel de inteligencia que le permite tomar decisiones sobre las acciones requeridas. El sistema presenta un alto nivel de automatización, donde el operador se limita a reprogramar al robot, y recibir información del mismo, es decir, monitorear su estado y acciones. El control se utiliza en aplicaciones donde el tiempo de respuesta es demasiado alto, especialmente en el campo aeroespacial, donde se desarrolló la programación de telesensores. En la Figura 14 se muestra al robot Justin, controlado por un astronauta alemán, Alexander Gerst. El robot se encuentra en la Tierra y es parte de un experimento para proporcionar asistencia médica en misiones en Marte.



**Figura 14.** Robot controlado por el astronauta alemán Alexander Gerst  
Fuente: (My Press, 2018)

El control directo por otra parte es un sistema en donde todas las acciones del robot son controladas por el operador, de tal manera que el robot no presenta ninguna autonomía en la toma de decisiones. Generalmente todos los sistemas presentan un control directo del robot, para evitar los problemas que pueden surgir en la autonomía local. En los sistemas que presentan control directo, el operador puede controlar la posición y velocidad del robot en todo momento, lo cual

requiere de una retroalimentación forzada al usuario, el cual controla al sistema con algún joystick o dispositivo de ingreso, y funciona como un master, y el robot como slave. La combinación de los dos sistemas es conocido como control compartido. Abbott et al (2007) hablan sobre el control compartido al compararlo con una regla:

Una línea recta dibujada por una persona con la ayuda de una regla es dibujada más rápido y recta que una línea dibujada a mano alzada. De manera similar, un robot puede aplicar fuerzas o posiciones a un operador humano para ayudarlo a dibujar una línea recta. (p. 1)

Básicamente esta definición habla sobre la ayuda que brinda el control compartido, al no dar toda la responsabilidad de las acciones al operador, el cual puede cometer errores de precisión. Las aplicaciones de estos sistemas se dan en procesos donde se requiera la precisión del robot, y el control del operador, como es el caso de los robots de cirugía, como el de la Figura 8. Otra aplicación son los sistemas conducidos, tales como los robots móviles, en donde el operador guía al robot y le da la trayectoria, mientras que el autómatas se encarga del control de los actuadores para seguir las instrucciones del operario. En la Figura 15 se muestra el robot aéreo PRODRONE, el cual se utiliza para la detección y detonación de minas personales, con la ayuda de dos brazos de agarre. Sistemas como el del robot PRODRONE tienen un control compartido en donde el operario da la trayectoria al robot, y este controla el funcionamiento de los motores para lograr alcanzar la posición deseada. En la Tabla 2 se muestra un resumen de los tipos de control dentro de la telerobótica: control directo, control asistido, control supervisado.



**Figura 15.** Robot aéreo controlado para la detección y detonación de minas personales  
Fuente: (Torrente Digital, 2016)

**Tabla 2**

*Ventajas y desventajas de los tipos de control de la telerobótica*

<b>Tipo de control</b>	<b>Ventajas</b>	<b>Desventajas</b>
<b>Directo</b>	Permite determinar el funcionamiento individual de los actuadores. No requiere lazos de control cerrado.	No es utilizable en aplicaciones reales. No permite monitorear el estado del robot.
<b>Asistido</b>	Permite un control de la posición del efector final del robot, independientemente del estado de los actuadores. Se utiliza en aplicaciones médicas y en robótica móvil.	Requiere cierto nivel de autonomía por parte del robot. El costo computacional puede ser elevado en ciertas ocasiones.
<b>Supervisado</b>	Los tiempos de retraso son bajos. Existe un monitoreo total y constante del robot, sus componentes y su entorno. Se utiliza en aplicaciones aeroespaciales y submarinas.	Requiere un alto nivel de autonomía por parte del robot. El costo económico y computacional es muy elevado.

### 2.2.3 Aplicaciones de la telerobótica

La telerobótica tiene ventajas, las cuales son aprovechadas en diferentes áreas, principalmente el mantener al operador lejos del área de trabajo o sitio remoto, lo que permite salvaguardar la seguridad del ser humano. Otras de las ventajas son el alcanzar espacios inaccesibles para las personas, aprovechar las características de fuerza y precisión de los robots, entre otras más. Los beneficios han hecho que varios campos se centren en el estudio y desarrollo de tecnología con la telerobótica.

El telerobot cirujano da Vinci, diseñado por Guthart y Salisbury (2000), que se muestra en la Figura 10, permite realizar cirugías mínimamente invasivas. El procedimiento presenta beneficios como cirugías con menor estancia hospitalaria, menor pérdida de sangre del paciente y por ende menor transfusiones necesarias. El sistema originalmente constaba de 3 brazos robóticos, manipulados por un cirujano a través de una consola, aunque los modelos más modernos presentan 4 brazos robóticos.

Las aplicaciones militares con tecnología de teleoperación se ven inmersas en varios de sus instrumentos y equipos. La detección de minas es un proceso crítico para la seguridad humana, por lo que se han diseñado robots que cumplan con este fin, como es el caso de PackBot, el cual se muestra en la Figura 16. Yamauchi (2004) expone en su trabajo el diseño de un robot que permita cumplir funciones de localización y desarme de minas u otros explosivos.



**Figura 16.** PackBot diseñado por iRobot  
Fuente: (TMCnet, 2010)

La robótica espacial también se ha beneficiado de los sistemas teleoperados, como es el caso del Rover de la NASA. En la obra de Wright et al (2005), se expone el uso de la teleoperación para la conducción del Rover, en Marte. El robot se muestra en la Figura 17, en el cual se hace uso de los sensores implementados en él para lograr visualizar el espacio del planeta, y conducir al vehículo por el mismo.



**Figura 17.** Robot Rover, situado en Marte, es un telerobot que permite estudiar el planeta Marte desde la Tierra  
Fuente: (Gizmodo, 2017)

## **2.3 Software de simulación**

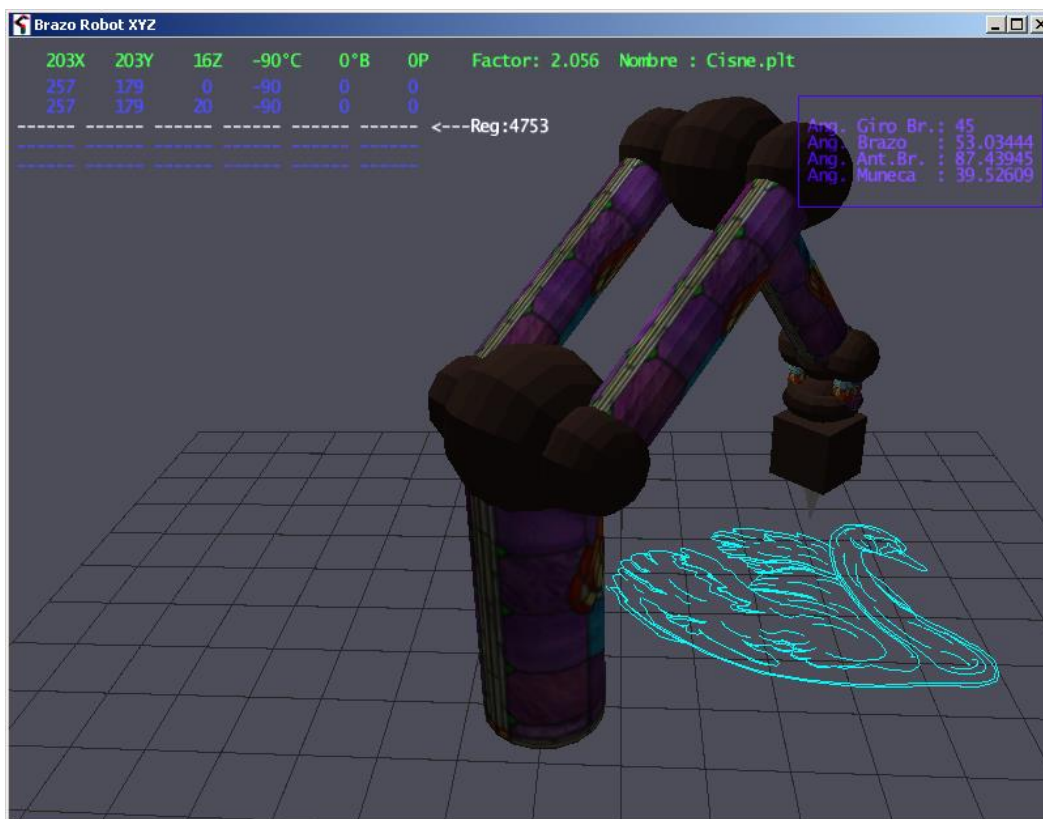
Una simulación es definida por Ingalls (2008) como diseñar el modelo de un sistema real de manera virtual, con el fin de entender el comportamiento del sistema, sin la necesidad de fabricarlo en la realidad. Una simulación puede entenderse como un sistema físico virtualizado, en el cual se puede realizar experimentos y mediciones, con el fin de obtener información real, sin la necesidad de construir o realizar las mediciones en el objeto o sistema físico real. Dentro de la robótica, la simulación es una herramienta poderosa.

Según Kumar y Narayan (2011) la simulación posee herramientas que permiten determinar el análisis cinemático y dinámico de robots, realizar programación fuera de línea, diseñar estructuras mecánicas, algoritmos de control, entre otras más. La programación fuera de línea constituye el estudio de esta investigación, debido a que permite realizar operaciones sin la necesidad del robot físico, lo que permite visualizar el comportamiento del autómeta, previa su implementación física. Otras de las características que brindan los simuladores actuales son el prototipado de manipuladores, renderizaciones realistas, animaciones estáticas y dinámicas, entre otras más.

### **2.3.1 Simuladores de brazos robóticos**

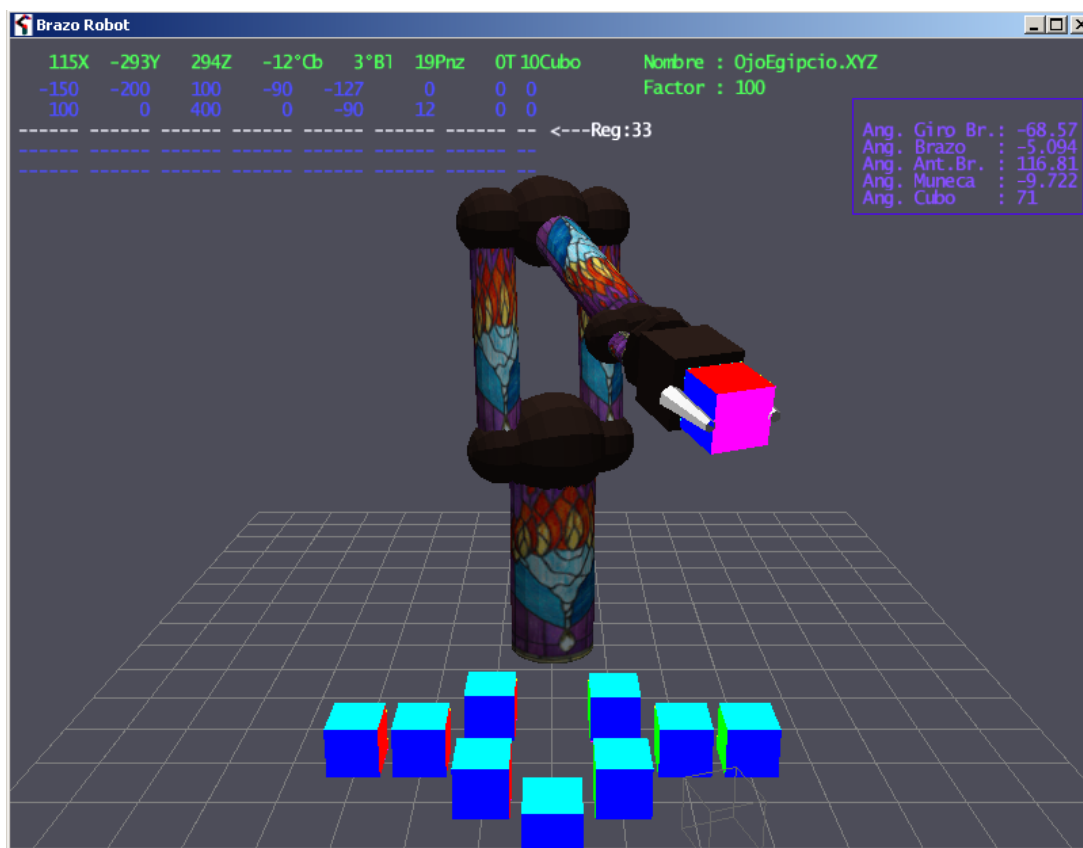
Los simuladores de brazos robóticos cumplen con la función general de un simulador, es decir, sustituir al robot físico por uno virtual, en específico un brazo robótico con seis grados de libertad. En la actualidad existen varios simuladores de brazos robóticos, como por ejemplo Brazo Robot, el cual es un software libre donde se pueden realizar programación para un manipulador robótico de seis grados de libertad. En la Figura 18 se muestra la interfaz del software. El simulador

permite ingresar las posiciones deseadas para el efector final, las graba en una memoria y genera la trayectoria según la información almacenada.



**Figura 18.** Simulador Brazo Robot, donde se muestra un brazo robótico virtual  
Fuente: (Google Sites, 2016)

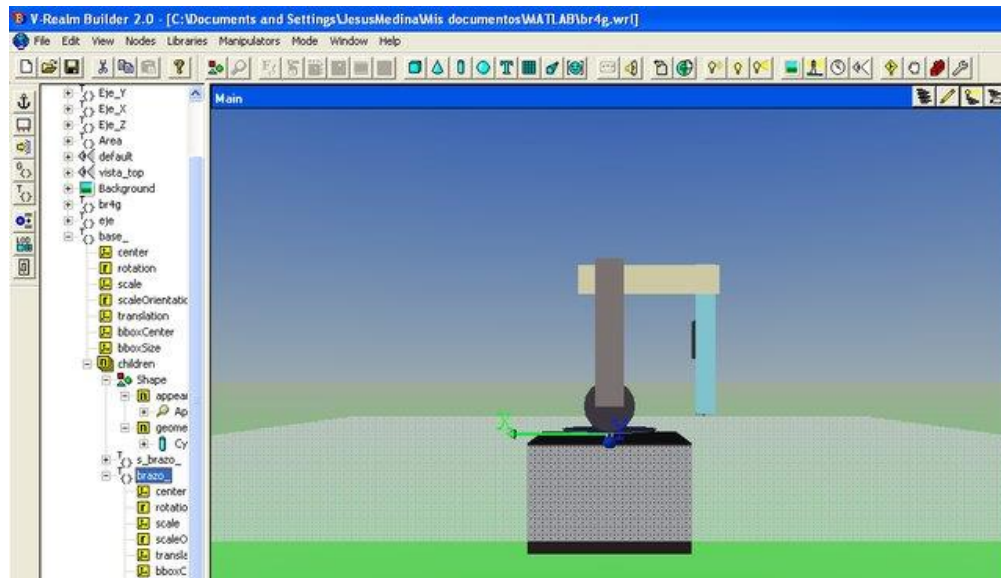
El mismo simulador tiene una variación donde permite la manipulación de cubos, en representación de una aplicación pick and place, aplicación muy empleada en el campo industrial para los brazos robóticos. En la Figura 19 se muestra la interfaz de esta variación del software de simulación Brazo Robot, conocido como DiezCubos. Este simulador presenta como principal característica su simplicidad al momento de interactuar con el manipulador virtual.



**Figura 19.** Variación del simulador Brazo Robot, conocida como DiezCubos, que simula aplicaciones de pick and place  
Fuente: (Google Sites, 2016)

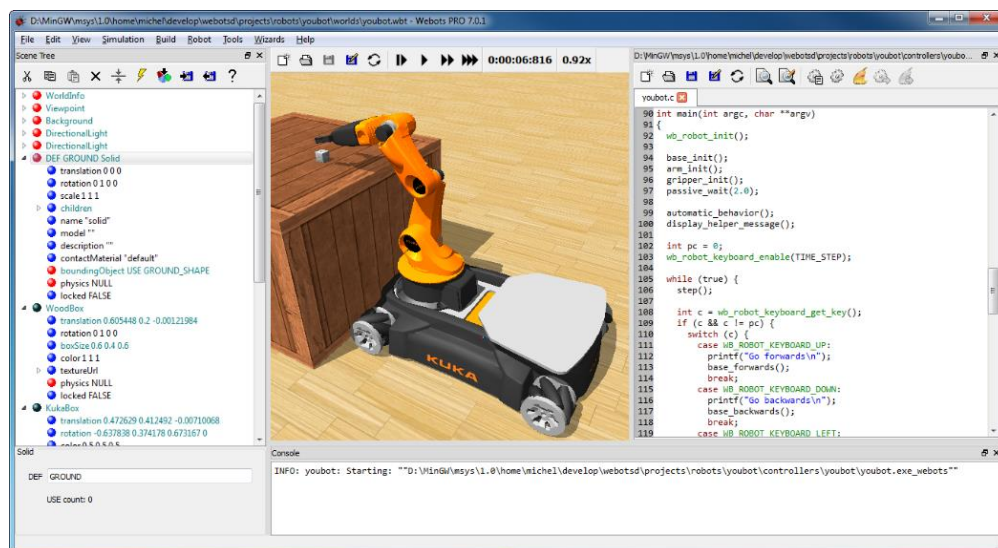
Cervantes et al (2007) presentan un simulador para un brazo robótico de cuatro grados de libertad, destinado para estudiantes de ingeniería, con el objetivo de brindar un servicio de simulación para manipuladores de cuatro grados de libertad. El programa se implementó en Matlab, utilizando la Toolbox de robótica. El software tiene la capacidad de funcionar en línea con el robot o fuera de línea (desconectado), presenta varios modos de operación para el control de la simulación. En la Figura 20 se muestra la interfaz del software de simulación, donde se incluye el modelo 3D del brazo robótico.





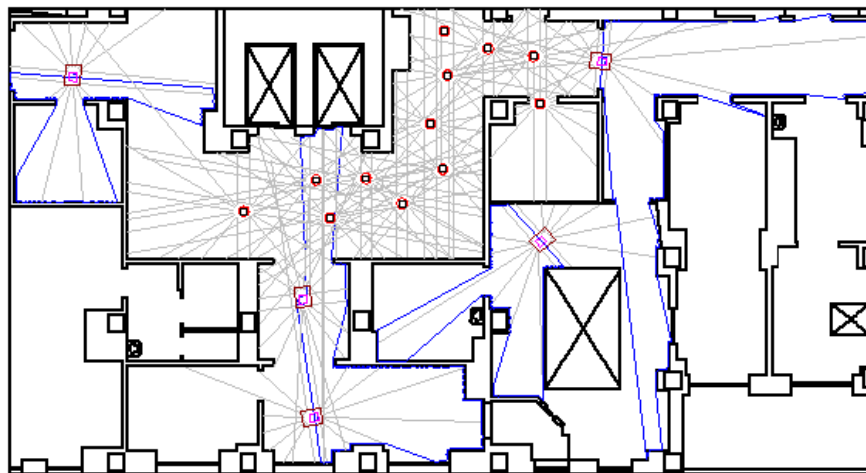
**Figura 20.** Interfaz del software de simulación de un brazo robótico de cuatro grados de libertad  
Fuente: (Cervantes et al, 2014)

Otro software de simulación es Webots, diseñado por Cyberbotics Ltd. (2004). A diferencia de Brazo Robot, Webots presenta una amplia gama de selección en robots, entre ellos robots articulares (piernas y brazos robóticos), robots móviles, robots aéreos, robots marítimos y robots modulares. En la Figura 21 se muestra la interfaz del software con un brazo robótico acoplado a un sistema de locomoción. Entre las ventajas que presenta este simulador es la amplia gama en actuadores y sensores para los robots, permite la construcción del robot, además de sus acabados (pintura y textura), la implementación de un sistema de detección de obstáculos, se puede generar animaciones ya grabarlas en formato MP4 o AVI, además de ser un programa multiplataforma, disponible en Windows, Apple y Linux.



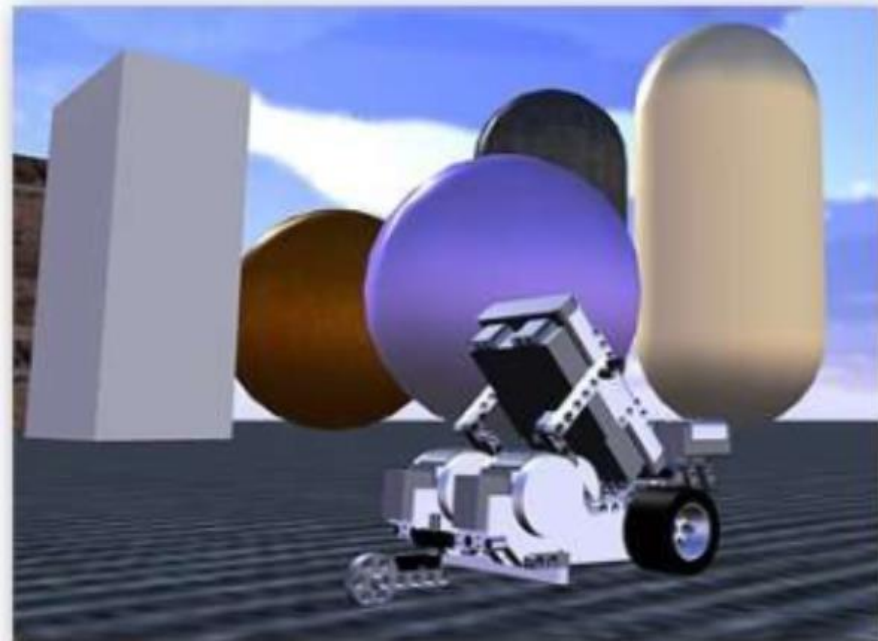
**Figura 21.** Interfaz del software de simulación Webots, desarrollado por Cyberbotics Ltd.  
Fuente: (Cyberbotics, 2004)

Uno de los softwares más utilizados es Player, junto a sus motores de simulación: Stage y Gazebo. Las ventajas de la interfaz Player es el hecho de ser de software libre, además de poseer una gran variedad de herramientas para el control de robots, como las múltiples conexiones simultáneas, y un amplio catálogo de robots móviles y accesorios. Sus simuladores, Stage y Gazebo, permiten representar robots virtuales en entornos bidimensionales y tridimensionales, respectivamente. En la Figura 22 se muestra un ejemplo de simulación en 2D realizada en Player Stage, donde se puede ver la generación de entornos y sensores de odometría, utilizados en su mayor parte para robótica móvil con sistemas de visión artificial.



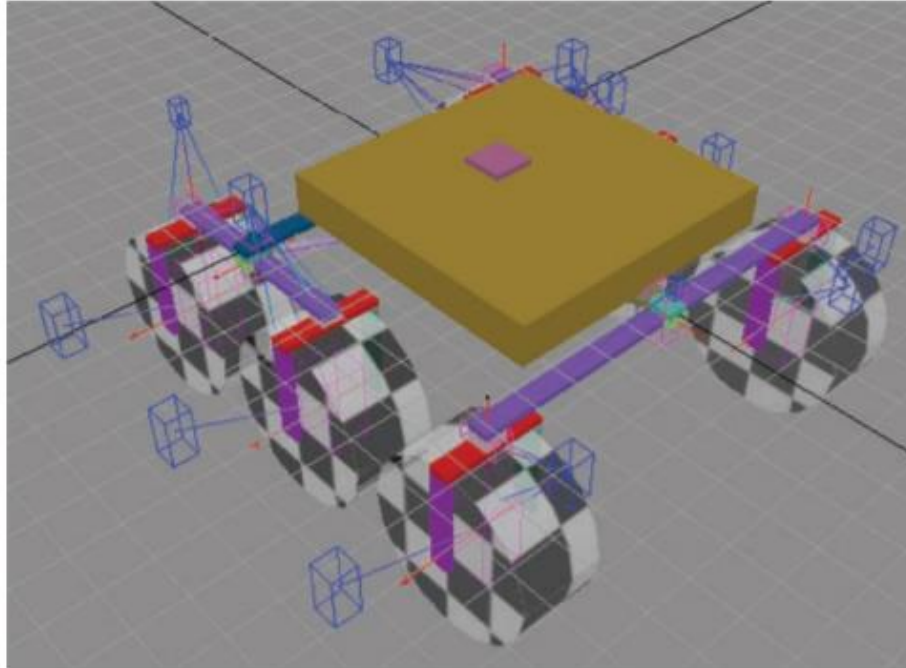
*Figura 22.* Simulación robótica 2D realizada en Player Stage  
Fuente: (USC, 2015)

Microsoft también presenta un software para el control y simulación de robots: Microsoft Robotics Studio. Arias et al. (2011) presentan en su artículo los resultados del uso de esta plataforma para la creación de una simulación de un robot móvil, como se presenta en la Figura 23. La principal ventaja del simulador es la inclusión de un entorno de programación basado en flujos. También posee una librería de robots para utilizarlos, o la posibilidad de crear autómatas propios.



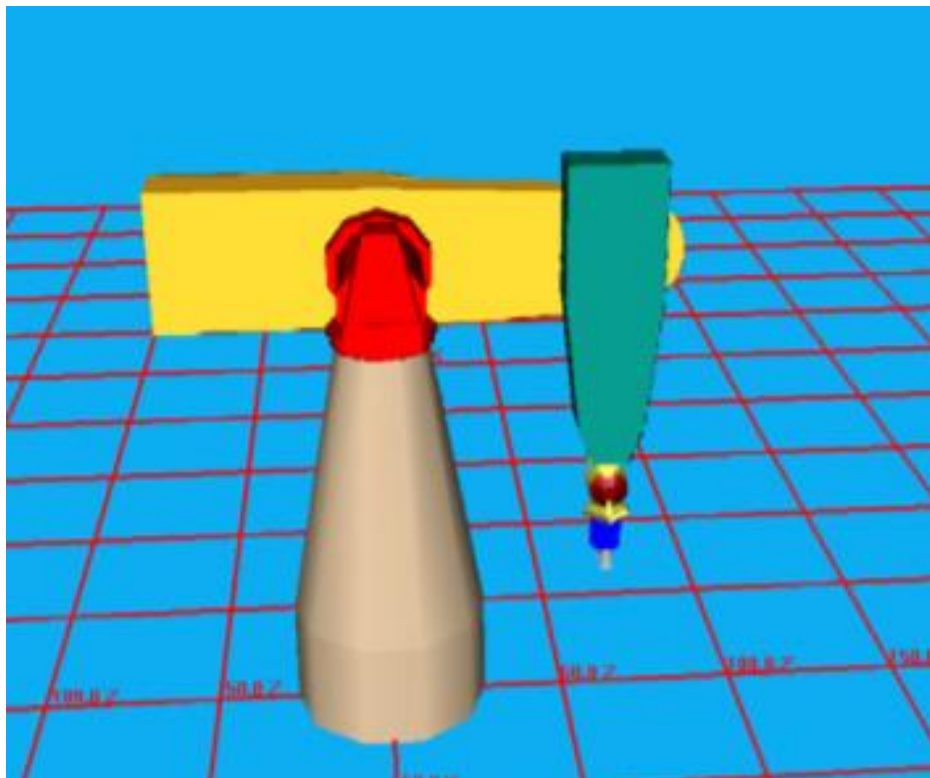
**Figura 23.** Simulación realizada en Microsoft Robotics Studio  
Fuente: (Arias et al: 2011)

El software Marilou Robotics Studio es un ambiente de simulación que permite modelar una gran variedad de robots, tales como robots móviles, humanoides, robots paralelos y brazos robóticos. Este programa acepta varios lenguajes de programación, como son C, C++, Java, entre otros más. Otra ventaja es la posibilidad de crear ambientes complejos, permitiendo definir situaciones más cercanas a la realidad, tales como obstáculos. En la Figura 24 se muestra un ejemplo de una simulación realizada por Martín et al. (2015) que representa un sistema de un vehículo prototipado para distribución automática de fertilizante.



**Figura 24.** Simulación de un robot en Marilou Robotics  
Fuente (Martín et al: 2015)

Otro programa para simular robots es Roboworks, el cual es un modelador en tres dimensiones para la simulación de elementos mecánicos. Una de las principales ventajas de este software es el modelado gráfico, además trabaja con programas de análisis y visualización, tales como Matlab, MathCad y LabVIEW. En la Figura 25 se muestra una simulación realizada por Muñoz et al (), que consiste en un brazo robótico modelado en 3D.



**Figura 25.** Brazo robótico simulado en Roboworks  
Fuente: (Muñoz et al, 2004)

En general, todos los simuladores de brazos robóticos presentan características que los diferencian. En la Tabla 3 se muestra la comparación de algunos simuladores con relación a ciertas características de los sistemas de simulación. La información se utilizó para definir los parámetros del sistema de simulación realizado en la presente investigación.

**Tabla 3**

*Comparación de diferentes softwares de simulación para brazos robóticos*

Software de simulación	Software gratuito	Simulación fuera de línea	Simulación en línea	Prototipado de robots	Sistema anticollisiones
Brazo Robot	Si	Si	No	No	No
V Realm Builder	Si	Si	Si	No	No
Webots	No	Si	Si	Si	Si
Actin	No	Si	Si	Si	Si
RoboDK	No	Si	No	No	No

## 2.4 Resumen

El capítulo expuso el fundamento teórico sobre el cual se asentó la investigación, para los diferentes componentes que intervienen: robot PUMA, teleoperación y sistemas de simulación. El robot PUMA es un robot antropomorfo, diseñado originalmente para aplicaciones de ensamblaje en industrias; más adelante se desarrolló su uso en otras áreas de la industria, como soldadura o pick and place, y en otros campos de la ciencia, como la medicina, donde fueron la base para los robots de operación asistida. Los sistemas teleoperados son aquellos donde el control por parte del usuario se realiza de forma remota con respecto a la máquina controlada. Los sistemas teleoperados se clasifican de acuerdo al tipo de control que presenten, lo que define su autonomía: directo, asistido y supervisado. Por último, la simulación es la virtualización de un sistema físico. Los sistemas de simulación se realizan principalmente para analizar el comportamiento y los cambios producidos de un sistema físico, sin la necesidad de construirlos. Los simuladores tienen sus propias características y herramientas dentro de su ambiente de trabajo, que los diferencia e identifica a cada uno de ellos.

## CAPÍTULO 3

### DISEÑO E IMPLEMENTACIÓN

El capítulo expone el diseño e implementación del sistema de simulación en tiempo real para un robot PUMA. El diseño del sistema se separó en subsistemas, de tal manera que se analizó a cada uno por separado. El primer subsistema es el robot físico, en el cual se considera su forma, material de los diferentes componentes; además se realizó el cálculo de los actuadores necesarios para generar los torques suficientes. El segundo subsistema es la teleoperación, donde se realizó una HMI de control, además de nodos de comunicación que permiten el paso de información desde la HMI hacia el robot físico y al robot virtual. El tercer subsistema es la simulación del robot PUMA, así como el código de control de movimientos para el robot virtual.

#### 3.1 Requerimientos del proyecto

Se utilizó la matriz QFD para poder interpretar los requerimientos del cliente en función de características técnicas de ingeniería. La Tabla 4 muestra esta matriz, donde se determinó las características técnicas más importantes, en función de los requisitos del cliente. Esto permitió dar prioridad a ciertas características del sistema sobre otras. Así mismo, fueron seleccionados subsistemas para el proyecto: Robot PUMA, Teleoperación y Simulación. Cada subsistema cumple con algunas de las características técnicas de la Tabla 4.



**Tabla 4**  
Matriz QFD para el Proyecto

		Importancia para el cliente (1-5)	Subsistema Robot PUMA					Teleoperación			Simulación	
			Material de la base del robot	Material del cuerpo del robot	Material de la muñeca del robot	Tipo de efector final	Tipo de actuadores	Tarjeta de control	Fuente de alimentación eléctrica	Interfaz de control de HMI	Software de simulación	Simulación estática
			1	2	3	4	5	6	7	8	9	10
1	Precio económico bajo	5	5	5	5	4	5	5	3	2	5	1
2	Transporte fácil	5	5	5	5	3	4	3	4	1	1	1
3	Facilidad de mantenimiento	2	3	3	4	3	5	4	3	1	1	1
4	Diseño tipo robot PUMA	4	4	4	4	5	3	2	1	1	1	1
5	Manipulador intercambiable	3	1	1	1	5	4	1	3	4	4	4
6	Funcionalidad similar a un robot industrial	4	4	4	4	5	5	3	5	3	1	1
7	Controlado por computadora	5	1	1	1	1	1	5	1	5	3	1
8	Control en tiempo real	5	1	1	1	1	4	5	3	5	4	4
9	Control manual de los actuadores	1	1	1	1	1	5	5	1	5	1	1
10	Alimentación eléctrica continua y confiable	4	1	1	1	1	5	5	5	1	1	1
11	Simulador del robot PUMA	5	2	2	2	3	1	1	1	4	5	5
12	Simulación de posición	3	1	1	1	1	1	1	1	4	5	5
13	Simulación y robot PUMA no necesariamente funcionan en conjunto	4	1	1	1	1	1	1	1	5	5	5
Importancia ponderada respecto a los requerimientos del cliente (absoluta)			123	123	125	133	161	158	127	156	152	122
Importancia ponderada respecto a los requerimientos del cliente (relativa)			1.01	1.01	1.02	1.1	1.3	1.3	1.04	1.3	1.2	1.00

En base a la información mostrada en la Tabla 4, se determinó los requerimientos que debe cumplir cada subsistema, en función de las características técnicas correspondientes, así:

#### Subsistema Robot PUMA

- Selección del material de la base del robot
- Selección del material del cuerpo del robot
- Selección del material de la muñeca del robot
- Selección del efector final
- Selección de los actuadores del sistema

#### Subsistema Teleoperación

- Selección de la tarjeta de control
- Selección del tipo de alimentación eléctrica
- Selección del programa para el diseño de la HMI de control

#### Subsistema Simulación

- Selección del programa para el diseño del simulador
- Diseño de una simulación estática

### **3.2 Subsistema Robot PUMA**

#### **3.2.1 Diseño del Robot PUMA**

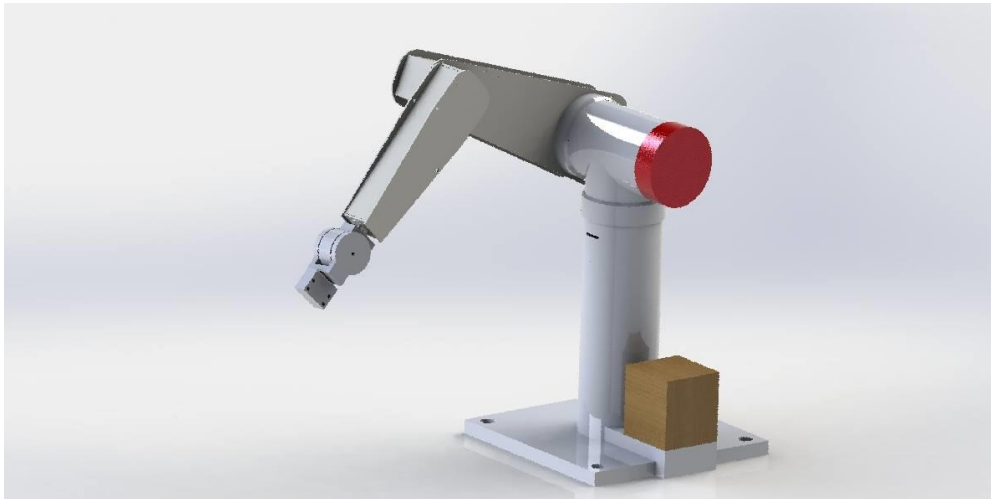
El primer paso del diseño del subsistema Robot PUMA fue determinar la forma y dimensión del mismo. La forma seleccionada se basa en el modelo base de un robot PUMA industrial, con la

diferencia que el antebrazo apunta hacia el suelo. En la Figura 26 se muestra el primer diseño CAD del prototipo, el cual se basó en la forma general del robot PUMA. La desventaja que presenta este primer boceto es la forma poco armoniosa, por lo cual se rediseñó el mismo.



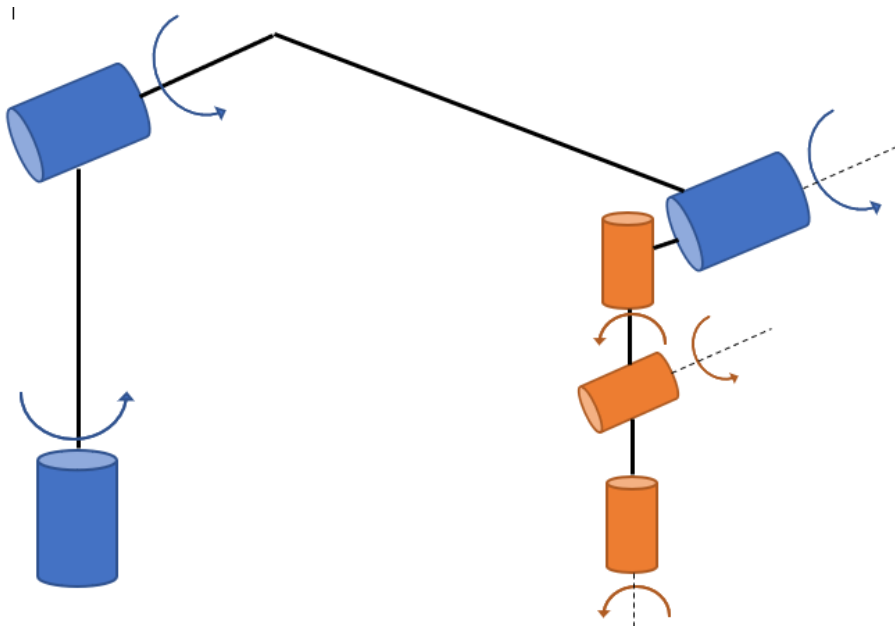
**Figura 26.** Primer diseño CAD del prototipo del robot PUMA

En la Figura 27 se muestra el segundo diseño CAD del prototipo, donde se corrigió la forma del robot para que presente una mayor armonía. Aquí se puede ver la posición que tiene el antebrazo respecto al brazo del robot, formando un ángulo de  $90^\circ$ , de tal manera que la muñeca se encuentre apuntando hacia el suelo. La razón para esta posición fue la aplicación del robot, la cual consiste en mantener al efector final perpendicular hacia el suelo.



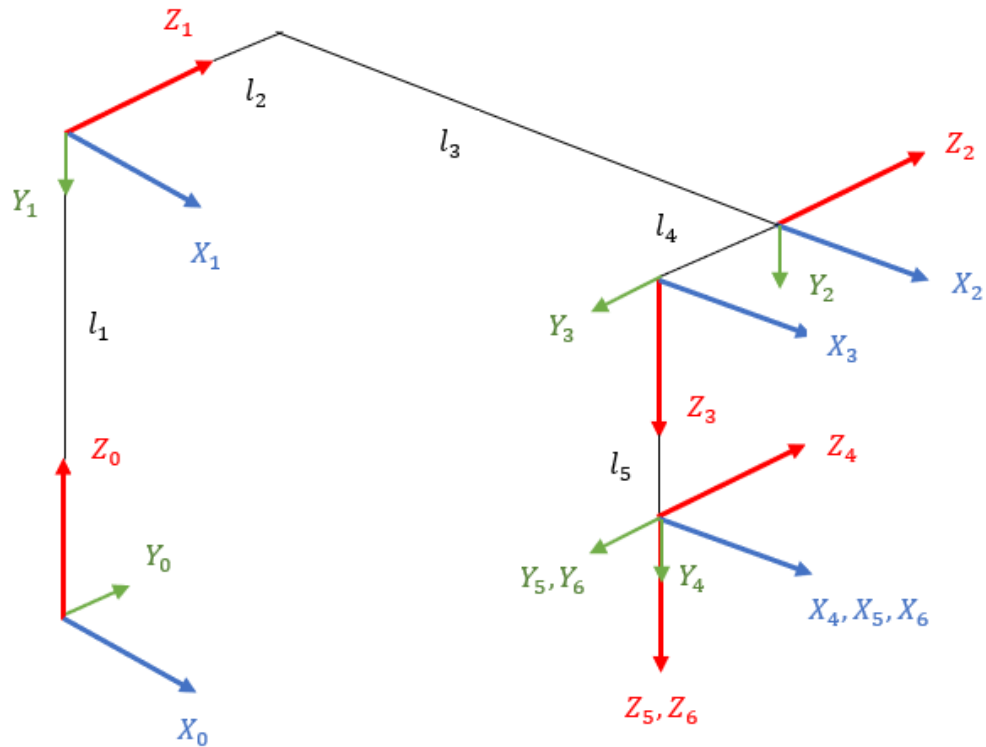
**Figura 27.** Segundo diseño CAD del prototipo del robot PUMA

En la Figura 28 se muestra de forma general los grados de libertad para el robot PUMA. Los cilindros en la Figura 28 representan una variable articular rotativa, por lo que se puede apreciar los seis cilindros que representan a los movimientos en conjunto del robot. Los tres primeros grados de libertad se encuentran representados con cilindros azules, mientras los últimos tres con cilindros naranjas, siendo las variables articulares de posición y orientación, respectivamente. Cada variable tiene un giro positivo, representado con las flechas curvas.



**Figura 28.** Representación de las variables articulares del robot PUMA

Una vez determinadas la forma y tamaño del robot, se realizó el cálculo de los actuadores utilizando la dinámica del sistema. En la Figura 29 se muestra la representación de los sistemas de referencia del robot PUMA, en base a las variables articulares vistas en la Figura 28. Se puede ver que la disposición corresponde a la orientación de los giros positivos designados previamente, así como a la convención Denavit-Hartenberg.



**Figura 29.** Representación de los sistemas de referencia para el robot PUMA

En la Tabla 5 se muestran los parámetros de Denavit-Hartenberg de acuerdo con la disposición de los ejes de la Figura 29. Considerando la convención clásica de Denavit-Hartenberg, los parámetros se colocaron en el siguiente orden de columnas en la tabla: giro sobre el eje 'z' ( $\theta$ ), desplazamiento del eje 'z' ( $d$ ), desplazamiento del eje 'x' ( $a$ ), giro sobre el eje 'x' ( $\alpha$ ). La ecuación (3-1) corresponde a la matriz de transformación homogénea de Denavit-Hartenberg, la cual se utilizó para determinar el cambio de los sistemas de referencia. La expresión  $C_{\theta_i}$  quiere decir el coseno del grado libertad  $\theta_i$ , y  $S_{\theta_i}$  representa el seno del grado de libertad  $\theta_i$ .

**Tabla 5**  
Parámetros de Denavit-Hartenberg para el robot PUMA

	$\theta$	$d$	$a$	$\alpha$
$H_0^1$	$\theta_1$	$l_1$	0	-90
$H_1^2$	$\theta_2$	$l_2$	$l_3$	0
$H_2^3$	$\theta_3$	$-l_4$	0	-90
$H_3^4$	$\theta_4$	$l_5$	0	90
$H_4^5$	$\theta_5$	0	0	-90
$H_5^6$	$\theta_6$	0	0	0

$$H_{i-1}^i = \begin{bmatrix} C_\theta & -C_\alpha S_\theta & S_\alpha S_\theta & aC_\theta \\ S_\theta & C_\alpha C_\theta & -S_\alpha C_\theta & aS_\theta \\ 0 & S_\alpha & C_\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-1)$$

Al sustituir los valores de la Tabla 5 en la ecuación (3-1) se obtuvo las matrices de transformación entre los sistemas de referencia. En la ecuación (3-2) se muestra la matriz de transformación desde el sistema de referencia '0' hasta el sistema de referencia '3', y en la ecuación (3-3) se muestra la matriz de transformación desde el sistema de referencia '3' hasta el sistema de referencia '6'. El primero corresponde a la posición del efector final, mientras que el segundo corresponde a la orientación del efector final. Se separó en dos sistemas para facilitar el análisis de la dinámica del sistema.

$$H_0^3 = \begin{bmatrix} C_{\theta_1} C_{\theta_{2+3}} & S_{\theta_1} & -C_{\theta_1} S_{\theta_{2+3}} & -(l_2 - l_4) S_{\theta_1} + l_3 C_{\theta_1} C_{\theta_2} \\ S_{\theta_1} C_{\theta_{2+3}} & -C_{\theta_1} & -S_{\theta_1} S_{\theta_{2+3}} & (l_2 - l_4) C_{\theta_1} + l_3 S_{\theta_1} C_{\theta_2} \\ -S_{\theta_{2+3}} & 0 & -C_{\theta_{2+3}} & l_1 - l_3 S_{\theta_2} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-2)$$

$$H_3^6 = \begin{bmatrix} C_{\theta_4} C_{\theta_5} C_{\theta_6} - S_{\theta_4} S_{\theta_6} & -C_{\theta_4} C_{\theta_5} S_{\theta_6} - S_{\theta_4} C_{\theta_6} & -C_{\theta_4} S_{\theta_5} & 0 \\ S_{\theta_4} C_{\theta_5} C_{\theta_6} + C_{\theta_4} S_{\theta_6} & -S_{\theta_4} C_{\theta_5} S_{\theta_6} + C_{\theta_4} C_{\theta_6} & -S_{\theta_4} S_{\theta_5} & 0 \\ S_{\theta_5} C_{\theta_6} & -S_{\theta_5} S_{\theta_6} & C_{\theta_5} & l_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-3)$$

El producto entre las matrices (3-1) y (3-2) es la matriz de transformación desde el origen del sistema de coordenadas hasta el efector final (3-3). La matriz (3-3) permite realizar el análisis cinemático directo e inverso, al considerar los seis grados de libertad del robot PUMA. Para el análisis dinámico se utiliza únicamente la matriz (3-1), debido a que la mayor cantidad de fuerzas y torques presentes en el subsistema se concentran en los grados de libertad de posición, es decir, las tres primeras rotaciones del robot.

$$H_0^6 = \begin{bmatrix} H_{11} & H_{12} & H_{13} & H_{14} \\ H_{21} & H_{22} & H_{23} & H_{24} \\ H_{31} & H_{32} & H_{33} & H_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-3)$$

$$H_{11} = (C_{\theta_1} C_{\theta_2+\theta_3} C_{\theta_4} + S_{\theta_1} S_{\theta_4}) C_{\theta_5} C_{\theta_6} - C_{\theta_1} S_{\theta_2+\theta_3} S_{\theta_5} C_{\theta_6} - C_{\theta_1} C_{\theta_2+\theta_3} S_{\theta_4} S_{\theta_6} + S_{\theta_1} C_{\theta_4} S_{\theta_6}$$

$$H_{12} = -(C_{\theta_1} C_{\theta_2+\theta_3} C_{\theta_4} + S_{\theta_1} S_{\theta_4}) C_{\theta_5} S_{\theta_6} + C_{\theta_1} S_{\theta_2+\theta_3} S_{\theta_5} S_{\theta_6} - C_{\theta_1} C_{\theta_2+\theta_3} S_{\theta_4} C_{\theta_6} + S_{\theta_1} C_{\theta_4} C_{\theta_6}$$

$$H_{13} = -(C_{\theta_1} C_{\theta_2+\theta_3} C_{\theta_4} + S_{\theta_1} S_{\theta_4}) S_{\theta_5} - C_{\theta_1} S_{\theta_2+\theta_3} C_{\theta_5}$$

$$H_{14} = p x_0^6 = -l_2 S_{\theta_1} + l_3 C_{\theta_1} C_{\theta_2} + l_4 S_{\theta_1} - l_5 C_{\theta_1} S_{\theta_2+\theta_3}$$

$$H_{21} = (S_{\theta_1} C_{\theta_2+\theta_3} C_{\theta_4} - C_{\theta_1} S_{\theta_4}) C_{\theta_5} C_{\theta_6} - S_{\theta_1} S_{\theta_2+\theta_3} S_{\theta_5} C_{\theta_6} - S_{\theta_1} C_{\theta_2+\theta_3} S_{\theta_4} S_{\theta_6} - C_{\theta_1} C_{\theta_4} S_{\theta_6}$$

$$H_{22} = -(S_{\theta_1} C_{\theta_2+\theta_3} C_{\theta_4} - C_{\theta_1} S_{\theta_4}) C_{\theta_5} S_{\theta_6} + S_{\theta_1} S_{\theta_2+\theta_3} S_{\theta_5} S_{\theta_6} - S_{\theta_1} C_{\theta_2+\theta_3} S_{\theta_4} C_{\theta_6} - C_{\theta_1} C_{\theta_4} C_{\theta_6}$$

$$H_{23} = -(S_{\theta_1} C_{\theta_2+\theta_3} C_{\theta_4} - C_{\theta_1} S_{\theta_4}) S_{\theta_5} - S_{\theta_1} S_{\theta_2+\theta_3} C_{\theta_5}$$

$$H_{24} = p y_0^6 = l_2 C_{\theta_1} + l_3 S_{\theta_1} C_{\theta_2} - l_4 C_{\theta_1} - l_5 S_{\theta_1} S_{\theta_2+\theta_3}$$

$$H_{31} = -S_{\theta_2+\theta_3} C_{\theta_4} C_{\theta_5} C_{\theta_6} - C_{\theta_2+\theta_3} S_{\theta_5} C_{\theta_6} + S_{\theta_2+\theta_3} S_{\theta_4} S_{\theta_6}$$



$$H_{32} = S_{\theta_2+\theta_3}C_{\theta_4}C_{\theta_5}S_{\theta_6} + C_{\theta_2+\theta_3}S_{\theta_5}S_{\theta_6} + S_{\theta_2+\theta_3}S_{\theta_4}C_{\theta_6}$$

$$H_{33} = S_{\theta_2+\theta_3}C_{\theta_4}S_{\theta_5} - C_{\theta_2+\theta_3}C_{\theta_5}$$

$$H_{34} = pz_0^6 = l_1 - l_3S_{\theta_2} - l_5C_{\theta_2+\theta_3}$$

Los valores de posición son la última columna de la Ecuación (3-3), descrito de manera separada en la Ecuación (3-4). Los valores de la posición del efector final son importantes en el análisis cinemático inverso, por lo que se escriben otra vez, pero sin la rotación del efector final respecto al origen de coordenadas.

$$\begin{bmatrix} px_0^6 \\ py_0^6 \\ pz_0^6 \end{bmatrix} = \begin{bmatrix} -l_2S_{\theta_1} + l_3C_{\theta_1}C_{\theta_2} + l_4S_{\theta_1} - l_5C_{\theta_1}S_{\theta_2+\theta_3} \\ l_2C_{\theta_1} + l_3S_{\theta_1}C_{\theta_2} - l_4C_{\theta_1} - l_5S_{\theta_1}S_{\theta_2+\theta_3} \\ l_1 - l_3S_{\theta_2} - l_5C_{\theta_2+\theta_3} \end{bmatrix} \quad (3-4)$$

Una vez determinados los valores de posición y rotación desde el origen hasta el efector final, se realizó el análisis cinemático inverso, con el fin de determinar las variables articulares en función de la posición del efector final. Para lograrlo, se partió de un análisis geométrico, con el cual se logró llegar a las Ecuaciones de posición y orientación del Robot PUMA (3-5), (3-6), (3-7), (3-8), (3-9), (3-10). Las ecuaciones que describen la cinemática inversa del Robot PUMA para la posición se basan en la Ecuación (3-4), mientras que la orientación se basa en la Ecuación (3-3) y los valores de los ángulos de la posición.

$$\theta_1 = \arctan\left(\frac{py_0^6}{px_0^6}\right) \quad (3-5)$$

$$\theta_2 = \arctan\left(\frac{pz_0^6}{\sqrt{(px_0^6)^2 + (py_0^6)^2}}\right) \quad (3-6)$$

$$\theta_3 = \arctan \left( \frac{\frac{(px_0^6)^2 + (py_0^6)^2 + (pz_0^6)^2 - l_5^2}{2l_5}}{\sqrt{1 - \left(\frac{(px_0^6)^2 + (py_0^6)^2 + (pz_0^6)^2 - l_5^2}{2l_5}\right)^2}} \right) \quad (3-7)$$

$$\theta_4 = \arctan \left( \frac{(H_{13}C_{\theta_1} + H_{23}S_{\theta_1})S_{\theta_2+\theta_3} + H_{33}C_{\theta_2+\theta_3}}{(H_{13}C_{\theta_1} + H_{23}S_{\theta_1})C_{\theta_2+\theta_3} + H_{23}S_{\theta_2+\theta_3}} \right) \quad (3-8)$$

$$\theta_5 = \arctan \left( \frac{\sqrt{1 - (H_{13}S_{\theta_1} - H_{23}C_{\theta_1})^2}}{H_{13}S_{\theta_1} - H_{23}C_{\theta_1}} \right) \quad (3-9)$$

$$\theta_5 = \arctan \left( \frac{H_{12}S_{\theta_1} + H_{22}C_{\theta_1}}{H_{11}S_{\theta_1} - H_{21}C_{\theta_1}} \right) \quad (3-10)$$

En la Tabla 6 se muestra el error porcentual que se cometió con las ecuaciones de la cinemática inversa. Para lograrlo, se ingresó la posición 200, 40, 125 como x, y, z respectivamente, en las ecuaciones (3-5), (3-6), (3-7), (3-8), (3-9), (3-10) y se obtuvieron los seis ángulos indicados en la segunda columna de la Tabla 6. Después, se usó los mismos ángulos para determinar la posición real en función de la cinemática directa, y se obtuvo la posición real 203.63, 39.62, 127.68 en x, y, z. se tomó estos valores como prácticos, y la posición ingresada en la cinemática inversa como real, se logró conseguir el error para cada componente de la posición, siendo de 2.144% la máxima, correspondiente a la componente z.

**Tabla 6**

*Cálculo del error porcentual para las ecuaciones de la cinemática inversa*

Posición deseada (mm)	VARIABLES ARTICULARES (cinemática inversa)	Posición real (mm)	Error porcentual (%)
200	-5.8	203.63	1.815
	-0.9		
40	21	39.62	0.95
	0		
125	-20.1	127.68	2.144
	5.8		

El último punto en el diseño del Robot PUMA es la dinámica del mismo. El análisis dinámico se realizó en base a la ecuación de Euler-Lagrange, que se muestra en la Ecuación (3-11).

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} = Q \quad (3-11)$$

Donde  $L$  es la diferencia entre la energía cinética ( $K$ ) y la energía potencial ( $U$ ), mostrada en la Ecuación (3-12).

$$L = K - U \quad (3-12)$$

La energía cinética se descompone en lineal y rotacional, mientras que la energía potencial corresponde al peso respecto a un nivel referencial de los componentes del sistema. Para encontrar las velocidades lineales y rotacionales, necesarias para la energía cinética, se partió de los jacobianos del sistema, mostrados a continuación:

$$J = \begin{bmatrix} J_V \\ J_W \end{bmatrix}$$

$$J_V = R_0^{i-1} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times (P_0^n - P_0^{i-1})$$

$$J_W = R_0^{i-1} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Utilizando los valores encontrados de los jacobianos para el Robot PUMA, y sustituyéndolos para aplicar la Ecuación (3-12) se encontró la Ecuación (3-13) de la diferencia de energías para la posición del efector final, y la Ecuación (3-15) de la diferencia de energías para la orientación del efector final. El análisis independiente entre posición y orientación se debe al desacople cinemático del Robot PUMA. En las ecuaciones se sustituyeron los valores de las longitudes y masas, de acuerdo con las características del Robot PUMA diseñado.

$$L = (0.275 + 0.02C_{\theta_2} + 0.03C_{\theta_2}^2 - 0.005S_{2\theta_1}C_{\theta_2})\dot{\theta}_1^2 \quad (3-13)$$

$$+ (0.21 + 0.04S_{\theta_1+\theta_2} + 0.03S_{\theta_1}C_{\theta_2})\dot{\theta}_2^2$$

$$+ (2329 + 0.03S_{\theta_1}C_{\theta_2})\dot{\theta}_3^2$$

$$+ (0.01 - 0.03(S_{\theta_1} + C_{\theta_1}) + 0.02S_{\theta_2}$$

$$- 0.065S_{2\theta_1}C_{\theta_2})\dot{\theta}_1\dot{\theta}_2 - 0.01(C_{\theta_1} + S_{\theta_1})\dot{\theta}_1\dot{\theta}_3$$

$$+ (0.12 + 0.06S_{\theta_1}C_{\theta_2})\dot{\theta}_2\dot{\theta}_3$$

$$L = (2.3065 * 10^{-4})\dot{\theta}_4^2$$

$$+ (4.6592 * 10^{-4}S_{\theta_4}^2 - 5.7795 * 10^{-5}S_{2\theta_4} + 4.7142$$

$$* 10^{-4}C_{\theta_4}^2)\dot{\theta}_5^2$$

$$+ (2.4576 * 10^{-4}S_{\theta_4} - 1.9494 * 10^{-4}C_{\theta_4})\dot{\theta}_4\dot{\theta}_5$$

$$- 0.04242S_{\theta_5}$$

(3-14)

Utilizando los valores encontrados en las Ecuaciones (3-13) y (3-14) en la Ecuación (3-11), se encontraron las Ecuaciones (3-15), (3-16), (3-17) (posición), (3-18) y (3-19) (orientación).

$$Q_1 = -0.01C_{2\theta_1}C_{\theta_2}\dot{\theta}_1^2 \quad (3-15)$$

$$\begin{aligned} &+ 0.02(C_{\theta_2} - 3.25S_{2\theta_1}S_{\theta_2} - 4C_{2\theta_1} - 1.5C_{\theta_1}C_{\theta_2})\dot{\theta}_2^2 \\ &+ 0.03C_{\theta_1}C_{\theta_2}\dot{\theta}_3^2 + 0.01(S_{2\theta_1}S_{\theta_2} - 4S_{\theta_2} - 6S_{2\theta_2})\dot{\theta}_1\dot{\theta}_2 \\ &- 0.06C_{\theta_1}C_{\theta_2}\dot{\theta}_2\dot{\theta}_3 \\ &+ 0.005(110 + 80C_{\theta_2} + 12C_{\theta_2}^2 - S_{2\theta_1}C_{\theta_2})\ddot{\theta}_1 \\ &+ 0.01[1 - 3(S_{\theta_1} + C_{\theta_1}) + 2S_{\theta_2} - 6.5S_{2\theta_1}C_{\theta_2}]\ddot{\theta}_2 \\ &- 0.01(S_{\theta_1} + C_{\theta_1})\ddot{\theta}_3 \end{aligned}$$

$$Q_2 = 0.005[6(S_{\theta_1} - C_{\theta_1}) - 26C_{2\theta_1} + 4S_{\theta_2} + 6S_{\theta_2} - S_{2\theta_1}S_{\theta_2}]\dot{\theta}_1^2 \quad (3-16)$$

$$\begin{aligned} &- 0.03S_{\theta_1}S_{\theta_2}(\dot{\theta}_2^2 - \dot{\theta}_3^2) + 0.04(C_{\theta_2} + 3.25S_{2\theta_1}S_{\theta_2})\dot{\theta}_1\dot{\theta}_2 \\ &+ 0.06C_{\theta_1}C_{\theta_2}\dot{\theta}_1\dot{\theta}_3 \\ &+ 0.01[1 - 3(S_{\theta_1} + C_{\theta_1}) + 2S_{\theta_2} - 6.5S_{2\theta_1}C_{\theta_2}]\ddot{\theta}_1 \\ &+ 0.06(7 + 1.33S_{2\theta_1} + S_{\theta_1}C_{\theta_2})\ddot{\theta}_2 + 0.06(2 + S_{\theta_1}C_{\theta_2})\ddot{\theta}_3 \\ &+ 1.02C_{\theta_2} \end{aligned}$$

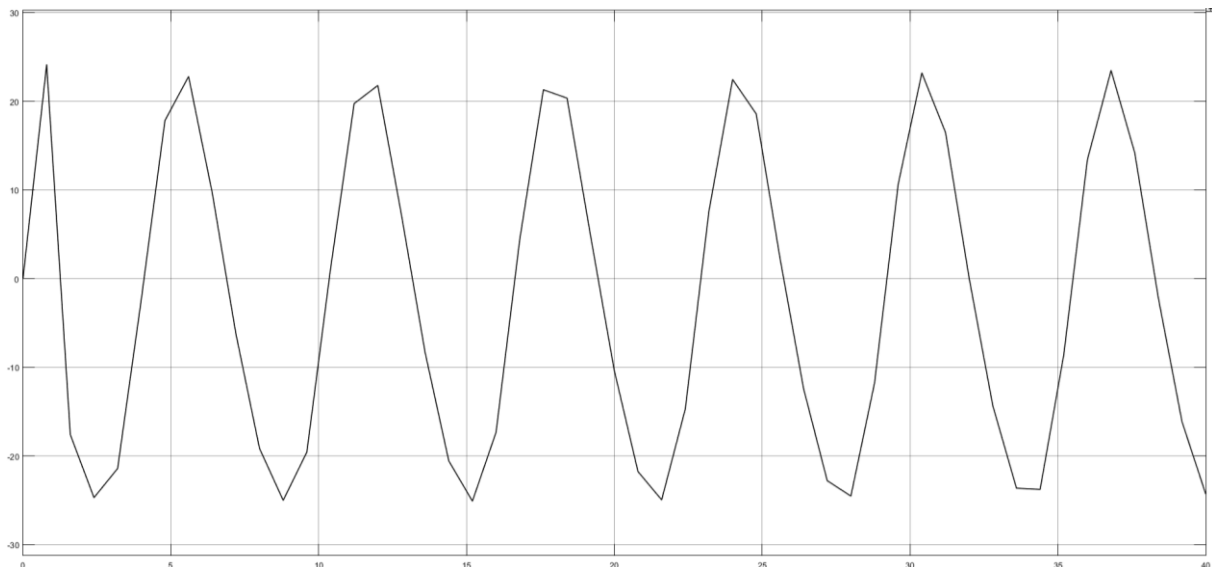
$$Q_3 = 0.01(S_{\theta_1} - C_{\theta_1})\dot{\theta}_1^2 - 0.06S_{\theta_1}S_{\theta_2}\dot{\theta}_2^2 + 0.06C_{\theta_1}C_{\theta_2}\dot{\theta}_1(\dot{\theta}_2 + \dot{\theta}_3) \quad (3-17)$$

$$\begin{aligned} &- 0.06S_{\theta_1}S_{\theta_2}\dot{\theta}_2\dot{\theta}_3 - 0.01(S_{\theta_1} + C_{\theta_2})\ddot{\theta}_1 \\ &+ 0.06(2 + S_{\theta_1}C_{\theta_2})(\ddot{\theta}_2 + \ddot{\theta}_3) + 0.71C_{\theta_3} \end{aligned}$$

$$Q_4 = 5.5 * 10^{-6}(S_{2\theta_4} + 10.508)\dot{\theta}_5^2 + 4.6130 * 10^{-4}\ddot{\theta}_4 + 1.994 * 10^{-4}(1.2607S_{\theta_4} - C_{\theta_4})\ddot{\theta}_5 \quad (3-18)$$

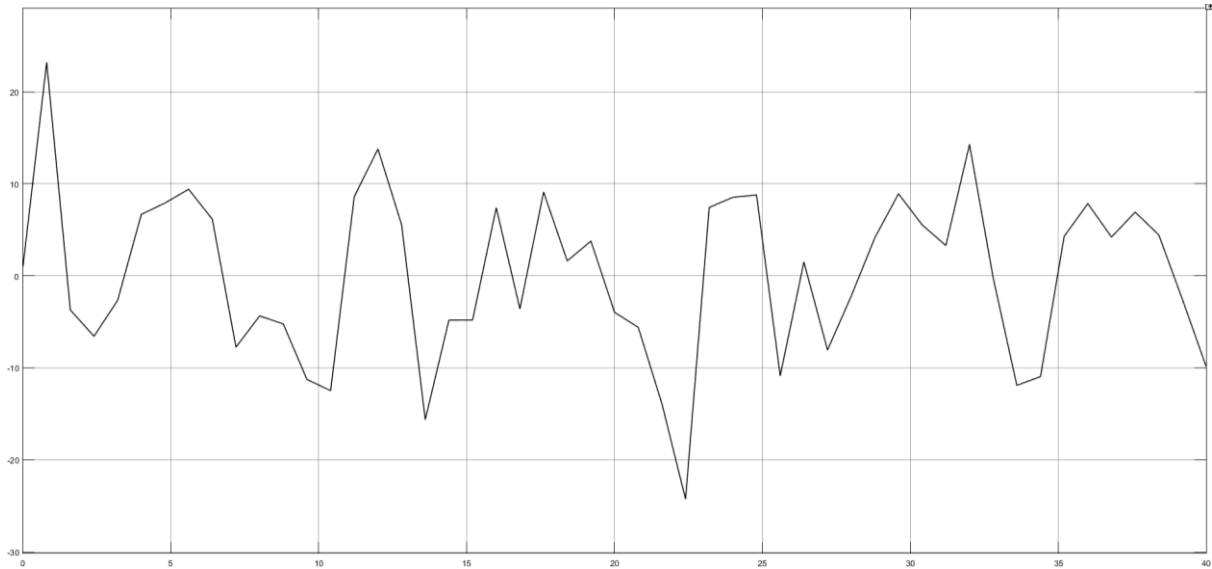
$$Q_5 = 1.9494 * 10^{-4}(S_{\theta_4} + 1.2607C_{\theta_4})\dot{\theta}_4^2 - 1.1 * 10^{-5}(S_{2\theta_4} + 10.5082C_{2\theta_4})\dot{\theta}_4\dot{\theta}_5 + 1.994 * 10^{-4}(1.2607S_{\theta_4} - C_{\theta_4})\ddot{\theta}_4 + 5.7794 * 10^{-5}(16.1233S_{\theta_4}^2 - S_{2\theta_4} + 16.3138C_{\theta_4}^2)\ddot{\theta}_5 + 0.04242C_{\theta_5}$$

Las ecuaciones que expresan la dinámica del sistema nos permiten dimensionar los motores necesarios para cada articulación. La Figura 30 muestra la gráfica del torque del primer servomotor en base a la Ecuación (3-15). Aquí se puede observar que el valor pico es de  $-25 \text{ kg.cm}$ , considerando este valor como el crítico en el análisis.



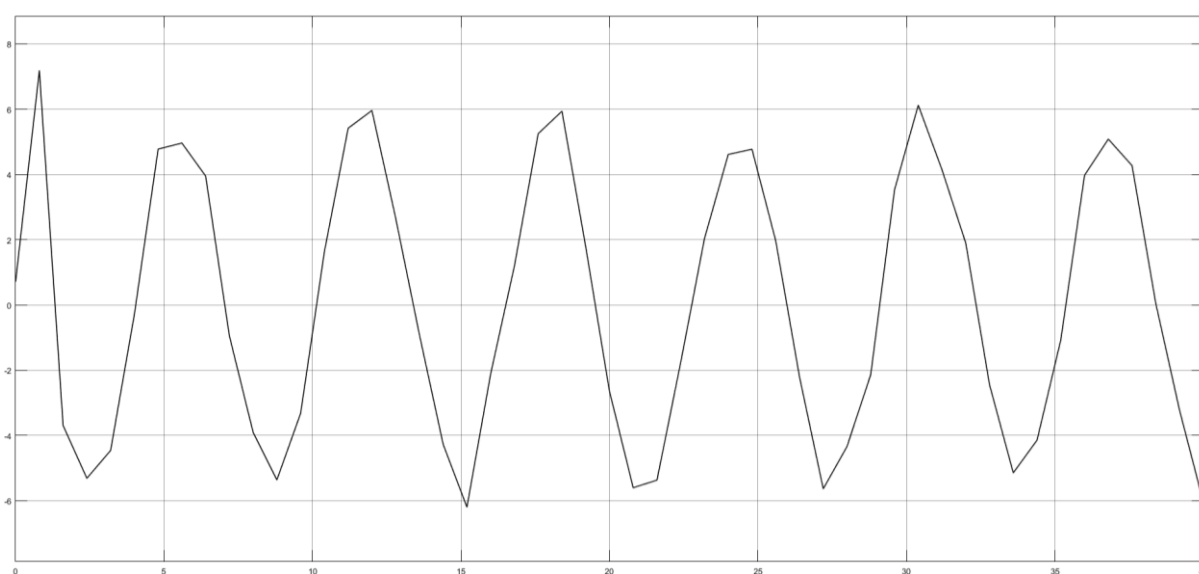
**Figura 30.** Torque del primer actuador en función de la dinámica

La Figura 31 muestra la gráfica del torque del segundo servomotor en base a la Ecuación (3-16). Aquí se puede observar que el valor pico es de  $-24 \text{ kg.cm}$ , considerando este valor como el crítico en el análisis.



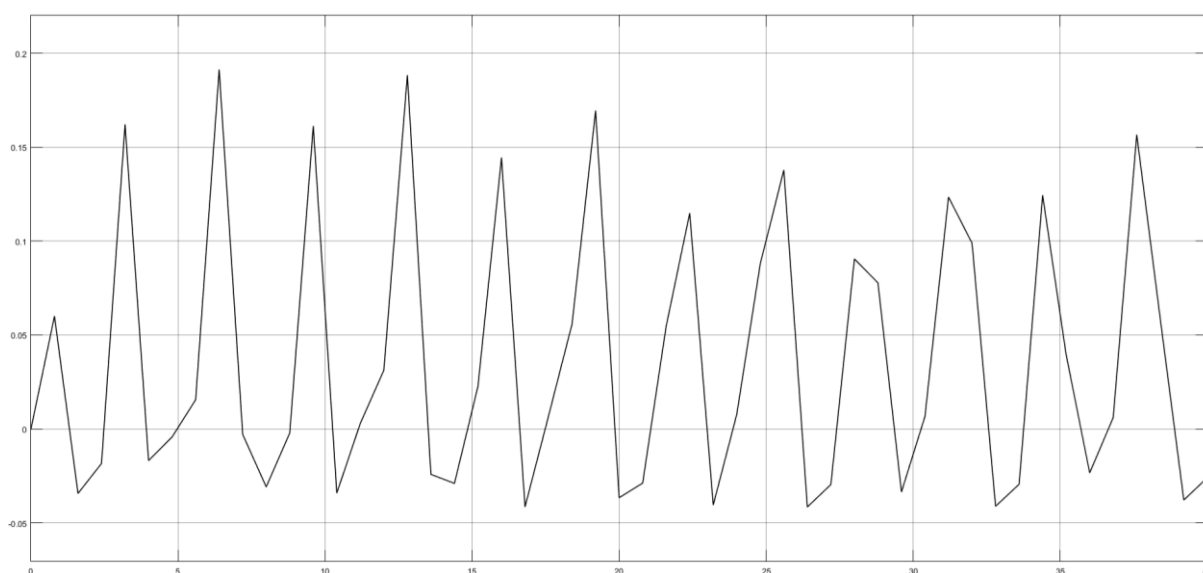
**Figura 31.** Torque del segundo actuador en función de la dinámica

La Figura 32 muestra la gráfica del torque del tercer servomotor en base a la Ecuación (3-15). Aquí se puede observar que el valor pico es de  $7 \text{ kg.cm}$ , considerando este valor como el crítico en el análisis.



**Figura 32.** Torque del tercer actuador en función de la dinámica

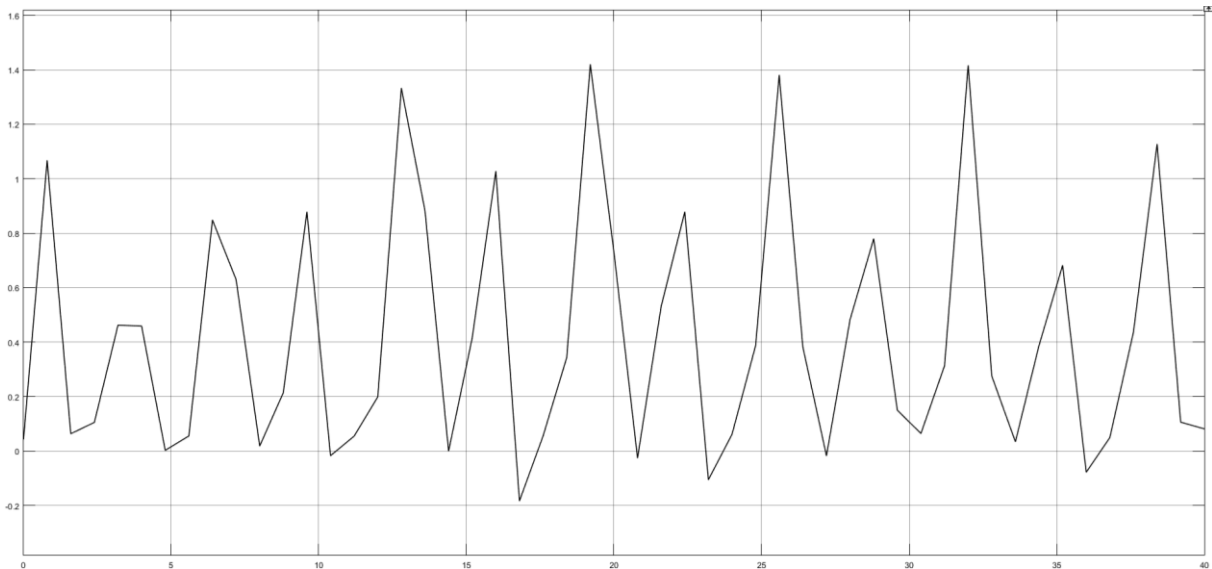
La Figura 33 muestra la gráfica del torque del cuarto servomotor en base a la Ecuación (3-15). Aquí se puede observar que el valor pico es de  $0.19 \text{ kg.cm}$ , considerando este valor como el crítico en el análisis.



**Figura 33.** Torque del cuarto actuador en función de la dinámica



La Figura 34 muestra la gráfica del torque del primer servomotor en base a la Ecuación (3-15). Aquí se puede observar que el valor pico es de  $1.4 \text{ kg.cm}$ , considerando este valor como el crítico en el análisis.



**Figura 34.** Torque del quinto actuador en función de la dinámica

En la Tabla 7 se muestran los valores críticos para los actuadores analizados, así como los servomotores definidos para cada articulación. La selección se realizó de acuerdo con un factor de trabajo del 70%, es decir, el valor crítico para cada actuador es el 70% del valor nominal del servomotor. En el análisis no se consideró el sexto actuador, debido a que el mismo trabaja libre de carga la mayor parte del tiempo, así que se seleccionó el mismo servomotor que en el caso de los actuadores 4 y 5. De igual manera, para el servomotor 4 se escogió el mismo que en el caso 5, para una facilidad en el diseño de la muñeca de Euler.

**Tabla 7***Selección de servomotores en función de los torques dinámicos*

Actuador	Torque crítico (kg.cm)	Servomotor escogido (código)	Torque nominal (kg.cm)	Factor de trabajo (%)
1	25	FT5335M	35	71.43
2	24	FT5335M	35	68.57
3	7	HS 705-MG	11	63.64
4	0.19	MG90S	2	9.5
5	1.4	MG90S	2	70
6	-	MG90S	2	-

### 3.2.2 Desarrollo de conceptos del subsistema Robot PUMA

El desarrollo de conceptos para el subsistema Robot PUMA se centró en cinco categorías: material del brazo, material de la muñeca, material de la base, aplicación del efector final y actuadores. La matriz morfológica de la Tabla 8 muestra las categorías junto a las opciones tentativas de cada una de ellas.

**Tabla 8***Matriz morfológica de los conceptos para el subsistema Robot PUMA*

Categoría	Opciones		
Material brazo	Aluminio	Acrílico	MDF
Material muñeca	Aluminio	PLA	ABS
Material base	Aluminio	Acero inoxidable	Madera
Efector final	Pick and Place	Ensamblaje	Soldadura
Actuador	Motor DC	Servomotor	Motor a pasos

#### 3.2.2.1 Categoría 'Material del brazo' del subsistema Robot PUMA

La Tabla 9 muestra la matriz de selección de criterios para la categoría 'Material del brazo' del subsistema Robot PUMA. Los criterios fueron seleccionados para cumplir con los requerimientos del subsistema Robot PUMA.

**Tabla 9**

*Matriz de selección de criterios para la categoría 'Material del brazo' del subsistema Robot PUMA*

	Peso	Costo	Manufactura	$\sum+1$	Ponderación
<b>Peso</b>	-	1	1	3	0.50
<b>Costo</b>	0	-	1	2	0.33
<b>Manufactura</b>	0	0	-	1	0.17
			Total	6	

En la Tabla 10 se muestra la calificación de las opciones de la categoría 'Material del brazo' considerando los criterios de la matriz de selección de criterios de la Tabla 9.

**Tabla 10**

*Calificación de las opciones de la categoría 'Material del brazo'*

Peso	Aluminio	Acrílico	MDF	$\sum+1$	Ponderación
<b>Aluminio</b>	-	0.5	0	1.5	0.25
<b>Acrílico</b>	0.5	-	0	1.5	0.25
<b>MDF</b>	1	1	-	3	0.50
			Total	6	
Costo	Aluminio	Acrílico	MDF	$\sum+1$	Ponderación
<b>Aluminio</b>	-	0.5	0	1.5	0.25
<b>Acrílico</b>	0.5	-	0	1.5	0.25
<b>MDF</b>	1	1	-	3	0.50
			Total	6	
Manufactura	Aluminio	Acrílico	MDF	$\sum+1$	Ponderación
<b>Aluminio</b>	-	1	0.5	2.5	0.42
<b>Acrílico</b>	0	-	0	1	0.17
<b>MDF</b>	0.5	1	-	2.5	0.42
			Total	6	

La Tabla 11 es la matriz de Holmes que permitió concluir que el material idóneo para el brazo del subsistema Robot PUMA es MDF. Los datos de las celdas para la matriz de Holmes se obtuvieron con el producto del valor de la ponderación de la matriz de criterio (Tabla 9) por el valor de la ponderación de la calificación de las opciones (Tabla 10).

**Tabla 11**

*Matriz de Holmes de la categoría ‘Material del brazo’ para el subsistema Robot PUMA*

	<b>Peso</b>	<b>Costo</b>	<b>Manufactura</b>	<b>Total</b>
<b>Aluminio</b>	0.125	0.083	0.069	0.278
<b>Acrílico</b>	0.125	0.083	0.028	0.236
<b>MDF</b>	0.25	0.167	0.069	0.486
			<b>Total</b>	<b>1</b>

Este proceso de selección se realizó para el resto de las categorías, en las cuales únicamente se incluye la matriz de Holmes y una explicación sobre los criterios de selección para cada categoría.

### 3.2.2.2 Categoría ‘Material de la muñeca’ del subsistema Robot PUMA

En la Tabla 12 se muestra la matriz de Holmes de la categoría ‘Material de la muñeca’. Los criterios de selección considerados fueron: peso, costo y manufactura, debido a que estas características abarcan mayoritariamente los requisitos del diseño. Se logró determinar que el material más adecuado para la construcción de la muñeca es plástico ABS, y el proceso más apto para este material es la impresión 3D.

**Tabla 12**

*Matriz de Holmes de la categoría ‘Material de la muñeca’ para el subsistema Robot PUMA*

	<b>Peso</b>	<b>Costo</b>	<b>Manufactura</b>	<b>Total</b>
<b>Aluminio</b>	0.083	0.042	0.125	0.25
<b>PLA</b>	0.167	0.104	0.042	0.313
<b>ABS</b>	0.25	0.104	0.083	0.438
			<b>Total</b>	<b>1</b>

### 3.2.2.3 Categoría ‘Material de la base’ del subsistema Robot PUMA

En la Tabla 13 se muestra la matriz de Holmes para la categoría ‘Material de la base’. Los criterios de selección fueron: peso, costo y manufactura. El material más idóneo para la

construcción de la base fue madera, para garantizar un equilibrio entre el peso del robot y la estabilidad de la base.

**Tabla 13**

*Matriz de Holmes de la categoría ‘Material de la base’ para el subsistema Robot PUMA*

	<b>Peso</b>	<b>Costo</b>	<b>Manufactura</b>	<b>Total</b>
<b>Aluminio</b>	0.167	0.111	0.069	0.347
<b>Acero inoxidable</b>	0.083	0.056	0.028	0.167
<b>Madera</b>	0.25	0.167	0.069	0.486
			<b>Total</b>	<b>1</b>

#### 3.2.2.4 Categoría ‘Aplicación del efector final’ del subsistema Robot PUMA

En la Tabla 14 se muestra la matriz de Holmes para la categoría ‘Aplicación del efector final’. Los criterios de selección fueron: peso, costo y manufactura. La aplicación seleccionada para este robot fue de simulación de soldadura. Sin embargo, el diseño del robot se enfoca en un efector intercambiable, para futuras investigaciones.

**Tabla 14**

*Matriz de Holmes de la categoría ‘Aplicación del efector final’ para el subsistema Robot PUMA*

	<b>Peso</b>	<b>Costo</b>	<b>Manufactura</b>	<b>Total</b>
<b>Pick and Place</b>	0.111	0.083	0.139	0.333
<b>Ensamblaje</b>	0.056	0.042	0.069	0.167
<b>Soldadura</b>	0.167	0.125	0.208	0.5
			<b>Total</b>	<b>1</b>

#### 3.2.2.5 Categoría ‘Actuadores’ del subsistema Robot PUMA

En la Tabla 15 se muestra la matriz de Holmes para la categoría ‘Actuadores’, que se refiere a la selección de los mismos. Los criterios de selección fueron: precisión, costo y control, características más importantes para estos elementos. Los actuadores seleccionados son

servomotores, debido a su alta precisión y fácil control, que garantiza un rendimiento óptimo del robot.

**Tabla 15**

*Matriz de Holmes de la categoría 'Actuadores' para el subsistema Robot PUMA*

	<b>Precisión</b>	<b>Costo</b>	<b>Control</b>	<b>Total</b>
<b>Motor DC</b>	0.042	0.104	0.083	0.229
<b>Servomotor</b>	0.104	0.042	0.25	0.396
<b>Motor a pasos</b>	0.104	0.104	0.167	0.375
			Total	1

### 3.1.2 Selección de conceptos del subsistema Robot PUMA

Como se determinó en la anterior sección, con la ayuda de las matrices de Holmes se logró concluir los conceptos para el subsistema Robot PUMA. La Tabla 16 muestra los conceptos seleccionados para tal fin.

**Tabla 16**

*Conceptos seleccionados para cada categoría del subsistema Robot PUMA*

<b>Categoría</b>	<b>Opción</b>
Material brazo	MDF
Material muñeca	ABS
Material base	Madera
Efector final	Soldadura
Actuador	Servomotor

## 3.3 Subsistema Teleoperación

### 3.3.1 Diseño de la Teleoperación

#### 3.3.1.1 Diseño de la HMI

El subsistema Teleoperación es aquel que permite realizar el control del subsistema Robot PUMA y el subsistema Simulador, y también permite la comunicación entre los dos subsistemas.

El software utilizado para el desarrollo de las ventanas y la programación de la HMI fue PyQt Designer 4, el cual se basa en el lenguaje de programación Python 2. En la Figura 35 se muestra la primera ventana de la HMI.



**Figura 35.** Ventana de inicio de la HMI de control

Para la ventana principal de la HMI se consideró los tipos de control para el movimiento del robot. Dentro de las configuraciones se implementó, implícitamente, dos de los tres tipos de control: directo y supervisado. En la Figura 36 se muestra las opciones presentes en la ventana principal. El control directo se utiliza en las opciones “DIRECTA”, “JOY” y “CAD”, el control supervisado se utiliza en las opciones “INVERSA” y “DIBUJO”.



**Figura 36.** Ventana principal de la HMI de control

La opción “DIRECTA” se refiere a la cinemática directa del robot, es decir que cumple con la ecuación (3-3) para la posición del efector final en función a las variables articulares. Al ingresar a la opción “DIRECTA” se despliega una ventana, como la que se muestra en la Figura 37. Aquí se ingresan los valores de los seis servomotores, identificados con las etiquetas \*\*, o se borran para generar los valores necesarios para ir a la posición inicial del robot. Además, se puede visualizar la posición del efector final alcanzada con dichos valores.



Online

### CINEMÁTICA DIRECTA DEL ROBOT PUMA

INGRESE LOS VALORES DE LOS ÁNGULOS DE CADA UNO DE LOS ESLABONES

Q1:  -60°..... 60°

Q2:  -60°..... 60°

Q3:  -60°..... 60°

Q4:  -60°..... 60°

Q5:  -60°..... 60°

Q6:  -60°..... 60°

Cordenadas del efector final

X: 300.00 mm

Y: 60.00 mm

Z: 105.00 mm

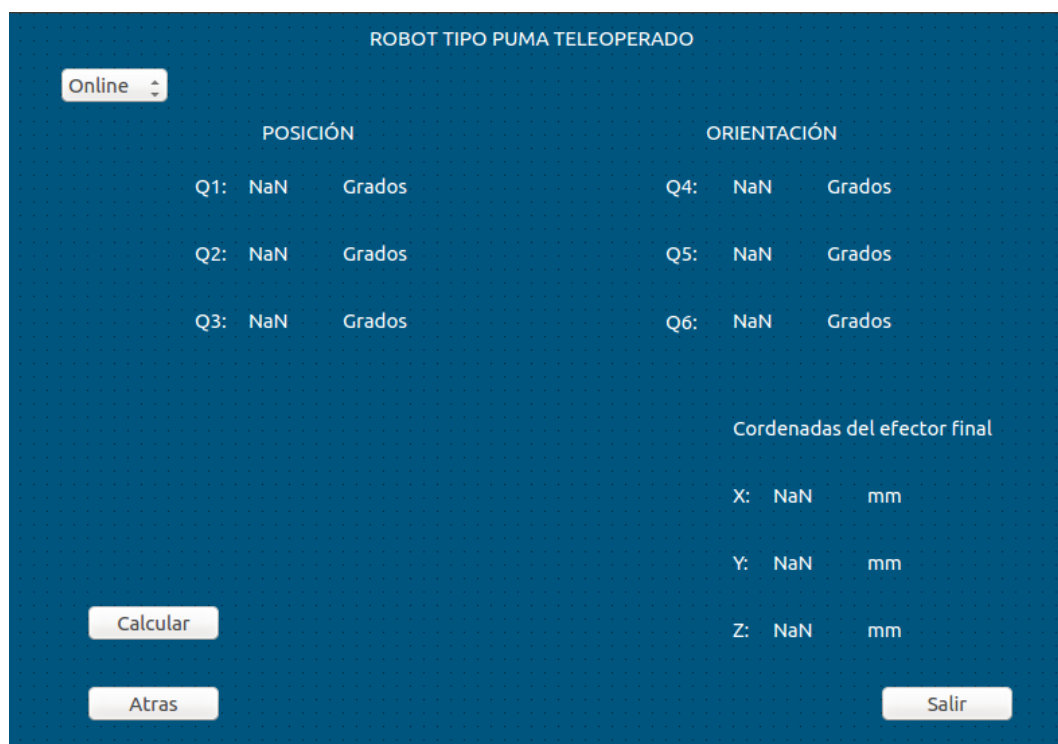
**Figura 37.** Ventana de cinemática directa del Robot PUMA de la HMI de control

La opción “INVERSA” se refiere a la cinemática inversa del robot, es decir que cumple con las ecuaciones (3-5), (3-6), (3-7), (3,8), (3-9) y (3-10) para el valor de las variables articulares en función de la posición del efector final. Al ingresar a la opción “INVERSA” se despliega una ventana como la que se muestra en la Figura 38. Aquí se ingresan la posición x, y, z del efector final para alcanzar dicha posición. Además, se puede visualizar el valor de los diferentes servomotores para alcanzar dicha posición.

The image shows a software interface titled "CINEMÁTICA INVERSA DEL ROBOT TIPO PUMA". It features a dark blue background with white text and input fields. At the top, it asks the user to "Ingrese las coordenadas en el espacio que se requiere del efector final" (Enter the coordinates in the space required for the final effector). There are three input fields for X, Y, and Z coordinates, each followed by "mm". The X field contains "300", the Y field contains "60", and the Z field contains "105". To the right of these fields is a dropdown menu set to "Online". Below the input fields are two buttons: "BORRAR" (Clear) and "ENVIAR" (Send). Underneath, it says "Los valores de los ángulos necesarios para alcanzar esa posición del efector final" (The values of the angles necessary to reach that position of the final effector). There are six rows of output, each labeled Q1 through Q6, followed by "NaN" and "Grados" (Degrees). At the bottom, there are two buttons: "Atras" (Back) and "Salir" (Exit).

**Figura 38.** Ventana de cinemática inversa del Robot PUMA de la HMI de control

La opción “JOY” permite controlar todos los servomotores del robot con una palanca joystick. El control es directo, ya que se envía directamente el valor de la palanca hacia los actuadores. A diferencia del resto de opciones, “JOY” únicamente permite controlar al Robot PUMA, no al simulador, debido a la alta demanda de memoria por la velocidad de transmisión de datos. En la Figura 39 se muestra la ventana de la opción mencionada.



**Figura 39.** Ventana de control por Joystick de la HMI de control

La opción “DIBUJO” permite realizar dibujos de figuras geométricas circunscritas. En la Figura 40 se muestra la ventana “DIBUJO”, en la cual se ingresa la figura a dibujar y las características del círculo circunscrito: centro y radio. El control de esta operación es asistido, y se compone de dos fases: en la primera se ubica al efector final en la posición del inicio del dibujo, y en la segunda se realizan pequeños pasos hasta completar la figura.

DIBUJANDO FIGURAS BASICAS CON EL ROBOT TIPO PUMA

Online ▾

Escoja la figura que requiera:  ▾

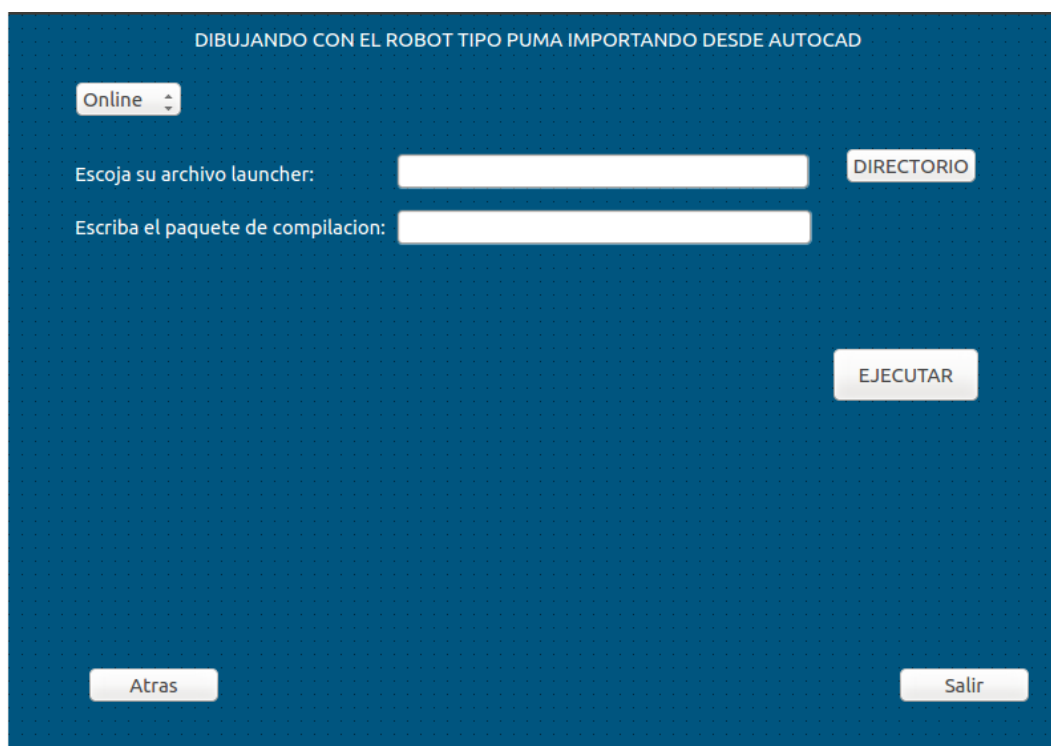
Ingrese el centro de la figura: x:  mm y:  mm

Ingrese las dimensiones de la figura: Radio:  mm

Ingrese el numero de lados del poligono: Lados:

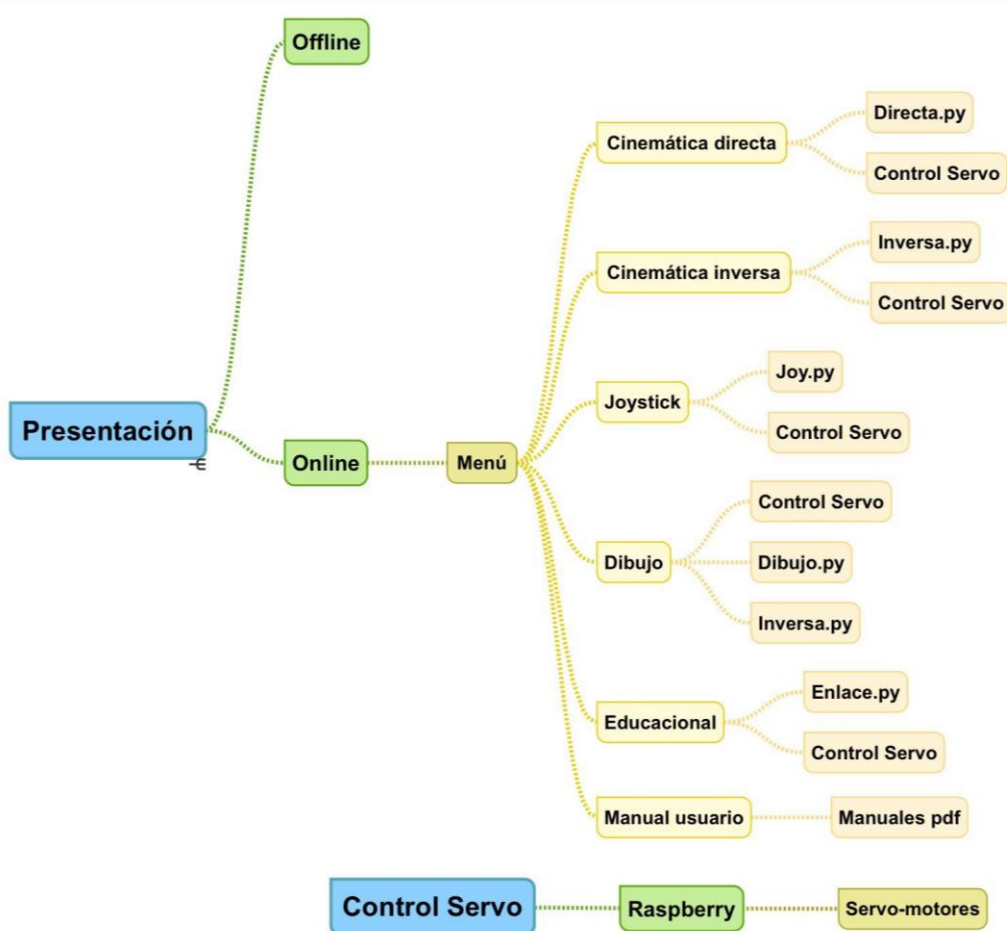
**Figura 40.** Ventana de Dibujo de la HMI de control

La opción “CAD” permite ingresar un archivo de control de movimiento, con una extensión del tipo .launch, de tal manera que los movimientos programadas en tal archivo se transmiten al Robot PUMA y al simulador. En la Figura 41 se muestra la interfaz de esta ventana, donde se carga el archivo y se genera el movimiento. El control para la opción “CAD” es directo, pues se controla el movimiento directamente con el manejo de las variables articulares.



**Figura 41.** Ventana de carga de archivos de la HMI de control

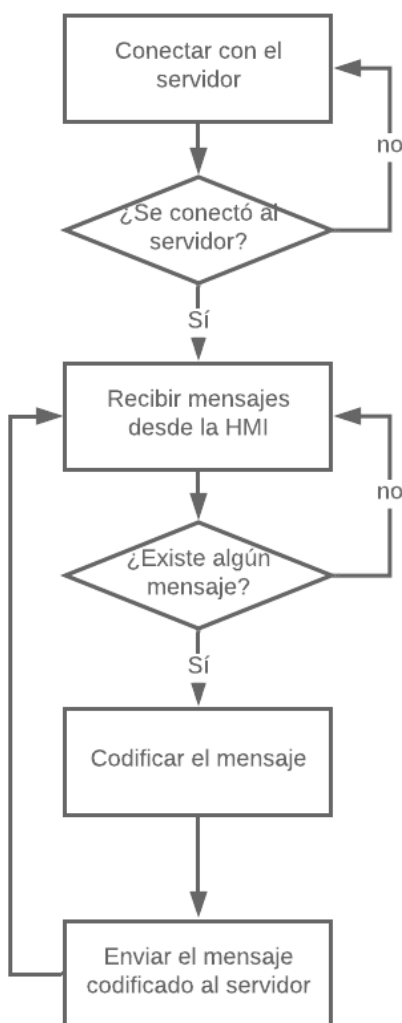
Todas las opciones presentan una opción “ONLINE/OFFLINE”, la cual permite realizar el control sobre los subsistemas Robot PUMA y Simulador simultáneamente (ONLINE) o únicamente sobre el Robot PUMA (OFFLINE), con excepción de la opción “JOY”. De igual manera, en todas las opciones existen restricciones sobre los valores de ingreso, de tal manera que únicamente se puede manejar ciertos rangos en cada opción, para salvaguardar la integridad del Robot PUMA y del operario. En la Figura 42 se muestra la arquitectura que siguen los nodos de control del subsistema Teleoperación.



**Figura 42.** Arquitectura de los nodos de control del subsistema Teleoperación

### 3.3.1.2 Diseño de los nodos de comunicación y control

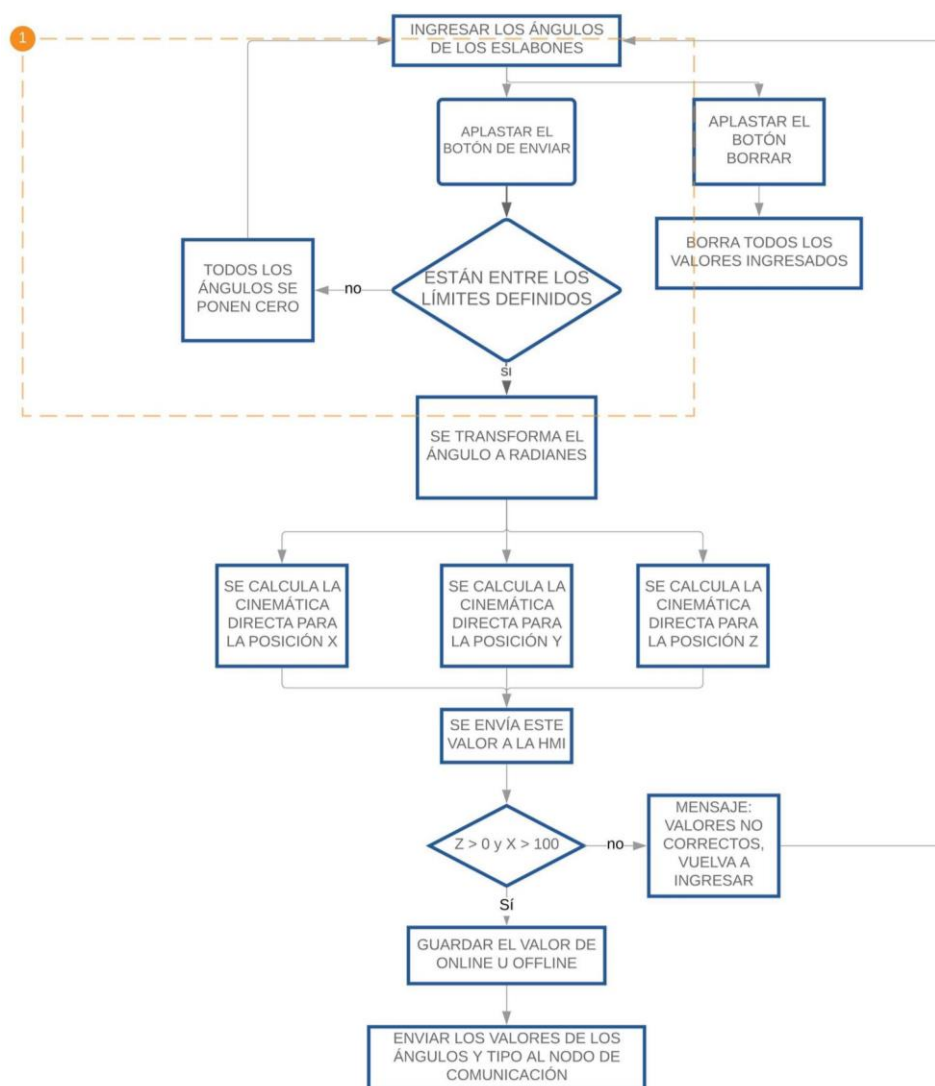
Después del diseño de la HMI, se procedió a realizar la comunicación entre ROS y Blender. Cabe indicar que los nodos de comunicación y control fueron diseñados en ROS Kinetic. La comunicación se basa en un sistema servidor – consumidor, donde el subsistema Simulación es el servidor, y la HMI es el consumidor. Para lograr la comunicación se utilizó un nodo en ROS que se basa en la lógica descrita por el diagrama de flujo de la Figura 43.



**Figura 43.** Diagrama de flujo del nodo de comunicación entre ROS y Blender

Una vez establecido el nodo de comunicación entre ROS y Blender, se procedió a realizar los nodos de control para las diferentes configuraciones del sistema. En la Figura 44 se muestra el diagrama de flujo en el cual se basa la cinemática directa. En primer lugar, se ingresan los valores de las variables articulares del Robot PUMA, los cuales están definidos por restricciones dentro de la programación. Después, se realiza una transformación de los ángulos a radianes para realizar el cálculo de la posición  $x$ ,  $y$ ,  $z$  del efector final, la cual es mostrada en la HMI. Por último, dependiendo la configuración establecida ONLINE/OFFLINE se procede a enviar los datos

correspondientes a los subsistemas Robot PUMA y Simulación, o únicamente al simulador, respectivamente.

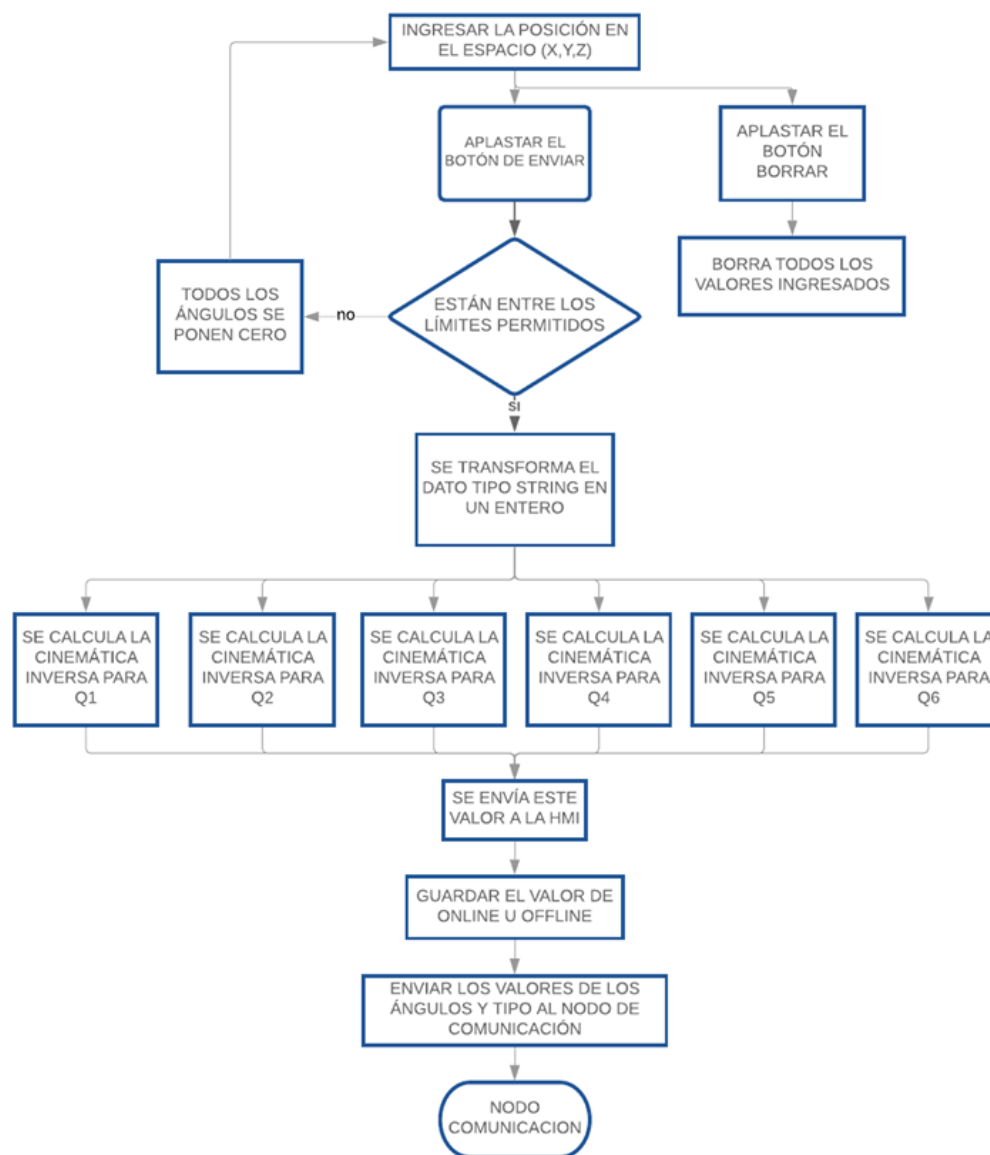


**Figura 44.** Diagrama de flujo de la cinemática directa del Robot PUMA

El nodo de cinemática inversa se basa en el algoritmo descrito en el diagrama de flujo de la Figura 45. En primer lugar, se ingresa la posición x, y, z del efector final, y al igual que en la cinemática directa, se verifican los límites permitidos para el ingreso. Una vez definida la posición deseada, se transforma las coordenadas x, y, z en las variables articulares, como se definió en las



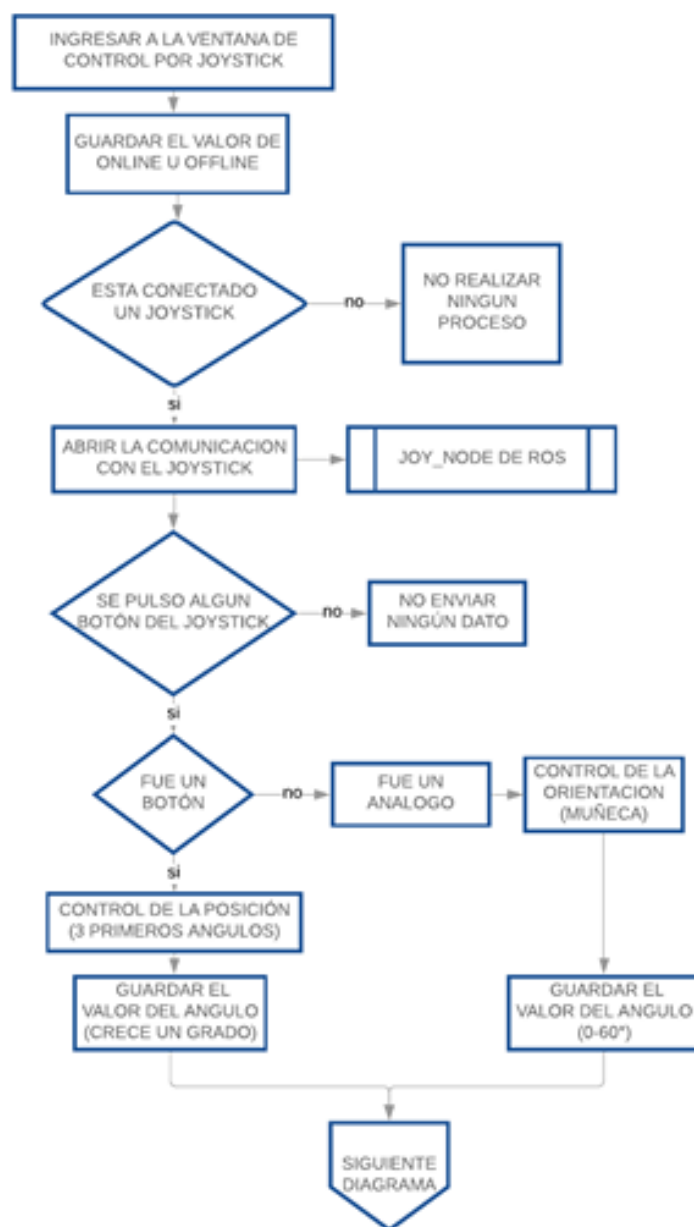
Ecuaciones (3-5), (3-6), (3-7), (3-8), (3-9), (3-10). Los datos obtenidos se muestran en la HMI y se envía al Robot PUMA y al simulador, de acuerdo con la configuración ONLINE/OFFLINE establecida.



**Figura 45.** Diagrama de flujo de la cinemática inversa del Robot PUMA

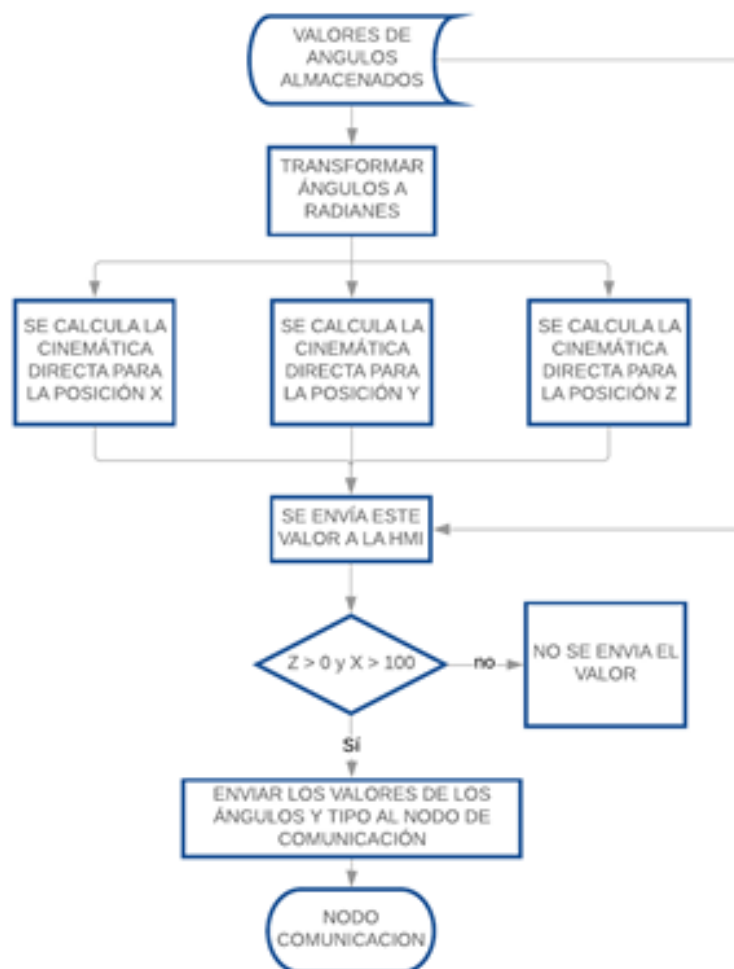
El nodo joystick, cuya lógica se muestra en las Figura 46 y Figura 47 posee la restricción inicial de verificar la conexión de una palanca joystick. En caso de no existir dicha conexión, o no

reconocer al dispositivo conectado, el nodo no realiza ningún proceso. En caso de establecer la conexión, se establece una configuración para definir la posición del Robot PUMA y su orientación: los botones de la palanca se encargan de establecer la posición del efector final, y los joysticks análogos establecen la orientación del mismo.



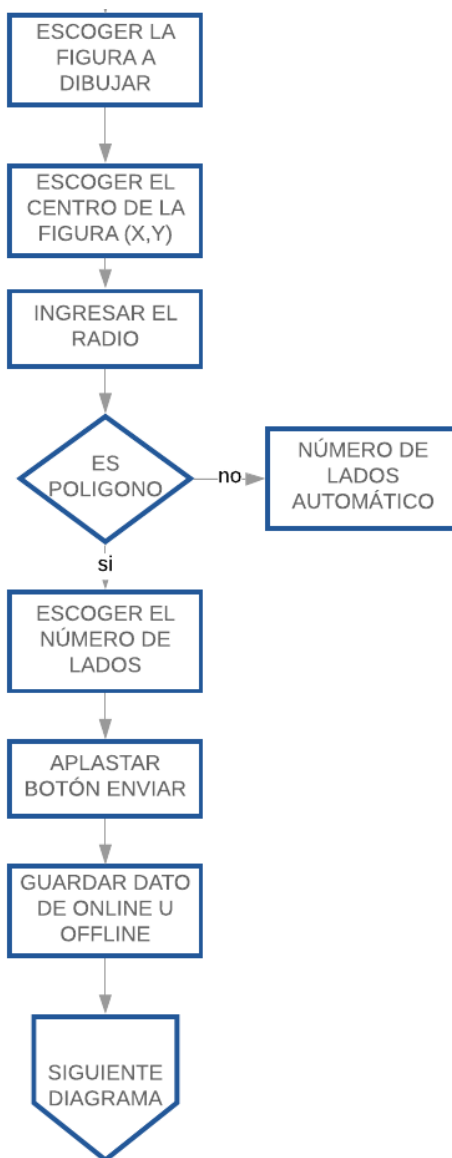
**Figura 46.** Diagrama de flujo del control por joystick del Robot PUMA (primera parte)

Después del reconocimiento entre posición y orientación de la palanca (botones y joysticks análogos, respectivamente) se transforman los datos en radianes. Para la posición se realiza un contador de incremento y decremento para los ángulos, de tal manera que cada pulso del botón de incremento o el botón de decremento cambien el contador de los ángulos, correspondientemente. En cambio, los joysticks análogos se multiplican por un valor determinado para cambiar de manera continua el estado de la rotación de la muñeca de Euler. Después, se procede de manera muy similar a la cinemática directa, donde se envía la información hacia los servos, y se imprime la posición del efector final en la HMI.



**Figura 47.** Diagrama de flujo del control por joystick del Robot PUMA (segunda parte)

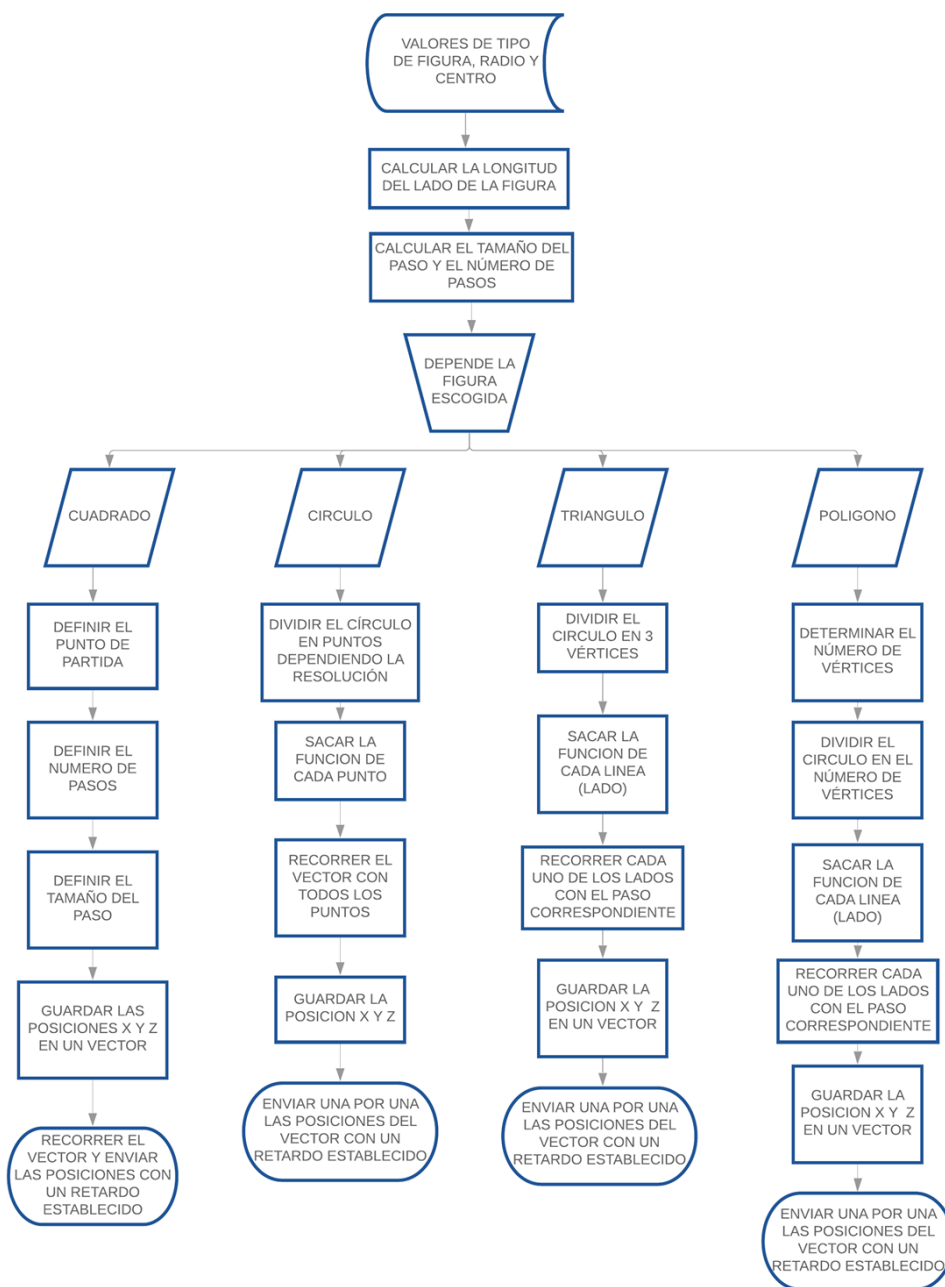
El nodo dibujo se basa en el algoritmo mostrado en la Figura 48, donde el primer paso es la selección de la figura a dibujar y sus características. Debido a que las figuras realizadas son circunscritas, las características son del círculo circunscrito para cada figura, es decir que se debe especificar el centro y el diámetro del mismo. En caso de que la figura a dibujar sea un polígono de 5 lados o más, se debe especificar el número de lados. Una vez enviada la información, se procede a confirmar la opción ONLINE/OFFLINE.



**Figura 48.** Diagrama de flujo del nodo de dibujo (primera parte)

En la Figura 49 se muestra la continuación del algoritmo del nodo dibujo. Una vez confirmadas todas las opciones de la HMI ingresadas por el operario, se procede a realizar los cálculos internos necesarios para la figura seleccionada, como la longitud del lado, el número de pasos necesarios para cada lado y la longitud de paso. Una vez establecidos los parámetros necesarios, se procede a realizar un cálculo de los puntos que se deben alcanzar, de acuerdo con

los pasos definidos, para lo cual se hace un análisis con la cinemática inversa. Al final, se envían los valores de rotación a los servomotores para alcanzar los puntos encontrados, y así, realizar las figuras correspondientes.



**Figura 49.** Diagrama de flujo del nodo de dibujo (segunda parte)

El nodo exportación de archivo se basa en el algoritmo mostrado en la Figura 50. A diferencia del resto de nodos, en este no se ingresa información de manera explícita, sino que el ingreso de información se lo realiza mediante la exportación de un archivo con extensión .launch, el cual debe contener una serie de valores preestablecidos, correspondientes a los movimientos de las articulaciones. El comportamiento es similar al del nodo directa, pues se asigna el valor de cada servomotor, mas no las posiciones a alcanzar.



**Figura 50.** Diagrama de flujo del nodo exportación de archivos

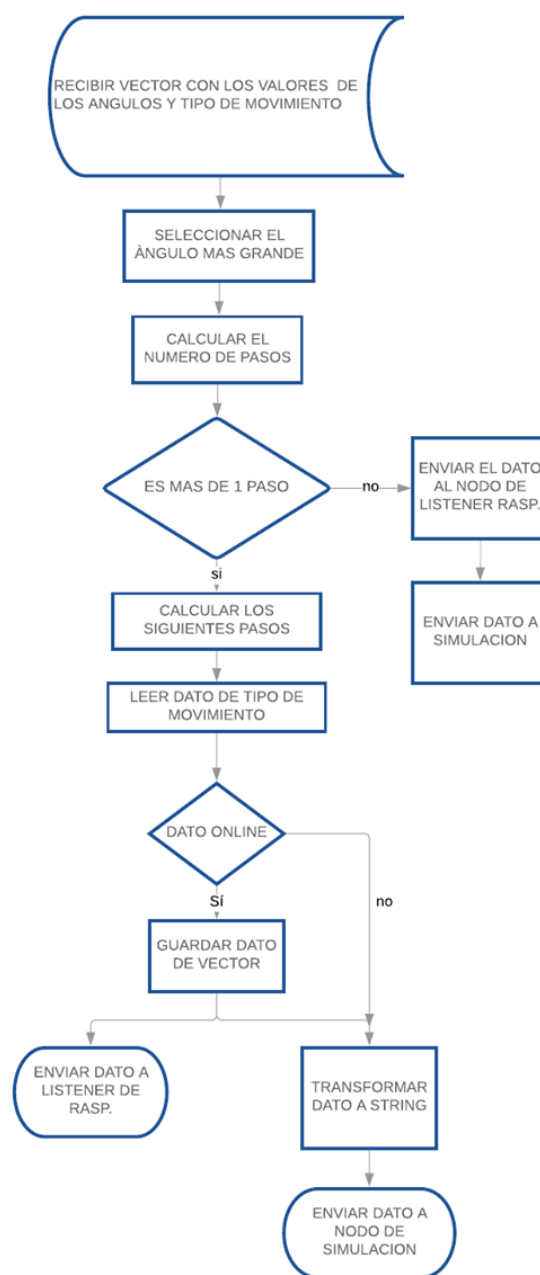
### *3.3.1.3 Comunicación y control desde la HMI hacia los servomotores*

El control de la simulación se realizó por medio del nodo de comunicación, cuyo algoritmo se mostró en la Figura 43. Sin embargo, para lograr la comunicación con el robot PUMA, fue necesario una placa de control que se encargue de la recepción de información de la HMI, y el envío de rotación hacia los servomotores. Dentro del proyecto se utilizó una placa Raspberry Pi 3 B+.

La placa Raspberry funciona como una computadora donde se corre ROS de manera simultánea con el ROS del ordenador central. Para lograrlo, se partió del concepto de los sistemas ROS Multiple Machine, el cual permite definir múltiples dispositivos que manejen sus propios nodos dentro de la plataforma ROS. El sistema es master – slave, donde en la computadora master (maestra) se corre el nodo maestro ‘roscore’, el cual permite correr todos los nodos basados en ROS. Por otro lado, la placa Raspberry se estableció como un slave, misma que utiliza el nodo maestro ‘roscore’ para utilizar sus propios nodos de ROS. Es importante recalcar que únicamente el nodo maestro debe correr en el dispositivo master (en la presente investigación, el ordenador), más no en los dispositivos tipo slave (Raspberry).

En función de lo indicado de los sistemas ROS Multiple Machine, se desarrolló dos nodos para la comunicación y control donde interviene la placa Raspberry: el primero envía la información desde la HMI hacia el robot PUMA y el simulador; y el segundo, escrito y manejado directamente en la placa Raspberry, conecta la HMI con los servomotores. En la Figura 51 se muestra el algoritmo en el cual se basó el primer nodo descrito.

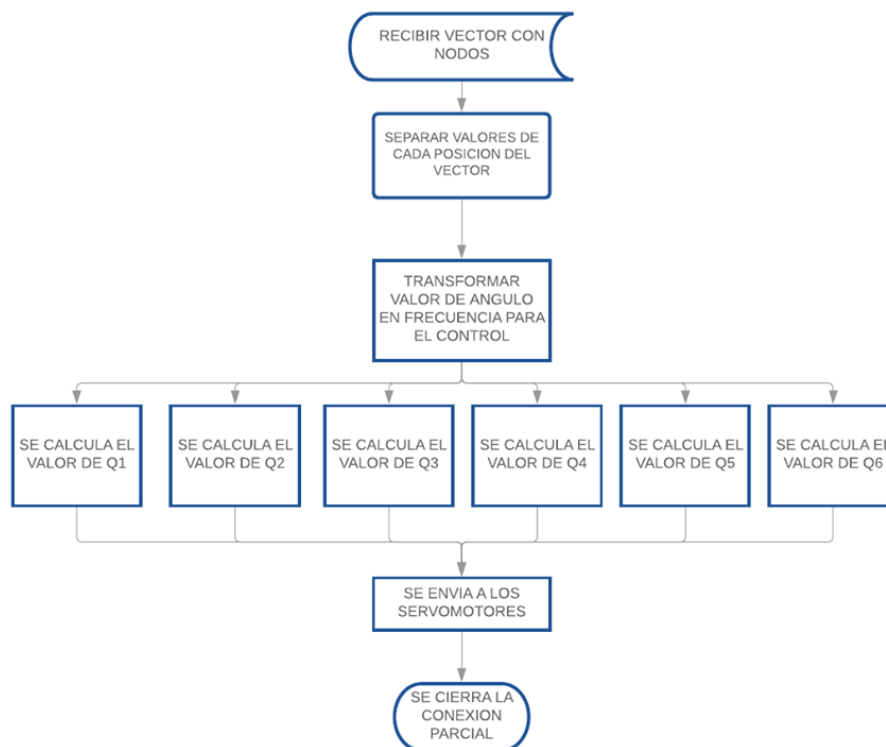




**Figura 51.** Diagrama de flujo del nodo de comunicación entre la HMI y la placa Raspberry /simulador

En la Figura 52 se muestra el algoritmo que se estableció para el nodo receptor de la placa Raspberry. La función de este nodo, como su nombre lo indica) es recibir la información proveniente de la computadora (master), más específicamente de la HMI. Una vez que recibe

información lo manipula, para lograr obtener los seis datos necesarios de movimiento de las articulaciones del robot PUMA.



**Figura 52.** Diagrama de flujo del nodo receptor de la placa Raspberry

### 3.3.2 Desarrollo de conceptos del subsistema Teleoperación

El desarrollo de conceptos para el subsistema Teleoperación se centró en tres categorías: Fase de control, Fase de potencia, HMI. La matriz morfológica de la Tabla 17 muestra las categorías junto a las opciones tentativas de cada una de ellas. La Fase de control hace referencia a la tarjeta que controla los actuadores del sistema, es decir, la interfaz entre el robot físico y la HMI. La Fase de potencia es la alimentación eléctrica de los actuadores. La HMI quiere decir el programa con el cual se desarrolló el diseño y la implementación de la interfaz máquina-hombre.

**Tabla 17**

*Matriz morfológica de los conceptos para el subsistema Teleoperación*

<b>Categoría</b>	<b>Opciones</b>		
<b>Fase de control</b>	Arduino	Raspberry	Microcontrolador
<b>Fase de potencia</b>	Pilas	Batería	Fuente DC
<b>HMI</b>	PyQt	Proface	ViewON3

### *3.3.2.1 Categoría 'Fase de control' del subsistema Teleoperación*

En la Tabla 18 se muestra la matriz de Holmes para la categoría 'Fase de control', la cual se basa en el control de los actuadores del sistema. El proceso se realizó de igual manera que en el diseño del subsistema Robot PUMA. Los criterios de selección para este caso fueron: costo, compatibilidad y confiabilidad. La tarjeta de control seleccionada para el diseño fue Raspberry, la cual permite un mejor control del robot.

**Tabla 18**

*Matriz de Holmes de la categoría 'Fase de control' para el subsistema Teleoperación*

	<b>Costo</b>	<b>Compatibilidad</b>	<b>Confiabilidad</b>	<b>Total</b>
<b>Arduino</b>	0.083	0.104	0.083	0.271
<b>Raspberry</b>	0.042	0.104	0.250	0.396
<b>Microcontrolador</b>	0.125	0.042	0.167	0.333
			<b>Total</b>	<b>1</b>

### *3.3.2.2 Categoría 'Fase de potencia' del subsistema Teleoperación*

En la Tabla 10 se muestra la matriz de Holmes para la categoría 'Fase de potencia', la cual es la selección de la fuente de alimentación eléctrica para el robot. Los criterios de selección para este caso fueron: costo, seguridad y disponibilidad. La fuente de alimentación seleccionada para la fase de potencia fue una fuente DC.

**Tabla 19**

*Matriz de Holmes de la categoría 'Fase de potencia' para el subsistema Teleoperación*

	<b>Costo</b>	<b>Seguridad</b>	<b>Disponibilidad</b>	<b>Total</b>
<b>Pilas</b>	0.125	0.083	0.125	0.333
<b>Batería</b>	0.063	0.167	0.063	0.292
<b>Fuente DC</b>	0.063	0.250	0.063	0.375
			<b>Total</b>	<b>1</b>

### 3.3.2.3 Categoría 'HMI' del subsistema Teleoperación

En la Tabla 20 se muestra la matriz de Holmes para la categoría 'HMI'. Esta categoría busca el software donde se desarrolle la interfaz de control para el robot. Los criterios de selección para este caso fueron: facilidad de programación, velocidad de respuesta y recursos informáticos. El programa seleccionado para el desarrollo de la HMI fue PyQt Designer.

**Tabla 20**

*Matriz de Holmes de la categoría 'HMI' para el subsistema Teleoperación*

	<b>Facilidad programación</b>	<b>Velocidad respuesta</b>	<b>Recursos</b>	<b>Total</b>
<b>PyQt</b>	0.125	0.250	0.125	0.500
<b>Proface</b>	0.063	0.125	0.063	0.250
<b>ViewON3DC</b>	0.063	0.125	0.063	0.250
			<b>Total</b>	<b>1</b>

### 3.3.3 Selección de conceptos del subsistema Teleoperación

Como se determinó en la anterior sección, con la ayuda de las matrices de Holmes se logró concluir los conceptos para el subsistema Teleoperación. La Tabla 21 muestra los conceptos seleccionados para tal fin.

**Tabla 21**

*Conceptos seleccionados para cada categoría del subsistema Teleoperación*

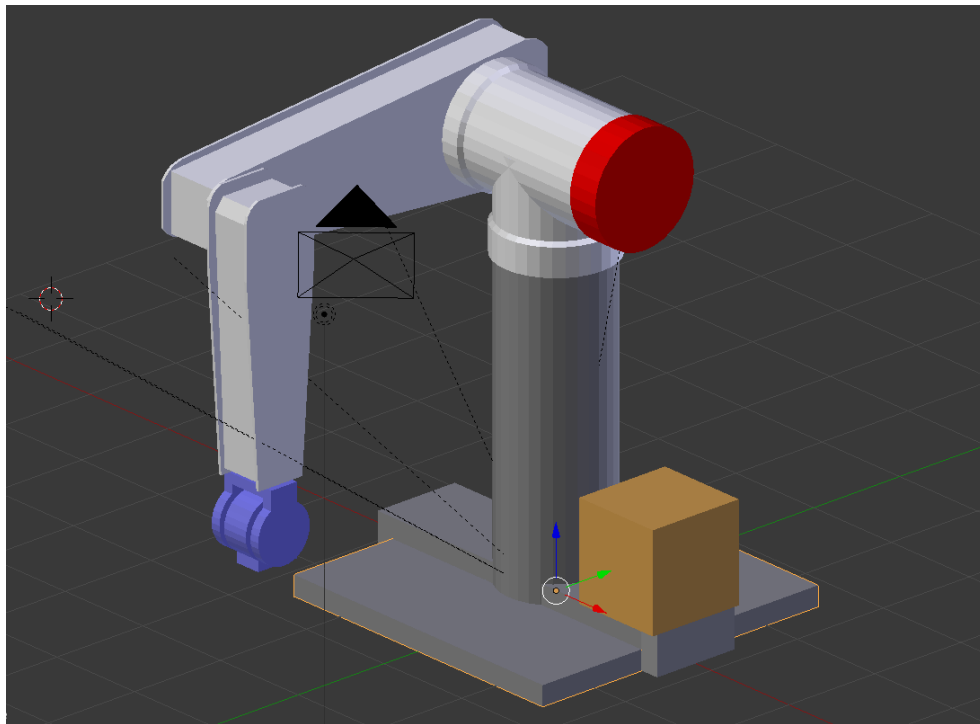
<b>Categoría</b>	<b>Opción</b>
Control	Arduino
Potencia	Pilas
HMI	PyQt

### 3.4 Subsistema Simulación

#### 3.4.1 Diseño del subsistema Simulación

##### 3.4.1.1 Diseño del simulador del Robot PUMA

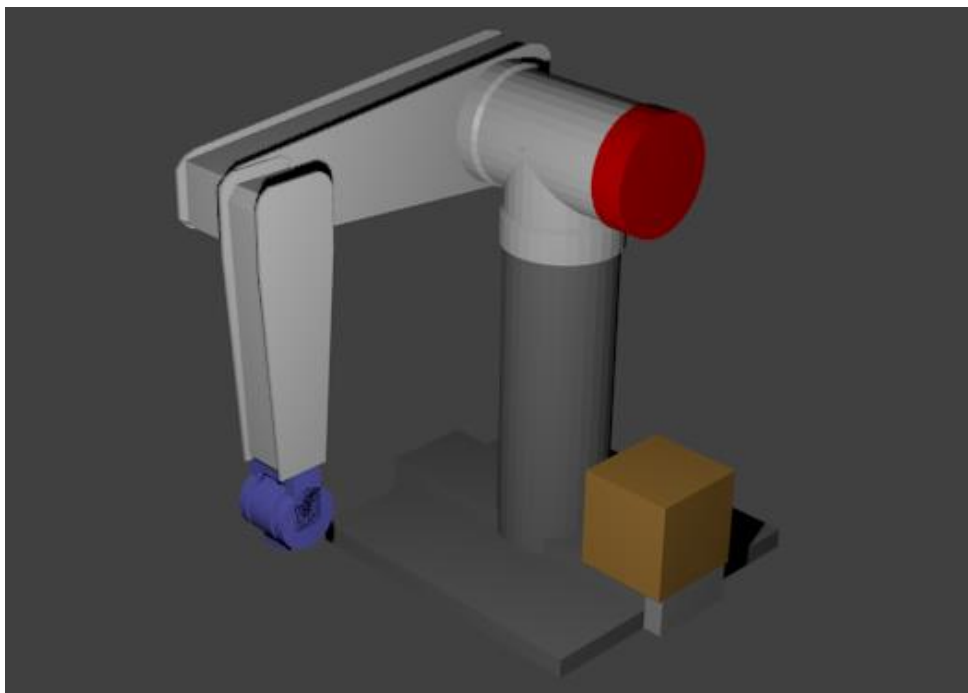
La simulación se realizó en el software Blender v2.79, el cual es un programa de modelado, renderización y animación de gráficos. En la Figura 53 se muestra el modelo realizado del robot. El modelo presenta las rotaciones correspondientes a los movimientos del robot PUMA, tanto para la posición y la orientación.



**Figura 53.** Modelo del simulador realizado en Blender v2.79

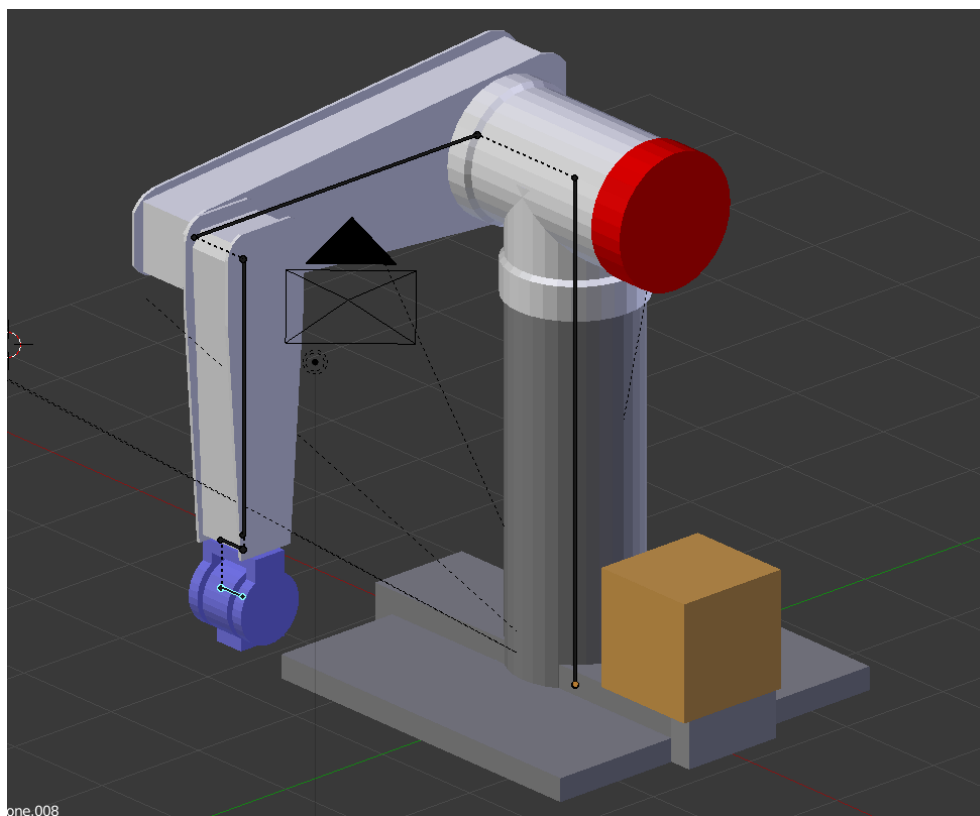
Una de las ventajas al utilizar Blender fue su motor de renderizado, el cual permitió realizar un modelo más realista. En la Figura 54 se muestra el modelo renderizado, utilizando el motor de

renderizado básico: Blender Render. Debido al consumo de memoria, Blender Render permite una locomoción con mayor fluidez, a diferencia del motor de arranque Blender Game.



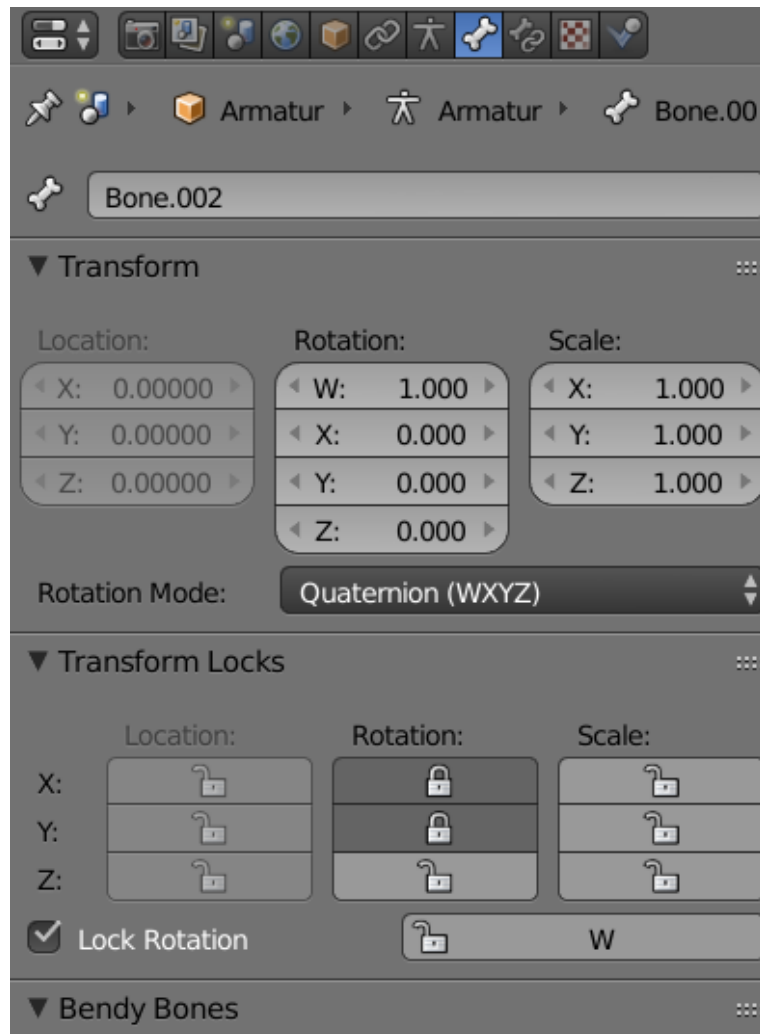
*Figura 54.* Modelo renderizado del simulador realizado en Blender v2.79

Los movimientos del robot son controlados por Bones, los cuales son herramientas que permiten dar una estructura al modelo, brindándole la capacidad de crear movimientos. En la Figura 55 se muestran los Bones del modelo montado sobre el mismo. La conexión entre un Bone y otro permite una articulación similar al esqueleto humano, y es la que permite los movimientos de rotación, que simulan al robot.



**Figura 55.** Modelo del simulador con armadura formada por Bones

El movimiento del modelo está restringido para generar únicamente las rotaciones del robot PUMA, es decir, una rotación sobre el plano Z, dos rotaciones sobre el plano X, y las tres rotaciones propias de la muñeca de Euler. En la Figura 56 se muestra los bloqueos realizados en la configuración de los Bones, en específico del que genera la rotación sobre Z. El bloqueo se realizó para facilitar el movimiento del simulador, sin que existan deformaciones en la geometría del mismo. Además, se realizó un bloque de los desplazamientos de todos los Bones, debido a que los movimientos necesarios únicamente son de rotación.



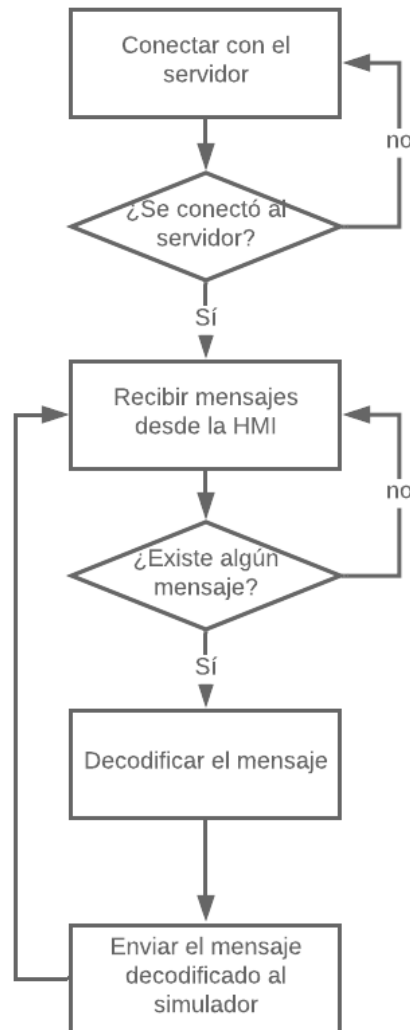
**Figura 56.** Bloqueo de rotaciones para los Bones del simulador

#### 3.4.1.2 Diseño del control del simulador mediante ROS

El objetivo del simulador es el recrear los movimientos del Robot PUMA, de tal manera que es necesario su control desde la HMI. Para tal fin, se creó un nodo de control dentro de Blender, en función al diagrama de flujo de la Figura 57. Primero, se define al simulador como un servidor, asignándole la dirección IP fija 192.168.1.1, de esta manera ROS puede vincularse a Blender como un consumidor. Después, Blender recibe los datos codificados, los decodifica y los manipula de tal manera que descompone el mensaje en seis valores enteros, correspondientes a las seis rotaciones



del Robot PUMA. Por último, los valores descompuestos se envían en forma de vector al simulador, para que los recreé, dando los movimientos correspondientes al robot simulado.



**Figura 57.** Diagrama de flujo del nodo de control del simulador

### 3.4.2 Desarrollo de conceptos del subsistema Simulación

El desarrollo de conceptos para el subsistema Simulación se centró en dos categorías: Software de simulación y Tipo de simulación. La matriz morfológica de la Tabla 22 muestra las opciones de las categorías mencionadas. La categoría Software de simulación es el programa

escogido para diseñar la simulación del robot PUMA. El Tipo de simulación hace referencia a una simulación estática, dinámica o mixta; es decir, las características propias de la simulación, como la forma o estudio de fuerzas, se mantiene constante en el tiempo (estática), si varía con el tiempo (dinámica) o si el operador puede elegir entre ambas (mixta).

**Tabla 22**

*Matriz morfológica de los conceptos para el subsistema Teleoperación*

<b>Categoría</b>	<b>Opciones</b>		
<b>Software de simulación</b>	SolidWorks	3D Grafter	Blender
<b>Tipo de simulación</b>	Estática	Dinámica	Mixta

#### *3.4.2.1 Categoría ‘Software de simulación’ del subsistema Simulación*

En la Tabla 23 se muestra la matriz de Holmes para la categoría ‘Software de simulación’. Esta categoría busca el software donde se desarrolle la simulación del robot. Los criterios de selección para este caso fueron: costo, facilidad de programación y recursos informáticos. El programa seleccionado para el desarrollo de la simulación fue Blender. Una característica muy importante del programa es la capacidad de trabajar con programación en Python, la cual es compatible con ROS (plataforma de robótica) y PyQt (software de diseño de HMI seleccionado).

**Tabla 23**

*Matriz de Holmes de la categoría ‘Software de simulación’ para el subsistema Simulación*

	<b>Costo</b>	<b>Facilidad programación</b>	<b>Recursos</b>	<b>Total</b>
<b>Webots</b>	0.104	0.104	0.069	0.278
<b>RoboDK</b>	0.104	0.104	0.069	0.278
<b>Blender</b>	0.208	0.208	0.028	0.444
			<b>Total</b>	<b>1</b>

### 3.4.2.2 Categoría 'Tipo de simulación' del subsistema Simulación

En la Tabla 24 se muestra la matriz de Holmes para la categoría 'Tipo de simulación'. Esta categoría hace referencia al comportamiento de la simulación (estática, dinámica o mixta). Los criterios de selección para este caso fueron: costo, facilidad de programación y estética. La simulación más adecuada es la estática, debido a los requerimientos del cliente.

**Tabla 24**

*Matriz de Holmes de la categoría 'Tipo de simulación' para el subsistema Simulación*

	Costo	Facilidad programación	Estética	Total
Estática	0.250	0.167	0.028	0.444
Dinámica	0.167	0.111	0.056	0.333
Mixta	0.083	0.056	0.083	0.222
			Total	1

### 3.4.3 Selección de conceptos del subsistema Simulación

Como se determinó en la anterior sección, con la ayuda de las matrices de Holmes se logró concluir los conceptos para el subsistema Simulación. La Tabla 25 muestra los conceptos seleccionados para tal fin.

**Tabla 25**

*Conceptos seleccionados para cada categoría del subsistema Teleoperación*

Categoría	Opción
Software de simulación	Blender
Tipo de simulación	Estática

## 3.5 Resumen

El capítulo expuso todas las consideraciones realizadas para el diseño del robot PUMA, su teleoperación y el simulador. Para el diseño del robot PUMA se utilizó las características del diseño CAD realizado, donde se determinó su forma y dimensiones; además, se realizó el análisis

matemático para encontrar la cinemática y la dinámica del sistema, herramientas que permitieron definir el posicionamiento del efector final y dimensionar los actuadores, respectivamente. El sistema de teleoperación utilizó dos terminales: un ordenador donde se encuentra la HMI, y una placa Raspberry que se encarga de controlar los movimientos del robot, en función a la información recibida. La teleoperación se basó en un sistema Multiple Machine, donde en ambas terminales se ejecutan nodos de ROS, pero únicamente el ordenador opera como nodo maestro. Por último, el simulador se desarrolló en Blender v2.79, empezando por el modelado del robot PUMA en la computadora. Después, se utilizó nodos para comunicar al simulador con la HMI, para lo cual se estableció un sistema servidor – consumidor, donde el simulador trabaja como servidor.

## CAPÍTULO 4

### PRUEBAS Y RESULTADOS

El capítulo expone las pruebas realizadas en base a los objetivos planteados al inicio de la investigación, así como los resultados obtenidos de los mismos. Las pruebas se basan en el funcionamiento del Robot PUMA diseñado, el funcionamiento de la simulación realizada, y la comunicación entre ambos sistemas al trabajar en conjunto.

#### 4.1 Robot PUMA

En la Figura 58 se muestra el robot PUMA realizado. Para las pruebas del mismo, se tomó en cuenta tres aspectos: velocidad de trabajo, espacio de trabajo y control de movimiento. En términos generales, para la primera prueba se buscó la velocidad de trabajo más adecuada, en el espacio de trabajo se comprobó que no existan golpes por parte del robot hacia sí mismo, y en el control del movimiento se buscó determinar la calidad del movimiento generado por el robot.



*Figura 58.* Robot PUMA diseñado

Dentro de las pruebas realizadas en el Robot PUMA, la más importante fue el control de la velocidad óptima de uso. En la Tabla 26 se muestran las velocidades nominales de los servomotores para las diferentes rotaciones del robot, con las cuales se realizó la primera prueba de velocidad. Se decidió utilizar velocidades menores debido a que las velocidades nominales no garantizaban la seguridad del operario ni del robot.

**Tabla 26**

*Velocidades nominales de los servomotores*

ARTICULACIÓN	SERVOMOTOR	VELOCIDAD (grados/s)
1	FT5335M	300
2	FT5335M	300
3	HS-705-MG	230
4	MG90S	600
5	MG90S	600
6	MG90S	600

Después de realizar la primera prueba, se redujo la velocidad de los motores dentro de la programación. Para la segunda prueba se utilizaron las velocidades de la Tabla 27, las cuales corresponden a un tiempo de espera de 300 milisegundos entre grados. Los movimientos del robot no generan grandes aceleraciones, de tal forma que se redujo la posibilidad de dañar al robot o al operario. Sin embargo, los movimientos perdieron fluidez y generan un mayor torque sobre los motores, de tal manera que no se seleccionaron estas velocidades.

**Tabla 27**

*Velocidades de los servomotores para la segunda prueba*

ARTICULACIÓN	SERVOMOTOR	VELOCIDAD (grados/s)
1	FT5335M	3.33
2	FT5335M	3.33
3	HS-705-MG	3.33
4	MG90S	3.33
5	MG90S	3.33
6	MG90S	3.33

En la tercera prueba se incrementó la velocidad de los motores dentro de la programación, reduciendo el tiempo de espera entre el envío de posición angular desde la HMI hacia la tarjeta Raspberry. En la Tabla 28 se muestra la velocidad de los servomotores para la tercera prueba. Los resultados fueron positivos para estos valores, debido a que la fluidez del robot mejoró en relación con la segunda prueba, además que la velocidad minimiza la posibilidad de lesión al robot o al operario.

**Tabla 28**

*Velocidades de los servomotores para la tercera prueba*

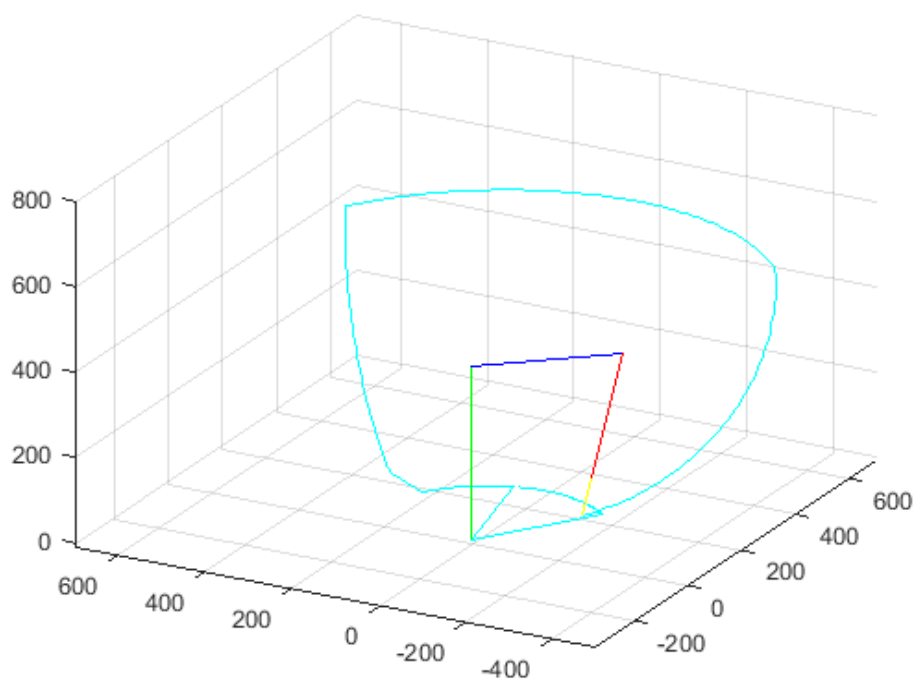
ARTICULACIÓN	SERVOMOTOR	VELOCIDAD (grados/s)
1	FT5335M	20
2	FT5335M	20
3	HS-705-MG	20
4	MG90S	20
5	MG90S	20
6	MG90S	20

La siguiente prueba consistió en una verificación de los rangos de los valores permitidos para el movimiento de los diferentes servomotores. En la Tabla 29 se muestran los rangos mencionados para cada articulación del robot. Con los rangos descritos el robot no presenta colisiones consigo mismo ni con su entorno, por tal motivo se limitó el movimiento a estos valores de rotación. En la Figura 59 se muestra el espacio de trabajo definido por los rangos de valores de rotación de los servomotores.

**Tabla 29**

*Rango de valores de rotación para los servomotores del Robot PUMA*

ARTICULACIÓN	SERVOMOTOR	RANGO (grados)	
		Mínimo	Máximo
1	FT5335M	-60	60
2	FT5335M	-20	20
3	HS-705-MG	-45	45
4	MG90S	-60	60
5	MG90S	-60	60
6	MG90S	-60	60

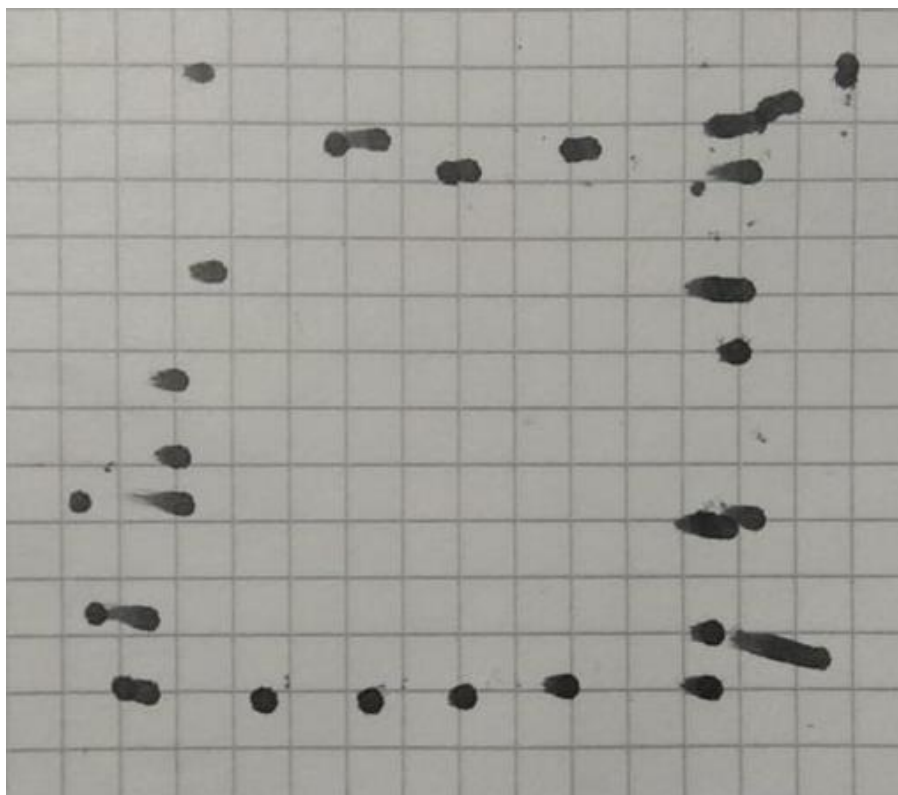


**Figura 59.** Espacio de trabajo para el robot PUMA

La última prueba realizada con el robot fue el control de movimiento al dibujar figuras geométricas. En la Figura 60 se muestra el resultado obtenido al realizar un cuadrado desde la HMI. El modo de operación del nodo Dibujo se basa en realizar puntos a través de la trayectoria del robot. Primero, se determina las coordenadas de los puntos a utilizar; después, con la cinemática inversa se obtienen los valores de las variables articulares para alcanzar los puntos, y todos los



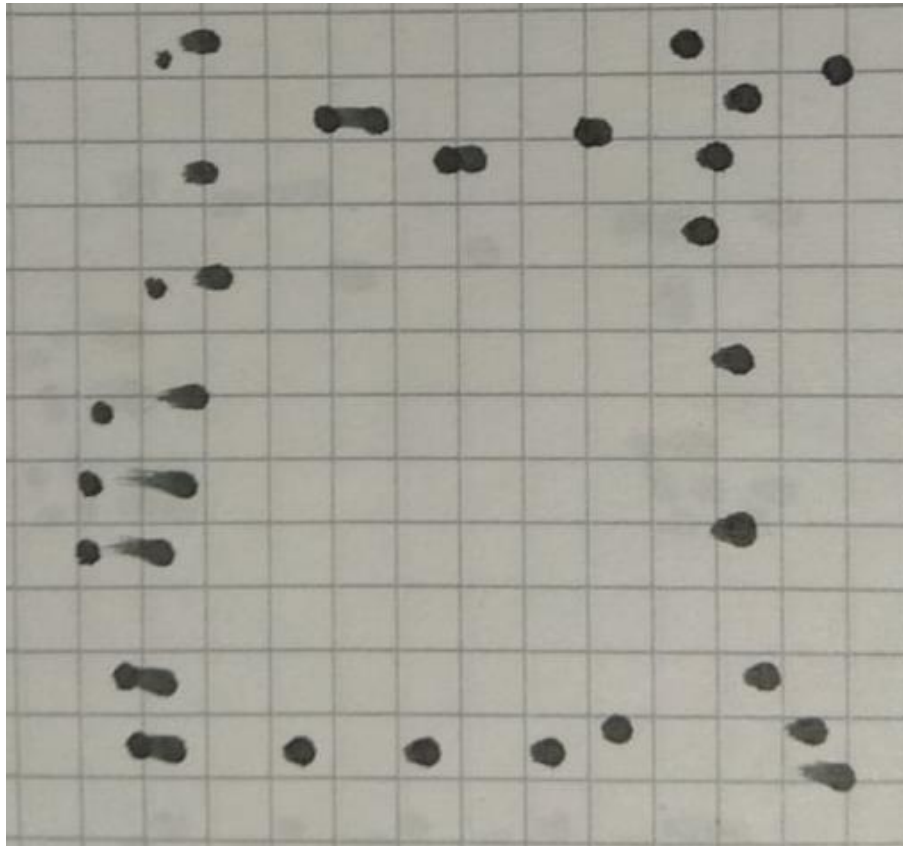
datos generados se almacenan en una matriz. Por último, se llama a cada conjunto de datos de la matriz para dibujar los puntos, con un tiempo de espera entre un punto y otro. Para la prueba, el largo del lado del cuadrado base fue de 5 centímetros.



**Figura 60.** Dibujo realizado por el Robot PUMA

Los resultados de la prueba de dibujo revelaron que los puntos se dispersan alrededor del cuadrado impuesto. La exactitud del sistema es alta, considerando que la dispersión de puntos es uniforme con respecto a la figura base. Sin embargo, el sistema no presenta una precisión adecuada, debido a los cambios pequeños entre los puntos, lo que provocó la dispersión del dibujo. En la Figura 61 se muestra una segunda prueba realizada con los mismos datos. La segunda prueba se realizó para determinar la repetitividad del sistema; al comparar las pruebas realizadas se ve que en la mayor parte del dibujo se tiene una alta repetitividad, mientras que en algunas zonas los

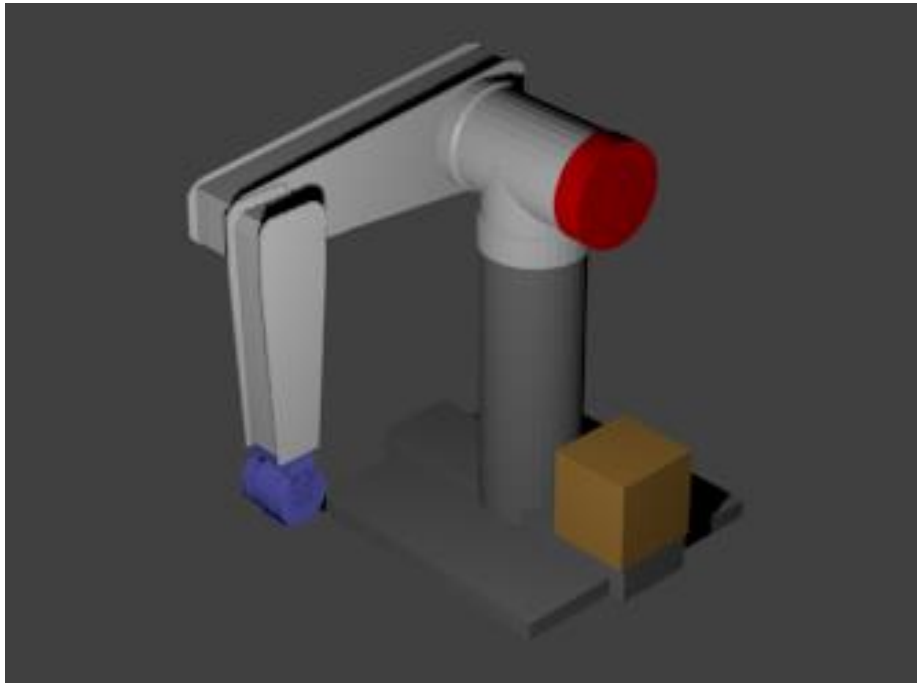
puntos se dispersaron. Las zonas donde la repetitividad es baja son aquellas donde el sistema utiliza una mayor velocidad para alcanzar las coordenadas de los puntos.



*Figura 61.* Segunda prueba de dibujo realizada con el robot PUMA

## 4.2 Simulador

Una vez realizado el simulador, no se realizaron muchas pruebas del mismo de forma independiente al Robot PUMA. La única prueba realizada únicamente con el simulador fue la de los movimientos de cada variable articular, es decir, cada rotación. En la Figura 62 se muestra el resultado final del diseño del simulador. Al realizar la prueba, se obtuvo el valor de memoria requerido por parte del ordenador, y fue de 19.71Mb



*Figura 62.* Renderizado final del simulador del Robot PUMA

Para realizar la prueba de rotación se escribió un código que permita controlar los movimientos desde el teclado del computador. Los resultados encontrados permitieron concluir que la fluidez, dirección y velocidad de los movimientos fue la correcta para proceder a la comunicación con el Robot PUMA. Una vez concluido la prueba, se procedió a implementar el código de comunicación Blender – ROS descrito en el capítulo anterior.

### **4.3 Comunicación**

El punto más importante en la comunicación entre el Robot PUMA y el simulador es la velocidad de respuesta de ambos al enviar la orden desde la HMI. Al realizar las pruebas en conjunto, se vio una simultaneidad en los movimientos del Robot PUMA y del simulador. Las pruebas se realizaron con todas las opciones de la HMI: cinemática directa (DIRECTA), cinemática inversa (INVERSA), joystick (JOY), dibujo de figuras geométricas (DIBUJO) y código (CAD).

En cada caso, los resultados fueron positivos, pues el simulador y el robot PUMA realizaron las acciones requeridas de manera síncrona. De esta manera, se estableció la velocidad de trabajo óptima que garantice la seguridad del operario, del robot y una fluidez en los movimientos del simulador. en la Tabla 30 se muestran los tiempos de respuesta del simulador y del robot. Para todos los casos, el robot PUMA tardó más en realizar las pruebas que el simulador.

**Tabla 30**

*Tiempo de respuesta del simulador y del robot PUMA*

Prueba	Tiempo de respuesta del simulador en segundos (valor teórico)	Tiempo de respuesta del robot en segundos (valor real)	Error porcentual (%)
1	1	1.14	14
2	1	1.11	11
3	1	1.09	9
4	1	1.07	7
5	1	1.08	8
6	1	1.05	5

#### 4.4 Resumen

El capítulo expuso las pruebas realizadas y el análisis de los resultados de cada una de ellas. Para el robot PUMA, las pruebas se centraron en tres puntos principales: velocidad de operación, espacio de trabajo y posicionamiento del efector final. La velocidad de operación para todos los actuadores se estableció en 20 °/s. Con respecto al espacio de trabajo, se definió los rangos de trabajo de todos los servomotores. Y en la prueba del posicionamiento del efector final se determinó que el sistema presenta buena exactitud y repetitividad, pero baja precisión. En cuanto al simulador, únicamente se realizó pruebas con respecto a su movimiento, para determinar que fuera correspondiente al sistema real. Por último, se observó el tiempo de respuesta del ambos sistemas, y se comparó entre ellos

## CAPÍTULO 5

### CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS

#### 5.1 Conclusiones

Se diseñó y construyó un robot PUMA teleoperado con un sistema de simulación en tiempo real. El primer paso del diseño fue el modelado CAD del robot para determinar las medidas, la forma y propiedades como la masa y la inercia. después se realizó el análisis matemático para dimensionar los actuadores en función de los torques máximos encontrados, de tal forma que se emplearon dos servomotores FT5335M, un servomotor HS 705-MG y tres servomotores MG90S, los tres primeros para posicionar al efector final, y los dos últimos para orientarlo.

Se determinó el espacio de trabajo del robot PUMA. Para lograrlo, se partió de la cinemática directa encontrada en el análisis matemático del robot, y se determinó que el espacio de trabajo correspondiente a una porción de una esfera de radio 53 centímetros. Los rangos para los ángulos limitaron la movilidad del robot, con el objetivo de salvaguardar la integridad del sistema y del operario.

Se realizó la HMI de control del sistema dentro de la teleoperación. El diseño de la HMI se realizó en PyQt Designer 4 y se basó un software por ventanas: una de presentación, una ventana principal y cinco ventanas de control. La ventana de presentación contiene la información general del proyecto. La ventana principal permite acceder a todos los controles del robot y el simulador. Las ventanas de control permiten el manejo del robot por parte del operario.

Se analizó la obtención del modelo estático y dinámico de un robot PUMA de 6 articulaciones rotatorias actuado por servomotores. El modelo estático del robot PUMA permitió

definir la cinemática directa y la cinemática inversa, necesarias para el desarrollo de los nodos de control y para el análisis dinámico del robot. En base al modelo estático, se desarrolló la dinámica del robot por medio del modelamiento de Euler – Lagrange, el cual permite igualar los torques requeridos para cada servomotor, con las fuerzas producidas por el movimiento, la velocidad y las aceleraciones de las articulaciones rotatorias. Para facilitar los cálculos se empleó el método del desacople cinemático, al separar el análisis de las tres primeras articulaciones, correspondientes al posicionamiento del efector final, y las tres últimas articulaciones, encargadas de orientar al efector final.

Se estudió e implementó un sistema de control basado en algoritmos de movimiento en base a la cinemática directa e inversa del robot PUMA. Los algoritmos utilizados se basan en el estudio del modelo estático del robot PUMA, del cual se partió para desarrollar la cinemática directa y la cinemática inversa, herramientas necesarias para definir los movimientos del robot, y por consiguiente su control. Dentro de los algoritmos de control se encuentran tanto la cinemática directa como la inversa, el primero para el envío de ángulos hacia los servomotores y la simulación, y el segundo para conocer la posición del efector final en cada momento.

Se desarrolló un sistema de simulación para el robot PUMA que permita realizar operaciones fuera de línea y su teleoperación. La simulación del robot PUMA se realizó en el software Blender v2.79, gracias a las ventajas que presenta el programa, principalmente la capacidad de realizar programación interna en lenguaje Python y que permita una comunicación con ROS. El primer paso fue el modelado del robot en la computadora, para después realizar el control del sistema virtual por parte de la HMI, para lo cual se estableció un sistema servidor –

consumidor, donde se definió al simulador como el servidor asignando una dirección IP fija, y el consumidor fue la HMI, mediante un nodo de comunicación.

## **5.2 Recomendaciones**

La forma y articulaciones del robot PUMA se basaron en el modelo original, es decir, forma antropomórfica y seis articulaciones, tres para posición y tres para orientación del efector final. Se recomienda realizar un diseño basado en un sistema redundante, al implementar otra variable articular al sistema, rotacional o prismática.

El espacio de trabajo se limitó en gran medida debido a las dimensiones de la estructura del robot. Se recomienda realizar un estudio de colisiones para el robot PUMA, con el objetivo de aprovechar el máximo espacio posible dentro de la propia geometría del autómeta.

La HMI de control permite operar el robot desde una posición remota, limitada por la conexión cableada entre las terminales. Se recomienda desarrollar un sistema inalámbrico entre el robot y el ordenador, con el fin de incrementar la distancia de la estación remota.

El análisis cinemático y dinámico se desarrolló mediante la aplicación de fórmulas y análisis matemático. Se recomienda utilizar herramientas y métodos computacionales para facilitar los cálculos necesarios para definir el movimiento del sistema.

El desarrollo de los algoritmos de control se basó en la cinemática directa e inversa para lograr alcanzar las posiciones establecidas. Se recomienda utilizar métodos iterativos para mejorar la predicción de la pose del efector final, en conjunto con los métodos analíticos utilizados en la presente investigación.

El control de los movimientos del simulador se realizó de forma directa. Se recomienda implementar un banco de datos donde se almacenen los movimientos de forma preestablecida, con el fin de requerir menor información por parte de la HMI de control.

### **5.3 Trabajos futuros**

La presente investigación se basó en desarrollar un robot PUMA y un simulador con propósitos educativos. Por esta razón se recomienda realizar investigaciones de aprendizaje de robótica por refuerzo tomando como referencia el simulador. El aprendizaje por refuerzo, considerando la presente investigación, consiste en trabajar con el simulador para realizar estudios de posicionamiento y trayectorias para el efector final, de tal manera que no exista el riesgo de cometer accidentes al trabajar con el robot PUMA. Una vez comprobado la factibilidad del comportamiento del simulador, se puede pasar al sistema real.



## BIBLIOGRAFÍA

- Abbott, J., Marayong, P., & Okamura, A. (2007). Haptic Virtual Fixtures for Robot-Assisted Manipulation. *Robotics Research*, 49-64.
- Andino, R. C. (2015). *Diseño e implementación de una estación de*. Universidad de las Fuerzas Armadas ESPE, Sangolqui. Retrieved April 15, 2018, from <http://repositorio.espe.edu.ec/bitstream/21000/9549/1/AC-EAC-ESPE-048541.pdf>
- Arias, J., Bernal, M., & Correa, E. (2011). MICROSOFT ROBOTICS DEVELOPER STUDIO UNA NUEVA TECNOLOGÍA DE SOFTWARE PARA EL DISEÑO, SIMULACIÓN E IMPLEMENTACIÓN DE PLATAFORMAS ROBÓTICAS. *Scientia et Technica*, 201-206.
- Asín, G., Fernández, F., García, R., López, D., Puertas, J., Rodrigo, E., . . . Yago, O. (2007). *Simuladores en Robótica*. Madrid, España.
- Avilés, O., Hernández Juan, & Olier, I. (1999, Noviembre). Una introducción a la robótica industrial. *Revista de la Facultad de Ingeniería*, 53-67. Retrieved April 1, 2018
- Basañez, L. (2015). *Avances en Telerobótica y Redes de Robots*. Cataluña, España.
- BBVA Innovation Center. (2015, mayo). *Realidad Virtual*. Retrieved April 8, 2018, from [www.centrodeinnovacionbbva.com](http://www.centrodeinnovacionbbva.com)
- Cerón, A. (2005, Noviembre). *Sistemas robóticos teleoperados*. *Ciencia e Ingeniería Neogranadina*, 62-72. Retrieved April 1, 2018

- Cervantes, J., Villafuerte, R., & Mejía, E. (2014). Simulador 3D para brazo robot de 4 grados de libertad. *Revista Iberoamericana para la Investigación y Desarrollo Educativo*, 1-19.
- Cortés, J. (2008). Cinemática del brazo articulado PUMA. España.
- Cyberbotics Ltd. (2004). Cyberbotics Ltd. Webots™: Professional Mobile Robot Simulation. *International Journal of Advanced Robotic Systems*.
- Davies, B., Hibberd, R., Ng, W., Timoney, A., & Wickham, J. (1991). The development of a surgeon robot for prostatectomies. *Journal of Engineering in Medicine*, 35-38.
- Denavit, J., & Hartenberg, R. (1955). Kinematic Modelling for Robot Calibration. *ASME Journals of Applied Mechanics*, 215-221.
- ENERGID. (2005). *ENERGID*. Retrieved from Control Adaptativo en Tiempo Real: <https://www.energid.com/actin/control-capabilities>
- Erfani, A., Rezaei, S., Pourseifi, M., & Derili, H. (2018). Optimal control in teleoperation systems with time delay: A singular perturbation approach. *Journal of Computational and Applied Mathematics*, 168-184. Recuperado el 8 de abril de 2018
- Ferrell, W., & Sheridan, T. (1967). Supervisory control of remote manipulation. *IEEE Spectrum*, 81-88.
- Gallo, V. M., & Saá, F. D. (2013). *Diseño e implementación de una celda de manufactura con brazo robótico centralizado y control de calidad con visión artificial en el Laboratorio de Robótica Industrial de la ESPE-L*. Latacunga: Latacunga ESPE. Retrieved abril 14, 2018, from <http://repositorio.espe.edu.ec/handle/21000/6788>

Ghosh, B., Xi, N., & Tarn, T. (1999). *Control in Robotics and Automation*. San Diego: Academic Press.

Goertz, R., & Thompson, W. (1954). Electronically controlled manipulator. *Nucleonics*.

*Google Sites*. (n.d.). Retrieved from Simuladores de Brazo Robot:  
<https://sites.google.com/site/proyectosroboticos/Descargar-Simuladores>

Harib, K., Ullah, S., Moustafa, K., & Zenieh, S. (2012). Parallel, Serial and Hybrid Machine Tools and Robotics Structures: Comparative Study on Optimum Kinematic Designs. (S. Kucuk, Ed.) *Serial and Parallel Robot Manipulators*. doi:10.5772/33797

Hendrix, A. (2019, Marzo 20). *Running ROS across multiple machines*. Retrieved from ROS.org:  
<http://wiki.ros.org/ROS/Tutorials/MultipleMachines?fbclid=IwAR2C-g7oOANgg2kcyqg-frV7dexms0kY7vxH4TTwNPia7Dm7zlEV-RQMKOI>

Hernandez, S., & Herrero, F. (2015). *Multi-master ROS systems*. Barcelona: Institut de Robòtica i Informàtica Industrial.

Hirzinger, G., Brunner, B., Dietrich, J., & Heindl, J. (1994). ROTEX-the first remotely controlled robot in space. *Proceedings of the 1994 IEEE International Conference on Robotics and Automation* (pp. 2604-2611). San Diego: IEEE.

Hirzinger, G., Landzettel, K., Reinstema, D., Preusche, C., Albu-Schäffer, A., Rebele, B., & Turk, M. (2005). ROCKVISS - Robotics component verification on ISS. *iSAIRAS*. Munich: IEEE.

- Ingalls, R. (2001). Introduction to simulation. *33rd conference on Winter simulation* (pp. 7-16).  
Arlington: IEEE.
- Jaramillo, A. (2005). Cinemática de Manipuladores Robóticos.
- Kumar, P., & Narayan, Y. (2011). Simulation in Robotics. *National Conference on "RECENT ADVANCES IN MANUFACTURING ENGINEERING & TECHNOLOGY"*, (pp. 24-30).  
Hyderabad.
- Larsen, L., Kim, J., Kupke, M., & Schuster, A. (2017). Automatic Path Planning of Industrial Robots Comparing Sampling-based and Computational Intelligence Methods. *Procedia Manufacturing*, 241-248. Retrieved Abril 1, 2018
- Martín, P., Hernández, O., & Toquica, J. (2015). Diseño y simulación de un vehículo prototipo para la distribución automática de fertilizantes agrícolas. *Épsilon*, 49-70.
- Ministerio de Salud Pública. (2015). *Sistema Quirúrgico Da Vinci*. Quito: Dirección de Inteligencia de Salud.
- Muñoz, E., Paruma, O., & Flores, J. (2004). Aplicaciones de las señales mioeléctricas para el control de interfaces hombre-máquina. *VI Congreso de Tecnologías Aplicadas a la Enseñanza de la Electrónica*.
- Quimbita, W. E., Galarza, O. V., Tumbaco, D. C., & Rivas, D. R. (2014). *Robot delta controlado con sistema operativo robotico (R.O.S.)*. Tesis Doctoral, Universidad de las Fuerzas Armadas ESPE, Sangolquí. Retrieved abril 14, 2018, from <http://revistas.ustabuca.edu.co/index.php/ITECKNE/article/view/821/636>

Ramírez, K. (2012). *Cinemática Inversa del Robot*. San José, Costa Rica.

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

Rückert, P., Wohlfromm, L., & Tracht, K. (2018). Implementation of virtual reality systems for simulation of human-robot collaboration. *Procedia Manufacturing*, 164-170. Retrieved April 1, 2018

Siciliano, B., & Khatib, O. (2016). *Springer Handbook of Robotics*. Berlín: Springer.

VDI-2206. (2014). Design methodology for mechatronic systems. Düsseldorf, Alemania.

Yamauchi, B. (2004). PackBot: a versatile platform for military robotics. *Defense and Security*. Orlando.