



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE MECÁNICA
CARRERA INGENIERÍA INDUSTRIAL

**“IMPLEMENTACIÓN DE UN SISTEMA PARA EL CONTROL DE
MOVIMIENTO REMOTO DEL BRAZO ROBÓTICO ANNO RV624
BASADO EN SEÑALES ELECTROMIOGRÁFICAS”**

Trabajo de Integración Curricular

Tipo: Propuesta Tecnológica

Presentado para optar al grado académico de:

INGENIERA/O INDUSTRIAL

AUTORES:

ADRIANA MARIBEL BAUTISTA GUANIN

ERMENSON DAVID OÑA CHANALUISA

Riobamba – Ecuador

2023



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE MECÁNICA
CARRERA INGENIERÍA INDUSTRIAL

**“IMPLEMENTACIÓN DE UN SISTEMA PARA EL CONTROL DE
MOVIMIENTO REMOTO DEL BRAZO ROBÓTICO ANNO RV624
BASADO EN SEÑALES ELECTROMIOGRÁFICAS”**

Trabajo de Integración Curricular

Tipo: Propuesta Tecnológica

Presentado para optar al grado académico de:

INGENIERA/O INDUSTRIAL

AUTORES: ADRIANA MARIBEL BAUTISTA GUANIN

ERMENSON DAVID OÑA CHANALUISA

DIRECTOR: ING. EDUARDO FRANCISCO GARCÍA CABEZAS

Riobamba – Ecuador

2023

© 2023, Adriana Maribel Bautista Guanin & Ermenson David Oña Chanaluisa

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

Nosotros, Adriana Maribel Bautista Guanin y Ermenson David Oña Chanaluisa, declaramos que el presente Trabajo de Integración Curricular es de nuestra autoría y los resultados de este son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autores asumimos la responsabilidad legal y académica de los contenidos de este Trabajo de Integración Curricular; el patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 29 de noviembre del 2023



Adriana Maribel Bautista Guanin
C.I: 185046861-0



Ermenson David Oña Chanaluisa
C.I: 210067830-5

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE MECÁNICA
CARRERA INGENIERÍA INDUSTRIAL

El Tribunal del Trabajo de Integración Curricular certifica que: El Trabajo de Integración Curricular; Tipo: Propuesta Tecnológica, “**IMPLEMENTACIÓN DE UN SISTEMA PARA EL CONTROL DE MOVIMIENTO REMOTO DEL BRAZO ROBÓTICO ANNO RV624 BASADO EN SEÑALES ELECTROMIOGRÁFICAS**”, realizado por la señorita: **ADRIANA MARIBEL BAUTISTA GUANIN** y el señor: **ERMENSON DAVID OÑA CHANALUISA**, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Integración Curricular, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal Autoriza su presentación.

FIRMA

FECHA

Dr. Miguel Ángel Pérez Bayas
PRESIDENTE DEL TRIBUNAL



2023-11-29

Ing. Eduardo Francisco García Cabezas
**DIRECTOR DEL TRABAJO DE
INTEGRACIÓN CURRICULAR**



2023-11-29

Ing. Bryan Guillermo Guananga Rodríguez
**ASESOR DEL TRABAJO DE
INTEGRACIÓN CURRICULAR**



2023-11-29

DEDICATORIA

El presente trabajo de titulación está dedicado en primer lugar a Dios por darme la fe y la fuerza necesaria para alcanzar mis metas tanto personales como académicas. A mis padres Juan Manuel Bautista y Rosa Delia Guanin, por el inmenso sacrificio que realizan cada día para brindarme un mejor porvenir; sus palabras y apoyo incondicional son el motivo y el pilar fundamental para lograr mis metas. A mis hermanos, Bety, Soña, Juan y Jhonny, fuente constante de aliento y respaldo, quienes han sido mi sostén en cada una de las etapas de mi vida, por último, a mis sobrinos, que me llenan de alegría cada uno de mis días y se convierten en la inspiración constante para ser mejor persona día con día.

Adriana

Quiero expresar mi profundo agradecimiento a Dios, mi mayor guía y fortaleza. A mi querida madre, Elsa Chanaluisa, y a mi hermano, Elder Chicaiza cuyo amor, apoyo incondicional y sacrificios han sido mi mayor motor. Gracias por ser mi ejemplo de perseverancia y dedicación. Su presencia ha hecho que este viaje sea más significativo y llevadero. A mi padre Luis Oña por sus enseñanzas, sabiduría y aliento constante. Su orientación ha sido esencial para mi desarrollo académico y personal. A mi querida amada Mayerli Lalangui por su paciencia, comprensión y apoyo inquebrantable. Tu presencia ha sido mi refugio en momentos de desafío y celebración.

David

AGRADECIMIENTO

Con sincero agradecimiento, deseo expresar mi reconocimiento a mi tutor el Ing. Eduardo García que con su ayuda, orientación y colaboración supo guiarme para el desarrollo y culminación del presente Trabajo de Integración Curricular. Agradezco de manera especial a mis amigos David, Jonathan y Genesis con quienes he compartido momentos agradables y que de una u otra manera siempre los llevaré en mi corazón. A mis mejores amigas Dayana y Jackeline, que, a pesar de la distancia, les agradezco por sus palabras motivadoras, su ejemplo ha sido un recordatorio constante del verdadero significado de una amistad.

Adriana

Quiero expresar mi sincero agradecimiento a mis queridos familiares, cuyo apoyo inquebrantable ha sido mi mayor fortaleza a lo largo de este arduo camino académico. Agradezco a mis amigos, quienes compartieron risas, desafíos y momentos inolvidables durante esta travesía, brindándome un apoyo invaluable. También, extendiendo mi gratitud a mis respetados docentes, cuya guía, conocimientos y dedicación han sido fundamentales en mi formación. Este logro no habría sido posible sin el respaldo de cada uno de ustedes.

David

ÍNDICE DE CONTENIDO

ÍNDICE DE TABLAS	xii
ÍNDICE DE ILUSTRACIONES.....	xiv
ÍNDICE DE ANEXOS	xvii
RESUMEN.....	xviii
SUMMARY	xix
INTRODUCCIÓN	1

CAPÍTULO I

1. DIAGNÓSTICO DEL PROBLEMA	2
1.1. Antecedentes.....	2
1.2. Planteamiento del problema	4
1.3. Justificación.....	5
1.4. Objetivos.....	6
1.4.1. <i>Objetivo General</i>	6
1.4.2. <i>Objetivos Específicos</i>	6
1.5. Alcance.....	6

CAPÍTULO II

2. MARCO TEÓRICO	8
2.1. Estado del arte.....	8
2.2. Robótica	10
2.2.1. <i>Impacto de la Robótica Industrial</i>	10
2.3. Robot.....	11
2.3.1. <i>Brazo Robótico</i>	11
2.3.2. <i>Grados de libertad</i>	11
2.3.3. <i>Componentes de un robot Industrial</i>	11

2.3.4.	<i>Tipos de brazos robóticos</i>	12
2.4.	Robot industrial	14
2.5.	Programación de un Robot Industrial	14
2.6.	Cinemática del robot	15
2.6.1.	<i>Cinemática directa</i>	16
2.7.	Señales Electromiográficas	18
2.7.1.	<i>Tipos de señales EMG</i>	18
2.7.2.	<i>Adquisición de señales EMGS</i>	19
2.7.2.1.	<i>Tipos de sensores para la adquisición de señales EMG</i>	20
2.7.3.	<i>Posicionamiento de electrodos</i>	22
2.7.4.	<i>Aplicaciones de las señales EMG</i>	23
2.7.4.1.	<i>Control de interfaces Hombre-Máquina</i>	23
2.7.4.2.	<i>Control de Exoesqueleto mediante señales electromiográficas</i>	24
2.7.4.3.	<i>Acondicionamiento de señales electromiográficas para el control de prótesis</i>	24
2.7.4.4.	<i>Control sistemas de movimiento para personas con discapacidades físicas</i>	24
2.8.	Comunicación inalámbrica	24
2.8.1.	<i>Ventajas de la Comunicación Inalámbrica en Brazos Robóticos</i>	25
2.8.2.	<i>Características de la Comunicación Inalámbrica en Brazos Robóticos</i>	26
2.8.3.	<i>Aplicaciones de la Comunicación Inalámbrica en Brazos Robóticos</i>	26

CAPÍTULO III

3.	MARCO METODOLÓGICO	28
3.1.	Enfoque de la investigación	28
3.2.	Nivel de investigación	28
3.3.	Tipo de investigación	28
3.3.1.	<i>Investigación documental</i>	28
3.3.2.	<i>Investigación Descriptiva</i>	29
3.3.3.	<i>Investigación de Campo</i>	29
3.3.4.	<i>Investigación experimental</i>	29

3.4.	Metodología para el desarrollo de la propuesta tecnológica	29
3.5.	Diagnóstico de la situación actual	31
3.5.1.	<i>Análisis situación actual Hardware</i>	32
3.5.2.	<i>Análisis situación actual Software</i>	33
3.5.2.1.	<i>Instalación Driver</i>	33
3.5.2.2.	<i>Interfaz gráfica para robot ANNO RV624</i>	34
3.5.2.3.	<i>Utilización de software RobotAnnoV1.0</i>	36
3.5.2.4.	<i>Robot Anno Simulator</i>	37
3.6.	Consolidación del sistema	38
3.7.	Arquitectura del sistema	39
3.7.1.	<i>Etapa 1 – Adquisición de señales electromiográficas EMG</i>	39
3.7.2.	<i>Etapa 2 – Sistema de Control</i>	40
3.7.3.	<i>Etapa 3 – Movimiento del brazo robótico</i>	40
3.8.	Definición de requerimientos	40
3.9.	Hardware	41
3.9.1.	<i>Brazo robótico ANNO RV624</i>	41
3.9.1.1.	<i>Servo motor integrado 57AIM30</i>	43
3.9.1.2.	<i>Servo motor integrado 60AIM25</i>	44
3.9.1.3.	<i>Servo motor integrado 42AIM30</i>	45
3.9.1.4.	<i>Servo motor integrado 42AIM15</i>	46
3.9.2.	<i>Controlador integrado</i>	47
3.9.2.1.	<i>Mini bomba y electroválvulas</i>	48
3.9.3.	<i>Sensor de señales EMG</i>	49
3.9.3.1.	<i>Selección de sensores EMG</i>	49
3.9.3.2.	<i>Brazalete MYO</i>	51
3.9.4.	<i>Computadora portátil</i>	52
3.9.5.	<i>Dispositivo HM-11</i>	53
3.10.	Cinemática directa del brazo robótico	54
3.11.	Software	61

3.11.1.	<i>Python</i>	61
3.11.2.	<i>Pycharm</i>	63
3.11.2.1.	<i>QT Designer</i>	63
3.11.3.	<i>MYO connect</i>	64
3.12.	Configuración del sistema	67
3.12.1.	Conexiones del Hardware	67
3.12.1.1.	<i>Comunicación alternativa de sensor EMG – Microcontrolador</i>	68
3.12.1.2.	<i>Calibración y conexión Brazaletes MYO con Python</i>	71
3.12.1.3.	<i>Algoritmo de aprendizaje</i>	73
3.12.1.4.	<i>Calibración y conexión Robot</i>	74
3.12.2.	Conexiones del Software	76
3.12.2.1.	<i>Programación Python</i>	76
3.12.2.2.	<i>Desarrollo de interfaz</i>	84

CAPÍTULO IV

4.	PROPUESTA TECNOLÓGICA	86
4.1.	Aplicación informática	86
4.1.1.	<i>Conexiones Iniciales</i>	87
4.1.2.	<i>Interfaz</i>	88
4.2.	Pruebas de operación	91
4.2.1.	<i>Entrenamiento de brazaletes MYO</i>	92
4.2.2.	<i>Pruebas de control</i>	93
4.3.	Análisis de reconocimiento de gesto - Brazaletes principal	97
4.3.1.	<i>Latencia de tiempo de respuesta - Brazaletes principal</i>	98
4.4.	Análisis de reconocimiento de gestos - Brazaletes secundario y Arduino	100
4.4.1.	<i>Latencia del tiempo de respuesta - Brazaletes secundario</i>	101
4.5.	Tiempo de respuesta a diferentes velocidades	102
4.6.	Evaluación de rendimiento del sistema de control	103
4.6.1.	<i>Tiempos de respuesta</i>	104

4.7.	Gestión del proyecto	105
4.7.1.	<i>Cronograma</i>	105
4.7.2.	<i>Costos</i>	105
4.7.3.	<i>Recurso humano</i>	106
4.7.4.	<i>Recurso material</i>	106

CAPÍTULO V

5.	CONCLUSIONES Y RECOMENDACIONES	107
5.1.	Conclusiones	107
5.2.	Recomendaciones	109

BIBLIOGRAFÍA

ANEXOS

ÍNDICE DE TABLAS

Tabla 2-1:	Tipos de brazos robóticos	13
Tabla 2-2:	Aplicaciones de la comunicación inalámbrica	27
Tabla 3-1:	Análisis de funcionamiento inicial de los componentes Hardware	32
Tabla 3-2:	Especificaciones del brazo robótico ANNO RV 624.....	42
Tabla 3-3:	Denominación de los motores paso a paso que contiene el brazo	43
Tabla 3-4:	Especificaciones técnicas del motor 57AIM30	44
Tabla 3-5:	Especificaciones técnicas del motor 60AIM25	45
Tabla 3-6:	Especificaciones técnicas del motor 42AIM30	46
Tabla 3-7:	Especificaciones técnicas del motor 42AIM15	47
Tabla 3-8:	Especificaciones de elementos de controlador del robot.....	47
Tabla 3-9:	Tabla de características de los sensores	50
Tabla 3-10:	Especificaciones del brazalete MYO Armband.....	51
Tabla 3-11:	Indicadores de conexión del brazalete	52
Tabla 3-12:	Especificaciones del equipo.....	53
Tabla 3-13:	Especificaciones de la tarjeta HM-11	54
Tabla 3-14:	Casos para la asignación del origen.....	56
Tabla 3-15:	Casos para la asignación del eje X	57
Tabla 3-16:	Tabla de Denavit-Hartenberg.....	59
Tabla 3-17:	Comparación de leguajes de programación.....	61
Tabla 3-18:	Características de PyCharm.....	63
Tabla 3-19:	Ventajas del Qt Designer	64
Tabla 3-20:	Máximos y mínimos de ángulo de giro	68
Tabla 3-21:	Conexiones entre Arduino y HM-11.....	69
Tabla 3-22:	Conexiones de Arduino	70
Tabla 4-1:	Descripción de gestos codificados para el brazo robótico.....	92
Tabla 4-2:	Coordenadas de la posición final de la junta 1	94
Tabla 4-3:	Coordenadas de la posición final de la junta 2	94

Tabla 4-4: Coordenadas de la posición final de la junta 3	95
Tabla 4-5: Coordenadas de la posición final de la junta 4	96
Tabla 4-6: Coordenadas de la posición final de la junta 5	96
Tabla 4-7: Coordenadas de la posición final de la junta 6	97
Tabla 4-8: Reconocimiento de gesto del brazaletes principal	97
Tabla 4-9: Tiempo de respuesta del brazaletes principal	98
Tabla 4-10: Reconocimiento de gestos de brazaletes secundario	100
Tabla 4-11: Tiempo de reconocimiento de brazaletes secundario	101
Tabla 4-12: Tiempo de respuesta de eslabones acorde a modificación de velocidad	102
Tabla 4-13: Reconocimiento de gestos	104
Tabla 4-14: Tiempo de respuesta	104
Tabla 4-15: Cronograma de las actividades	105
Tabla 4-16: Costos totales para la implementación del proyecto	106

ÍNDICE DE ILUSTRACIONES

Ilustración 2-1:	Sistema de un Robot Industrial	12
Ilustración 2-2:	Componentes del brazo robótico.....	14
Ilustración 2-3:	Cinemática directa e inversa	16
Ilustración 2-4:	Señales electromiográficas.....	18
Ilustración 2-5:	Myoware AT- 04-001.....	20
Ilustración 2-6:	Myo Armband	21
Ilustración 2-7:	Sensor ECG /EMG 3,3v AD8832	22
Ilustración 2-8:	Posición de electrodos.....	23
Ilustración 2-9:	Tecnología de comunicación inalámbrica	25
Ilustración 3-1:	Diagrama de flujo de la metodología	30
Ilustración 3-2:	Prerrequisitos de instalación.....	33
Ilustración 3-3:	Seleccionar Prolific USB-to serial comm Port (COM19).....	34
Ilustración 3-4:	Instalación del software ANNOSTM32	34
Ilustración 3-5:	Interfaz de trabajo Robot Anno	35
Ilustración 3-6:	Interfaz de trabajo RobotAnnoV1.0	35
Ilustración 3-7:	Ventana de trabajo para controlar el brazo robótico.....	36
Ilustración 3-8:	Ventana de control del brazo robótico.....	37
Ilustración 3-9:	Interfaz gráfica ANNO-VRSimulator.....	37
Ilustración 3-10:	Diseño del sistema.....	38
Ilustración 3-11:	Arquitectura del sistema.....	39
Ilustración 3-12:	Robot ANNO RV624.....	41
Ilustración 3-13:	Servo motor integrado 57AIM30	43
Ilustración 3-14:	Servo motor integrado 60AIM25	44
Ilustración 3-15:	Servo motor integrado 42AIM30	45
Ilustración 3-16:	Servo motor integrado 42AIM15	46
Ilustración 3-17:	Elementos del controlado del robot ANNO RV624	48
Ilustración 3-18:	Electroválvula	49

Ilustración 3-19: Brazaletes MYO Armband.....	51
Ilustración 3-20: Dell Inspiron 15-5576	53
Ilustración 3-21: Módulo Bluetooth HM-11	53
Ilustración 3-22: Brazo ANNO RV624.....	54
Ilustración 3-23: Asignación de eslabones y articulaciones	55
Ilustración 3-24: Asignación del eje <i>Zi</i>	56
Ilustración 3-25: Asignación de los puntos de origen.....	56
Ilustración 3-26: Diagrama completo de Denavit-Hartenberg.....	57
Ilustración 3-27: Diagrama de cuerpo libre del brazo robótico	58
Ilustración 3-28: Conexión entre el brazaletes y el dispositivo.....	65
Ilustración 3-29: Opciones de selección de trabajo	65
Ilustración 3-30: Proceso de conexión MYO a Bluetooth	66
Ilustración 3-31: Crear un nuevo dispositivo.....	66
Ilustración 3-32: Emparejar Bluetooth	67
Ilustración 3-33: Pantalla de fin de carga del firmware para el Bluetooth HM-11	69
Ilustración 3-34: Conexión entre Myo y ordenador.....	69
Ilustración 3-35: Conexiones del Arduino	70
Ilustración 3-36: Envío de datos Arduino-MYO	71
Ilustración 3-37: Bibliotecas de Python.....	71
Ilustración 3-38: Librería PYOMYO.....	72
Ilustración 3-39: Código para la conexión del Brazaletes y Python	72
Ilustración 3-40: Medición de los 8 sensores EMG.....	73
Ilustración 3-41: Algoritmo de aprendizaje Classifier.....	73
Ilustración 3-42: Datos generados por los sensores.....	74
Ilustración 3-43: Recopilación y almacenamiento de datos	74
Ilustración 3-44: Comandos del brazo robótico.....	75
Ilustración 3-45: Algoritmo de conexión.....	75
Ilustración 3-46: Diagrama de flujo – Interfaz de usuario.....	76
Ilustración 3-47: Diagrama de flujo del proceso inicial.....	77

Ilustración 3-48:	Diagrama de flujo del menú principal.....	78
Ilustración 3-49:	Diagrama de flujo de la calibración	79
Ilustración 3-50:	Diagrama de almacenamiento de gestos.	80
Ilustración 3-51:	Proceso de control manual para el robot	81
Ilustración 3-52:	Proceso de control automático del robot	82
Ilustración 3-53:	Sincronización de la operación del robot por control automático	83
Ilustración 3-54:	Desarrollo de interfaz en QT Designer.....	85
Ilustración 4-1:	Sistema de control del brazo robótico ANNO RV624.....	86
Ilustración 4-2:	Conexión inicial con el robot	87
Ilustración 4-3:	Elección del puerto para la conexión del brazo robótico	88
Ilustración 4-4:	Selección de las opciones de control.....	89
Ilustración 4-5:	Entrenamiento de gestos para cada grado de libertad	89
Ilustración 4-6:	Interfaz de control manual del brazo.....	90
Ilustración 4-7:	Interfaz para el control automático.....	91
Ilustración 4-8:	Tipos de archivos de Python	91
Ilustración 4-9:	Posición inicial de robot.....	93
Ilustración 4-10:	Movimiento de la junta 1	94
Ilustración 4-11:	Movimiento de la junta 2	94
Ilustración 4-12:	Movimiento de la junta 3	95
Ilustración 4-13:	Movimiento de la junta 4	95
Ilustración 4-14:	Movimiento de la junta 5	96
Ilustración 4-15:	Movimiento de la junta 6	96
Ilustración 4-16:	Control del gripper	97
Ilustración 4-17:	Eficacia-Brazaletes secundario	99
Ilustración 4-18:	Eficacia-Brazaletes secundario	101
Ilustración 4-19:	Tiempo respuesta a distinta velocidad.....	102
Ilustración 4-20:	Tiempo respuesta a una velocidad del 75%	103
Ilustración 4-21:	Evaluación de rendimiento.....	104

ÍNDICE DE ANEXOS

- ANEXO A:** TIEMPO DE RECONOCIMIENTO DE BRAZALETE SECUNDARIO
- ANEXO B:** TIEMPOS DE LATENCIA PARA UNA VELOCIDAD DEL 5%
- ANEXO C:** GRÁFICAS DE TIEMPO DE RESPUESTA A UNA VELOCIDAD DEL 5%
- ANEXO D:** TIEMPOS DE LATENCIA PARA UNA VELOCIDAD DEL 20%
- ANEXO E:** GRÁFICAS DE TIEMPO DE RESPUESTA A UNA VELOCIDAD DEL 20%
- ANEXO F:** TIEMPOS DE LATENCIA PARA UNA VELOCIDAD DEL 50%
- ANEXO G:** GRÁFICAS DE TIEMPO DE RESPUESTA A UNA VELOCIDAD DEL 50%
- ANEXO H:** TIEMPOS DE LATENCIA PARA UNA VELOCIDAD DEL 75%
- ANEXO I:** GRÁFICAS DE TIEMPO DE RESPUESTA A UNA VELOCIDAD DEL 75%
- ANEXO J:** TIEMPOS DE LATENCIA PARA UNA VELOCIDAD DEL 100%
- ANEXO K:** GRÁFICAS DE TIEMPO DE RESPUESTA A UNA VELOCIDAD DEL 100%
- ANEXO L:** RECONOCIMIENTO DE GESTOS
- ANEXO M:** TIEMPO DE RESPUESTA DE CADA ESLABÓN
- ANEXO N:** EXTRACTO DE LA PROGRAMACIÓN EN PYTHON
- ANEXO O:** MATRICES DE DENAVIT-HARTENBERG

RESUMEN

La adopción de tecnologías de automatización industrial, como robots, ha enfrentado desafíos en la interfaz hombre-máquina, especialmente en adaptabilidad y respuesta en tiempo real. En Ecuador, a pesar del crecimiento en la adopción de estas tecnologías, persisten obstáculos críticos en la integración efectiva de la intervención humana, por lo que nace la necesidad de desarrollar un sistema de control humano eficaz y seguro para robots industriales, abordando la seguridad y adaptabilidad en entornos complejos. Por lo tanto, en el presente trabajo de titulación se implementó un sistema para el control remoto del brazo robótico ANNO RV624 basado en señales electromiográficas, buscando mejorar la competitividad empresarial mediante la combinación de eficiencia automatizada con adaptabilidad humana. La metodología abordó la evaluación del estado inicial del brazo, la consolidación y arquitectura del sistema, lo que permitió definir sus requerimientos necesarios para el sistema. Por otra parte, para la configuración de hardware y software, se empleó dos brazaletes MYO encargados de captar las señales EMG y transmitir los datos mediante bluetooth, complementando al brazalete secundario con un módulo de comunicación alternativa conformada por Arduino y un HM-11. El desarrollo del algoritmo de control se llevó a cabo con Python y sus bibliotecas como QYQT5, PYOMYO, SERIAL, junto con Qt Designer para la creación de la interfaz de control. En cuanto a los resultados, las pruebas de funcionalidad demostraron la capacidad del sistema de control del brazo robótico ANNO RV624, registrando tiempos de respuesta con una latencia de 2145 a 2510 ms a una velocidad del 75%, destacando el funcionamiento del gripper con 2145 ms, seguido por el segundo GDL con 2147 ms, y el sexto GDL con la mayor latencia de 2510 ms. La eficacia del sistema, evaluada a través de 1600 muestras, alcanzó un 98,86%.

Palabras clave: <SISTEMA DE CONTROL> <ROBOT INDUSTRIAL> <SEÑALES ELECTROMIOGRÁFICAS> <AUTOMATIZACIÓN INDUSTRIAL> <ADAPTABILIDAD HUMANA>.

2123-DBRA-UPT-2023



SUMMARY

The implementation of industrial automation technologies, such as robots, has faced challenges in the human-machine interface, particularly in terms of adaptability and real-time response. In Ecuador, despite the growth in the implementation of technologies, there are still serious obstacles in the effective integration of human intervention, so there is a need to develop an effective and safe human control system for industrial robots, with safety and adaptability in complex environments. Therefore, in this research a system for remote control of ANNO RV624 robotic arm based on electromyographic signals was implemented in order to improve business competitiveness through the combination of automated efficiency with human adaptability. The methodology of the research included the initial evaluation state of the arm, the consolidation and architecture of the system, which allowed defining the necessary requirements for the system. On the other hand, for the hardware and software configuration, two MYO bracelets were used to capture the EMG signals and transmit the data via Bluetooth, complementing the secondary bracelet with an alternative communication module consisting of Arduino and HM-11. The implementation of the control algorithm was based on Python and its software libraries such as QYQTS, PYOMYO, SERIAL, together with Qt Designer for the creation of the control interface. As for the actual results, the performance tests showed the capability of the ANNO RV624 robotic arm control system, reporting response times with latency ranging from 2145 to 2510 ms at 75% speed, with the gripper performance standing out with 2145 ms, followed by the second GDL with 2147 ms, and the sixth GDL with the highest latency of 2510 ms. The efficiency of the system, evaluated across 1600 samples, reached 98.86%.

Keywords: <CONTROL SYSTEM> <INDUSTRIAL ROBOT> <ELECTROMIOGRAPHIC SIGNALS> <INDUSTRIAL AUTOMATION> <HUMAN ADAPTABILITY>.



Mgs. Mónica Paulina Castillo Niama.
C.I. 060311780-5

INTRODUCCIÓN

La robótica industrial constituye una disciplina de la ingeniería que se enfoca en el diseño, desarrollo y fabricación de robots diseñados específicamente para automatizar procesos en entornos industriales. Estos robots desempeñan funciones en cadenas de producción, llevando a cabo tareas de manera continua y eficiente, sin interrupciones, con el propósito fundamental de elevar la productividad, optimizar la gestión de recursos y minimizar errores en los procesos de fabricación. En la actualidad, la robótica industrial no se limita únicamente a la mejora de la productividad; también exhibe una versatilidad que abarca diversos sectores industriales. (Digital 360, 2023)

La robótica industrial se enfrenta a desafíos cruciales en términos de seguridad y adaptabilidad. La necesidad de ejecutar tareas con extrema exactitud, garantizar la seguridad en la interacción humano-robot, y adaptarse de manera ágil a cambios en entornos de fabricación son aspectos clave. Superar estos desafíos es esencial para aprovechar plenamente el potencial de la robótica industrial en la mejora continua de los procesos productivos.

Una alternativa innovadora para abordar estos desafíos es la implementación de un sistema de control de movimiento remoto a distancia. En este contexto el presente trabajo de titulación tiene como fin la implementación de un sistema para el control de movimiento remoto del brazo robótico ANNO RV624 basado en señales electromiográficas, este enfoque permite una interacción más intuitiva y precisa con el robot, la precisión se mejora significativamente al traducir las señales EMG en comandos de movimiento lo que facilita la ejecución precisa de tareas delicadas.

Desde el punto de vista de la seguridad, este sistema ofrece una solución al permitir que los operadores controlen los robots a distancia, reduciendo la exposición directa a entornos potencialmente peligrosos. En términos de adaptabilidad, el control remoto basado en señales EMG permite una configuración rápida y cambios dinámicos en las tareas asignadas al robot. Los operadores pueden ajustar la ejecución de las funciones del robot según las necesidades cambiantes del entorno de trabajo, lo que mejora la flexibilidad y la eficacia operativa.

Por otra parte, se diseñó una estructura de soporte para la fijación del brazo y el controlador, también se desarrolló un módulo para habilitar la conectividad inalámbrica con el brazaletes MYO, utilizando Arduino y el módulo HM-11. La integralidad de este sistema está bajo el control de Python, que a través de una programación específica presenta una interfaz gráfica para el control del ROBOT ANNO RV624, fundamentada en señales electromiográficas.

CAPÍTULO I

1. DIAGNÓSTICO DEL PROBLEMA

1.1. Antecedentes

La Escuela Superior Politécnica de Chimborazo ha establecido un laboratorio de vanguardia en automatización y control de procesos industriales y de manufactura, con el propósito de brindar a sus estudiantes las herramientas necesarias para desarrollar habilidades clave en áreas como mecánica, electrónica, robótica, automatización y programación. En el ámbito de la automatización industrial, estos laboratorios son fundamentales, ya que cuentan con sistemas de fabricación que permiten la simulación y modelado de procesos relacionados con líneas de producción.

Dentro de este entorno, destaca la presencia de un robot ANNO RV624, lo que plantea la oportunidad de implementar un sistema de control de movimiento remoto del brazo robótico utilizando señales electromiográficas (EMG). La automatización industrial se beneficia enormemente de los laboratorios, y esta investigación aborda un enfoque innovador al introducir la interacción entre el ser humano y la tecnología robótica a través de señales bioeléctricas.

De acuerdo con el artículo realizado por (Aguirre et al., 2015: pp.8-13) titulado “*Control de Señales EMG para el Movimiento de un Brazo Robótico de Tres Grados de Libertad*” presentó un sistema diseñado para capturar señales electromiográficas (EMG) a través de cinco canales. Estas señales son generadas por los músculos y se obtienen mediante electrodos adheridos a la piel. Una vez adquiridas, estas señales son amplificadas, filtradas y convertidas en datos digitales para su posterior análisis en una computadora. Se utilizó un enfoque de control difuso que se basó en 23 reglas con valores lingüísticos, estas señales electromiográficas (EMG) son utilizadas para emular el acto de doblar el brazo, así como para doblar y separar el hombro en un brazo robótico que posee tres grados de libertad. Los hallazgos de este análisis demostraron que es factible lograr una precisión de más del 90% en la recreación de estos movimientos bajo este enfoque.

En el trabajo de (Asnalema, 2017) titulado “*Desarrollo de un sistema de visión artificial a través de una tarjeta de desarrollo para controlar el brazo robótico Mitsubishi*”, se menciona que el brazo robótico tiene la capacidad de mimetizar los movimientos del brazo humano, lo cual es especialmente valioso en situaciones con un alto riesgo para la seguridad de las personas. Por lo tanto, se desarrolló un sistema de visión artificial que puede identificar y reproducir los movimientos del brazo humano en un brazo robótico de Mitsubishi.

Este proceso implicó la creación de un ambiente de detección en un cuarto oscuro, donde se identificaron dos puntos clave para establecer la posición del brazo humano. La implementación del software de visión artificial se llevó a cabo utilizando una placa de desarrollo Raspberry Pi 3. Se realizaron pruebas en terreno para validar la efectividad del controlador y analizar los resultados. Al comparar los ángulos medidos en el brazo humano en el cuarto oscuro con los ángulos generados por el robot, se reveló un margen de error del 7.96% en la reproducción de movimientos. Además, se evaluó el tiempo promedio que requería el robot Mitsubishi para comenzar a replicar los ángulos, y se encontró un retraso de 4.23 segundos

En el trabajo de titulación *“Almacén inteligente mediante robot omnidireccional y visión artificial”* realizado por (Calvopiña, 2018), señala que la manipulación de sustancias químicas conlleva potenciales riesgos para los seres humanos. Es fundamental poseer un conocimiento preciso de los procedimientos adecuados para su manejo, a fin de prevenir la ocurrencia de incidentes provocados por estos peligros. Se implementó un sistema que se encarga de almacenar y distribuir productos mediante el uso de un robot omnidireccional, dado que el robot es una máquina, no está sujeto a las repercusiones que este tipo de tareas tendrían en un ser humano. A la vez, se empleó una tarjeta de desarrollo Raspberry Pi 3 para llevar a cabo la captura, detección y decodificación de códigos QR. Los resultados obtenidos de esta investigación mostraron que la efectividad del proceso de almacenamiento y distribución de productos alcanza el 83%. No obstante, se identificó un margen de error del 17%.

Según (Vera, 2021) en su trabajo titulado *“Desarrollo de un sistema de visión artificial asociado con el robot ANNO RV624 para reconocimiento y clasificación de piezas”* desarrolló un sistema de visión artificial asociado al robot ANNO RV624. Este procedimiento fue llevado a cabo mediante la identificación de objetos utilizando una cámara web, para ello, la imagen capturada fue procesada mediante un algoritmo de visión artificial, lo que permitió el reconocimiento y la ubicación de las piezas. Luego, las coordenadas resultantes se transmitieron al controlador y al brazo robótico, lo que permitió que el robot organizara las piezas de manera sistemática y organizada. Además, se concluyó que el software creado presenta un retraso temporal promedio de 1.08 segundos al llevar a cabo secuencias de movimientos. En cuanto a la ejecución completa del sistema, se observó un tiempo aproximado de 1.85 segundos. Estos lapsos se deben tanto a la velocidad de la conexión serial como a las características del hardware utilizado.

En el trabajo titulado *“Control del movimiento remoto del brazo robótico Mitsubishi usando el sistema operativo Ros a través de la Web”* realizado por (Solano, 2023: pp.51-64), menciona que los entornos industriales enfrentan desafíos constantes que pueden resolverse a través de un análisis profundo del caso de estudio, por lo tanto, resulta esencial crear simulaciones que propongan los

procedimientos inherentes a los sectores industriales. En este contexto, se ha desarrollado un entorno virtual para el brazo robótico Mitsubishi RV-2AJ en la plataforma Linux, empleando el Sistema Operativo Robótico (ROS). Esta plataforma facilita la interacción con el modelo robótico mediante herramientas como MoveIt y RViz, permitiendo la visualización y el desarrollo de movimientos y rutas en el modelo del robot Mitsubishi RV-2AJ. Esta iniciativa da lugar a un sistema de control remoto.

1.2. Planteamiento del problema

Desde una perspectiva empresarial enfocada en la búsqueda constante de eficacia y seguridad, la automatización industrial se ha considerado como una solución esencial, llevando a las empresas a adoptar tecnologías avanzadas para controlar y optimizar sus procesos. Entre estas tecnologías, los robots industriales se destacan como herramientas vitales para realizar tareas especializadas que van más allá de la capacidad humana. Sin embargo, a pesar de los avances en robótica, persisten desafíos significativos en la interfaz hombre-máquina, especialmente en términos de adaptabilidad y respuesta en tiempo real.

En el contexto de Ecuador, la adopción de estas tecnologías ha experimentado un crecimiento considerable, aunque las empresas están adoptando medidas para automatizar sus operaciones, aún enfrentan desafíos críticos en la integración efectiva de la intervención humana. Esto es especialmente relevante en situaciones que requieren toma de decisiones y adaptabilidad en tiempo real, un aspecto aún no completamente resuelto por las soluciones automatizadas actuales.

En este escenario, surge un problema central: la necesidad de desarrollar un sistema de control humano eficaz y seguro para robots industriales. Uno de los desafíos más destacados y prometedores es la implementación de un control preciso de robots mediante señales electromiográficas generadas por la actividad muscular del operario. Aunque estas señales ofrecen la promesa de una interfaz más intuitiva y adaptable, su aplicación efectiva plantea interrogantes de seguridad y adaptabilidad, especialmente en entornos industriales complejos y cambiantes.

Por lo tanto, esta tesis se propone investigar y desarrollar un sistema para el control de un brazo robótico mediante el uso de señales electromiográficas, superando las limitaciones actuales en la interfaz hombre-máquina, permitiendo un control más seguro, preciso y adaptable del brazo robótico en entornos industriales. Este avance podría significar un paso crítico para mejorar la competitividad de las empresas en Ecuador, al combinar eficiencia automatizada con la adaptabilidad humana.

1.3. Justificación

La automatización industrial ha revolucionado los procesos productivos, brindando eficacia, seguridad y consistencia en las operaciones de las empresas. Sin embargo, la naturaleza dinámica y a menudo impredecible de los entornos industriales presenta desafíos que no siempre pueden ser abordados eficientemente por sistemas puramente automatizados. Situaciones inesperadas, cambios en las condiciones de trabajo o problemas atípicos requieren una intervención humana ágil y precisa para garantizar la continuidad y seguridad del proceso.

En este contexto, la integración de tecnologías biomédicas y robóticas emerge como una solución prometedora. El control de brazos robóticos mediante señales electromiográficas representa una innovación en este ámbito. Estas señales, derivadas de la actividad muscular del operador, ofrecen una interfaz intuitiva que permite al usuario interactuar y controlar el robot de manera más natural y directa, combinando la precisión de la maquinaria con la adaptabilidad humana.

La implementación de este sistema no solo potencia la eficiencia del proceso al permitir ajustes en tiempo real, sino que también eleva los estándares de seguridad. Al permitir que un operador humano intervenga y controle el brazo robótico de manera remota, se pueden anticipar y evitar posibles accidentes o daños a los equipos. Además, esta integración aprovecha la destreza, experiencia y juicio del operador para mejorar la eficiencia y calidad del proceso industrial.

Este estudio representa un significativo aporte investigativo tanto para Ecuador en general como para la ESPOCH en particular. Al proporcionar un entendimiento detallado del robot ANNO RV624 y su viabilidad de control a través de señales electromiográficas, se sientan las bases para futuras investigaciones y aplicaciones en el campo. Así también al desarrollar una aplicación informática con herramientas Open Source, no solo beneficia a la comunidad de software libre, sino que también brinda soluciones accesibles para investigadores e instituciones con recursos limitados.

La importancia de este estudio no se limita únicamente a su valiosa contribución técnica, sino que también radica en el impacto significativo que podría tener en la industria ecuatoriana. Al evidenciar la eficacia del sistema propuesto y sus tiempos de respuesta óptimos, se abre la puerta para futuras adaptaciones o implementaciones en el sector industrial local.

1.4. Objetivos

1.4.1. Objetivo General

Implementar un sistema para el control de movimiento remoto de un brazo robótico ANNO RV624 basado en señales electromiográficas.

1.4.2. Objetivos Específicos

- Determinar el principio de funcionamiento de los sistemas del robot ANNO RV624.
- Seleccionar el hardware adecuado para la adquisición inalámbrica y procesamiento de las señales electromiográficas.
- Desarrollar una aplicación informática para la calibración del sistema de señales electromiográficas empleando una herramienta Open Source.
- Vincular el sistema de control basado en señales electromiográficas al sistema de control del brazo robótico ANNO RV624.
- Realizar pruebas de funcionalidad del sistema para verificación de eficacia y tiempos de respuesta.

1.5. Alcance

La propuesta tecnológica plantea la implementación de un sistema para el control de movimiento remoto del brazo robótico ANNO RV624 basado en señales electromiográficas, el trabajo especifica acciones para la adquisición y procesamiento de señales EMG.

Seleccionar el hardware y software necesario para capturar y procesar las señales electromiográficas generadas por los músculos del brazo y su vinculación a un algoritmo de control para el manejo del brazo robótico ANNO RV624.

Mediante el desarrollo de algoritmos de procesamiento de señales, se analizará y extraerá información relevante de las señales electromiográficas. Esto puede incluir la detección de patrones de activación muscular, la identificación de movimientos específicos y la clasificación de diferentes estados musculares.

Se buscará establecer la comunicación entre el sistema de adquisición de señales EMG y el brazo robótico, permitiendo la transmisión de comandos de movimiento basados en las señales EMG capturadas.

Será necesario codificar algoritmos de control que traduzcan las señales EMG en comandos de movimiento precisos para el brazo robótico. Esto implica establecer relaciones entre las señales EMG y los movimientos deseados, así como también ajustar los parámetros de control para lograr una respuesta adecuada.

Finalmente, se llevarán a cabo pruebas y experimentos para evaluar la capacidad de respuesta y la eficacia del sistema de control de movimiento remoto. Esto puede involucrar la comparación de los movimientos realizados por el brazo robótico con los movimientos deseados y la evaluación del rendimiento en diferentes condiciones.

CAPÍTULO II

2. MARCO TEÓRICO

2.1. Estado del arte

Acorde al trabajo de Cedeño (2017) titulada "*Diseño de un sistema de control de movimientos de agarre para una prótesis de mano robótica con brazalete Myo*", se ha creado un sistema que posibilita el control de movimientos esenciales de agarre en una prótesis de mano asequible. El propósito de este sistema es facilitar a un joven, debido a una malformación congénita, carece de dedos en su mano derecha, para llevar a cabo dichos movimientos. Se tuvieron en cuenta los movimientos completos de su muñeca. Para iniciar, se desarrolló un primer modelo de prótesis de mano que emplea el brazalete Myo. Este brazalete incluye ocho sensores mioeléctricos no invasivos ubicados en el antebrazo. Además, el brazalete Myo está equipado con un giroscopio y un acelerómetro, lo que posibilita la captura de datos en los tres ejes (x, y, z) y la transmisión de información de posición. Estas características abren un amplio abanico de posibilidades para desarrolladores en todo el mundo (Cedeño, 2017).

L La investigación titulada "*Implementación de un sistema de clasificación de gestos de manos humanas usando MYO Armband para el control remoto de un brazo robótico 3GDL*" analiza la implementación de un sistema para clasificar gestos de manos humanas usando señales de electromiografía (EMG) obtenidas con un dispositivo MYO Armband comercial. Este sistema se utiliza para controlar de forma remota un brazo robótico de 3 grados de libertad. Para la clasificación de gestos se desarrolló una interfaz gráfica en Matlab para obtener las señales EMG correspondientes a cada gesto que se clasifica y se repitió cinco veces durante el entrenamiento. La clasificación se basó en la obtención de la distancia DTW (Dynamic Time Warping) entre la señal EMG del gesto desconocido y todas las señales EMG del conjunto de entrenamiento. El algoritmo kNN (k Nearest Neighbors) se utiliza para la clasificación de movimiento para validar el sistema y evaluar su velocidad y tiempo de respuesta, se realizaron pruebas en 10 voluntarios. A cada voluntario se le pidió que repitiera cada gesto 30 veces. Estas pruebas lograron una tasa de clasificación del 86 % y un tiempo de respuesta de menos de 250 ms (Zea, 2017: pp.16-76).

En la tesis "*Desarrollo de una arquitectura software para teleoperación y monitoreo del robot ANNO RV624*" se enfoca en el desarrollo de un software que permita el control y monitoreo remoto del robot ANNO RV624. Al analizar el software original del brazo robótico, se

encontraron problemas en la comunicación con la mano, lo que llevó a la necesidad de una interfaz que se comunique con diferentes sistemas operativos. Además, existía la necesidad de un software que pudiera formar una interfaz gráfica de usuario con un soporte de datos externo y una plataforma que sirviera como soporte para el brazo.

En cuanto al hardware, se utilizaron los mismos elementos que estaban disponibles en el brazo robótico. Se utilizó Visual Basic para implementar la interfaz de usuario y descargar los algoritmos. Este software se encarga de enviar señales e instrucciones al controlador y establecer una conexión entre la plataforma Ubidots IoT y el controlador. El controlador, a su vez, procesa los datos recibidos y los transmite a los controladores de los accionamientos eléctricos. Luego de la realización del prototipo de interfaz de usuario, se logró el 100% de funcionalidad de los datos transmitidos relacionados con los movimientos de cada actuador. En cuanto a la operación y el monitoreo remotos, transcurrieron entre 8,37 y 2,62 segundos desde la activación hasta la llegada de los datos (Mera y Quicaliquin, 2021: pp.25-75).

En el estudio denominado: *"Implementación de un sistema inclusivo basado en el uso del brazo robótico ANNO RV624 para la alimentación de personas con problemas de movilidad"*, se orientó en brindar una solución a un conjunto de personas consideradas vulnerables aplicando los conocimientos obtenidos en la formación académica. El estudio desarrolló un algoritmo que detecta y reconoce el área facial de una persona que está siendo alimentada, especialmente la boca, usando visión artificial. Para la producción de los datos se recurrió a las técnicas de cámara y visión artificial. El software central fue Python con las bibliotecas OpenCV y MediaPipe Face Mesh, que se enfocan en analizar imágenes reales y traducirlas a un lenguaje binario que la computadora pueda entender para realizar las acciones programadas del algoritmo. En términos de tiempo, se encontró que la velocidad al 20% era subóptima para la función de alimentación, provocando pérdida de comida tomada de la cuchara y vibración de la mesa y desaceleraciones repentinas de las trayectorias. Tales situaciones pueden dañar los servomotores del brazo robótico. Además, se recomienda configurar el brazo robótico ANNO RV624 en la posición "HOME" o cero antes de usarlo, pues por defecto, hará el último movimiento antes de apagar el dispositivo (Pástor, 2022: pp.18-73).

Esta investigación propone el desarrollo de un sistema interactivo que tiene como objetivo establecer comunicación entre humanos y computadoras a través de la clasificación de señales electromiográficas (EMG). Crea un sistema automático para controlar un brazo robótico utilizando Gesture Control Bar (MYO), y técnicas de aprendizaje automático. Para ello, existe la obligación de desarrollar técnicas que permitan la generación de modelos predictivos para un aprendizaje automático preciso. En este contexto, se crea un algoritmo que utiliza técnicas de

aprendizaje automático para obtener un modelo idóneo para la clasificación de las señales EMG y reconocer los gestos asociados. Inicialmente, este modelo se entrena y prueba utilizando bases de datos de señales bioeléctricas almacenadas en archivos. Posteriormente se agrega al proceso una tira de control de gestos MYO, lo que permite utilizar la actividad eléctrica de la mano de una persona para crear una base de datos propietaria. Finalmente, después de reconocer el gesto, se envía la acción de control correspondiente y se muestra en el brazo robótico simulado en el entorno de desarrollo integrado. De esta forma, es posible establecer una comunicación efectiva entre el usuario y el brazo robótico clasificando las señales EMG y utilizando la tira MYO (Posso et al., 2020: pp.1-3).

2.2. Robótica

La robótica es una disciplina científica que combina diversas ramas tecnológicas con el fin de crear máquinas automatizadas que tengan la capacidad de realizar tareas de forma automática, con más rapidez, menos riesgosa y más productiva, basándose en las capacidades de su software. Esta disciplina abarca diferentes áreas, tales como la mecánica, la electrónica, la inteligencia artificial e informática, y busca crear robots capaces de interactuar con su entorno y llevar a cabo diversas actividades, desde tareas industriales y médicas hasta exploración espacial y asistencia en el hogar (Revista de Robots, 2023).

2.2.1. Impacto de la Robótica Industrial

Durante la tercera Revolución Industrial, la automatización se hizo presente en las fábricas a través de máquinas cada vez más sofisticadas e inteligentes. Lo que antes parecía un concepto perteneciente a nuestra imaginación, ahora se ha vuelto fundamental para los procesos industriales de las empresas. En la actualidad, debido a un mercado altamente competitivo y a consumidores cada vez más exigentes, las empresas se enfrentan al desafío de adaptar sus procesos de producción utilizando tecnologías más avanzadas. Esto contribuye a lograr una mayor sostenibilidad y optimización en sus operaciones (Paka, 2019)

La robótica dentro de la sociedad y en las industrias ha ido aumentando día con día. La creciente presencia de robots con mayor autonomía está generando cambios no solo en la forma de trabajar, sino también en la forma de vivir y relacionarse con el entorno. El impacto de la robótica se hace cada vez más evidente en todos los ámbitos profesionales, puesto a que los robots posibilitan un aumento en la productividad, eficiencia y rentabilidad de las empresas, lo que motiva a las empresas industriales a integrar tecnologías en sus procesos de producción (CE.Entel, 2022).

2.3. Robot

Un robot se refiere a una entidad autónoma que consta de componentes de mecánica artificial y un sistema electromecánico. Lo que distingue a un robot está conformado por mecanismos físicos y tangibles (Revista de Robots, 2023).

2.3.1. Brazo Robótico

Un brazo robótico o manipulador se describe como un conjunto de componentes electromecánicos que permiten el desplazamiento de un extremo utilizando tecnología robótica, con funciones que se pueden programar por completo. El brazo robótico es autónomo en su operación y se crea con el objetivo de replicar las habilidades de un brazo humano y la destreza de una mano humana. Los tipos más comunes son los robots articulados, que cuentan con la capacidad de trabajar en diferentes posiciones y ángulos sin fatigarse (Revista de Robots, 2023).

2.3.2. Grados de libertad

Es la capacidad que tiene cada robot para realizar movimientos independientes con respecto a la articulación anterior. La estructura mecánica y el tipo de robot se definen según el tipo de las tres primeras articulaciones, que pueden ser rotacionales, planares, prismáticas, esféricas, cilíndricas o de tornillo. Estas articulaciones proporcionan una configuración específica a los tres ejes principales del robot, los cuales determinan la posición de la herramienta en el espacio, así como el tipo de coordenadas utilizadas para calcular esta posición o localización (Reynoso, 2021: pp.28-29).

2.3.3. Componentes de un robot Industrial

Un robot industrial tiene la capacidad de realizar diversas funciones desde tareas sencillas hasta tareas complejas y peligrosas para el ser humano, para esto deben contar con los siguientes componentes básicos:

- **Estructura mecánica.** La configuración mecánica de un robot industrial se compone principalmente de segmentos articulados, la parte fundamental en la base y la pinza. Esta estructura puede estar construida en acero, aluminio u otros metales resistentes.
- **Actuadores.** - Son los componentes que permiten generar y controlar el movimiento del robot, dentro de los actuadores más importantes en la construcción de un brazo robótico son los motores (eléctricos, hidráulicos o neumáticos), cilindros, juntas, etc.

- **Sensores.** - Los robots industriales pueden estar equipados con diferentes tipos de sensores, estos pueden ser internos o externo. Los sensores suelen ser de proximidad, sensores de fuerza, sensores de visión, entre otros, que tienen como fin detectar y reaccionar en el entorno donde se encuentra.
- **Control a la Computadora.** - Según la tarea a realizar, la computadora controla y coordina las distintas funciones llevadas a cabo por el usuario, y al mismo tiempo, opera como los sistemas de control del robot.
- **Los elementos de la posición.** - El componente clave de los elementos de posición, sin considerar la dirección, tiene la capacidad de desplazarse a diferentes ubicaciones en el espacio.

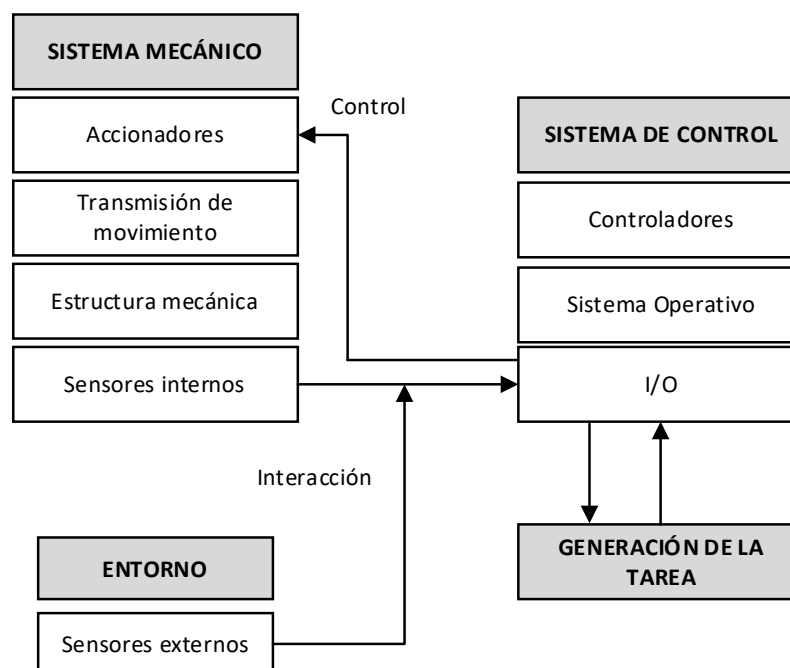


Ilustración 2-1: Sistema de un Robot Industrial

Fuente: (Caparroso, Avilés & Hernández, 1999)

2.3.4. Tipos de brazos robóticos

Los brazos robóticos, como protagonistas fundamentales de la revolución tecnológica y la automatización industrial, han trascendido de las páginas de la ciencia ficción para convertirse en una realidad palpable en diversos campos de la vida moderna. Estos ingeniosos dispositivos mecánicos, diseñados con la capacidad de replicar y, en muchos casos, superar las destrezas humanas en términos de precisión y fuerza, han encontrado aplicaciones en industrias que van desde la manufactura y la medicina hasta la exploración espacial y la investigación científica. La tecnología robótica está en constante evolución, por lo que los detalles específicos pueden

cambiar con el tiempo (Mera y Quicaliquin, 2021: pp.7-10). En la Tabla 2-1, se detalla los principales tipos de brazos robóticos.

Tabla 2-1: Tipos de brazos robóticos

TIPO DE ROBOT	CARACTERÍSTICAS PRINCIPALES	APLICACIONES
Brazo Robótico Industrial	<ul style="list-style-type: none"> •Varios grados de libertad para movimientos precisos •Alta capacidad de carga •Programables para tareas específicas •Utilizados en entornos de fabricación y producción 	<ul style="list-style-type: none"> •Ensamblaje de productos •Soldadura y corte •Manipulación de materiales •Paletización y empaquetado
Robot de Servicio	<ul style="list-style-type: none"> •Diseñados para interactuar y ayudar a las personas •Variedad de formas y tamaños •Realizan tareas como limpieza, acompañamiento y cuidado 	<ul style="list-style-type: none"> •Asistencia a personas con discapacidad •Entrega de medicamentos en entornos hospitalarios •Apoyo en la atención al cliente en tiendas
Brazo antropomórfico o brazo articulado	<ul style="list-style-type: none"> •Imita la anatomía humana con múltiples articulaciones similares a las de un brazo y mano humanos. •Generalmente entre 4 y 7 grados de libertad. •Cada articulación está equipada con actuadores. •Movimientos precisos y flexibilidad. •Se pueden acoplar diversos tipos de herramientas y dispositivos en el extremo del brazo. 	<ul style="list-style-type: none"> •Ensamblaje industrial. •Soldadura y pintura. •Manipulación de materiales. •Cirugía asistida por robots. •Articulaciones.
Brazo Robótico Delta	<ul style="list-style-type: none"> •Movimientos rápidos en espacio 3D •Actuadores lineales para movimiento •Amplio espacio de trabajo 	<ul style="list-style-type: none"> •Ensamblaje de productos con alta cadencia. •Picking y placing en líneas de producción. •Empaquetado y manipulación en la industria alimentaria.
Robot Colaborativo (Cobot)	<ul style="list-style-type: none"> •Diseñados para trabajar junto a humanos •Sensores de seguridad para evitar colisiones •Fácil programación y configuración •Aplicaciones seguras en entornos de trabajo compartido 	<ul style="list-style-type: none"> •Ensamblaje colaborativo •Inspección y prueba de productos •Proceso de laboratorio y análisis •Tareas de carga y descarga
Robot Móvil Autónomo	<ul style="list-style-type: none"> •Equipados con sensores y sistemas de navegación •Capacidad para moverse y navegar de manera autónoma •Pueden llevar a cabo tareas en diversos entornos 	<ul style="list-style-type: none"> •Logística y transporte en almacenes •Vigilancia y seguridad en espacios públicos •Entrega y recogida de productos

Robot Industrial Cartesiano	<ul style="list-style-type: none"> •Movimientos lineales precisos y repetibles •Uso en aplicaciones con trayectorias rectilíneas •Ideal para procesos de mecanizado y grabado 	<ul style="list-style-type: none"> •Fabricación de productos lineales •Inspección y control de calidad en líneas de montaje •Corte y fresado en procesos de manufactura
Brazo Robótico SCARA	<ul style="list-style-type: none"> •Configuración en forma de "L" o "U" •Movimientos rápidos en plano horizontal •Alta precisión en el plano XY 	<ul style="list-style-type: none"> •Ensamblaje de componentes electrónicos •Operaciones de picking y placing •Procesamiento ligero

Fuente: (Mera y Quicaliquin, 2021).

Realizado por: Bautista A., & Oña D., 2023

2.4. Robot industrial

Un robot industrial es una máquina programable diseñada para realizar tareas en entornos industriales de manera autónoma o semiautónoma. Está compuesto por diversas partes y sistemas que trabajan en conjunto para permitir su funcionamiento eficiente y preciso. En la Ilustración 2-2, se muestra los componentes del brazo robótico.

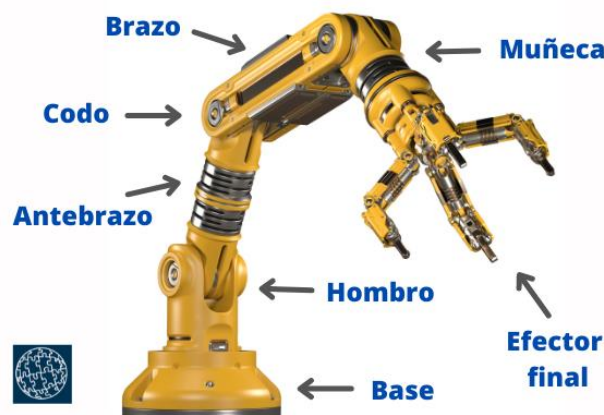


Ilustración 2-2: Componentes del brazo robótico

Fuente: (Kumar, 2010)

2.5. Programación de un Robot Industrial

La programación de un robot industrial se centra en la creación de directrices de control adaptadas a un sistema robótico específico. Estas directrices tienen el propósito de definir la secuencia de pasos a seguir durante una tarea que el robot tiene la intención de ejecutar en un entorno de fabricación o automatización. Estos pasos pueden comprender desde desplazamientos hacia puntos previamente establecidos, hasta la manipulación de objetos, interacción y sincronización con el entorno circundante (Escandón R, 2013).

La programación de un robot industrial implica varios aspectos como:

1. **Definición de acciones:** En primer lugar, se debe determinar qué tarea o conjunto de acciones debe realizar el robot, esto implica comprender las necesidades del proceso industrial y las restricciones del entorno de trabajo.
2. **Creación del programa:** Se desarrolla un programa que consiste en una serie de comandos y movimientos que el robot debe realizar para llevar a cabo la tarea, por lo que puede incluir movimientos de las articulaciones, desplazamiento del efector final, ajuste de velocidad, control de herramientas o sensores, y toma de decisiones en tiempo real si el robot es autónomo. Esta programación puede estar basada en lenguajes de programación tradicionales como C++, Python o Java para crear algoritmos y secuencias de comandos que controlan el comportamiento del robot.
3. **Verificación y simulación:** Antes de que el robot realice la tarea en el entorno de producción, es común verificar y simular el programa en un entorno de prueba para asegurarse de que funcione correctamente y cumpla con los requisitos de seguridad.
4. **Operación y supervisión:** El robot se opera y supervisa durante la ejecución de la tarea, donde los operadores pueden monitorear el rendimiento del robot y tomar decisiones si surgen problemas.

2.6. Cinemática del robot

La cinemática representa un campo de la mecánica que se concentra en la exploración de los aspectos geométricos relacionados con el desplazamiento de objetos, prescindiendo de las razones que lo originan. El estudio cinemático comprende conceptos tales como la localización, alineación, velocidad y aceleración de los diversos elementos dentro de un sistema. En el caso de los robots, el modelo cinemático busca comprender cómo se mueven los eslabones dentro de la estructura robótica, analizando las capacidades de movimiento de cada articulación que controla dichos eslabones.

Este análisis cobra importancia al permitir el estudio integral de los movimientos individuales en cada articulación y su integración en el sistema, considerando que las articulaciones y eslabones deben trabajar de manera conjunta para realizar una tarea específica. Si se cuenta con un modelo cinemático que describe la interacción entre los componentes del sistema, un ingeniero puede diseñar métodos y algoritmos para controlar la operación del robot, controlando cada una de las articulaciones. Esto incluye cumplir con restricciones particulares que cada tarea exige, como evitar posiciones prohibidas, controlar velocidades o aceleraciones máximas, y limitar fuerzas, entre otros aspectos (Pérez et al., 2014: pp.111-112).

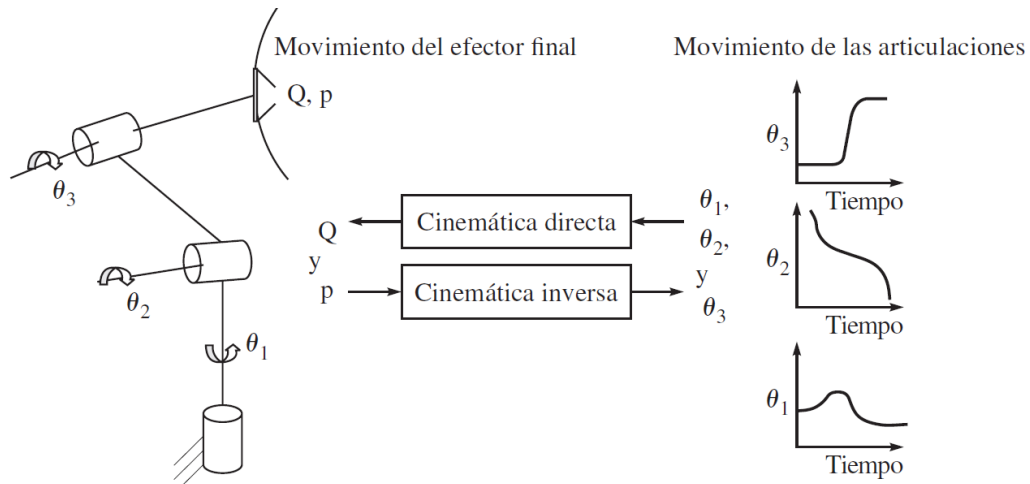


Ilustración 2-3: Cinemática directa e inversa

Fuente: (Kumar, 2010)

2.6.1. Cinemática directa

La solución al desafío de la cinemática directa involucra determinar la ubicación y el alineamiento alcanzados por el extremo del robot, teniendo conocimiento de los valores que establecen la ubicación y el alineamiento de las articulaciones. Estas variables de las articulaciones son captadas de manera directa por los sensores del robot y procesadas por la unidad de control. El modelo de cinemática directa se utiliza para varios fines, entre ellos, proporcionar al usuario información sobre la posición del extremo del robot. (Barrientos et al., 2010).

Dentro de la cinemática directa de posición, se cuentan las ubicaciones de las articulaciones, lo cual incluye los ángulos de las articulaciones rotatorias y los desplazamientos de las articulaciones prismáticas. De esta manera, se han seleccionado las coordenadas cartesianas y los ángulos como la representación de la posición y orientación del extremo de un robot de seis grados de libertad, la solución a la cinemática directa se obtendrá de las correspondientes relaciones establecidas.

$$x = f_x(q_1, q_2, q_3, q_4, q_5, q_6)$$

$$y = f_y(q_1, q_2, q_3, q_4, q_5, q_6)$$

$$z = f_z(q_1, q_2, q_3, q_4, q_5, q_6)$$

$$\alpha = f_\alpha(q_1, q_2, q_3, q_4, q_5, q_6)$$

$$\beta = f_\beta(q_1, q_2, q_3, q_4, q_5, q_6)$$

$$\gamma = f_\gamma(q_1, q_2, q_3, q_4, q_5, q_6)$$

La meta es descubrir la disposición final del efector, lo cual abarca tanto su posición como su alineación. Este resultado se logra utilizando las ecuaciones de cierre, siguiendo los siguientes pasos:

a) Se asigna un sistema de coordenadas individual a cada uno de los eslabones del robot, siendo en total $n + 1$ sistemas. El sistema 1 se vincula al sistema de coordenadas inmóvil, mientras que el sistema $n + 1$ se conecta al efector final o al n -ésimo componente del cuerpo (Kumar, 2010).

b) Se definen los parámetros de Denavit-Hartenberg (DH)

c) Se escriben las matrices de transformación homogénea, como $T_1, T_2 \dots T_n$, donde T_i para $i = 1, \dots, n$ representa la transformación del cuerpo i o sistema $i + 1$ con respecto a su cuerpo anterior $i - 1$, o al sistema al que está adjunto, es decir, el sistema i .

d) La matriz de transformación homogénea del sistema de coordenadas del efector final con respecto al sistema 1, denotada como T , se obtiene multiplicando en secuencia las transformaciones homogéneas individuales T_1, T_2, \dots, T_n que se presentaron anteriormente (Kumar 2010). Es decir, la matriz T se calcula como:

$$T = T_1 * T_2 * \dots * T_n.$$

Es relevante resaltar que en cada transformación individual T_i , esta se expresa en el sistema de coordenadas que está conectado al cuerpo $i-1$ o sistema i . Por otro lado, el lado izquierdo de la matriz T se representa en relación con el sistema inamovible, específicamente el sistema 1. Por lo tanto, se requiere precaución para transferir de manera precisa los vectores y matrices asociados a los sistemas sucesivos antes de llevar a cabo multiplicaciones o sumas en los sistemas finales. Esta ecuación es conocida como la ecuación de cierre del robot en cuestión. Al sustituir la expresión de la matriz de transformación homogénea T_i en la ecuación, resultan dos relaciones diferentes en términos de la orientación de los eslabones de las articulaciones y las ubicaciones de los puntos de origen de los sistemas vinculados a los eslabones.(Kumar, 2010). Las relaciones son las siguientes:

$$Q = Q_1 Q_2 \dots Q_n$$

$$p = a_1 + Q_1 a_2 + \dots + Q_1 \dots Q_{n-1} a_n$$

Al contrastar ambos lados de la ecuación, se puede notar que el primer lado resulta más intuitivo, mientras que las últimas ecuaciones son más eficaces desde un punto de vista computacional. Utilizando la ecuación, se necesita calcular 16 parámetros de matrices en formato 4×4 , en comparación con los únicamente 12 parámetros empleados, con nueve de ellos asignados a matrices en formato 3×3 y tres relacionados con vectores de posición. Es relevante resaltar que, durante el análisis de la cinemática directa de posición, los componentes de los lados derechos de la ecuación se establecen como datos de entrada, mientras que los elementos de los lados izquierdos se obtienen a través de cálculos como resultados (Kumar, 2010).

2.7. Señales Electromiográficas

Las señales electromiográficas son impulsos eléctricos que surgen como resultado de la actividad muscular. Estas señales están compuestas por los movimientos fisiológicos que ocurren en las membranas celulares de las fibras musculares. Las fibras musculares están compuestas por un conjunto de hilos que son conocidos como miofibrillas o también llamados filamentos, mismos que están compuestos por actina y miosina que cuando se estimulan mediante un potencial de acción estos filamentos tienden a contraerse. La actividad eléctrica generada por la contracción muscular puede ser capturada mediante electrodos colocados en la piel cerca del grupo muscular en cuestión (Avila y Fajardo, 2019).

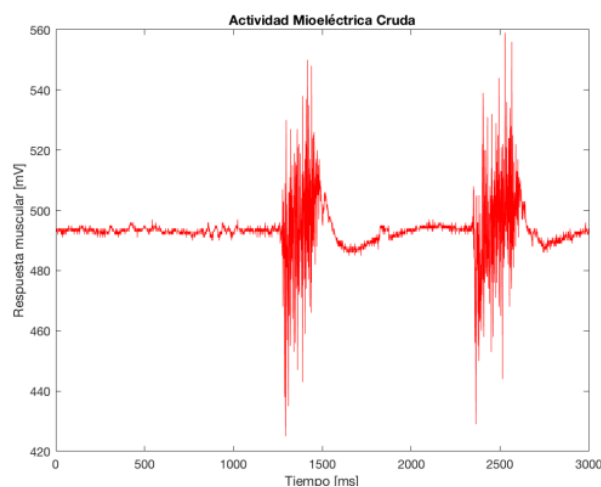


Ilustración 2-4: Señales electromiográficas

Fuente: (Ortega et al., 2018)

La electromiografía se enfoca en registrar la actividad eléctrica del músculo en respuesta a un potencial de acción, es decir, la activación generada por la célula muscular durante la excitación. Si bien las señales de electromiografía son relativamente fáciles de medir, requieren sistemas sólidos para su interpretación cuantitativa y garantizar la repetibilidad de las mediciones mediante la estandarización del proceso de adquisición. En general, para medir la electromiografía, se utiliza una configuración de electrodos de superficie en una disposición bipolar, junto con sistemas de amplificación diferencial en pruebas kinesiológicas (Proaño, 2019: p. 20).

2.7.1. Tipos de señales EMG

Los principales tipos de señales electromiográficas que se estudian son:

- **Electromiografía de fibra única (SFEMG).** Esta técnica se basa en el uso de un electrodo de agujas concéntrico especial con una superficie de registro activa de 25 μm

de diámetro, ubicada a 3 mm de la punta del electrodo. Al utilizar este electrodo especial, se logra un registro preciso y detallado de las señales eléctricas generadas por las fibras musculares, lo que brinda información valiosa sobre la función y la actividad muscular a nivel microscópico (Gomez y Treto, 2000).

- **Potencial de acción de la unidad motora (MUAP).** La electromiografía invasiva se emplea para capturar el potencial de acción particular que es generado por una unidad motora específica. Para alcanzar este propósito, se introduce un electrodo en el interior del músculo mediante agujas. Estos registros son el producto de la combinación de diversos potenciales de acción generados por fibras musculares que se contraen de manera sincronizada (Guzmán y Méndez, 2018).
- **EMG de superficie (EMGs).** Es una herramienta esencial y no invasiva que permite analizar la actividad muscular y comprender el papel de músculos específicos tanto en reposo como durante el movimiento. Esta técnica proporciona información sobre la actividad eléctrica asociada a la contracción muscular y se obtiene mediante el uso de electrodos superficiales colocados en la piel. (García, 2017).

2.7.2. Adquisición de señales EMGS

La adquisición de señales EMG se lleva a cabo mediante el uso de electrodos bipolares que se colocan sobre la piel. Estos electrodos capturan los cambios generados por la actividad eléctrica de las fibras musculares, conocidos como potenciales de acción de unidad motora. En condiciones normales, estos potenciales tienen una amplitud promedio de alrededor de 0.5 mV y una duración que varía entre 8 y 14 ms, dependiendo del tamaño de las unidades motoras. Sin embargo, la señal obtenida de los electrodos es débil y está sujeta a ruido, lo cual requiere un proceso de adecuación antes de poder ser analizada. Para mejorar la calidad de la señal EMG en términos de relación señal-ruido (Signal to Noise Ratio, SNR), es crucial utilizar un hardware apropiado que maximice la captura y preservación de la señal de interés mientras minimiza la interferencia del ruido (Moreno, 2017: p. 19).

Además de los factores que se detallaron previamente y que impactan en la relación señal-ruido (SNR), en el espectro de frecuencia se pueden identificar las siguientes interferencias:

- Ruido del equipo electrónico: desde frecuencias bajas (0 Hz) hasta varios miles de Hz.
- Interferencias ambientales: emanadas de fuentes de radiación electromagnética a 50 Hz, originadas en las fuentes de suministro eléctrico.
- Artefactos de movimiento: perturbaciones que surgen durante el desplazamiento, típicamente entre el electrodo y el cable, con una frecuencia en el rango de 0 a 20 Hz.

- Variabilidad en la señal: dentro del intervalo de 0 a 20 Hz (Moreno, 2017)

2.7.2.1. Tipos de sensores para la adquisición de señales EMG

a) Electrodo

Un electrodo superficial para EMG es un sensor que registra la actividad eléctrica de un músculo o transduce la corriente iónica en el tejido en corriente electrónica a través de cables metálicos. Estos electrodos se caracterizan por su forma, dimensiones, tecnología y materiales utilizados. Son capaces de adquirir los potenciales de acción muscular, lo que permite analizar diferentes aspectos de las señales electromiográficas que se desean estudiar. Estos electrodos son fundamentales para capturar y medir la actividad eléctrica generada por las fibras musculares, brindando información valiosa sobre la función muscular y permitiendo investigar y diagnosticar condiciones neuromusculares (Moreno, 2017: p. 17).

b) Sensor muscular Myoware AT- 04-001.

El Sensor Muscular Myoware AT-04-001 es una herramienta valiosa para la captura y análisis de señales electromiográficas (EMG) en aplicaciones de control y monitoreo muscular. Este sensor se destaca por su capacidad para recibir señales separadas de los electrodos Middle, End y Reference, lo que permite una mayor precisión en la captura de la actividad muscular. Además, su habilidad para leer la señal cruda (RAW) y rectificarla antes de enviar los datos de pico a un microprocesador lo convierte en un recurso eficaz para el análisis de la intensidad y amplitud de la actividad muscular. A través de su diseño compacto y su compatibilidad con microprocesadores como Arduino, el Sensor Muscular Myoware AT-04-001 presenta ventajas notables en aplicaciones que requieren control preciso y detallado de la actividad muscular, como en el campo de la robótica, la rehabilitación y la investigación médica. (Electronics, 2023)



Ilustración 2-5: Myoware AT- 04-001

Fuente: (Electrostore, 2019)

c) MYO ARMBAN

El MYO Armband es un dispositivo innovador fabricado por Thalmic Labs que revoluciona la forma en que interactuamos con la tecnología. Este brazalete de sensores permite reconocer y traducir los movimientos musculares realizados por el brazo y antebrazo en comandos digitales. Su característica distintiva radica en su capacidad para permitir a los usuarios controlar diversos dispositivos y sistemas de forma inalámbrica mediante gestos musculares naturales, eliminando la necesidad de interfaces físicas o táctiles.

La combinación de sensores electromiográficos (EMG) junto con giroscopios, acelerómetros y magnetómetros habilita la identificación precisa de una amplia variedad de gestos, permitiendo una interacción fluida y eficiente con la tecnología. Con aplicaciones que abarcan desde el control de dispositivos electrónicos hasta la investigación médica y la rehabilitación, el MYO Armband representa un avance significativo en la interfaz humano-computadora, ofreciendo una experiencia de usuario intuitiva y vanguardista. (Copaci et al., 2022)



Ilustración 2-6: Myo Armband

Fuente: (Copaci et al., 2022)

d) SENSOR MUSCULAR ECG/EMG 3,3V AD8832.

El Sensor Muscular ECG/EMG 3,3V AD8832 es una solución versátil diseñada para medir señales electromiográficas (EMG) y electrocardiográficas (ECG). Este dispositivo se caracteriza por su capacidad de amplificación integrada, permitiendo detectar señales musculares y cardíacas débiles para su posterior análisis. Con una alimentación de 3,3V, es compatible con diversas plataformas electrónicas y su salida analógica facilita la integración con microcontroladores y sistemas de procesamiento de señales. Aunque su tamaño compacto lo hace adecuado para dispositivos portátiles, su uso puede requerir conocimientos técnicos en electrónica y anatomía, así como la calibración periódica para mantener mediciones precisas. A pesar de estas

consideraciones, el Sensor Muscular ECG/EMG 3,3V AD8832 se destaca como una herramienta valiosa para aplicaciones médicas y de investigación, brindando una mayor comprensión de la actividad eléctrica en músculos y corazón. (Electrostore, 2019)



Ilustración 2-7: Sensor ECG /EMG 3,3v AD8832

Fuente: (Electrostore, 2019)

2.7.3. Posicionamiento de electrodos

Según la norma SENIAM (Surface ElectroMyoGraphy for the NonInvasive Assessment of Muscles), se recomienda que la distancia entre los electrodos de superficie sea de 20 mm. Sin embargo, en el caso de músculos pequeños, esta distancia no debe exceder el $\frac{1}{4}$ de la longitud del músculo en cuestión. Es fundamental ubicar correctamente los sensores para asegurar una detección de alta calidad de las señales EMG de superficie (Avila y Fajardo, 2019: p. 16).

Para ello, se requiere consultar un atlas anatómico que proporcione información precisa sobre la ubicación, origen, inserción y función del músculo estudiado, así como de los músculos cercanos que puedan generar interferencias no deseadas. Muchas recomendaciones para la colocación de electrodos se basan en referencias anatómicas, como áreas óseas destacadas o estructuras palpables. Además, es necesario posicionar al menos un electrodo de referencia neutral por sujeto.

- Colocar los sensores a lo largo del eje longitudinal medio del músculo objetivo.
- Evitar ubicar el sensor fuera de los límites del músculo, ya que esta área puede captar interferencias de músculos adyacentes.
- Posicionar el sensor entre un punto motor (zona de innervación) y un tendón de inserción, o entre dos puntos motores.
- Evitar colocar el sensor directamente sobre o cerca del punto motor, ya que esto puede generar interferencias o cancelar las señales generadas.

- Evitar situar el sensor sobre o cerca de un tendón muscular, ya que a medida que las fibras musculares se acercan al tendón, se vuelven más delgadas y menos numerosas, lo que reduce la amplitud de la señal EMG.
- Una vez determinada la ubicación general del sensor, es importante realizar contracciones musculares para asegurar la calidad de las señales detectadas (Proaño, 2019: p. 4).

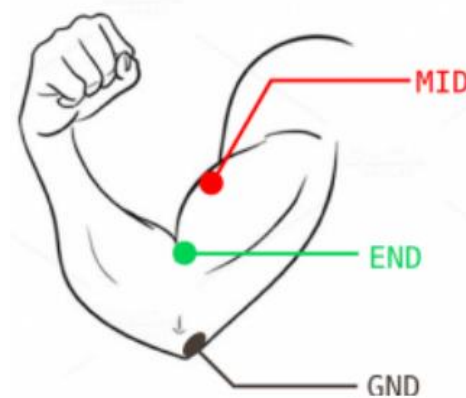


Ilustración 2-8: Posición de electrodos

Fuente: (Ortega et al., 2018)

2.7.4. Aplicaciones de las señales EMG

Las señales electromiográficas tienen diversas aplicaciones, principalmente se utilizan en el campo de la medicina con el fin de detectar patologías en los órganos de origen, así también son utilizadas mediante interfaces donde las señales EMG permiten controlar dispositivos electromecánicos, por lo tanto, las áreas de aplicación pueden ser la industria, la medicina, el hogar y el entretenimiento (Proaño, 2019). Algunas de las aplicaciones son las siguientes:

2.7.4.1. Control de interfaces Hombre-Máquina

La aplicación dentro de interfaces hombre – máquina posibilita controlar dispositivos tecnológicos como ratones, teclados, robots móviles, brazos robóticos, entre otros, en lo que concierne al control de brazos robóticos se lo realiza mediante un software y un robot móvil, donde el usuario cuenta con una interfaz gráfica que le permite entrenar el sistema y acceder a diferentes opciones de control. Las señales utilizadas para el control se obtienen a partir de tres estados diferentes que representan la actividad muscular del brazo del usuario, siendo estas el reposo, contracción media y contracción fuerte. Todos los movimientos realizados por los sistemas controlados dependen de la programación e interfaz de entrenamiento dependiendo de los grados de libertad que posea (Florez et al., 2004: p. 2-3).

2.7.4.2. Control de Exoesqueleto mediante señales electromiográficas

Las señales EMG generadas por ciertos músculos para el control de un exoesqueleto basándose en las señales musculares del bíceps y tríceps, además estas señales permiten conocer el ángulo del brazo al que se maneja en todo momento. Para lograr se emplea sensores táctiles de presión y actuadores con memoria de forma, mismos que permitirán la adaptabilidad del exoesqueleto para realizar diversas actividades, el objetivo principal de esta aplicación es lograr el movimiento de un brazo humano de una manera más realista y natural, en base a señales EMG (Suberviola, 2019).

2.7.4.3. Acondicionamiento de señales electromiográficas para el control de prótesis

El análisis electromiográficas en el control electrónico de una prótesis se lo realiza mediante las señales electromiográficas superficiales (EMGS) las cuales resultan del monitoreo superficial de la piel y de la comunicación entre neuronas y motoneuronas al interior de los músculos. Las ventajas que las señales EMGS proporcionan a una prótesis de mano pueden ser la disminución en las limitaciones físicas del usuario, reacciones en tiempo real de los músculos del individuo, mejora la sensibilidad y precisión con la que se desea realizar una actividad (Ortega et al., 2018: pp.1).

2.7.4.4. Control sistemas de movimiento para personas con discapacidades físicas

Una de las aplicaciones potenciales implica emplear estas señales con el propósito de permitir que individuos que enfrentan discapacidades físicas significativas puedan manejar sus sillas de ruedas de manera autónoma. Esto habilita el control de la dirección de desplazamiento, así como la capacidad de frenar y arrancar la silla. Igualmente, estas señales podrían utilizarse para operar sistemas de software capaces de recitar opciones de un menú en la pantalla de una computadora, que se encuentra enlazada a un sistema que adquiere señales de este tipo. Para lograrlo, es esencial asociar señales electromiográficas específicas con los movimientos de un cursor en la pantalla de la computadora, lo cual permite controlar sus trayectorias. Adicionalmente, otra aplicación similar se encuentra en el ámbito de la medicina deportiva, vinculada con la activación muscular (Rubio, 1999: pp.1).

2.8. Comunicación inalámbrica

La comunicación inalámbrica se refiere a la transferencia de información entre dispositivos sin la necesidad de cables físicos o conexiones por cable. Utiliza señales electromagnéticas, como ondas de radio, microondas, infrarrojos y otros métodos, para transmitir datos a través del aire o del espacio libre, esta tecnología ha revolucionado la forma en que los dispositivos intercambian

información, ya que elimina las limitaciones físicas de los cables permitiendo la movilidad y la conectividad en una variedad de aplicaciones. Por ende, se refiere a la transmisión de datos y comandos entre el controlador y el brazo robótico a través de señales electromagnéticas, como ondas de radio o señales de infrarrojos. En términos generales, se tienen los siguientes tipos de comunicación inalámbricas:

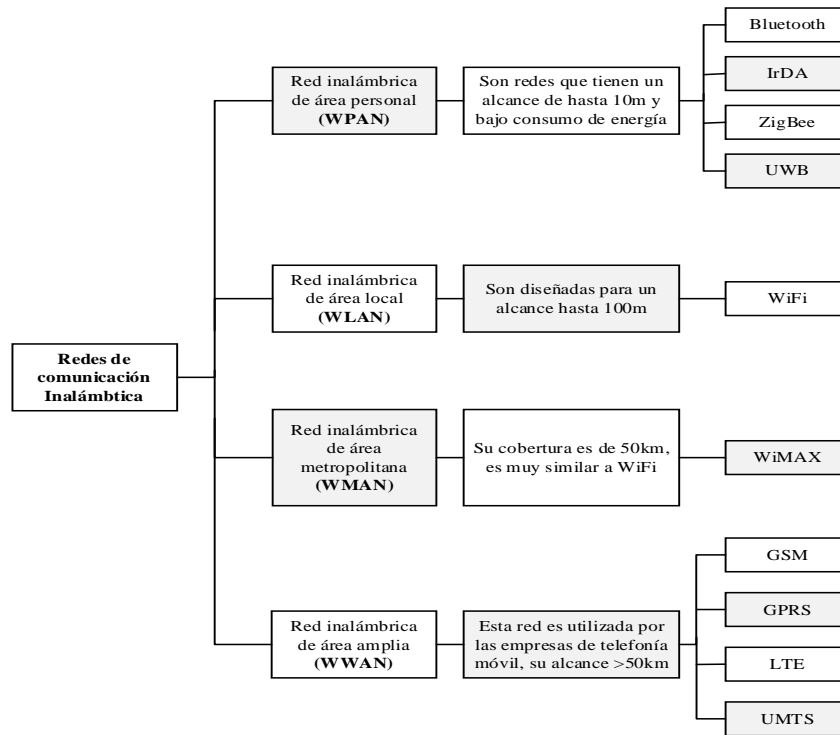


Ilustración 2-9: Tecnología de comunicación inalámbrica

Fuente: (SEO, 2020)

2.8.1. Ventajas de la Comunicación Inalámbrica en Brazos Robóticos:

La comunicación inalámbrica se ha consolidado como una tecnología necesaria en ciertos ámbitos, brindando ventajas como:

- **Flexibilidad:** La comunicación inalámbrica permite una mayor flexibilidad en la ubicación del controlador y el brazo robótico. No es necesario estar conectado físicamente a través de cables, lo que facilita la operación en espacios estrechos o difíciles de alcanzar.
- **Movilidad:** Los brazos robóticos inalámbricos son más móviles, ya que no están limitados por cables que puedan enredarse o restringir el movimiento del robot.
- **Fácil Instalación:** La instalación de sistemas inalámbricos es más sencilla y rápida en comparación con la configuración de cables y conexiones físicas.

- **Menos Interferencias:** Los sistemas inalámbricos modernos utilizan técnicas para reducir las interferencias, lo que mejora la confiabilidad de la comunicación en entornos con otros dispositivos inalámbricos.
- **Escalabilidad:** Es más fácil agregar más brazos robóticos a un sistema inalámbrico sin preocuparse por la longitud de los cables.
- **Seguridad Industrial:** Al eliminar los cables y conexiones físicas, se disminuye el riesgo de tropiezos o caídas en el lugar de trabajo, lo que es esencial para mantener la seguridad de los trabajadores. Así también la comunicación inalámbrica permite que el operador controle el brazo robótico a una distancia segura, reduciendo el riesgo de lesiones o accidentes en caso de situaciones de alto riesgo o de un mal funcionamiento del robot. Por otra parte, los cables y las conexiones físicas pueden interferir con otros equipos o maquinaria en el entorno de trabajo por lo que la comunicación inalámbrica minimiza este tipo de interferencias, y en cuanto al mantenimiento al eliminar la necesidad de desconectar y conectar cables, se reduce el riesgo de accidentes y se facilita el mantenimiento seguro del robot.

2.8.2. Características de la Comunicación Inalámbrica en Brazos Robóticos

- **Latencia.** Es el tiempo que lleva que un comando transmitido inalámbricamente llegue al brazo robótico y se ejecute. La baja latencia es crucial para aplicaciones que requieren una respuesta rápida y precisa.
- **Ancho de Banda.** Un mayor ancho de banda permite transmitir datos más rápidamente, lo que es esencial para enviar información detallada de sensores o imágenes en aplicaciones de visión.
- **Seguridad.** La seguridad de la comunicación inalámbrica es crítica para evitar la interferencia no autorizada o el acceso a los sistemas robóticos. El cifrado y otros métodos de seguridad son fundamentales.
- **Robustez.** Los sistemas inalámbricos deben ser robustos ante posibles interferencias y pérdida de señal, especialmente en entornos industriales.

2.8.3. Aplicaciones de la Comunicación Inalámbrica en Brazos Robóticos

La comunicación inalámbrica en un brazo robótico tiene una amplia gama de aplicaciones en entornos industriales y no industriales, en la Tabla 2-2 se presentan detalladamente algunas de las aplicaciones más comunes:

Tabla 2-2: Aplicaciones de la comunicación inalámbrica

APLICACIÓN	DESCRIPCIÓN
Manufactura y Automatización	Los brazos robóticos inalámbricos son útiles en líneas de producción, donde se pueden reconfigurar y desplegar fácilmente según las necesidades cambiantes.
Inspección Remota	En entornos peligrosos o difíciles de alcanzar, como la exploración de estructuras en la industria de la construcción o la inspección de equipos en la industria petrolera.
Teleoperación	Permite controlar un brazo robótico a distancia para tareas delicadas o de alta precisión.
Salud y Cuidado Personal	En rehabilitación, los brazos robóticos inalámbricos pueden ser utilizados para asistir en terapias físicas y ocupacionales.
Investigación Científica	En laboratorios y entornos de investigación, la comunicación inalámbrica facilita la operación de brazos robóticos en experimentos y estudios.

Fuente: (Detta, 2020)

Realizado por: Bautista A., & Oña D., 2023

La comunicación inalámbrica ofrece una amplia gama de beneficios en el campo de los brazos robóticos, mejorando la movilidad, la flexibilidad y la eficiencia en diversas aplicaciones. Sin embargo, es importante considerar las necesidades específicas de cada aplicación y las limitaciones técnicas al implementar sistemas de comunicación inalámbrica en brazos robótico.

CAPÍTULO III

3. MARCO METODOLÓGICO

Para el desarrollo de esta propuesta tecnológica se analizaron varios conceptos descritos en el capítulo anterior necesarios para la implementación del sistema de control por medio de electromiografía.

3.1. Enfoque de la investigación

El enfoque de esta propuesta tecnológica es cuantitativo, consiste en la implementación de un sistema que permita controlar el movimiento de un brazo robótico modelo ANNO RV624, utilizando señales electromiográficas (EMG). Por lo tanto, se busca capturar las señales EMG a través de electrodos colocados en la superficie de la piel, con el fin de procesar, analizar estas señales y utilizar los resultados para controlar los movimientos del brazo robótico.

3.2. Nivel de investigación

La presente propuesta tecnológica es de nivel aplicativo, ya que busca poner en práctica un sistema de control de movimiento del brazo robótico ANNO RV624, basado en señales electromiográficas dentro de la industria manufacturera como alternativa de control a distancia durante procesos o actividades peligrosas, difíciles o que necesitan de gran precisión, por otra parte también es de tipo experimental debido a que se realizará pruebas para evaluar el rendimiento y la eficacia del sistema implementado, estas pruebas pueden incluir comparación, evaluación de la precisión y la velocidad de respuesta del sistema del control basado en la EMG.

3.3. Tipo de investigación

3.3.1. *Investigación documental*

La investigación es de tipo documental o bibliográfica porque se busca información para comparar, evaluar y comprender el comportamiento del brazo robótico mediante señales electromiográficas por lo que se ha buscado información en diferentes documentos como: libros, tesis, artículos científicos y páginas web, esta información es fundamental para obtener todos los conocimientos previos sobre señales electromiográficas, métodos de adquisición, algoritmos de control, por lo tanto, la revisión literaria existente permite identificar las mejores opciones, metodología para aplicarlo en esta investigación.

3.3.2. Investigación Descriptiva

Se busca describir las características y propiedades del sistema de control de movimiento remoto del brazo robótico y así como de las señales electromiográficas empleadas en dicho proceso. Se busca analizar las especificaciones técnicas del sistema, incluyendo aspectos como la velocidad, seguridad y capacidad de respuesta del control de movimiento, con el fin de comprender a fondo su desempeño operativo.

3.3.3. Investigación de Campo

La investigación implica la recolección de datos y observaciones dentro de un entorno operativo real con el propósito fundamental de evaluar la eficacia y aplicabilidad del sistema de control de movimiento remoto implementado en conjunto con las señales electromiográficas, durante la investigación de campo, se dará prioridad a la evaluación de la capacidad de respuesta temporal del sistema ante comandos de control remoto. Se registrarán y analizarán datos relacionados con la ejecución de movimientos específicos, la adaptabilidad del sistema a diversas velocidades y la velocidad de respuesta a cambios inesperados.

3.3.4. Investigación experimental

Esta investigación es de tipo experimental, ya que se llevarán a cabo una serie de pruebas para evaluar el rendimiento del sistema de control remoto del brazo robótico ANNO RV624. Este sistema se fundamenta en la utilización de señales electromiográficas, las cuales permiten la captura de datos provenientes de los músculos, fundamentales para el procesamiento, diseño e implementación del algoritmo de control. En el transcurso de esta fase experimental, se llevarán a cabo pruebas destinadas a evaluar la capacidad de respuesta del sistema ante una variedad de comandos de control remoto. Se prestará especial atención a la eficacia en la ejecución de movimientos y la adaptabilidad con el entorno. Además, se explorará la eficacia del algoritmo de control en la interpretación de las señales electromiográficas, evaluando su capacidad para traducir de manera precisa la intención del usuario en acciones del brazo robótico.

3.4. Metodología para el desarrollo de la propuesta tecnológica

El diseño de la metodología para la implementación de un sistema del control de movimiento remoto del brazo robótico seleccionado basado en señales electromiográficas se muestra en la Ilustración 3-1.

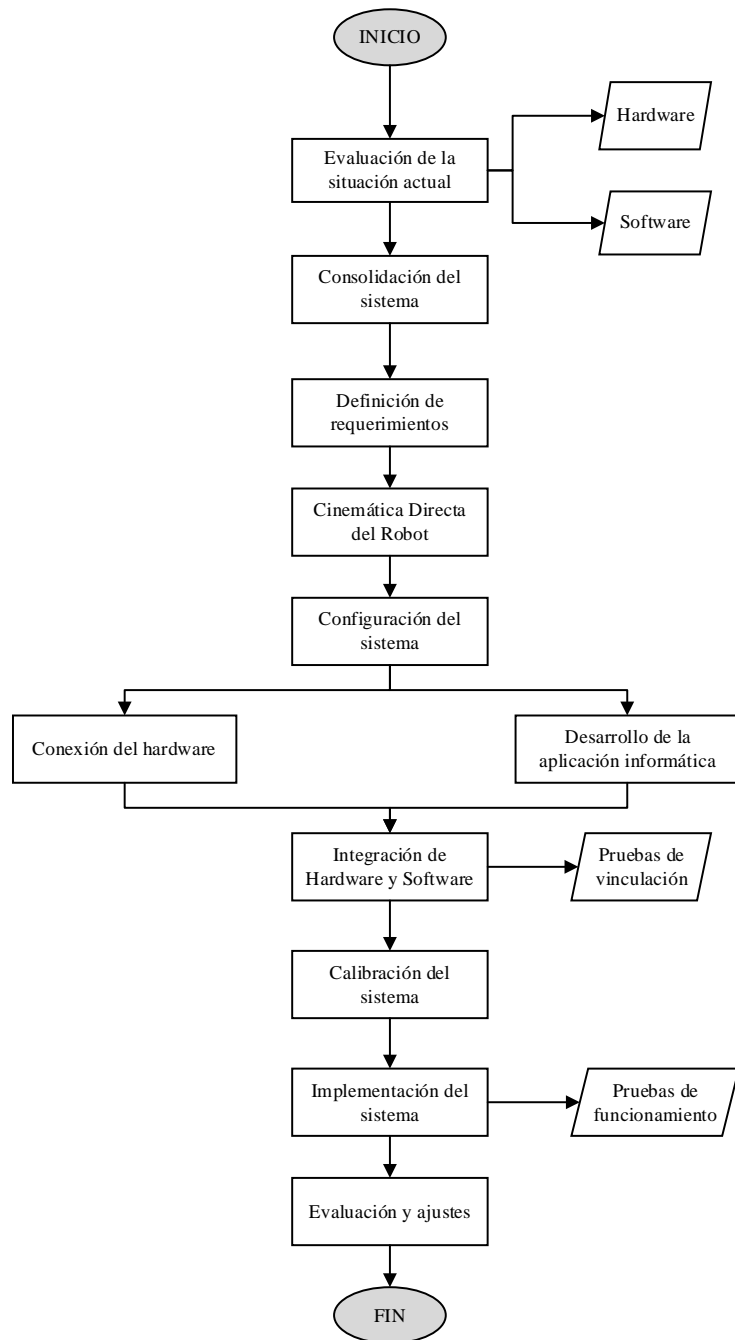


Ilustración 3-1: Diagrama de flujo de la metodología

Realizado por: Bautista A., & Oña D., 2023

Dentro de la metodología empleada, se llevó a cabo una evaluación preliminar del estado actual del brazo robótico, donde se analizó tanto la parte de hardware como de software, dentro del hardware se realizó una valoración de todos los elementos que integran el brazo y en la parte de software interactuando con la plataforma de control propia del brazo robótico.

Una vez que se ha detallado el estado actual del hardware y software, se pudo definir la consolidación del sistema, donde se ha delineado el alcance del proyecto, dando paso a la

arquitectura del sistema, este proceso implica la identificación de las etapas específicas involucradas en el control del brazo robótico, detallando cómo se realizó es la adquisición de señales EMG, el sistema de control y el movimiento del brazo robótico. La consolidación y arquitectura del sistema permiten, a su vez, definir los requerimientos necesarios para garantizar la viabilidad y eficacia del brazo robótico.

Al trabajar con un elemento robótico se consideró también hacer el levantamiento de información sobre su cinemática directa, este proceso resulta fundamental ya que posibilita determinar la posición y orientación final del extremo del brazo también conocido como herramienta o efector final. Posteriormente se realizó la configuración de los elementos del sistema, tanto a nivel hardware como software, en lo que respecta a la configuración hardware, se identificaron los equipos o dispositivos necesarios para la adquisición de las señales biométricas asociadas a las señales EMG. Además, se buscó la manera de procesarlas dentro de una aplicación informática, una vez lograda la integración de estos elementos se realizó pruebas de adquisición de datos para verificar la vinculación y conexión entre los componentes hardware y la interfaz, con base a esta etapa, se llevó a cabo un proceso de calibración del sistema. Finalmente, se ingresó a la fase de evaluación y ajuste, sometiendo al sistema a pruebas, con el fin de discernir su eficacia y realizar ajustes necesarios para optimizar su desempeño.

3.5. Diagnóstico de la situación actual

El Grupo de Investigación "Tecnologías de la Información, Comunicación y Procesos Industriales AUTOPRO" de la Facultad de Mecánica de la ESPOCH dispone de un brazo robótico ANNO RV624, que se empleó para el desarrollo de la propuesta tecnológica, considerándolo como un actuador ideal, con el fin de implementar su control mediante señales electromiográficas.


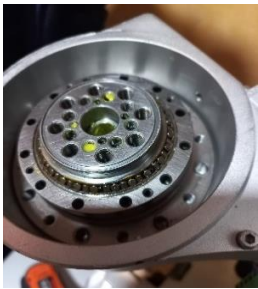

Este análisis se basa en la revisión de investigaciones previas que utilizaron el brazo robótico ANNO RV624 en diversas implementaciones. En este contexto, cabe destacar que este robot ha desempeñado un papel clave en investigaciones anteriores, incluyendo la creación de una arquitectura de software para teleoperación y supervisión, la implementación de un sistema inclusivo destinado a personas con dificultades de movilidad, y el desarrollo de un sistema de visión artificial para el reconocimiento y clasificación de piezas.

Se llevó a cabo una evaluación del estado inicial del brazo robótico abarcando tanto sus componentes como los elementos esenciales para su correcto funcionamiento. A continuación, se presenta un análisis detallado de la situación actual de los componentes identificados de hardware y software.

3.5.1. Análisis situación actual Hardware

En cuanto a los componentes físicos del robot, dado a su diseño de seis grados de libertad, consta de seis servomotores, un gripper y su conjunto controlador, cabe destacar que la estructura en su totalidad está fabricada en aluminio.

Tabla 3-1: Análisis de funcionamiento inicial de los componentes Hardware

COMPONENTE	ELEMENTOS	ESTADO		OBSERVACIÓN	DEFECTO	
		Bueno	Malo			
Servo motor integrado	57AIM30 (J1)	X				
	60AIM25 (J2)	X				
	57AIM30 (J3)			X	Inicialmente el motor de la junta 3 se activó de manera involuntaria, quedando en una posición incorrecta. En esta situación, no era posible efectuar ningún tipo de movimiento en dicha junta.	
					Se desmontó la junta, y se determinó que la falta de movimiento se debía al posicionamiento incorrecto de su rodamiento.	
	57AIM30 (J4)	X				
	42AIM30 (J5)		X	En ocasiones no se reciben las señales enviadas correctamente, debido a la presencia de un cable suelto.		
	42AIM15 (J6)	X				
Conjunto Gripper	Pinza		X	Se encuentra dañada, lo que imposibilita tareas de sujeción. Se plantea el diseño de una nueva herramienta.		
	Electroválvula	X				
	Mini bomba	X				
Conjunto controlador	Controlador de movimientos	X				
	Fuente de poder	X				
	Placa de relés	X				
	Regulador de voltaje	X				

	Placa de conexión (DB25-M)	X			
--	----------------------------	---	--	--	--

Realizado por: Bautista A., & Oña D., 2023

Resultado del análisis de la información recolectada en la Tabla 3-1 revela que, en su mayor parte, el brazo mantiene un funcionamiento satisfactorio en la mayoría de sus elementos. En aquellos casos en los que se identificaron fallos, estos fueron fácilmente corregibles, abarcando desde problemas en la recepción de señales hasta desajustes en el posicionamiento de la arandela. No obstante, se detectó un daño que amerita el rediseño y la fabricación de una pieza específica.

3.5.2. *Análisis situación actual Software*

Con el objetivo de analizar los parámetros de funcionamiento del software actual, se lleva a cabo un análisis de las características del programa propio. Este software está dotado de una interfaz gráfica que facilita la interacción entre el usuario y el robot, así como de un programa de simulación que posibilita el estudio de la cinemática del sistema robótico, a continuación, se presenta la descripción de:

3.5.2.1. *Instalación Driver*

Para el control del Robot ANNO RV624 se cuenta con dos aplicaciones, sin embargo para poder ejecutar las aplicaciones se requiere contar con la instalación de los drivers, por lo tanto, durante este proceso se realizaron pruebas con varios controladores, y el que resultó más adecuado fue el denominado "PL2303_Prolific_GPS_1013_20090319", como se aprecia en la Ilustración 3-2, es relevante señalar que este controlador no se encuentra disponible en la página oficial de Prolific. Como resultado, se requirió una búsqueda detallada para localizar este controlador específico.

 PL2303_Prolific_GPS_1013_20090319.exe	19/3/2009 14:16	Aplicación	2.987 KB
 releasenote.txt	18/8/2009 15:37	Documento de tex...	2 KB

Ilustración 3-2: Prerrequisitos de instalación

Realizado por: Bautista A., & Oña D., 2023

En el proceso de configuración, es necesario seleccionar el dispositivo adecuado para establecer la conexión entre el equipo y el brazo robótico. En este caso, se debe escoger "Prolific USB-to serial comm Port (COM19)" para garantizar una comunicación fluida y efectiva. Esta selección permite que el software del brazo robótico interactúe de manera óptima con el dispositivo a través del puerto serial virtual COM19, ver Ilustración 3-3.

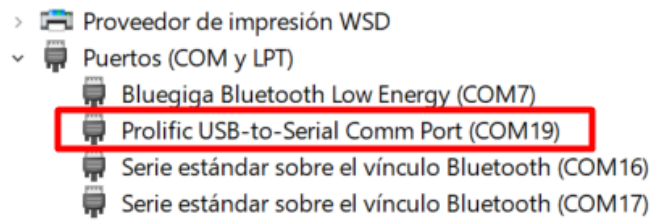


Ilustración 3-3: Seleccionar Prolific USB-to serial comm Port (COM19)

Realizado por: Bautista A., & Oña D., 2023

3.5.2.2. Interfaz gráfica para robot ANNO RV624

El software para evaluar el rendimiento del robot es proporcionado de fábrica y ofrece dos opciones: una aplicación básica y otra completa, tal como se muestra en la Ilustración 3-4. Una vez manipulado el robot con ambas alternativas, se ha observado que la aplicación básica brinda una respuesta óptima y eficaz en comparación con la versión completa. La aplicación elegida para su uso es denominada "RobotAnnoV2.0".

📁 Languages	28/6/2023 15:47	Carpeta de archivos	
📄 DrawImageUseRobotArm.exe	6/11/2018 21:02	Aplicación	19 KB
📄 halcon.dll	6/11/2018 21:02	Extensión de la ap...	31.564 KB
📄 halcondotnet.dll	6/11/2018 21:03	Extensión de la ap...	1.250 KB
📄 halcondotnetxl.dll	6/11/2018 21:03	Extensión de la ap...	1.250 KB
📄 hdevenginedotnet.dll	6/11/2018 21:02	Extensión de la ap...	54 KB
📄 hdevenginedotnetxl.dll	6/11/2018 21:02	Extensión de la ap...	54 KB
📄 Robot AnnoSTM32V2.0.exe	15/1/2020 16:07	Aplicación	261 KB
📄 RobotAnnoTestTools.exe	6/11/2018 21:03	Aplicación	192 KB

Ilustración 3-4: Instalación del software ANNOSTM32

Realizado por: Bautista A., & Oña D., 2023

Dentro de la aplicación propia del brazo robótico, la interfaz se presenta como una pantalla principal, con la que el operador puede interactuar directamente con el brazo, esta interfaz está equipada con todos los elementos esenciales para programar la trayectoria del robot, junto con sistemas de alarma pertinentes y una gama de opciones de navegación.

Sin embargo, es importante señalar que las dos aplicaciones vienen preconfiguradas en el idioma mandarín, lo que complica la interacción con el operador. Para resolver este problema, se puede considerar la posibilidad de ajustar la interfaz a un idioma más familiar y accesible para el operador.

En ambas ilustraciones se muestran las dos aplicaciones que permiten dar funcionamiento al brazo robótico Anno RV624, apoyadas en la existencia de controles análogos, tales como la regulación

de la velocidad, la gestión de ángulos de desplazamiento, la especificación de coordenadas, y la transmisión directa de datos hacia el robot.

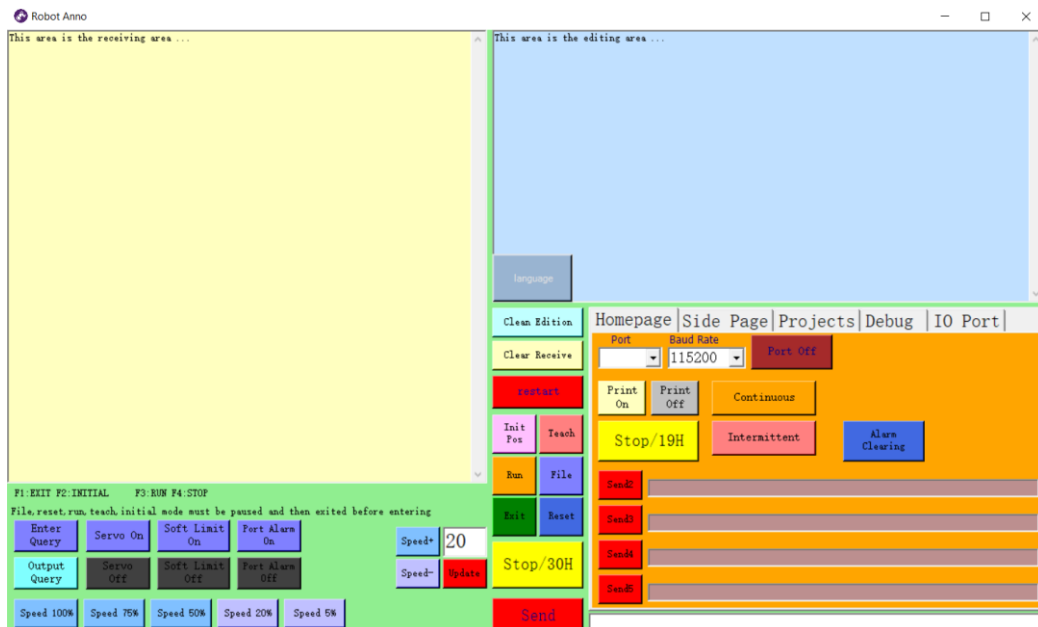


Ilustración 3-5: Interfaz de trabajo Robot Anno

Realizado por: Bautista A., & Oña D., 2023

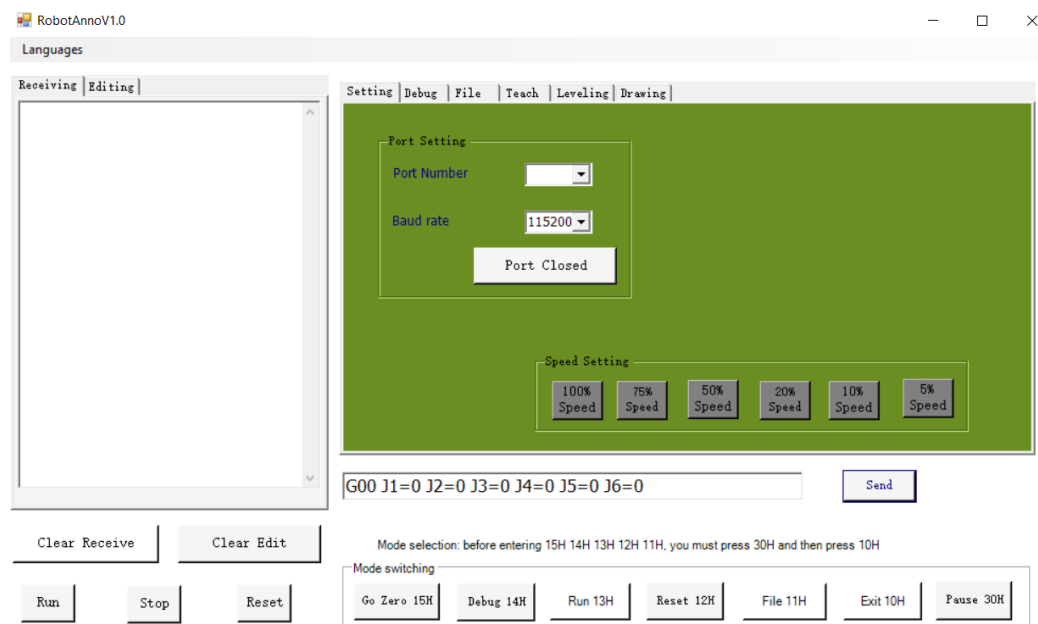


Ilustración 3-6: Interfaz de trabajo RobotAnnoV1.0

Realizado por: Bautista A., & Oña D., 2023

Se evidencia que la primera aplicación, identificada como Robot Anno, ostenta una complejidad operativa más pronunciada, en contraposición a la aplicación Robot AnnoV1.0, la cual se presenta como una interfaz notablemente más simplificada y accesible para el operario.

3.5.2.3. Utilización de software RobotAnnoV1.0

Después de haber instalado el controlador correcto se ingresa a la sección de puertos COM, que presentará el puerto correspondiente para establecer la conexión, es necesario seguir una secuencia predefinida para poder controlar el brazo robótico de manera efectiva, tal como:

- Seleccionar "Pause 30H": Esta instrucción permite pausar el funcionamiento del robot durante un periodo específico, que en este caso corresponde a 30 segundos. Esto puede ser útil para realizar ajustes o intervenciones necesarias durante la operación.
- "Exit 10H": Al seleccionar esta opción, el robot ejecutará la acción de salir o detener su operación actual, es decir para interrumpir una tarea en curso o para finalizar una secuencia de movimientos.
- "Go zero 15H": Esta instrucción dirige al brazo robótico a su posición inicial o "cero". Es útil para restablecer la posición del robot antes de iniciar una nueva tarea o movimiento.

Una vez realizado los pasos antes mencionados ya es posible controlar el brazo robótico de manera precisa y confiable, asegurando un funcionamiento seguro en las tareas que se requieran realizar.

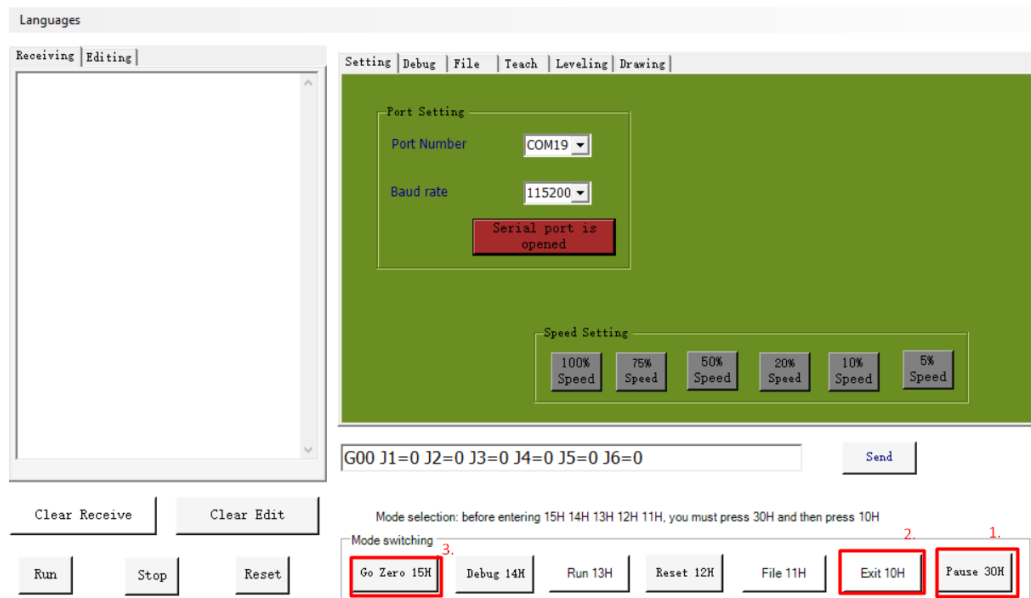


Ilustración 3-7: Ventana de trabajo para controlar el brazo robótico

Realizado por: Bautista A., & Oña D., 2023

La opción "Teach", permite guiar y programar los movimientos de los eslabones del robot de manera gradual y precisa como se muestra en la Ilustración 3-8. Estos movimientos pueden ser grabados y repetidos posteriormente para automatizar tareas específicas.

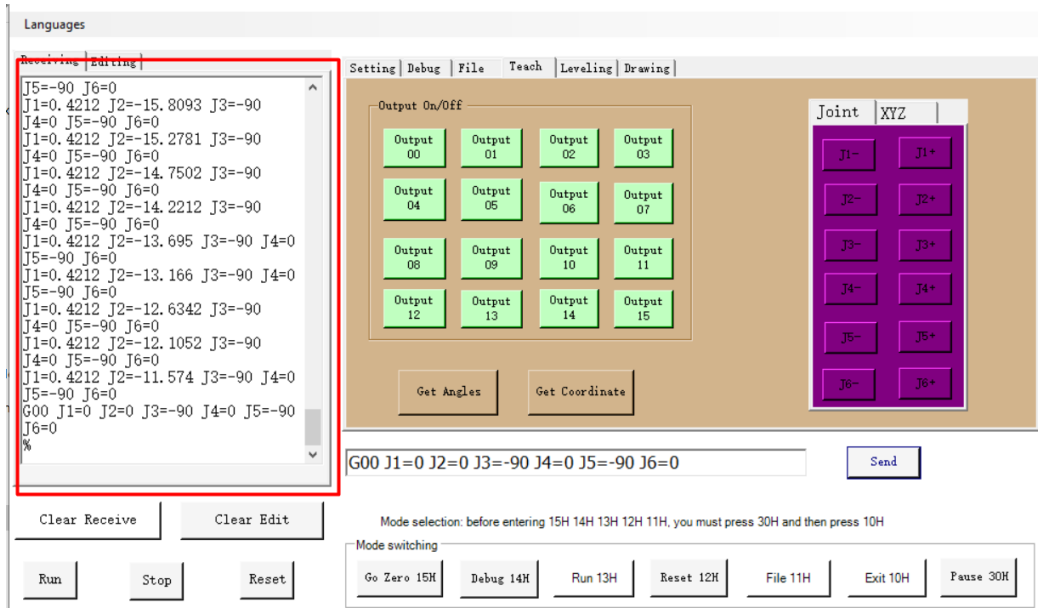


Ilustración 3-8: Ventana de control del brazo robótico

Realizado por: Bautista A., & Oña D., 2023

3.5.2.4. Robot Anno Simulator

Para evaluar el desempeño del brazo robótico ANNO RV624, se aprovechó la información proporcionada por los fabricantes, incluyendo controladores y una guía de usuario, así como la disponibilidad de un simulador. Este simulador se utilizó como una herramienta introductoria para explorar la cinemática del brazo robótico y entender las limitaciones físicas en su manipulación en diferentes ejes.

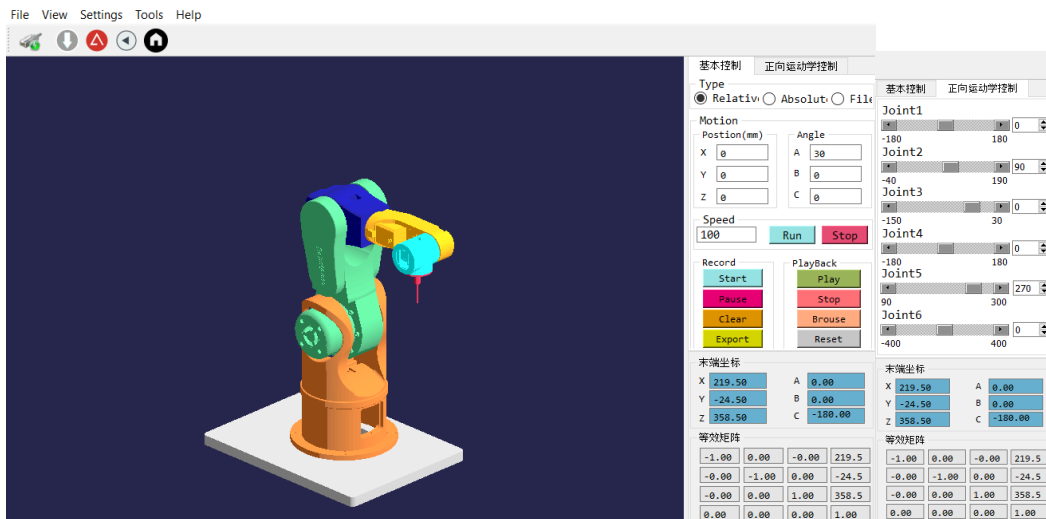


Ilustración 3-9: Interfaz gráfica ANNO-VRSimulator

Realizado por: Bautista A., & Oña D., 2023

Como se visualiza en la Ilustración 3-9 la interfaz consta de dos pestañas: la primera se utiliza para controlar el brazo a través de coordenadas, mientras que la segunda posibilita la manipulación utilizando las articulaciones mecánicas del brazo.

3.6. Consolidación del sistema

Para el desarrollo del sistema de control de movimiento remoto del brazo robótico seleccionado basado en señales electromiográficas se requirió determinar el funcionamiento del sistema, se debe combinar el hardware y software apropiados para el control de las funciones del brazo robótico por medio de un canal de comunicación idóneo, este proceso se visualiza en la Ilustración 3-10.

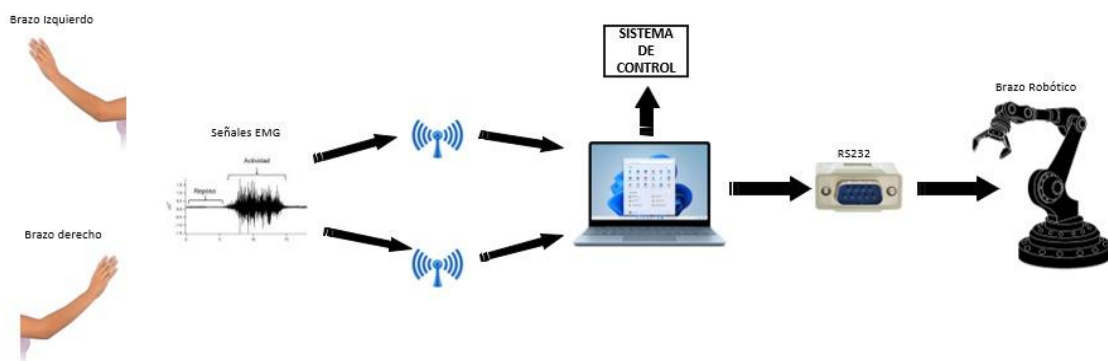


Ilustración 3-10: Diseño del sistema

Realizado por: Bautista A., & Oña D., 2023

El sistema representado en la Ilustración 3-10 se compone de dos sensores diseñados para captar señales electromiográficas (EMG), los cuales asumen la doble función de interpretar gestos y controlar los grados de libertad del robot, estos dispositivos generan y transmiten inalámbricamente datos hacia un entorno de programación, donde son analizados y procesados. Los sensores se encuentran ubicados estratégicamente, uno en el brazo izquierdo y otro de carácter secundario en el brazo derecho, con el propósito de discernir patrones de actividad muscular. Por último, los resultados de este proceso se utilizan para controlar las acciones del robot, sincronizándolas con los movimientos y gestos identificados y decodificados.

Los brazaletes portadores de los sensores EMG desempeñan un papel crucial al permitir el reconocimiento de movimientos musculares con un enfoque bidireccional, esto implica que los sensores EMG, además de ofrecer una supervisión de la actividad muscular, están habilitados para operar en una función de control. Este control se proyecta sobre los seis grados de libertad del brazo robótico, brindando la capacidad de maniobrar en múltiples direcciones de manera precisa

y coordinada. Además, se configura la facultad para interpretar gestos, una característica que confiere la habilidad de dirigir el gripper del brazo robótico.

Las señales captadas y registradas por los sensores que desempeñan una función cardinal al determinar patrones direccionales de movimiento. Estos patrones, codificados en las señales EMG, se procesan y traducen para transmitir información direccional específica, la transferencia de esta información se ejecuta a través de una comunicación serial, estableciendo una conexión efectiva y de bajo nivel entre los sensores y un entorno de programación. Esta última entidad se encarga de recibir y decodificar la información, habilitando un entorno propicio para la emisión de comandos.

3.7. Arquitectura del sistema

La arquitectura del sistema desarrollado se estructura a través de tres etapas principales, cuya disposición y funcionalidad se detallan en la Ilustración 3-11. Cada una de estas etapas desempeña un papel importante en la consolidación y eficacia del sistema en su conjunto.

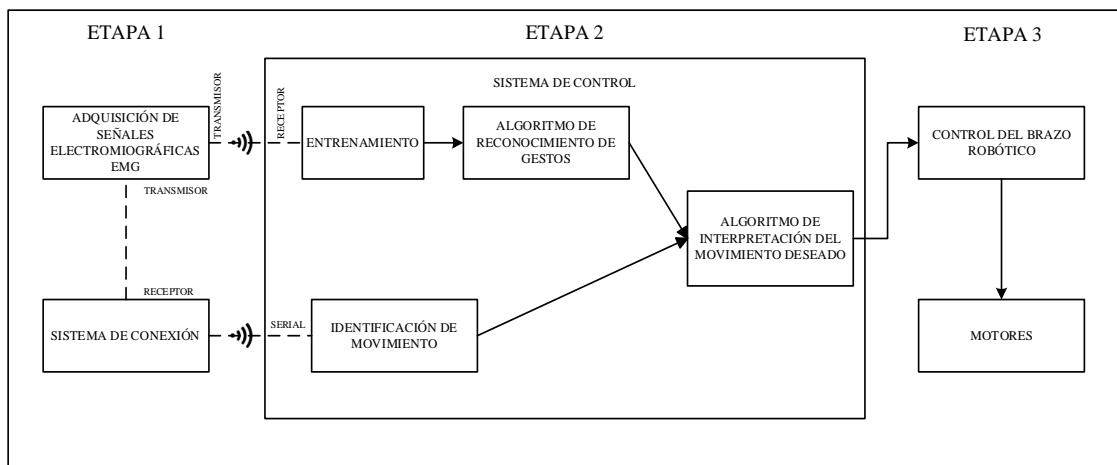


Ilustración 3-11: Arquitectura del sistema

Realizado por: Bautista A., & Oña D., 2023

A continuación, se detallan las etapas que interactúan en la arquitectura del sistema:

3.7.1. Etapa 1 – Adquisición de señales electromiográficas EMG

En la etapa inicial, el sistema realizó la adquisición de las señales electromiográficas EMG, donde los sensores detectarían la contracción y relajación muscular, capturando estas señales a través de electrodos en la piel. Estos datos de sensores son procesados en tiempo real mediante algoritmos avanzados, interpretando tanto las señales eléctricas musculares como los patrones de movimiento

para identificar gestos y acciones específicas realizadas por el usuario. Los sensores se comunicarán a través de un transmisor hacia un receptor lo que facilita la transmisión de los datos procesados a dispositivos externos.

3.7.2. Etapa 2 – Sistema de Control

La siguiente etapa es ejecutada por un lenguaje de programación, este sistema se estructura en torno a tres bloques principales; el primer bloque integra un algoritmo de reconocimiento de señales que tiene la capacidad de identificar los gestos almacenados por el usuario; el segundo bloque contiene un intérprete de gestos destinados exclusivamente al sensor; este intérprete es capaz de discernir si el usuario está proporcionando movimientos direccionales hacia la derecha o la izquierda. El tercer bloque engloba un algoritmo de interpretación que procesa las salidas derivadas de los dos bloques anteriores y genera la instrucción de movimiento destinada al robot mediante una conexión serial.

3.7.3. Etapa 3 – Movimiento del brazo robótico

La última etapa se enfoca en el movimiento real del robot, establece una comunicación bidireccional con el sistema a través de enlaces seriales. Esto implica la capacidad tanto de transmitir como de recibir información, en este caso recibe datos codificados en formato G-code que son interpretados para generar movimientos específicos, para luego estos ser transmitidos a los motores del robot, permitiendo que realice las acciones requeridas.

Para culminar, el sistema inicia su proceso detectando los movimientos o gestos registrados por los sensores electromiográficas. Estos movimientos son luego cotejados con la base de datos de calibración y en función de esta comparación, el sistema toma decisiones sobre los movimientos del brazo robótico. Finalmente, la información correspondiente a estos movimientos se traduce en códigos que el robot interpreta y ejecuta, completando así el ciclo de funcionamiento.

3.8. Definición de requerimientos

Dado que el enfoque principal del presente proyecto radica en el control del brazo robótico ANNO RV624 a través de señales electromiográficas, resulta esencial considerar la identificación y establecimiento de los requisitos tanto de hardware como software, mismos que permitirán el correcto funcionamiento del sistema y se detallan a continuación:

- Determinar a detalle los componentes del Robot ANNO RV624.
- Identificar el dispositivo encargado de captar y decodificar las señales electromiográficas.
- Plantear un sistema alternativo de comunicación del sensor EMG a un microcontrolador.
- Identificar el software capaz de codificar el algoritmo requerido para la adquisición y procesamiento de señales electromiográficas.
- Desarrollar una interfaz gráfica que facilite la calibración del sistema de manera efectiva.
- Establecer una conexión inalámbrica entre el sistema de control de señales electromiográficas y el sistema de control del brazo robótico ANNO RV624.

3.9. Hardware

Para la implementación del proyecto se cuenta con los siguientes elementos: brazo robótico ANNO RV624, controlador integrador, sensor de señales electromiográficas y un procesador. A continuación, se detallan las características del hardware empleado.

3.9.1. *Brazo robótico ANNO RV624*

El brazo robótico ANNO RV624 es un robot industrial que combina la versatilidad y la seguridad para llevar a cabo una amplia gama de tareas en entornos de producción y fabricación. El sistema está diseñado por módulos con el propósito de adaptarse a diferentes estructuras y aplicaciones industriales.

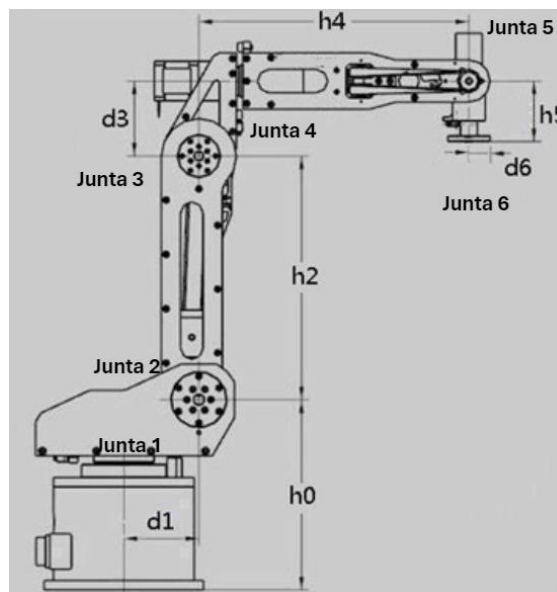


Ilustración 3-12: Robot ANNO RV624

Fuente: (Manual, 2023)

La integración de sensores y sistemas de detección en el brazo robótico agrega una capa adicional de seguridad, lo que resulta en un rendimiento optimizado en diversas tareas. Además, su capacidad para recibir comandos a distancia, como los generados por señales electromiográficas (EMG) provenientes de brazaletes, lo convierte en una opción prometedora para aplicaciones de control remoto. En la Ilustración 3-12 se observa las articulaciones y los tipos de rotación los cuales son controlados mediante motores paso a paso.

Como se puede observar en la Ilustración 3-12, el robot posee articulaciones rotacionales que son operadas por una variedad de motores paso a paso de diferentes modelos. A continuación, se presenta las especificaciones del brazo robótico ANNO RV624:

Tabla 3-2: Especificaciones del brazo robótico ANNO RV 624

CARACTERÍSTICA	DESCRIPCIÓN
Marca	Robot Anno
Modelo	RV624
Tipo de controlador	<ul style="list-style-type: none"> • Servo de 6 ejes o motor paso a paso y control de bucle semicerrado por pulsos • Detección de cero de 6 vías • Interruptor de límite positivo de 6 vías • Interruptor de límite negativo de 6 vías
Parámetro de control	<ul style="list-style-type: none"> • Puntos de entrada y salida de E/S: • Entrada dedicada de 24 puntos • Salida dedicada de 24 puntos • Interfaz de punto de entrada/salida de transmisión en serie del extremo remoto de 256 puntos • Voltaje de alimentación: 24V
Funciones de la tarjeta de control	<ul style="list-style-type: none"> • Algoritmo de control del manipulador conjunto de la serie de 6 grados de libertad. • Además de las funciones comunes, proporciona descripciones y ejemplos de API de interfaz de control, adecuados para usuarios como la integración de líneas de producción automatizadas y el desarrollo de su propio software operativo. • Proporciona una interfaz de función DLL para el desarrollo secundario y proporciona un código fuente de rutina. • El software de control proporcionado se utiliza directamente para admitir operaciones de código G.
Software	<p>Admite una variedad de sistemas operativos, Windows, WinCE, Linux, Android, etc., se prefiere el desarrollo de sistemas integrados.</p> <ul style="list-style-type: none"> ➢ Windows10 de 32,64 bits ➢ Windows8 32,64 bits ➢ Windows7 de 32,64 bits ➢ Windows XP y XP de 64 bits ➢ Servidor 2003, XP y Servidor 2008 ➢ Windows CE 4.2, 5.0,6.0 y 7.0

Fuente: (RobotAnno, 2021)

Realizado por: Bautista A., & Oña D., 2023

El sistema del brazo robótico se encuentra conformado por un total de seis motores de tipo paso a paso, complementados por un dispositivo de sujeción conocido como "gripper", situado en la parte superior del brazo. Este último componente tiene la función de operar las tenazas destinadas a la tarea de sujeción. Cada uno de los motores posee su propia designación específica, que se describe en la Tabla 3-3.

Tabla 3-3: Denominación de los motores paso a paso que contiene el brazo

MOTOR	DENOMINACIÓN	CARACTERÍSTICAS
Servo motor integrado 57AIM30	J1	50W, 2.2-4.4A, 24-36v, 500KHz
Servo motor integrado 60AIM25	J2	7A, 36V, 200 W
Servo motor integrado 57AIM30	J3	50W, 2.2-4.4A, 24-36v, 500KHz
Servo motor integrado 57AIM30	J4	50W, 2.2-4.4A, 24-36v, 500KHz
Servo motor integrado 42AIM30	J5	Nema 34, 50w, 1.5-4.4 A, 24V, 500 KHz
Servo motor integrado 42AIM15	J6	Nema 34, 100w, 1.5-4.4 A, 24V, 500 KHz

Fuente: (Mera y Quicaliquin, 2021)

Realizado por: Bautista A., & Oña D., 2023

3.9.1.1. Servo motor integrado 57AIM30

El servo motor integrado 57AIM30 se encuentra situado en la articulación 1 (J1), articulación 3 (J3) y articulación 4 (J4), con el propósito de proporcionar movimiento a dichas articulaciones, sus especificaciones técnicas se muestran en Tabla 3-4.

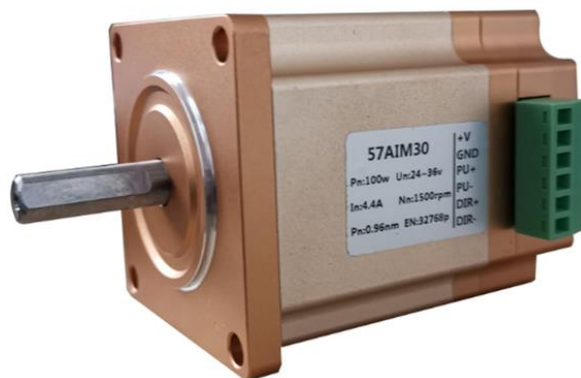


Ilustración 3-13: Servo motor integrado 57AIM30

Fuente: (Yizhi, 2020)

Tabla 3-4: Especificaciones técnicas del motor 57AIM30

Fuente de alimentación	Voltaje	24~36 VDC
	Amperaje	4.4 A
Parámetros del Motor	Esfuerzo de Torsión	0.96 NM
	Velocidad Nominal	1000 RPM
	Velocidad Máxima	1500 RPM
	Potencia	100 W
Señal de retorno	Encoder absoluto de múltiples vueltas (una sola vuelta de 32768 pulsos, una sola vuelta de 15 bits)	
Método de Enfriamiento	Enfriamiento natural	
Peso		
Modo de control de posición	Frecuencia máxima de entrada de pulsos	500 KHz
	Modo de instrucción de pulsos	Pulso + dirección, fase A + fase B
	Relación de transmisión electrónica	Ajustar rango de 1-65535 a 1-65535
	Muestreo de ubicación	2 KHz
Función Protectora	Alarma de Bloqueo	
Interfaz de Comunicación	RS485, (modbus RTU 19200,8, N,1)	
Ambiente de operación	Temperatura Ambiente	0~40°
	Temperatura máxima admisible del motor	85°
	Humedad	5~95%

Fuente: (Manual Anno RV624, 2020)

Realizado por: Bautista A., & Oña D., 2023

3.9.1.2. Servo motor integrado 60AIM25

El servo motor integrado 60AIM25 se encuentra ubicado en la articulación 2 (J2) del robot y es responsable de dar movimiento a la misma, las especificaciones técnicas del servo motor se detallan en la Tabla 3-5.



Ilustración 3-14: Servo motor integrado 60AIM25

Fuente: (Yizhi, 2020)

Tabla 3-5: Especificaciones técnicas del motor 60AIM25

Fuente de alimentación	Voltaje	36 VDC \pm 10%
	Amperaje	7A
Parámetros del motor	Esfuerzo de Torsión	2 NM
	Velocidad Nominal	1000 RPM
	Velocidad Máxima	1500 RPM
	Potencia	250 W
Señal de retorno	Codificador magnetoelectrónico de 15 bits de una sola vuelta (32768 pulsos/vuelta)	
Método de enfriamiento	Enfriamiento natural	
Modo de control de posición	Frecuencia máxima de entrada de pulsos	500 KHz
	Modo de instrucción de pulsos	Pulso + dirección, fase A + fase B
	Relación de transmisión electrónica	Ajustar rango de 1-65535 a 1-65535
	Muestreo de ubicación	2 KHz
	Frecuencia	
Función protectora	Alarma de bloqueo	
Interfaz de comunicación	RS485, (modbus RTU 19200,8,N,1)	
Ambiente de operación	Temperatura Ambiente	0~40°
	Temperatura máxima admisible del motor	85°
	Humedad	5~95%

Fuente: (Manual Anno RV624, 2020)

Realizado por: Bautista A., & Oña D., 2023

3.9.1.3. Servo motor integrado 42AIM30

El servo motor integrado 42AIM30 se ubica en la articulación 5 (J5) y brinda movimiento a dicha articulación, en la Tabla 3-6 se muestran sus respectivas especificaciones técnicas.

**Ilustración 3-15:** Servo motor integrado 42AIM30

Fuente: (Yizhi, 2020)

Tabla 3-6: Especificaciones técnicas del motor 42AIM30

Fuente de alimentación	Voltaje	24~36 VDC
	Amperaje	4.4A
Parámetros del Motor	Esfuerzo de Torsión	0.96NM
	Velocidad Nominal	1000 RPM
	Velocidad Máxima	1500 RPM
	Potencia	100 W
	Resistencia	1,3Ω
	Inductor	0.5 mH
Señal de retorno	15 codificador de valor absoluto multigiros Una vuelta 15 Bits, varias vueltas 9 Bits, total 24 BIIts (32768 pulsos por vuelta)	
Método de Enfriamiento	Enfriamiento natural	
Modo de control de posición	Frecuencia máxima de entrada de pulsos	500 KHz
	Modo de instrucción de pulsos	Pulso + dirección, fase A + fase B
	Relación de transmisión electrónica	Ajustar rango de 1-65535 a 1-65535
	Muestreo de ubicación	2 KHz
Función Protectora	Alarma de Bloqueo	
Interfaz de Comunicación	RS485, (modbus RTU 19200,8,N,1)	
Ambiente de operación	Temperatura Ambiente	0~40°
	Temperatura máxima admisible del motor	85°
	Humedad	5~95%

Fuente: (Manual Anno RV624, 2020)

Realizado por: Bautista A., & Oña D., 2023

3.9.1.4. Servo motor integrado 42AIM15

El servo motor integrado 42AIM15 se sitúa en la articulación 6 (J6) y las especificaciones técnicas se encuentran detalladas en la Tabla 3-7.



Ilustración 3-16: Servo motor integrado 42AIM15

Fuente: (Yizhi, 2020)

Tabla 3-7: Especificaciones técnicas del motor 42AIM15

Fuente de alimentación	Voltaje	24 VDC \pm 10%
	Amperaje	2.2 A
Parámetros del Motor	Esfuerzo de Torsión	0.48 NM
	Velocidad Nominal	1000 RPM
	Velocidad Máxima	1500 RPM
	Potencia	50 W
Señal de retorno	Codificador magnetoelectrónico de 15 bits de una sola vuelta (32768 pulsos por vuelta)	
Método de Enfriamiento	Enfriamiento natural	
Modo de control de posición	Temperatura Ambiente	500 KHz
	Temperatura máxima admisible del motor	Pulso + dirección, fase A + fase B
	Humedad	Ajustar rango de 1-65535 a 1-65535
	Temperatura Ambiente	2 KHz
Función Protectora	Alarma de Bloqueo	
Interfaz de Comunicación	RS485, (modbus RTU 19200,8, N,1)	
Ambiente de operación	Temperatura Ambiente	0~40°
	Temperatura máxima admisible	85°
	Humedad	5~95%

Fuente: (Manual Anno RV624, 2020)

Realizado por: Bautista A., & Oña D., 2023

3.9.2. Controlador integrado

El controlador integrado cuenta con componentes electrónicos dispuestos para operar de manera coordinada y asegurar la transmisión de señales, hacia los actuadores correspondientes. Por ende, esta entidad desempeña un papel central y vital, casi equiparable al cerebro del robot ANNO RV624, los elementos que forman parte del controlador integrado se detallan en la Tabla 3-8.

Tabla 3-8: Especificaciones de elementos de controlador del robot.

ELEMENTO	DETALLE
Controlador de movimiento de seis ejes	Controlador de movimiento diseñado para el control preciso de robots con hasta seis ejes de movimiento, compatible con motores de paso a paso y servomotores. Marca: ANNO. Utilizable con múltiples sistemas operativos.
Fuente de Poder	Fuente de alimentación necesaria para suministrar energía a los componentes del controlador es de 220V.
Placa de Relés	Dispositivo que permite el control simultáneo de varias cargas utilizando un único procesador. Compacta y versátil. Ubicada debajo del regulador de voltaje.
Regulador de Voltaje	Componente que garantiza la estabilidad de voltaje, previniendo picos y fluctuaciones causadas por variaciones en la corriente eléctrica. Asegura el correcto funcionamiento de los elementos.

Placa de Conexión para Puerto Paralelo	Interfaz de conexión que facilita la comunicación entre el controlador y otros dispositivos como la electroválvula mediante un puerto paralelo.
---	---

Fuente: (Mera y Quicaliquin, 2021)

Realizado por: Bautista A., & Oña D., 2023

La Ilustración 3-17 muestra la distribución de componentes esenciales que componen el conjunto controlador del robot ANNO RV624. Este conjunto está diseñado meticulosamente para coordinar y gestionar las operaciones del brazo robótico. Cada elemento desempeña un papel importante en el funcionamiento del sistema.

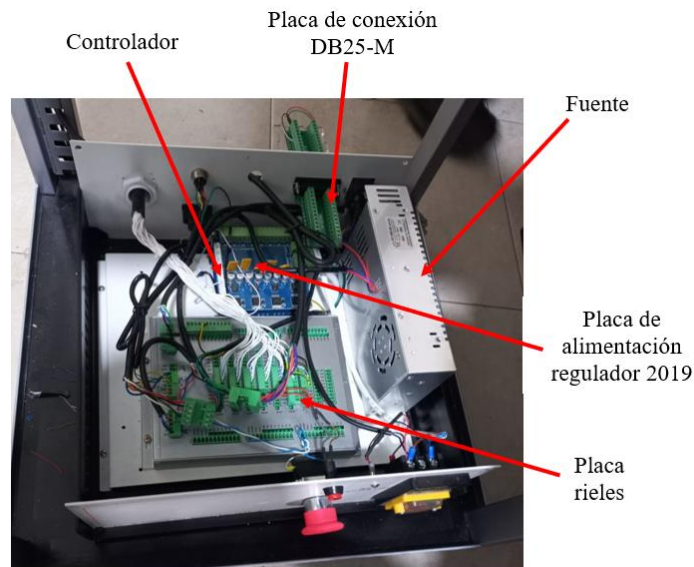


Ilustración 3-17: Elementos del controlado del robot ANNO RV624

Realizado por: Bautista A., & Oña D., 2023

3.9.2.1. Mini bomba y electroválvulas

La electroválvula junto con la mini bomba desempeña un papel crucial al posibilitar la operación precisa del Gripper, permitiéndole sujetar objetos de manera efectiva. Estos dos componentes trabajan en conjunto para garantizar que el proceso de agarre se lleve a cabo de manera controlada y eficiente.

La electroválvula 5/2-1/4, en este contexto, actúa como un interruptor controlado eléctricamente que regula el flujo de aire o fluido hidráulico hacia la mini bomba. Cuando se activa, permite que el fluido sea dirigido a la mini bomba, lo que a su vez activa el sistema y genera la fuerza necesaria para que el Gripper realice el agarre. Esta activación controlada es esencial para asegurar que el objeto se sujete con la cantidad justa de fuerza y seguridad, evitando daños o problemas de manipulación.

La mini bomba, por su parte, es responsable de generar la presión necesaria para operar el sistema de agarre. Su diseño compacto y eficiente le permite producir la fuerza requerida para que el Gripper funcione correctamente. Al trabajar en conjunto con la electroválvula, la mini bomba garantiza que la operación de sujeción se ajuste a las necesidades específicas del objeto a agarrar, proporcionando un control preciso y fiable sobre el proceso.



Ilustración 3-18: Electroválvula

Realizado por: Bautista A., & Oña D., 2023

3.9.3. *Sensor de señales EMG*

Es posible obtener señales electromiográficas (EMG) para el análisis del movimiento mediante el empleo de electrodos, los cuales pueden ser aplicados tanto en la superficie de la piel como intramusculares, generalmente en configuraciones bipolares. La amplitud y la banda de frecuencia de la señal EMG no están exclusivamente determinados por las fuentes electrofisiológicas y sus distancias respecto a los electrodos, sino que también se ven influenciados por las características y dimensiones de los electrodos utilizados. Por consiguiente, para el presente proyecto, se ha considerado realizar una selección de sensores que permitan captar señales electromiográficas.

3.9.3.1. *Selección de sensores EMG*

Para la selección de los sensores se ha considerado tres tipos de sensores de superficie que se colocan directamente en la región muscular. Estos electrodos superficiales permiten investigar la actividad de la totalidad del músculo que se encuentra en la superficie, la separación entre los electrodos incide en el volumen de la captación de la señal del tejido, presentando una relación directa entre separaciones más reducidas y una captación más específica. Por lo tanto, en la Tabla 3-9, se presenta la comparación entre el sensor muscular MYOWARE AT-04-001, el MYO Armband y el sensor muscular ECG EMG 3,3V AD8832, con el propósito de evaluar y contrastar las características distintivas de estos dispositivos realizando una revisión de los aspectos como

la naturaleza de las señales capturadas, conectividad, compatibilidad con software, el diseño y otras especificaciones técnicas.

Tabla 3-9: Tabla de características de los sensores

CARACTERISTICAS	MYOWARE AT-04-001	MYO ARMBAND	SENSOR MUSCULAR ECG EMG 3,3V AD8832
Captura de Señales	EMG	EMG y Movimiento	ECG y EMG
Conexión a Placas de Desarrollo	Posiblemente compatible con Arduino	Compatible con Arduino	No especificado
Interfaz de Salida	Señales EMG (Voltajes o digitales)	Señales EMG y de movimiento	Señales ECG y EMG
Electrodos	Requiere electrodos	No especificado	No especificado
Diseño	Compacto y posiblemente ergonómico	Ergonómico para antebrazo	No especificado
Conectividad	No especificado	Inalámbrica	No especificado
Compatibilidad con Software	No especificado	Posiblemente compatible con software específico y plataformas de desarrollo	No especificado
Alimentación	3,1V – 6,3V	5V (USB)	3,3V
Amplificación y Filtrado Interno	No especificado	No especificado	Posiblemente incluido
Aplicaciones	No especificado	Control gestual, aplicaciones biomédicas	Aplicaciones biomédicas

Fuente: (Electrostore, 2019; Copaci et al., 2022)

Realizado por: Bautista A., & Oña D., 2023

La comparación de los tres dispositivos permitió discernir las fortalezas y limitaciones de cada dispositivo, proporcionando una base para la selección adecuada según las necesidades específicas de aplicación, donde el MYO Armband se destaca por ser un sensor versátil para aplicaciones que van desde la realidad virtual hasta el control de dispositivos electrónicos mediante gestos y movimientos musculares. Además, su integración de sensores EMG, giroscopios, acelerómetros y magnetómetros proporciona una visión completa de los movimientos y permite una interacción fluida con el dispositivo. El hecho de ser un dispositivo portátil y fácil de usar bajo diversos sistemas operativos lo convierte en una opción atractiva para aplicaciones de control.

El brazalete MYO es un dispositivo wearable desarrollado por la empresa Thalmic Labs (ahora parte de North) que permite controlar dispositivos electrónicos a través de gestos y señales musculares. La producción se discontinúa en el año 2020, por ello, se suspenden las ventas y el soporte del brazalete MYO, sin embargo, se cuenta con dos brazaletes MYO adquiridos dentro

del mercado nacional, debido a sus múltiples bondades, tales como, el control intuitivo y la personalización de las aplicaciones.

3.9.3.2. Brazaletes MYO

Una vez seleccionado el brazalete MYO Armband como dispositivo para la adquisición de las señales electromiográficas requeridas, debido a su capacidad para detectar señales eléctricas generadas por la musculatura del antebrazo, esta tecnología se emplea con el fin de ofrecer un control inalámbrico a otros dispositivos, en este caso, al brazo robótico ANNO RV624.

Otro motivo de seleccionar el MYO Armband radica en la compatibilidad con el sistema operativo del brazo robótico, además de su capacidad de lograr una comunicación inalámbrica a través de la tecnología Bluetooth. Destacando aún más su funcionalidad, ya que incorpora un SDK (Software Development Kit) que simplifica significativamente la gestión de la comunicación entre el brazalete y el software MyoConnect, Este kit incluye librerías, documentación y controladores esenciales. Las características del Brazaletes MYO Armband se presentan en la Tabla 3-10.



Ilustración 3-19: Brazaletes MYO Armband

Fuente: (Rodríguez, 2021)

Tabla 3-10: Especificaciones del brazalete MYO Armband

CARACTERÍSTICAS	DESCRIPCIÓN
Sensores Médicos EMG (8 sensores)	Frecuencia de muestreo: 200 Hz
IMU de 9 Ejes	Frecuencia de muestreo: 50 Hz
- Giroscopio de 3 Ejes	
- Acelerómetro de 3 Ejes	
- Magnetómetro de 3 Ejes	
Indicador LED	Proporciona retroalimentación visual

Procesador ARM de 1200 MHz	Potente procesador para el procesamiento
Retroalimentación Háptica	Proporciona retroalimentación táctil
Conexión Bluetooth	Comunicación inalámbrica
Batería Recargable de Litio	Fuente de energía recargable

Fuente: (Cedeño, 2017)

Realizado por: Bautista A., & Oña D., 2023

El MYO posee dos indicadores LED que brindan información sobre el estado de la batería del brazalete, además los indicadores ofrecen una manera de identificar el estado de conexión del brazalete, como se detalla en la Tabla 3-11.

Tabla 3-11: Indicadores de conexión del brazalete

ED LOGO THALMIC	LED ESTADO	ESTADO DE CONEXIÓN
Constante azul	Constante azul	Bluetooth conectado y sincronización realizada
Constante azul	Pulso azul	Bluetooth desconectado y sincronización realizada
Pulso azul	Apagado	Bluetooth desconectado y sin sincronización (Modo reposo)
Apagado	Apagado	Modo suspensión (Activación por movimiento)
Destello azul	Constante azul	Modo calentando (MYO vibrará al terminar)

Fuente: (Cedeño, 2017)

Realizado por: Bautista A., & Oña D., 2023

El brazalete Myo, opera por comunicación inalámbrica Bluetooth. Para este proyecto se utilizó dos brazaletes, uno de ellos opera normalmente con comunicación bluetooth, pero en el caso, del segundo dispositivo no opera por bluetooth, por ello, se realiza una adaptación de comunicación por medio de medio de una tarjeta HM-11 y codificación con Arduino para la recepción de las señales EMG.

3.9.4. Computadora portátil

Como unidad de procesamiento se utilizó un laptop Dell Inspiron 15-5576, misma que procesó el algoritmo y la información adquirida por el MYO. A continuación, en la Tabla 3-12 se presenta un resumen de las principales especificaciones y características de esta laptop, desde su procesador y memoria hasta su capacidad gráfica, conectividad y mucho más.



Ilustración 3-20: Dell Inspiron 15-5576

Fuente: (Laptomedia, 2019)

Tabla 3-12: Especificaciones del equipo

CARACTERÍSTICA	DESCRIPCIÓN
Modelo del equipo	Inspiron 15-5576
Procesador	AMD FX 9830P RADEON R7, 12 COMPUTE CORES 4C+8G
Conjunto de chips	Integrado en el procesador
Disco Duro	Unidad de disco duro hasta 1 TB + SSD de hasta 256 GB
RAM	8 GB
Sistema operativo	Windows 10 Home 64 bits
BIOS	1.07
Pantalla	Pantalla no táctil FHD de 15,6 pulgadas
Resolución (máxima)	1920 x 1080
Cámara	1280 x 720 (HD) a 30 fps
Audio	Realtek ALC3246 con Waves MaxxAudio
Conexión inalámbrica	Wi-Fi 802.11ac - Bluetooth 4.1
Puertos y Unidad óptica	HDMI / BT / USB 3 puertos 3.0 / RJ45 / SD / Audio y Mic
Velocidad	2400 MHz
Peso	2,57 kg

Fuente: (Dell, 2019)

Realizado por: Bautista A., & Oña D., 2023

3.9.5. Dispositivo HM-11

El dispositivo HM-11 funciona mediante el microcontrolador CC2541, el cual integra la avanzada tecnología Bluetooth de Baja Energía (BLE). Esta característica permite establecer conexiones eficientes con una amplia variedad de dispositivos, como PCs, smartphones y otros dispositivos compatibles con Bluetooth V4.0.

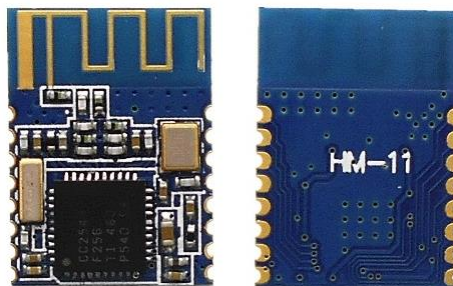


Ilustración 3-21: Módulo Bluetooth HM-11

Fuente: (TURIBOT, 2023)

Un aspecto destacado de este dispositivo es su total personalización a través de comandos AT, lo que brinda la flexibilidad necesaria para ajustar su funcionamiento de acuerdo con requisitos específicos. Es posible modificar parámetros clave, como la velocidad de comunicación, el nombre asignado al dispositivo e incluso la potencia de transmisión. En la Tabla 3-13, se detalla las especificaciones de la tarjeta HM-11.

Tabla 3-13: Especificaciones de la tarjeta HM-11

PARÁMETRO	REFERENCIA
Fuente de alimentación:	3.3 VDC 50 mA
Microprocesador:	CC2541
Frecuencia de trabajo:	Banda ISM de 2,4 GHz
Modulación:	GFSK (modulación por desplazamiento de frecuencia gaussiana)
Protocolo de comunicación:	UART (TTL)
Potencia transmitida:	-23 DBM, 6 DBM, 0 DBM, 6 DBM, se puede cambiar a través de comandos AT
Sensibilidad:	-84 DBM BER a 0.1%
Velocidad de transmisión:	Asíncrono/ Síncrono: 6 KBPS

Fuente: (TURIBOT, 2023)

Realizado por: Bautista A., & Oña D., 2023

3.10. Cinemática directa del brazo robótico

Con el propósito de llevar a cabo el cálculo cinemático, se efectuó un análisis del movimiento del brazo robótico con relación a un sistema de coordenadas ubicado en su base por lo que se empleó la cinemática directa, la cual permite determinar la posición y orientación del efector final del brazo robótico en relación con el sistema de coordenadas de referencia. En este contexto, se seleccionó el método de Denavit-Hartenberg como algoritmo para obtener una solución.



Ilustración 3-22: Brazo ANNO RV624

Realizado por: Bautista A., & Oña D., 2023

El método de Denavit-Hartenberg permitió la conexión entre el sistema de referencia del elemento i y el sistema de referencia del elemento $i-1$ a través de cuatro transformaciones fundamentales. A fin de obtener la cinemática directa, se detalla a continuación mediante los siguientes pasos:

1. Localizar y asignar tanto los eslabones como las articulaciones del brazo robótico.

En la Ilustración 3-23 se puede observar los seis eslabones (l_i) y las seis articulaciones (q_i), donde q_1 pertenece a una proyección del brazo sobre un plano horizontal, q_2 sobre el eslabón l_2 , q_3 se obtiene de la proyección del eslabón l_2 hacia l_3 , q_4 tiene un movimiento rotacional, dando movimiento a los dos últimos eslabones, q_5 sobre un plano formado sobre l_5 y finalmente al mover q_6 solamente se acciona la herramienta.

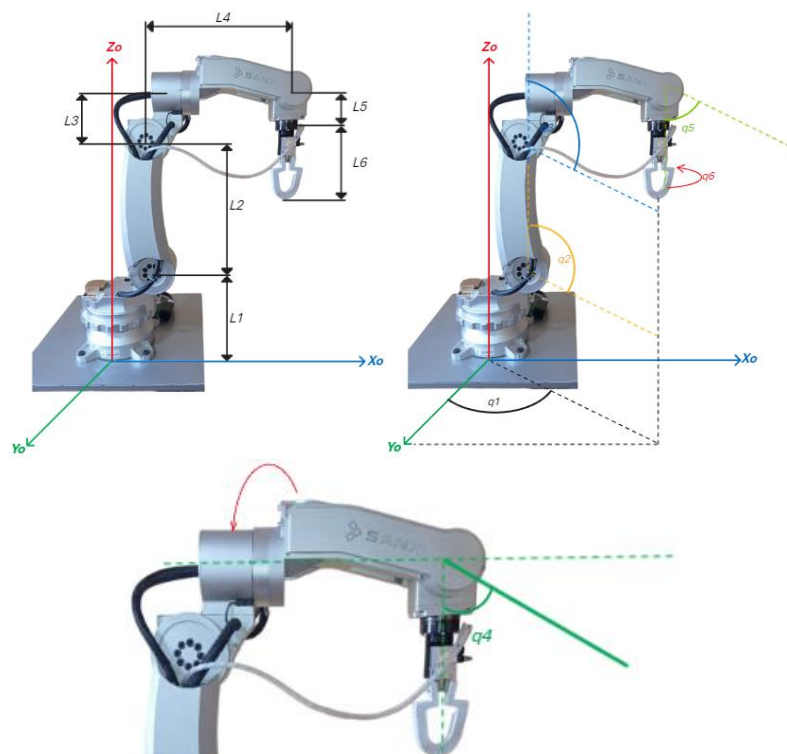


Ilustración 3-23: Asignación de eslabones y articulaciones

Realizado por: Bautista A., & Oña D., 2023

2. Se asignó el eje Z colineal a cada eje de cada articulación.

En la Ilustración 3-24 se muestra la designación del eje para z cada una de las articulaciones ubicadas anteriormente, siendo así, z_0 sobre el eje z, el z_1 , z_2 , z_3 sobre la articulación q_2 , q_3 y q_4 respectivamente, se observa z_4 colineal a la articulación q_5 y colineal a la articulación q_5 se tiene z_5 . Por último, se observa intersectarse a los ejes z_3 , z_4 y z_5 en un solo punto y z_6 se encuentra en la misma dirección que z_5 .

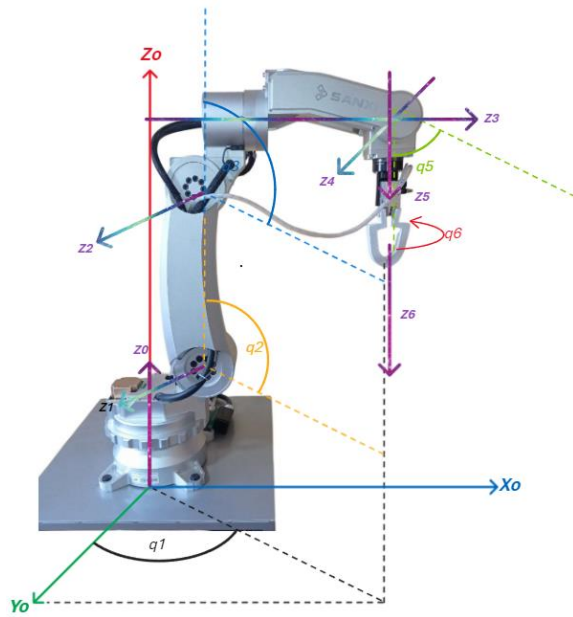


Ilustración 3-24: Asignación del eje Z_i

Realizado por: Bautista A., & Oña D., 2023

3. En el siguiente paso se asignó el origen a cada trama, dependiendo de cada caso.

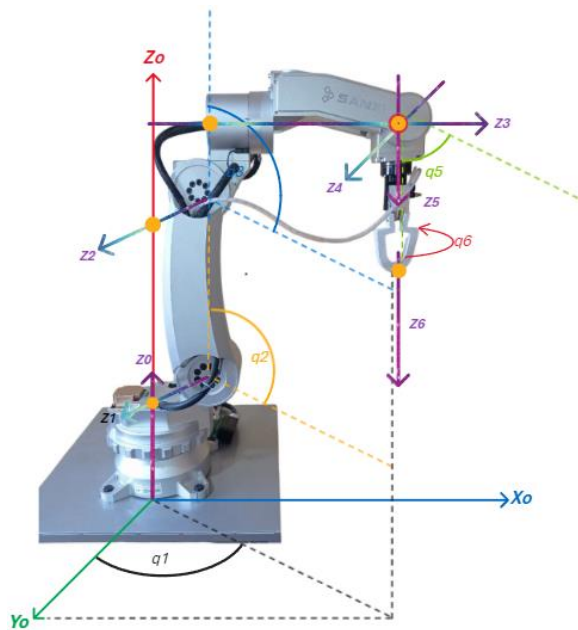
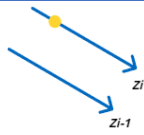
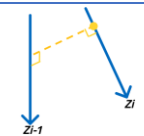


Ilustración 3-25: Asignación de los puntos de origen

Realizado por: Bautista A., & Oña D., 2023

Tabla 3-14: Casos para la asignación del origen

	DESCRIPCIÓN	INTERSECCIÓN
CASO 1	En el punto de intersección entre z_i y z_{i-1}	

CASO 2	En cualquier punto a lo largo de z_i se situará en la articulación $i+1$.	
CASO 3	En la intersección entre z_i y la línea perpendicular que lo une a z_{i-1} .	

Fuente: (Ding y Liu, 2018)

Realizado por: Bautista A., & Oña D., 2023

Como se observa en la Ilustración 3-25 se asignó el origen para z_0 , en la trama z_0-z_1 se intersecan a lo largo de z_1 correspondiendo al primer caso, en la siguiente trama de z_1-z_2 se observa que los dos ejes son paralelos, por ende, se coloca el origen en cualquier punto a lo largo de z_2 , es recomendable colocarlo en el centro de la articulación, posteriormente en la trama entre z_2-z_3 se intersecan en este mismo punto y su origen coincide con el anterior, continuando con z_3-z_4 y z_4-z_5 nuevamente se observa el caso número uno, para la trama entre z_5-z_6 se observa que estos dos ejes son colineales y se asigna el origen en cualquier punto a lo largo de z_6 .

4. Continuando con el algoritmo de Denavit-Hartenberg se asignó los ejes X , dependiendo de cada caso.

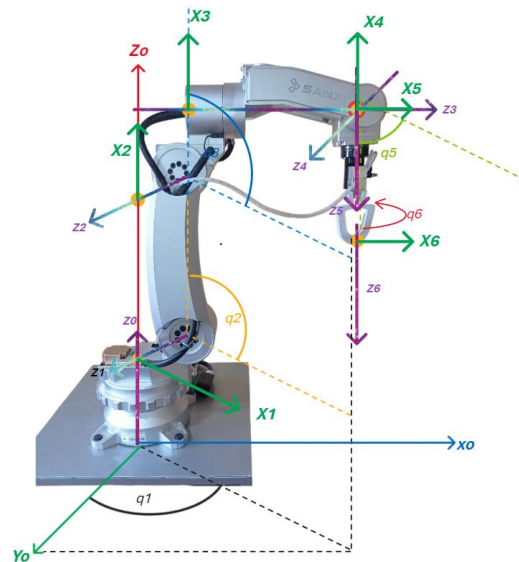



Ilustración 3-26: Diagrama completo de Denavit-Hartenberg

Realizado por: Bautista A., & Oña D., 2023

Tabla 3-15: Casos para la asignación del eje X

	DESCRIPCIÓN	INTERSECCIÓN
CASO 1	X_i será perpendicular al plano formado por Z_i y Z_{i-1}	

CASO 2	X_i será perpendicular a Z_i y a Z_{i-1}	
CASO 3	X_i será colineal con la única línea perpendicular a Z_i y Z_{i-1}	

Fuente: (Ding y Liu, 2018)

Realizado por: Bautista A., & Oña D., 2023

La Ilustración 3-16 muestra la asignación del eje X , donde el eje X_0 es la trama principal, el eje X_1 se colocó perpendicular al plano formado entre el Z_0-Z_1 , para Z_1-Z_2 se presenta el segundo caso por lo tanto se colocó el eje X_2 perpendicular a ambos, para el eje X_3 se considera el plano formado por Z_2-Z_3 , para eje X_4 se tomó en cuenta el primer caso donde Z_3-Z_4 forman un plano de tal manera que se coloca a partir del origen, el eje X_5 se colocó perpendicular al plano formado por la trama Z_4-Z_5 y finalmente para colocar X_6 , se observa que Z_5-Z_6 son colineales por lo cual, el X_6 es perpendicular a ambos Z , por otra parte el método de Denavit-Hartenberg se basa en la observación de las tramas con los ejes X y Z .

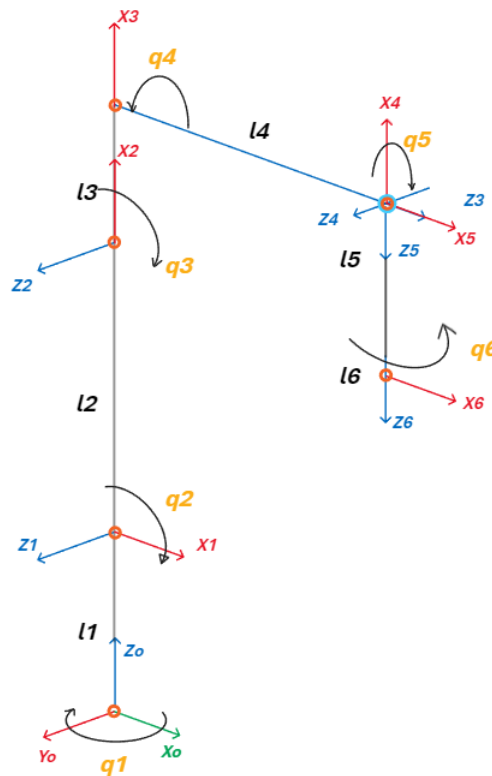


Ilustración 3-27: Diagrama de cuerpo libre del brazo robótico

Realizado por: Bautista A., & Oña D., 2023

5. Se procede a colocar cada uno de los parámetros obtenidos anteriormente en la tabla de Denavit-Hartenberg.

Tabla 3-16: Tabla de Denavit-Hartenberg

TRAMA	θ_x	d_x	a_z	α_z
0-1	q_1	l_1	0	$\pi/2$
1-2	$90+q_2$	0	l_2	0
2-3	q_3	0	l_3	$\pi/2$
3-4	q_4	l_4	0	$-\pi/2$
4-5	$q_5-\pi/2$	0	0	$\pi/2$
5-6	q_6	l_5+l_6	0	0

Realizado por: Bautista A., & Oña D., 2023

En la Tabla 3-16 se detallan los valores para teta “ θ ” ángulo de rotación en el eje x, “ d ” de traslación en el eje x, “ a ” de traslación en el eje z y alfa “ α ” ángulo de rotación en el eje z, es decir se observa si existe un ángulo y distancia entre los ejes x sobre el eje z anterior, como también si existe alguna distancia y ángulo entre los ejes z sobre el x actual, cabe recalcar que cada una de las filas corresponde a una matriz de Denavit-Hartenberg.

6. Por último, se obtuvo las matrices de Denavit-Hartenberg.

$$H = \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Para obtener las matrices de Denavit-Hartenberg, se utilizó la matriz (H) como fórmula, donde se sustituyen los valores determinados en los pasos anteriores, así también se mencionó que cada fila de la Tabla 3-16 corresponde a una matriz, por ende, en este caso se obtiene 6 matrices.

$$H_1^0 = \begin{bmatrix} \cos(q_1) & 0 & \sin(q_1) & 0 \\ \sin(q_1) & 0 & -\cos(q_1) & 0 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_2^1 = \begin{bmatrix} -\sin(q_2) & \cos(q_2) & 0 & -l_2 \sin(q_2) \\ \cos(q_2) & -\sin(q_2) & 0 & l_2 \cos(q_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_3^2 = \begin{bmatrix} \cos(q_3) & 0 & \sin(q_3) & l_3 \cos(q_3) \\ \sin(q_3) & 0 & -\cos(q_3) & l_3 \sin(q_3) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_4^3 = \begin{bmatrix} \cos(q_4) & 0 & -\sin(q_4) & 0 \\ \sin(q_4) & 0 & \cos(q_4) & 0 \\ 0 & -1 & 0 & l_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_5^4 = \begin{bmatrix} \sin(q_5) & 0 & -\cos(q_5) & 0 \\ -\cos(q_5) & 0 & -\sin(q_5) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_6^5 = \begin{bmatrix} \cos(q_6) & -\sin(q_6) & 0 & 0 \\ \sin(q_6) & \cos(q_6) & 0 & 0 \\ 0 & 0 & 1 & l_5 + l_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

La matriz resultante es una matriz de 4x4, obteniendo valores de posición en (x, y, z) que pertenecen a la posición final del “gripper” del brazo robótico y se hallan mediante la siguiente fórmula:

$$H_6^0 = H_1^0 * H_2^1 * H_3^2 * H_4^3 * H_5^4 * H_6^5$$

En la multiplicación de matrices se sustituyeron las variables asociadas a las longitudes de los eslabones (l_i) y los ángulos de las articulaciones (q_i). Como resultado de esta sustitución, se obtuvo la matriz resultante que representa la transformación compuesta de los diferentes eslabones y sus respectivas articulaciones, obteniendo así la siguiente matriz:

$l_1=240$	$q_1=0$
$l_2=350$	$q_2=0$
$l_3=120$	$q_3=0$
$l_4=355$	$q_4=0$
$l_5=72$	$q_5=0$
$l_6=195$	$q_6=0$

$$H_6^0 = \begin{bmatrix} 1 & 0 & 0 & 355 \\ 0 & 0 & 0 & 209,5 \\ 0 & 0 & 0 & 433 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A partir de la matriz resultante, podemos concluir que los valores de los ejes x, y, z toman valores de 355, 209.5 y 433 respectivamente, además, los ángulos A, B y C son determinados como 0 ya se ha evaluado en la posición inicial y se hallaron mediante la aplicación de las siguientes ecuaciones:

$$A = \arctan \left(\frac{h_{21}}{h_{11}} \right)$$

$$B = \arctan \left(\frac{-h_{31}}{\sqrt{h_{32} * h_{32} + h_{33} * h_{33}}} \right)$$

$$C = \arctan \left(\frac{h_{32}}{h_{33}} \right)$$

3.11. Software

Para cumplir con los objetivos que abarcan desde el desarrollo de una aplicación informática para calibrar el sistema de señales electromiográficas hasta el desarrollo de un algoritmo para la adquisición y procesamiento inalámbrico, se hacen necesarios los siguientes requerimientos.

3.11.1. Python

La Tabla 3-17 comparativa de lenguajes de programación reveló que Python emerge como la elección preferida para implementar el sistema, gracias a sus notables fortalezas. Una de las razones primordiales es la rica disponibilidad de librerías que ofrecen soluciones y funcionalidades específicas. Estas librerías, en constante evolución conforme avanza la tecnología, son un factor determinante en la elección de Python como base para el proyecto.

El uso de librerías en Python simplificó y agilizó el desarrollo, permitiendo aprovechar funcionalidades preconstruidas sin la necesidad de reinventar la rueda. Esto se torna esencial en casos como la comunicación con el brazalete MYO, ya que la existencia de varias librerías dedicadas a este propósito brinda un enfoque más directo y eficiente.

Python no solo garantiza una curva de aprendizaje suave, sino que también abre la puerta a la exploración y experimentación en la interacción con el brazalete MYO. A medida que las librerías se actualizan y optimizan, el desarrollo se beneficia de mejoras continuas, alineándose con los avances tecnológicos.

Tabla 3-17: Comparación de lenguajes de programación

LENGUAJE DE PROGRAMACIÓN	CARACTERÍSTICAS	FORTALEZA	DEBILIDAD
C++	<ul style="list-style-type: none"> • Orientada a objetos • Rápido 	<ul style="list-style-type: none"> • Ideal para sistemas robustos 	<ul style="list-style-type: none"> • No soporta creación de aplicaciones web • No es atractivo visualmente

C	<ul style="list-style-type: none"> • Popular para la creación de software de sistema 	<ul style="list-style-type: none"> • Orientado a la implementación de sistemas operativos Unix • Rápido • Eficiente 	<ul style="list-style-type: none"> • Sintaxis compleja • No es popular para la creación de aplicación
C#	<ul style="list-style-type: none"> • Orientado a objetos • Estandarizado por Microsoft como parte de su plataforma Net. 	<ul style="list-style-type: none"> • Se desempeña de forma plena en los sistemas operativos Windows. • Posibilidad de realizar aplicaciones web, de escritorio y móviles. 	<ul style="list-style-type: none"> • Requiere un mínimo de 4 gb para su instalación. • Contar con Windows NT 4 o superior,
Python	<ul style="list-style-type: none"> • Permite la creación de todo tipo de programas incluso sitios web • No requiere de compilación es un código interpretado 	<ul style="list-style-type: none"> • Libre y código fuente abierto • Lenguaje de propósito general • Multiplataforma • Orientado a objetos • IA y aprendizaje automático • Sintaxis clara y fácil de leer • Abundantes bibliotecas y frameworks 	<ul style="list-style-type: none"> • Los lenguajes interpretados pueden ser un poco lentos
Ruby	<ul style="list-style-type: none"> • Orientado a objetos • Lenguaje de alto nivel • Sintaxis similar a Python y Perl • Opensource 	<ul style="list-style-type: none"> • Diferencia entre mayúsculas y minúsculas • Puede cargar librerías si el sistema operativo lo permite • Multiplataforma 	<ul style="list-style-type: none"> • Es relativamente nuevo y no cuenta con mucha documentación • No está muy difundido con relación a otros lenguajes.
Java	<ul style="list-style-type: none"> • Es orientado a objetos • Multiplataforma 	<ul style="list-style-type: none"> • Al ser orientado a objetos permite su modularización. • Creación de aplicaciones de escritorio. • Tiene soporte a desarrollo de aplicaciones móviles y web. 	<ul style="list-style-type: none"> • Es un lenguaje interpretado por lo que es lento en comparación con otros lenguajes.
PHP	<ul style="list-style-type: none"> • Genera páginas web dinámicas • No se puede ver el código PHP únicamente reciben en sus navegadores código HTML • Lenguaje de alto nivel 	<ul style="list-style-type: none"> • Su sintaxis es muy similar a otros lenguajes • Multiplataforma • Tiene frameworks que facilitan el desarrollo en este lenguaje. • Fácil y rápido 	<ul style="list-style-type: none"> • Necesita un servidor para funcionar • La programación orientada a objetos es deficiente para aplicaciones grandes. • Todo el trabajo se realiza el en servidor y mucha información o solicitudes pueden ser ineficiente.

Fuente: (García, 2021)

Realizado por: Bautista A., & Oña D., 2023

Se eligió Python ya que es altamente versátil y se adapta a una amplia gama de aplicaciones, desde desarrollo web hasta el análisis de datos y automatización, esto debido a que es un lenguaje efectivo para la enseñanza de programación y se destaca en campos como la inteligencia artificial y el aprendizaje automático. Su naturaleza interpretada, compatibilidad multiplataforma consolidan a Python como una opción adecuada para llevar a cabo dicha propuesta tecnológica.

3.11.2. Pycharm

PyCharm es un software complementario que se utilizó como herramienta para mejorar la eficiencia del código generado en Python, como se puede apreciar en la Tabla 3-18, PyCharm ofrece un conjunto completo de herramientas y características diseñadas específicamente para agilizar el proceso de desarrollo eficiente en Python. Con un soporte integral para el lenguaje, análisis inteligente de código, autocompletado avanzado, depuración integrada y un sólido control de versiones, PyCharm simplifica tanto la creación como el mantenimiento de proyectos. Además de su integración extensa con frameworks web y bases de datos, así como su enfoque en el desarrollo web, brinda una versatilidad inigualable. Añadiendo a esto su comunidad activa, opciones de personalización y configuración flexibles, PyCharm se eleva como un entorno de desarrollo integrado (IDE) robusto y altamente efectivo que permite elevar la calidad de su código en Python.

Tabla 3-18: Características de PyCharm

CARACTERÍSTICA	PYCHARM
Precio	Gratuito para Community, de pago para Professional
Desarrollo Python	Soporte completo
Desarrollo web (HTML, CSS, JS)	Soporte completo
Control de versiones (Git, etc.)	Soporte completo
Autocompletado	Sí
Depuración	Sí
Refactorización	Sí
Pruebas unitarias	Sí
Análisis de código estático	Sí
Integración con frameworks	Amplio soporte a través de plugins
Soporte de bases de datos	Amplio soporte
Desarrollo científico	No es el enfoque principal
Soporte para lenguajes adicionales	Alguno (según edición)
Personalización	Amplias opciones de configuración

Fuente: (Datascientest, 2023)

Realizado por: Bautista A., & Oña D., 2023

3.11.2.1. QT Designer

Qt Designer es una herramienta gráfica dentro del framework Qt que permitió diseñar interfaces de usuario de manera visual, esto simplificó la creación de ventanas, botones y elementos interactivos en aplicaciones al generar automáticamente el código correspondiente. Como

resultado, agilizó el desarrollo y disminuye el esfuerzo necesario para crear aplicaciones con interfaces atractivas y funcionales. En la Tabla 3-19 se detallan las ventajas de Qt Designer.

Tabla 3-19: Ventajas del Qt Designer

VENTAJAS	DESCRIPCIÓN
Diseño visual intuitivo	Permite diseñar interfaces gráficas de usuario de manera visual, facilitando la creación.
Generación automática de código	Genera automáticamente el código correspondiente a la interfaz diseñada, ahorrando tiempo.
Componentes predefinidos	Ofrece una amplia variedad de componentes listos para usar, como botones, ventanas y más.
Personalización	Permite ajustar el diseño y estilo de los elementos para adaptarse a la identidad de la app.
Agiliza el desarrollo	Simplifica y acelera la creación de interfaces, lo que permite enfocarse en la lógica de la app.
Integración con el framework Qt	Se integra perfectamente con el framework Qt, facilitando la construcción de aplicaciones.
Reducción de errores de diseño	Al visualizar la interfaz antes de implementarla, se reducen los errores y ajustes posteriores.
Actualización dinámica de la interfaz	Permite ver los cambios en tiempo real, facilitando la iteración y mejora del diseño.
Compatibilidad multiplataforma	Las interfaces diseñadas en Qt Designer son compatibles con diferentes sistemas operativos.
Mejora la colaboración entre desarrolladores	Facilita la comunicación entre diseñadores y desarrolladores al visualizar la interfaz.

Fuente: (Herrmann, 2023)

Realizado por: Bautista A., & Oña D., 2023

3.11.3. MYO connect

Otro requisito importante es el MYO connect, este software actúa como un intermediario fundamental entre el brazalete Myo y las aplicaciones en una computadora. Su función es simplificar y optimizar el proceso de enlace del brazalete con el dispositivo, permitiendo una calibración precisa para detectar movimientos individuales, así como la configuración de gestos personalizados que controlan diversas funciones en aplicaciones compatibles.

Además, este software proporciona actualizaciones de firmware para el brazalete, lo que garantiza su óptimo rendimiento y funcionalidad continua. Es esencial para establecer la conexión entre el brazalete y el dispositivo a través de Bluetooth, como se muestra en la Ilustración 3-28.

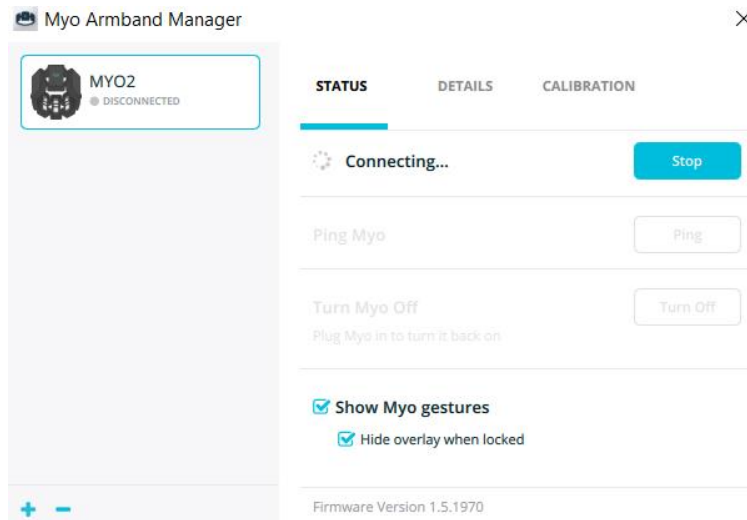


Ilustración 3-28: Conexión entre el brazalete y el dispositivo

Realizado por: Bautista A., & Oña D., 2023

Este software despliega una etapa de calibración exclusiva, que permite ajustar y controlar diversos aspectos y tareas de la computadora mediante movimientos específicos, tal como se observa en la Ilustración 3-29. Esta etapa de calibración es esencial para asegurar la precisión y sensibilidad en la detección de los gestos realizados con el brazalete. Además, la herramienta presenta una gama de 5 gestos principales predefinidos que los usuarios pueden emplear para interactuar con las aplicaciones y funciones del sistema. Estos gestos se han diseñado para proporcionar una experiencia fluida y cómoda, aumentando la eficiencia en la interacción entre el usuario y la tecnología.

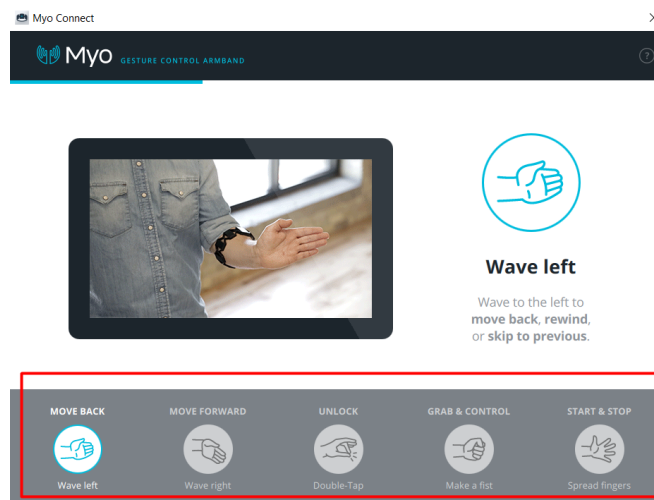


Ilustración 3-29: Opciones de selección de trabajo

Realizado por: Bautista A., & Oña D., 2023

Pasos para conectar MYO Connect a Bluetooth:

1. Se inicia la aplicación MYO Connect Manager asegurándose de que el dispositivo Bluetooth esté conectado previamente a la computadora.

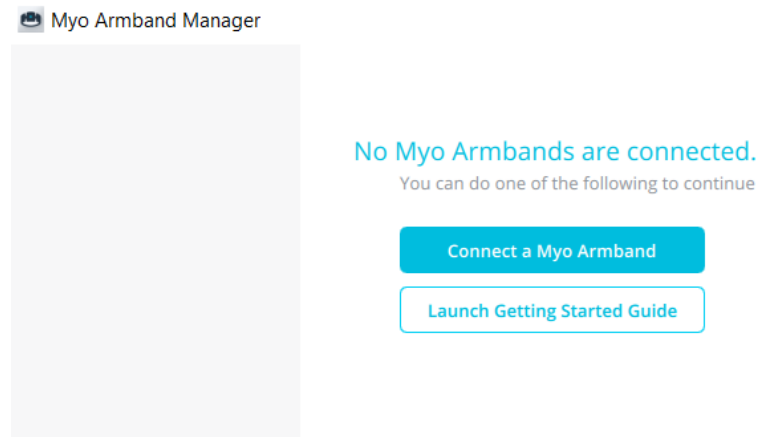


Ilustración 3-30: Proceso de conexión MYO a Bluetooth

Realizado por: Bautista A., & Oña D., 2023

2. Una vez dentro de la aplicación, se sigue los pasos para crear un nuevo dispositivo. Este proceso permite establecer la comunicación entre el brazalete Myo y su computadora a través de Bluetooth.

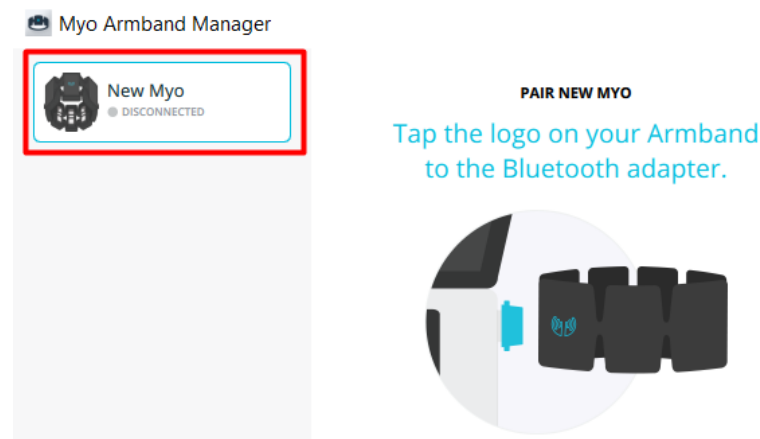


Ilustración 3-31: Crear un nuevo dispositivo

Realizado por: Bautista A., & Oña D., 2023

3. Durante este proceso de configuración, es posible que se solicite habilitar la función Bluetooth en el ordenador y buscar los dispositivos disponibles, para luego seleccionar el brazalete Myo de la lista de dispositivos detectados.

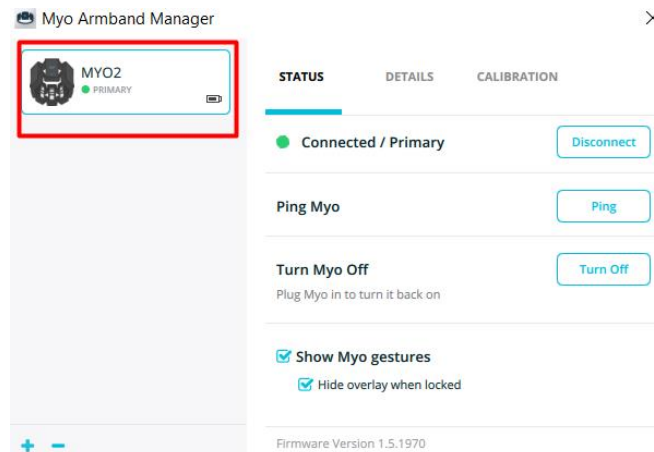


Ilustración 3-32: Emparejar Bluetooth

Realizado por: Bautista A., & Oña D., 2023

- Una vez seleccionado el brazalete Myo, la aplicación MYO Connect Manager inicia automáticamente el proceso de emparejamiento. Este proceso permite que el brazalete se conecte al Bluetooth de la computadora.

Durante el proceso de emparejamiento, el brazalete Myo y el ordenador intercambiarán la información necesaria para establecer una conexión segura y estable. Una vez completado el emparejamiento, la aplicación MYO Connect Manager indica que el brazalete Myo está conectado exitosamente a través de Bluetooth, lo que permitirá utilizarlo con aplicaciones compatibles.

3.12. Configuración del sistema

Para el control del brazo robótico es necesario establecer la comunicación y la interpretación de las señales electromiográficas, esto incluye la selección y configuración de los componentes hardware y software necesarios, como los sensores de EMG, el procesador de señales, el sistema de comunicación y el software de control.

3.12.1. Conexiones del Hardware

En este contexto, con el robot ANNO RV624 funcionando adecuadamente se buscó su control de movimientos de forma inalámbrica por medio de señales EMG, para esto se establecen parámetros esenciales, específicamente como lo son los límites angulares, que definen los rangos extremos de ángulos que cada eslabón puede adoptar. Estos valores, tanto los máximos como los mínimos, se encuentran detallados en la Tabla 3-20 es relevante mencionar que el robot está equipado con un sistema de protección para sus motores, basado en la implementación de sensores inductivos,

sensores que cumplen la función de interrumpir la operación del robot en el caso de que el ángulo de giro transmitido a través de la comunicación serial supere el ángulo máximo permisible para su funcionamiento.

Tabla 3-20: Máximos y mínimos de ángulo de giro

ESLABÓN	MÍNIMO	MÁXIMO
J1	-90	90
J2	-80	50
J3	-150	0
J4	-90	90
J5	-135	-45
J6	-90	90

Realizado por: Bautista A., & Oña D., 2023

Además, considerando la elección de sensores de señales EMG anteriormente realizada, se utilizarán dos dispositivos MYO Armband, uno en el antebrazo derecho y otro en el izquierdo, por ende, es imprescindible mencionar que uno de los brazaletes no dispone de conectividad bluetooth, lo que implica la necesidad de establecer una comunicación alternativa entre el sensor EMG y un microcontrolador.

3.12.1.1. *Comunicación alternativa de sensor EMG – Microcontrolador*

La comunicación de los brazaletes MYO, se establece a través de Bluetooth con la computadora, sin embargo, para el brazaletes derecho, se ha optado por utilizar un microcontrolador ARDUINO como receptor de las señales EMG provenientes del brazaletes. Este microcontrolador establecerá la comunicación a través de un dispositivo Bluetooth capaz de conectarse al brazaletes, por lo que se ha seleccionado el módulo HM-11 considerando las características de la Tabla 3-13.

Dada la limitación en el conjunto de características ofrecidas por la mayoría de los modelos Bluetooth, que les impide establecer una conexión directa con el brazaletes Myo, se hace indispensable la utilización de un Arduino con firmware personalizado para el módulo Bluetooth HM-11. Este firmware se carga mediante el software CCLoader, para ejecutar este procedimiento, se debe primero cargar el programa CCLoader.ino en la placa Arduino, a continuación, se establecieron las conexiones físicas entre la placa Arduino y el módulo HM-11, siguiendo las especificaciones de los pines definidos en la Tabla 3-21. Tras cargar el firmware y completar las conexiones físicas, el HM-11 queda habilitado para comunicarse con el brazaletes Myo, como resultado se obtiene la pantalla que se muestra en la Ilustración 3-33.

Tabla 3-21: Conexiones entre Arduino y HM-11

PIN HM-11	ARDUINO UNO
Vcc	3.3V
Gnd	GND
Debug data	PIN 6
Debug clock	PIN 5
reset	PIN 4

Realizado por: Bautista A., & Oña D., 2023

```
Request sent already!
/*****
* If there is no respond last for 3s, please press "Ctrl+C" to exit!
* And pay attention to :
* 1. The connection between computer and Arduino;
* 2. The connection between Arduino and CC2540;
* 3. Whether the device you using is Leonardo or not;
* 4. Other unexpected errors.
*****/

waiting for respond from Arduino...

Uploading firmware...
1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33
3  34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
3  64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93
3  94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118
119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142
143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166
167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190
191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214
215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238
239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262
263 264 265 266 267 268 269 270 271 272 273 274 Upload successfully!
File closed!
Comport closed!
```

Ilustración 3-33: Pantalla de fin de carga del firmware para el Bluetooth HM-11

Realizado por: Bautista A., & Oña D., 2023

Para confirmar que el software cargó adecuadamente, se realizó una pequeña prueba en Arduino mediante la librería <MyoBridge.h>, donde se obtiene un mensaje de búsqueda y conexión como se muestra en la Ilustración 3-34.

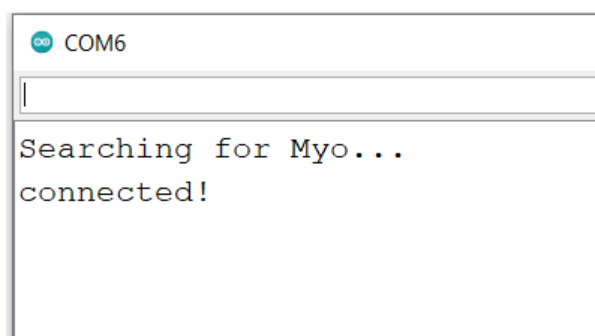


Ilustración 3-34: Conexión entre Myo y ordenador

Realizado por: Bautista A., & Oña D., 2023

Las conexiones para implementar el sistema de comunicación inalámbrica conformado por el módulo HM-11 y Arduino se puede observar en la Ilustración 3-35, también posee 3 leds indicadores, los cuales tienen como función la visualización del estado como son, el led verde

informa si la comunicación con el brazalete MYO secundario se efectuó con normalidad, el led naranja indica si el gesto proporcionado por el brazalete es hacia la izquierda y el led rojo indica si el gesto pertenece al movimiento hacia la derecha, en la Tabla 3-22 se detalla las conexiones del Arduino con el módulo HM-11.

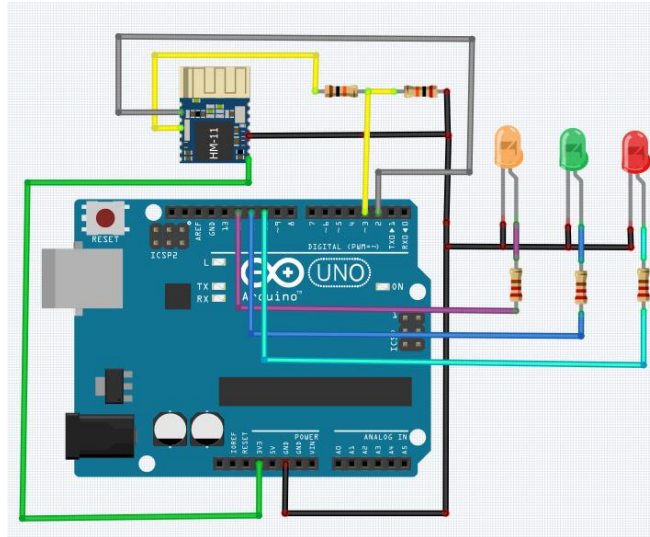


Ilustración 3-35: Conexiones del Arduino

Realizado por: Bautista A., & Oña D., 2023

Tabla 3-22: Conexiones de Arduino

PIN ARDUINO	CONEXIÓN	FUNCIÓN
3.3v	Vcc-HM11	Alimenta el módulo HM-11
GND	GND -HM11 y cátodo leds	Alimenta con GND el módulo HM-11 y los diodos leds
Pin 2	TX-HM11	Transmisión de datos
Pin3	RX-HM11	Recepción de datos
Pin12	Led verde	Indica si el MYO se conecto
Pin 11	Led naranja	Indica si el gesto es izquierdo
Pin 10	Led rojo	Indica si el gesto es derecho

Realizado por: Bautista A., & Oña D., 2023

Una vez completadas las conexiones y la elaboración de la PCB, procedimos a llevar a cabo pruebas de envío de datos, como se evidencia en la Ilustración 3-36, estas pruebas permitieron obtener los datos provenientes de los sensores del brazalete, los cuales pueden ser utilizados para interpretación de gestos.

```
COM6
Searching for Myo...
connected!
-99 13 97 102 102 13 -87 -50
-10 14 1 -38 -11 -11 36 -1
31 -5 -30 -47 -31 -5 21 15
-13 -8 -2 6 0 -3 -20 -11
20 -4 -18 -13 -19 -4 14 10
15 1 -6 -1 -4 2 7 7
9 0 -10 -12 -13 -1 15 8
-1 -1 1 8 4 -1 -4 2
14 0 -12 -12 -4 1 17 7
1 1 1 0 2 0 -1 -2
6 0 -4 -5 -5 -3 6 2
0 1 0 2 5 -1 1 0
```

Ilustración 3-36: Envío de datos Arduino-MYO

Realizado por: Bautista A., & Oña D., 2023

La librería <MyoBridge.h>, también cuenta con herramientas básicas para la detección de gestos elementales, por lo que se recurrió a este recurso con el fin de simplificar el manejo del robot; el brazalete secundario se emplea para controlar de manera individual el movimiento de cada junta o eslabón. Los datos son transmitidos de manera serial a la aplicación desarrollada en Python, la cual interpreta y ejecuta las acciones correspondientes.

3.12.1.2. Calibración y conexión Brazalete MYO con Python

Gracias a Python y su activa comunidad, existen diversas bibliotecas diseñadas para la administración y la conexión con el brazalete MYO, tal como se muestra en la Ilustración 3-37 adjunta.

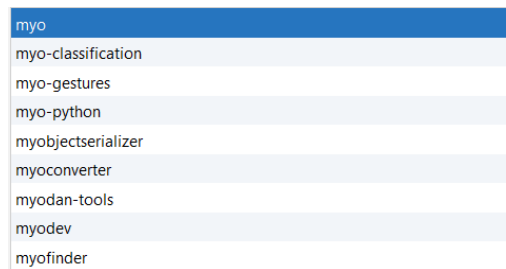


Ilustración 3-37: Bibliotecas de Python

Realizado por: Bautista A., & Oña D., 2023

Se ha optado por utilizar la librería PYOMYO, cuya principal ventaja radica en su método de entrenamiento de gestos. Con la capacidad de almacenar hasta 10 gestos mediante su base de datos interna y la facultad de acceder a los valores de todos los sensores y acelerómetros, esta librería brinda la posibilidad de crear un algoritmo personalizado conforme a nuestras necesidades. A raíz de su notable nivel de desarrollo y considerando los resultados obtenidos en las pruebas realizadas, se puede concluir que la opción más favorable es emplear su propio sistema de entrenamiento.

pyomop
 pyomorfi
 pyomovi
 pyompa
 pyomu
pyomyo
 pyon
 pyon-lib
 pyonapp
 pyonboard
 pyonce
 pyoncoprint
 pyone
 pyoneall

Ilustración 3-38: Librería PYOMYO

Realizado por: Bautista A., & Oña D., 2023

En la Ilustración 3-39 se puede observar el código para la conexión del brazaletes con Python y la lectura de todos los sensores.

```

if __name__ == '__main__':
    # Start Myo Process
    p = multiprocessing.Process(target=worker, args=(q,))
    p.start()
    while (q.empty()):
        # Wait until we actually get data
        continue
    anim = animation.FuncAnimation(fig, animate, blit=False, interval=2)

1 usage
def on_close(event):
    p.terminate()
    raise KeyboardInterrupt
    print("On close has ran")

fig.canvas.mpl_connect('close_event', on_close)
try:
    plt.show()
except KeyboardInterrupt:
    plt.close()
    p.close()
    quit()

def animate(i):
    # Myo Plot
    while not (q.empty()):
        myox = list(q.get())
        if (emg_queue.full()):
            emg_queue.get()
            emg_queue.put(myox)
        channels = np.array(emg_queue.queue)

    if (emg_queue.full()):
        for i in range(0, SENSORS):
            channel = channels[:, i]
            lines[i].set_ydata(channel)
            subplots[i].set_ylim(0, max(1024, max(channel)))
  
```

Ilustración 3-39: Código para la conexión del Brazaletes y Python

Realizado por: Bautista A., & Oña D., 2023

El resultado obtenido se refleja en una representación gráfica que exhibe las mediciones adimensionales de los 8 sensores EMG, tal como se aprecia en la Ilustración 3-40. Para la interpretación directa de estos valores se optó por aprovechar el algoritmo de aprendizaje interno

que brinda la librería, ya que es la más adecuada para lograr un sistema funcional de reconocimiento de gestos.

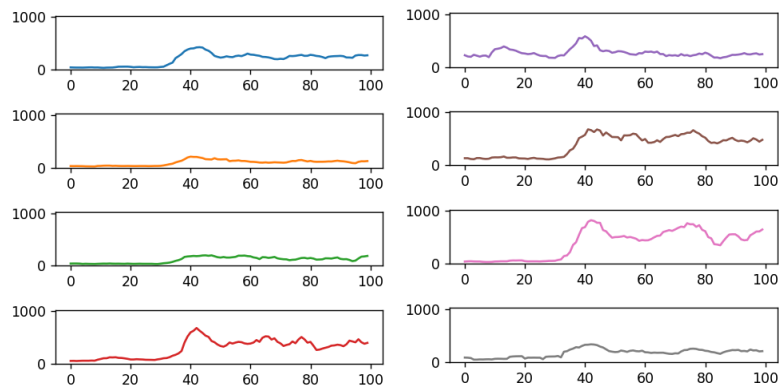


Ilustración 3-40: Medición de los 8 sensores EMG

Realizado por: Bautista A., & Oña D., 2023

3.12.1.3. Algoritmo de aprendizaje

El algoritmo posee una herramienta de aprendizaje cuyo nombre es Classifier, mismo que permitió identificar y clasificar gestos en tiempo real. Este algoritmo puede diferenciar entre diversos tipos de gestos, asignando etiquetas o categorías correspondientes a cada uno, así también facilita la adaptabilidad a nuevas muestras y variaciones en el movimiento.

```
from sklearn.svm import SVC
from sklearn.pipeline import make_pipeline
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from xgboost import XGBClassifier
from sklearn.naive_bayes import GaussianNB

from pyomyo import Myo, emg_mode
from pyomyo.Classifier import Live_Classifier, MyoClassifier, EMGHandler
```

Ilustración 3-41: Algoritmo de aprendizaje Classifier

Realizado por: Bautista A., & Oña D., 2023

Esta herramienta facilita la capacidad de almacenar hasta 10 gestos en una base de datos, como se puede apreciar en la Ilustración 3-42. Los datos generados por los sensores se capturaron en formato "dat", y esta información fue procesada por la librería, que se encarga de identificar y reconocer el gesto almacenado en función de estos datos.

Nombre	Fecha de modificación	Tipo	Tamaño
vals0.dat	6/8/2023 11:01	Archivo DAT	7 KB
vals1.dat	6/8/2023 10:47	Archivo DAT	5 KB
vals2.dat	11/2/2023 8:36	Archivo DAT	0 KB
vals3.dat	11/2/2023 8:36	Archivo DAT	0 KB
vals4.dat	11/2/2023 8:36	Archivo DAT	0 KB
vals5.dat	11/2/2023 8:36	Archivo DAT	0 KB
vals6.dat	11/2/2023 8:36	Archivo DAT	0 KB
vals7.dat	11/2/2023 8:36	Archivo DAT	0 KB
vals8.dat	11/2/2023 8:36	Archivo DAT	0 KB
vals9.dat	11/2/2023 8:36	Archivo DAT	0 KB

Ilustración 3-42: Datos generados por los sensores

Realizado por: Bautista A., & Oña D., 2023

Para poner a prueba esta herramienta, se desarrolló un algoritmo que permite la recopilación y almacenamiento de los datos correspondientes, además de la revelación del gesto en cuestión, tomando como ejemplo la Ilustración 3-43 donde el gesto 0, cuenta con 405 elementos almacenados, la barra de color morado actúa como indicador para determinar su correspondencia con el gesto específico; en este caso, se almacenó el gesto de "puño", resulta notorio que el algoritmo es capaz de identificar el gesto. En el apartado de pruebas y resultados, se expondrán las muestras recabadas durante el proceso, acompañadas por el tiempo de respuesta correspondiente.



Ilustración 3-43: Recopilación y almacenamiento de datos

Realizado por: Bautista A., & Oña D., 2023

Una vez concluido el proceso de entrenamiento, donde se guardó los gestos tanto para los seis grados de libertad como para la apertura y cierre del gripper, se abre la posibilidad de interpretar los datos para guiar el movimiento del robot de manera coherente y eficiente.

3.12.1.4. Calibración y conexión Robot

La estructura fundamental del sistema robótico se compone principalmente de un controlador basado en una STM32, este controlador desempeña un papel clave al recibir códigos G, también conocidos como G-Code. La interacción entre el robot y el ordenador se efectuó mediante una

conexión serial, solo que a diferencia de enviar caracteres puros se envían comandos G-Code, dichos comandos se encuentran disponibles en el manual como se muestra en la Ilustración 3-44, mismos abarcan diversas acciones. Por ejemplo, el comando "G06 O=P1.1" señala la activación de la salida 1 en el controlador, no obstante, es esencial tener en cuenta que la utilización de una serie de comandos requiere precaución, particularmente con relación a las velocidades, que constituyen un aspecto crucial a considerar.

```
G06 directive
G06 T=500 delay 500 milliseconds
G06 t=VXX Delay VXX content milliseconds (for example: G08 MOV V400=#2000 G06 t=V400 is a
delay of 2000 milliseconds)
G06 I=P1.1 Wait for P1 to be high
G06 I=P2.0 Wait for P2 to be low
G06 O=P1.1 makes output port P1 high (different port from input port)
G06 O=P2.0 makes the output port P2 low
G06 SCAN=I Read input port value, stored in V144-V150 unit
G06 SCAN=O Read output port value, stored in V160-V166 unit
G06 SCAN=RTC Read system clock value and store it in V176-V179 unit
G06 DEGREE=ARC robot takes a three-point arc
G06 DEGREE=35.2 The robot walks 35.2 degrees
```

Ilustración 3-44: Comandos del brazo robótico

Fuente: (Manual, 2023)

La Ilustración 3-45 presenta el código que permite la conexión serial con el robot, uno de los aspectos clave a tener en cuenta es la velocidad de transmisión medidos en baudios, la velocidad predefinida del robot se establece en 115 200 baudios, aunque existe la posibilidad de modificarla, no obstante, se aconseja evitar realizar dicho cambio a menos que sea estrictamente necesario.

```
# Open serial port
s = serial.Serial('COM19', 115200)
s = serial.Serial(args.port, 115200)
print
'Opening Serial Port'

# Open g-code file
f = open('/media/UNTITLED/shoulder.g', 'r');
f = open(args.file, 'r');
print
'Opening gcode file'
```

Ilustración 3-45: Algoritmo de conexión

Realizado por: Bautista A., & Oña D., 2023

En el proceso de calibración, una de las ventajas fundamentales proviene del sistema intrínseco de posicionamiento del robot, es decir el sistema permite al robot tener conocimiento en tiempo real de su posición inicial, así como de los grados de rotación de cada eslabón. Por esta razón, al llevar a cabo la calibración, también conocida como establecimiento de la posición cero o Home, se utilizó las medidas específicas de los eslabones, estos puntos angulares se convierten en la posición de referencia, o punto Home, dentro de nuestro sistema.

$$G00 \gg J1 = 0 \quad J2 = 0 \quad J3 = -90 \quad J4 = 0 \quad J5 = -90 \quad J6 = 0$$

3.12.2. Conexiones del Software

Como se ha destacado previamente, se optó por Python como el lenguaje de programación fundamental para el sistema, una elección respaldada por su adaptabilidad a nuestras necesidades específicas y sus características, detalladas en la Tabla 3-17. Python facilitó el establecimiento de conexiones con el brazalete Myo y la creación de algoritmos para la lectura de señales EMG. Esto se logra a través del uso de librerías especializadas como PyoMyo, PYQT5, Classifier, y se complementa con la herramienta de desarrollo PyCharm, que ofrece análisis de código inteligente.

3.12.2.1. Programación Python

Para facilitar la comprensión del sistema, se han delineado diversos procedimientos de acuerdo con el número de interfaces gráficas. El diagrama de flujo que muestra en la Ilustración 3-46 corresponde al método de control para el brazo robótico, detallando cada una de las pantallas generadas, partiendo de la calibración de los brazaletes, control manual y automático. La etapa de calibración de los brazaletes adquiere una gran importancia en el manejo del sistema, ya que se fundamenta en la recopilación y almacenamiento de datos relativos a los seis grados de libertad.

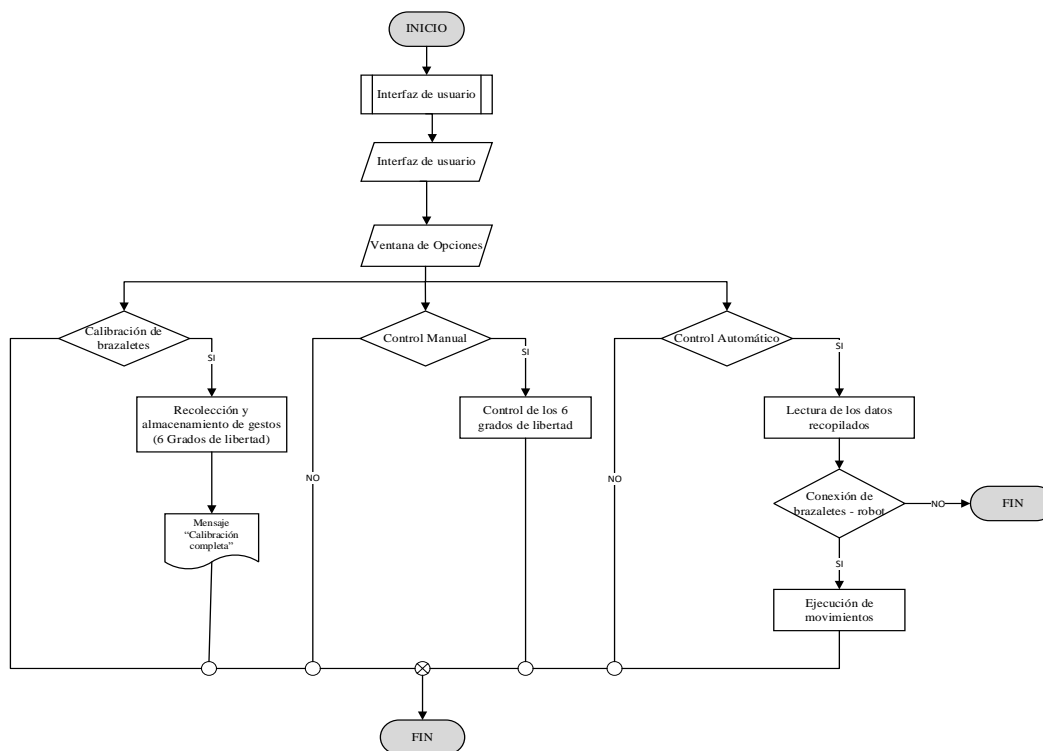


Ilustración 3-46: Diagrama de flujo – Interfaz de usuario

Realizado por: Bautista A., & Oña D., 2023

Adicionalmente, incorpora una interfaz destinada al control manual y otra para el control automático, el control manual permite manipular directamente el robot, mientras que el control automático posibilita la manipulación del brazo empleando señales electromiográficas. Asimismo, permite la visualización de la dirección de los movimientos y determina la posición final del gripper aplicando la cinemática directa. Cada uno de estos procesos se detallan en las ilustraciones subsecuentes mediante diagramas de flujo.

a) **Inicio.** En la etapa inicial se estableció la comunicación inalámbrica de los brazaletes, utilizando la librería PYOMYO, y con el robot a través de una conexión serial. Simultáneamente, se activó la interfaz de la pantalla inicial, una vez que esta interfaz es desplegada se mantiene visible hasta que el usuario haga clic en el botón “Empezar”. Este proceso permite establecer la comunicación del robot por medio de la programación efectuada, dando inicio el proceso de operación, como se detalla en la Ilustración 3-47.

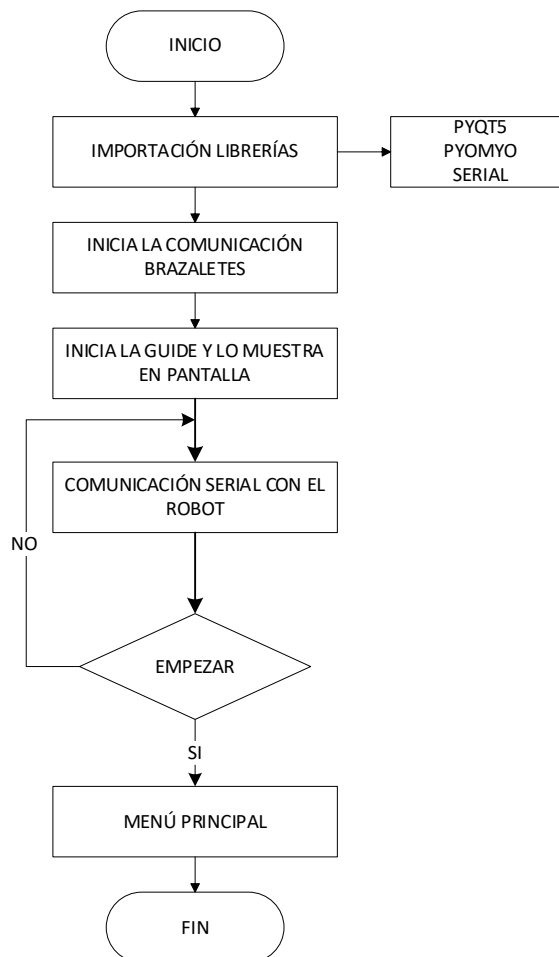


Ilustración 3-47: Diagrama de flujo del proceso inicial

Realizado por: Bautista A., & Oña D., 2023

b) **Menú Principal.** En la fase de implementación, se destaca la agilidad del código, ya que se centra en la creación de un menú de opciones mediante la biblioteca PyQt en Python. Este menú, implementado en la clase menú, ofrece al usuario tres opciones fundamentales para la operación del robot, reflejando la comunicación y señales provenientes de los brazaletes. La primera opción, gestionada por el método **c1**, se enfoca en la calibración, permitiendo ajustar los seis grados de libertad y el gripper. La segunda opción, manejada por **m1**, habilita el control manual para la manipulación de cada articulación del brazo robótico. La tercera opción, manejada por **a1**, activa el control automático para operar el robot de manera remota.

Es crucial recalcar que, previo al uso de la opción de control automático, se debe realizar la calibración del dispositivo MYO para habilitar el control remoto, como se evidencia en el código mediante las funciones de conexión a través de puertos serie con **ser.write()** y la ejecución en segundo plano mediante hilos. Este diseño modular y la integración de las funciones esenciales resaltan la eficiencia y claridad en la implementación. Estas acciones se sintetizan en el diagrama de la Ilustración 3-48.

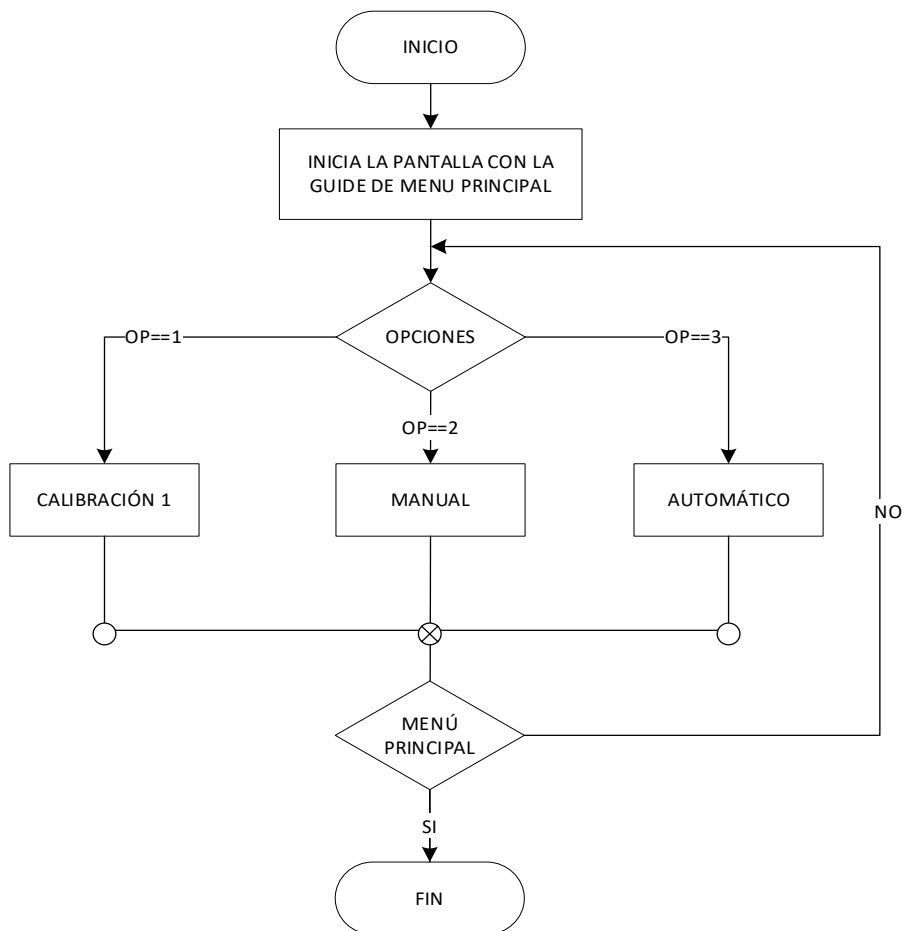


Ilustración 3-48: Diagrama de flujo del menú principal

Realizado por: Bautista A., & Oña D., 2023

c) **Calibración:** En la etapa de calibración guarda similitudes con la fase anterior: se despliega una interfaz en pantalla en la que el usuario debe ajustar el tiempo necesario para almacenar los datos. Para mantener la simplicidad del diagrama, se optó por generalizarlo, donde la fase de almacenamiento se inicia con un mensaje en el cual se señala el comienzo de la lectura de muestras. Basándonos en pruebas realizadas, se observa que un rango óptimo para la toma de muestras por gesto es de 1800 a 2400, tras completar el almacenamiento de datos para el movimiento del robot, se muestra un mensaje de conclusión que da por terminado el proceso de almacenamiento.

Como función adicional, se incluye un botón para restablecer “reset” la base de datos, es importante destacar que esta lógica de programación se replica para cada grado de libertad y para el Gripper en uso, además, la captura de datos se basa principalmente en un cronograma temporal. El proceso de calibración de cada grado de libertad se sintetiza en la Ilustración 3-49.

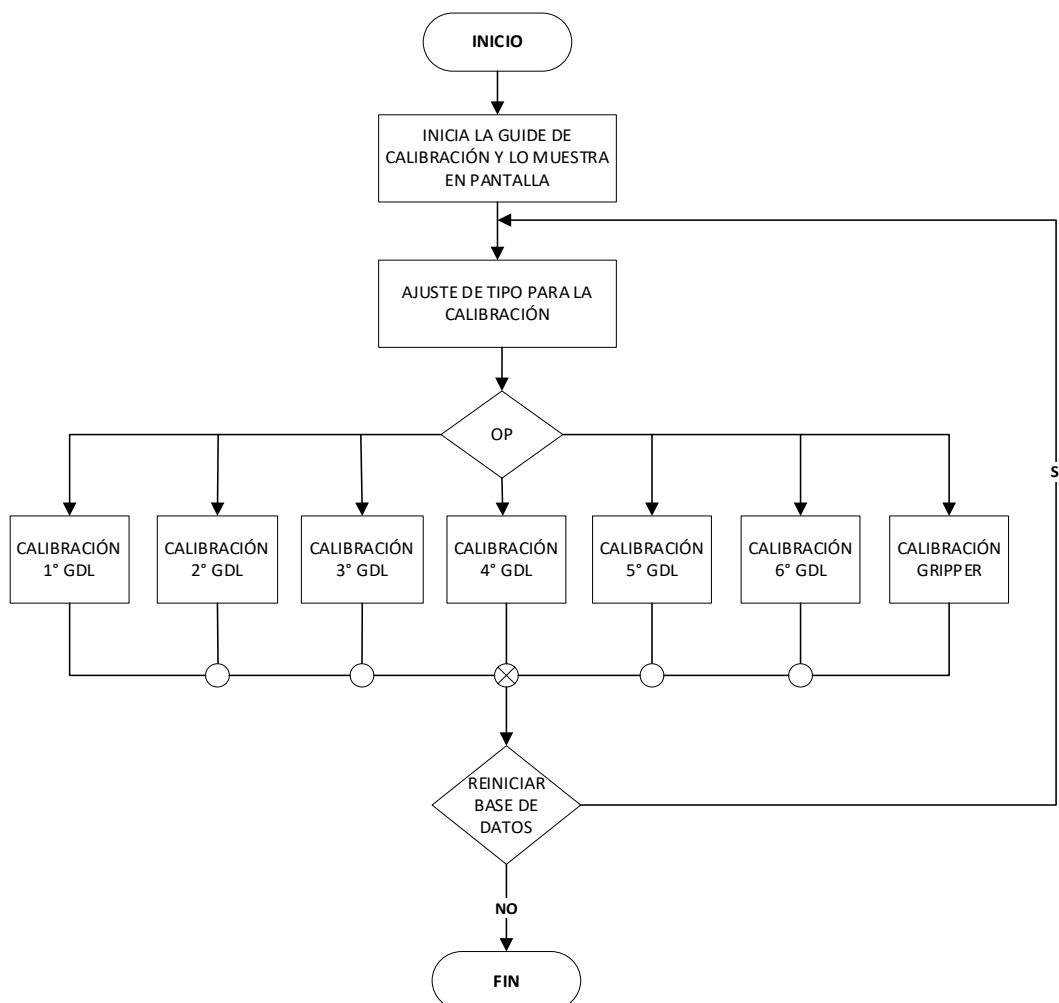


Ilustración 3-49: Diagrama de flujo de la calibración

Realizado por: Bautista A., & Oña D., 2023

Para la calibración del robot, es apropiado llevar a cabo la sincronización y el almacenamiento de gestos mediante señales EMG, de esta forma se calibra cada grado de libertad en relación con los gestos emulados en el robot. Este procedimiento se detalla en el diagrama de la Ilustración 3-50, partiendo con la configuración del tiempo apropiado para la recopilación de datos, durante este tiempo las señales EMG captadas por los sensores del MYO son almacenadas en una base de datos, tras el transcurso del tiempo establecido se dará por finalizado el almacenamiento de datos correspondientes a un gesto específico y en caso que el tiempo transcurrido sea menor al tiempo establecido se volverá a leer las señales proporcionadas por el sensor.

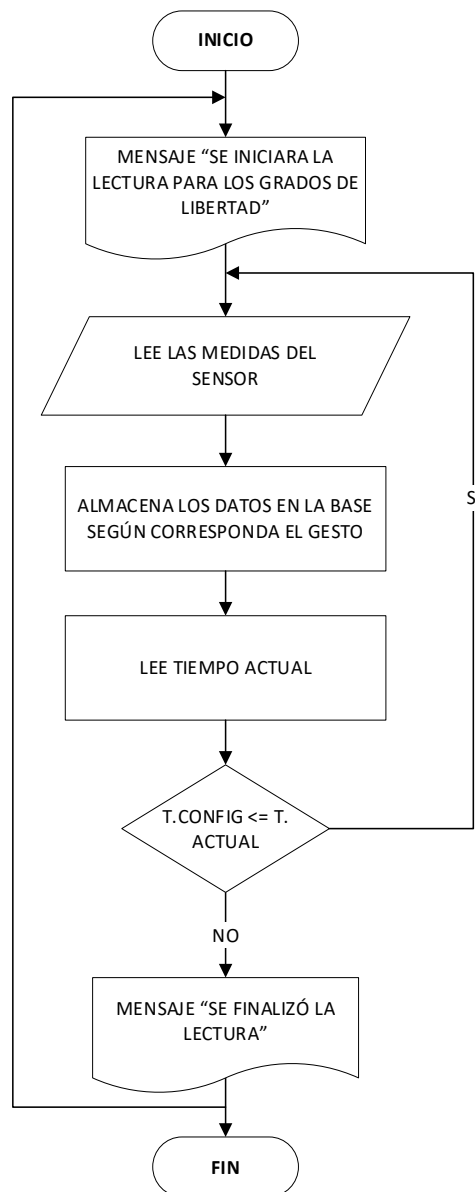


Ilustración 3-50: Diagrama de almacenamiento de gestos.

Realizado por: Bautista A., & Oña D., 2023

d) **Control Manual:** En relación con el control manual, este componente representa un controlador para un manipulador robótico a través de una interfaz gráfica implementada en Python con PyQt. La clase `manual1`, heredera de `QMainWindow`, gestiona eventos de botones para controlar las articulaciones del robot, empleando una interfaz de aproximadamente 15 botones, divididos equitativamente en 6 de signo positivo y 6 de signo negativo. En tiempo real, el usuario puede ajustar los ángulos de las articulaciones mediante estos botones. Al presionar cualquiera de ellos, el sistema lee el ángulo actual del eslabón correspondiente. Dependiendo de si se pulsa un botón de signo positivo o negativo, se incrementará o disminuirá el ángulo respectivo, como se describe en el diagrama de flujo generalizado en la Ilustración 3-51. Estos métodos no solo modifican los ángulos de las articulaciones, sino que también envían comandos al dispositivo para reflejar dichos cambios. La información se procesa y se prepara para ser enviada al robot en formato G-code, dando lugar a un ciclo continuo de acciones que persistirá de manera indefinida hasta que se abandone el modo de control manual.

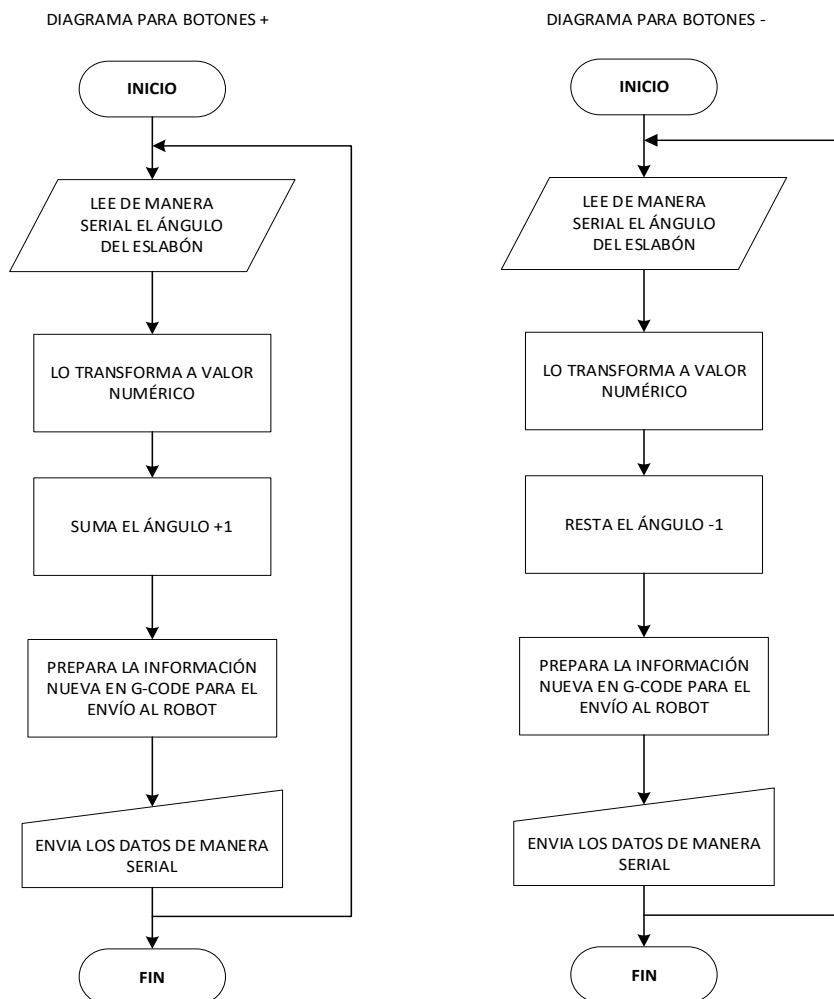


Ilustración 3-51: Proceso de control manual para el robot

Realizado por: Bautista A., & Oña D., 2023

- e) **Control Automático:** Para el proceso de control automático, se inicia activando la captura de datos de los sensores. Luego, estos datos son comparados con la información almacenada en la base de datos, si el gesto capturado coincide con uno previamente registrado, se genera un valor numérico como resultado, es importante mencionar que a pesar de que se dispone de 10 gestos almacenados por sensor según la documentación, en la implementación actual se utilizan únicamente 7 gestos. De estos, 6 corresponden a los Grados de Libertad (GDL), mientras que 1 se asigna al Gripper, por lo tanto, el valor resultante oscilará en el rango de 1 a 7, identificando así el gesto específico que fue reconocido.

Una vez que el gesto es identificado, se procede a medir el ángulo del eslabón relacionado con dicho gesto, posteriormente, se realiza una operación de suma o resto en una unidad. La determinación de sumar o reiniciar depende de la información proporcionada por otro sensor, el cual indica si se debe realizar un movimiento en dirección positiva o negativa, el valor resultante de esta operación se prepara para ser transmitido al robot en forma de un mensaje de tipo G-code. La Ilustración 3-52, muestra el flujo del sistema, desde la captura del gesto hasta la comparación en la base de datos y la posterior selección de funciones, a la vez, muestra una visualización de la ejecución del movimiento de los eslabones en función de los datos procesados.

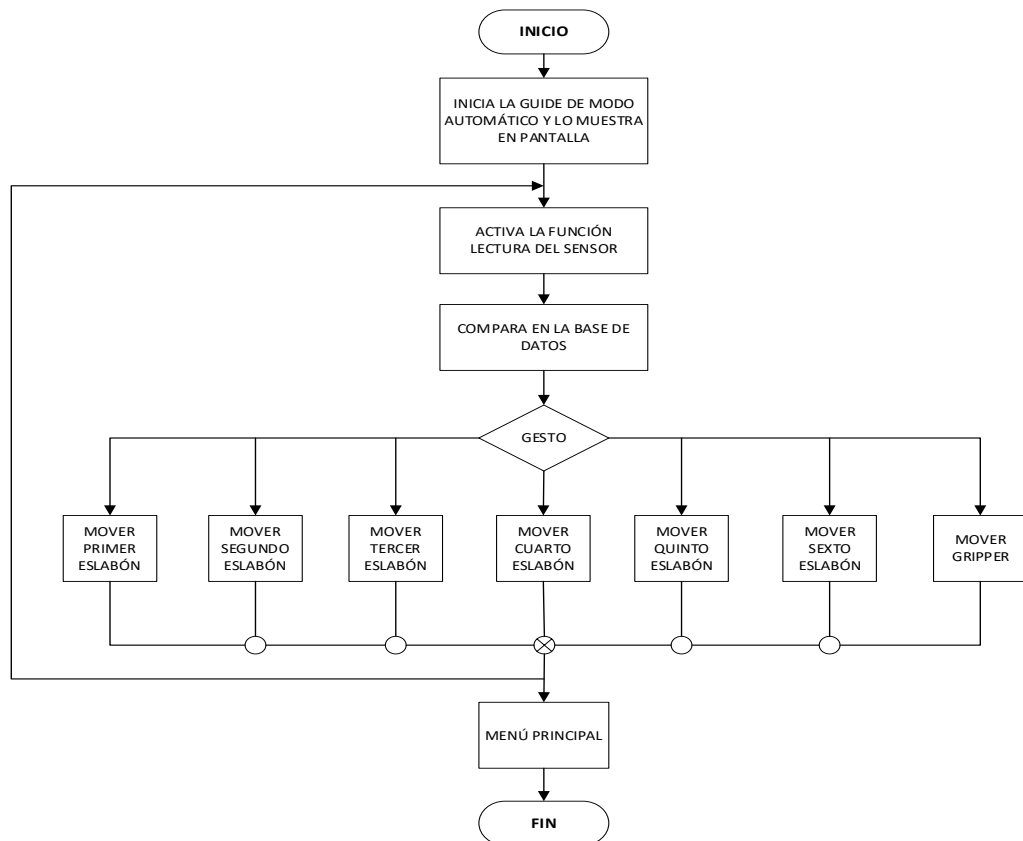


Ilustración 3-52: Proceso de control automático del robot

Realizado por: Bautista A., & Oña D., 2023

Después de haber identificado el gesto o el número del eslabón a activar, se procederá a exhibir en pantalla la selección del eslabón correspondiente, a continuación, se establece una comunicación serial con un Arduino que alberga los datos del segundo brazaletes, en esta etapa, se especifica la dirección del movimiento deseado: positivo o negativo. Una vez que estos datos son interpretados, se realiza la preparación necesaria antes de enviarlos al robot, estos valores se transforman en instrucciones G-code, un lenguaje que el robot puede entender y ejecutar los movimientos deseados de manera precisa. El detalle de la operación se muestra en el diagrama de la Ilustración 3-53.

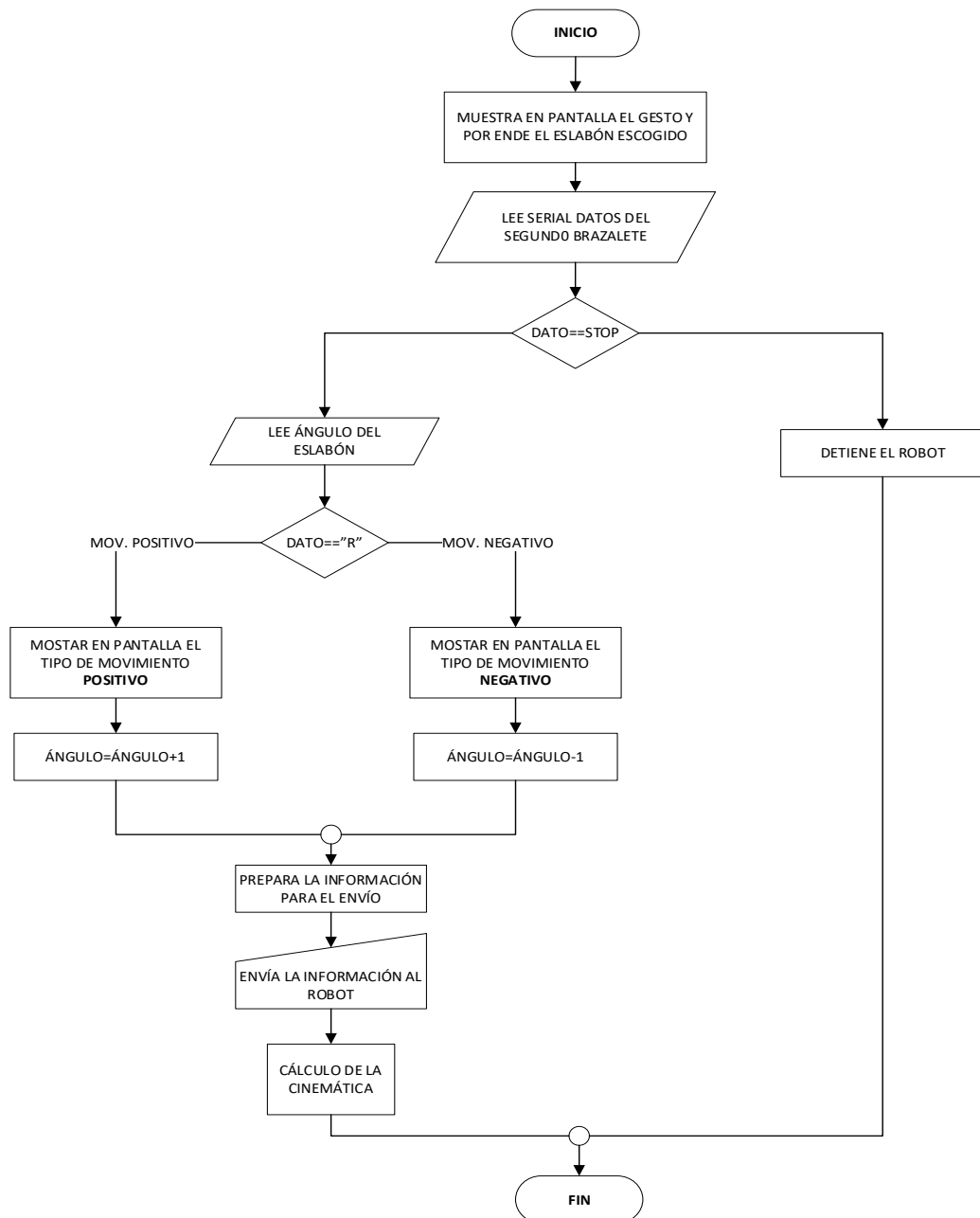


Ilustración 3-53: Sincronización de la operación del robot por control automático

Realizado por: Bautista A., & Oña D., 2023

El diagrama de flujo presentado en la Ilustración 3-53 representa la sincronización de las operación del robot mediante un control automático, en este proceso se empleó funciones específicas para Iniciar (clnician, I_rob), Detener (S_rob) y Controlar el robot (para), el comando **clnician** y **I_rob** inicia el sistema mediante el establecimiento de self.actT en verdadero, mostrando un cuadro de mensaje de "SISTEMA INICIADO". Por otro lado el comando **S_rob**, detiene el modo iniciado por **I_rob** por ende, self.actR se vuelve falso y muestra un mensaje de confirmación.

Otra función empleada en el sistema es la de Actualización (actualize), la cual opera obteniendo un valor entero dentro de un bloque **if**, este bloque utiliza una declaración **match** para comparar el valor de estado desencadenando así la modificación de ángulos de cada articulación mientras se genera distintos movimientos. Además, esta función facilita la lectura de datos almacenados por el myo_arduino y la interfaz gráfica se ajusta en consecuencia, para esto si self.actR es verdadero, se envían comandos al robot según los ángulos actuales de las articulaciones y estos ángulos se visualizan en la interfaz gráfica.

En cuanto al último bloque del cálculo de la cinemática, se utilizó varias ecuaciones matemáticas para determinar las coordenadas (X, Y, Z) y los ángulos (A, B, C) del sistema en función de las variables de ángulo proporcionadas y longitudes de los eslabones, estos cálculos se redondean y luego se muestran en la interfaz gráfica a través de las funciones display y print.

3.12.2.2. *Desarrollo de interfaz*

Para el desarrollo de la interfaz, se empleó el programa QT Designer, el cual brinda componentes como botones, etiquetas (labels), visualizadores LCD, entre otros, estos elementos son seleccionados y configurados de manera similar a la creación de una interfaz gráfica, como podría ser el caso de una GUIDE en Matlab, como se muestra en la Ilustración 3-54, es fundamental tener en cuenta el identificador (ID) o nombre asignado a cada objeto, ya que estos se utilizarán posteriormente en Python para el envío y la obtención de información correspondiente.

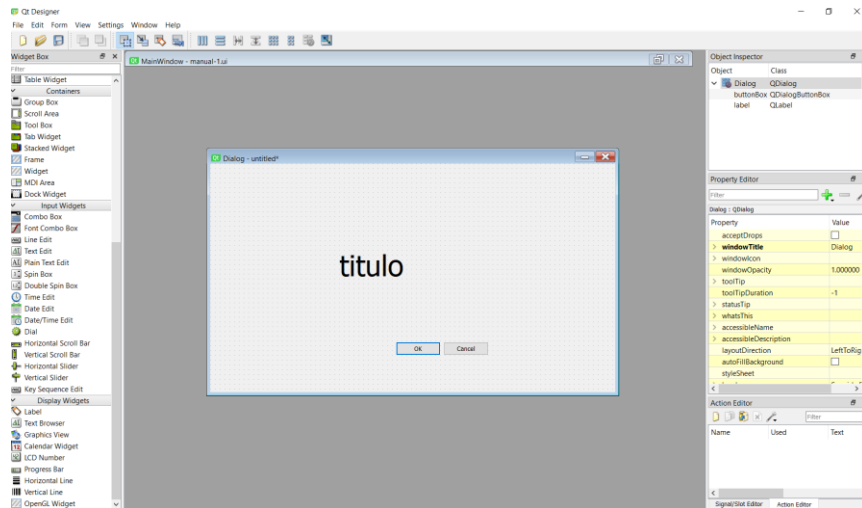


Ilustración 3-54: Desarrollo de interfaz en Qt Designer

Realizado por: Bautista A., & Oña D., 2023

Además, Qt Designer es una herramienta gráfica que se utilizó junto con Python para crear interfaces de usuario, este software permitió diseñar interfaces de usuario de forma visual, arrastrando y soltando componentes gráficos en un lienzo, también facilita la separación entre la lógica de la aplicación y la presentación de la interfaz de usuario. Esto sigue el principio de diseño de software conocido como el patrón Modelo-Vista-Controlador (MVC).

CAPÍTULO IV

4. PROPUESTA TECNOLÓGICA

En este capítulo, se detallan los resultados obtenidos de la implementación del sistema de control remoto para el robot ANNO RV624, basado en la captura y procesamiento de señales electromiográficas.

4.1. Aplicación informática

El algoritmo del sistema se desarrolló en PYTHON ya que es un programa en código abierto, donde el sistema recibe las señales tanto del Myo izquierdo como del derecho mediante comunicación bluetooth. Posteriormente el algoritmo se procesa y se compara con la base de datos de la calibración y según estos se toma la decisión del movimiento del brazo enviando esta información mediante código G, para la operación del robot. El desarrollo de la interfaz fue realizado en QT Designer.

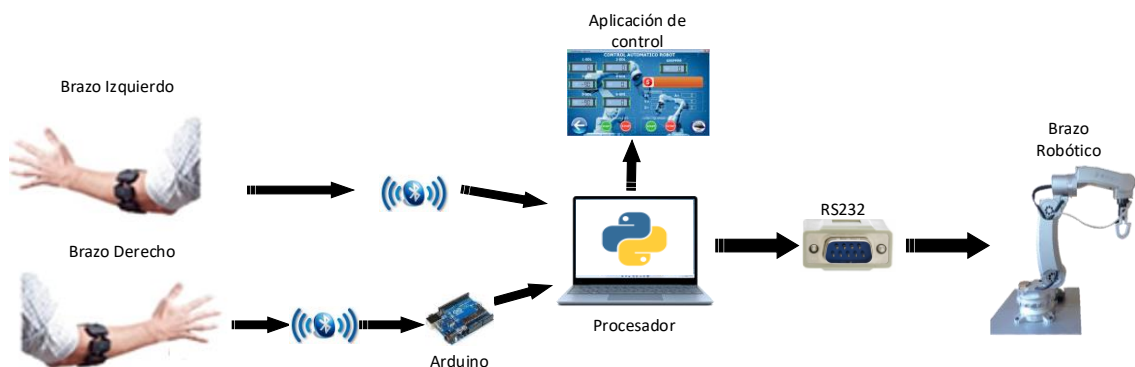


Ilustración 4-1: Sistema de control del brazo robótico ANNO RV624

Realizado por: Bautista A., & Oña D., 2023

Cómo se puede observar en la Ilustración 4-1 el sistema implementado consta de dos brazaletes MYO los cuales poseen sensores EMG que detectan las señales electromiográficas generadas por la contracción y relajación muscular, capturando estas señales a través de electrodos diferenciales de superficie en la piel. Además, incluye acelerómetros y giroscopios para detectar el movimiento, la orientación del brazo y la muñeca, lo que le permite entender los gestos y la posición en el espacio tridimensional. El brazalette se conecta de forma inalámbrica al procesador, a través de Bluetooth, lo que facilita la transmisión de los datos procesados. Es importante mencionar que el MYO derecho se conecta al Arduino donde se encuentra el módulo HM 11 para posteriormente ser conectada mediante un cable al procesador.

La interfaz sirve inicialmente como un portal de bienvenida, ofreciendo información contextual sobre el sistema en su totalidad. El menú principal se subdivide en tres pantallas clave:

1. Una pantalla de calibración para el receptor principal, que regula los ángulos de actuación del sistema robótico.
2. Una interfaz de control manual que otorga al usuario la capacidad de manejar el robot de forma directa.
3. Una pantalla de control automático que recopila y procesa datos de los receptores seleccionados. El algoritmo desarrollado se encarga de interpretar estos datos para determinar las acciones de movimiento y los resultados son transmitidos al robot a través de una conexión RS232.

4.1.1. Conexiones Iniciales

En la Ilustración 4-2 se muestra la conexión inicial del robot a través de la interfaz de inicio, donde se procede a seleccionar el puerto serial pertinente, seguido de la verificación del estado de conexión. Previo a esta configuración, es necesario establecer la conexión física entre los brazaletes y el robot con el ordenador, una vez completada y finalizada esta acción, se puede iniciar el control remoto del robot.



Ilustración 4-2: Conexión inicial con el robot

Realizado por: Bautista A., & Oña D., 2023

4.1.2. Interfaz

Para la calibración, operación y control robótico, se diseñó una interfaz que se ajusta a la metodología propuesta y desarrollada. La interfaz se compone de un conjunto de pantallas o guídes divididas en 6 etapas o procesos. A continuación, se presenta un análisis detallado del funcionamiento de estas pantallas, junto con la información correspondiente acerca de cada elemento empleado en la interfaz.

1. Pantalla principal

La pantalla principal permite establecer la conexión entre el robot y el sistema de control desarrollado, para esto se selecciona el puerto COM4 y se presiona el botón “Star”. Una vez que la conexión del robot se ha efectuado se emite un mensaje indicando “Conexión exitosa”, lo que desbloquea el acceso al menú principal.



Ilustración 4-3: Elección del puerto para la conexión del brazo robótico

Realizado por: Bautista A., & Oña D., 2023

2. Menú principal

La segunda interfaz, como se muestra en la Ilustración 4-3, se encuentra jerarquizada donde brinda al usuario tres opciones de trabajo, la primera opción posibilita la calibración del brazaletes, la segunda alternativa habilita el control manual del robot empleando botones y la última opción es para el control automático a través de la integración de señales EMG. Este diseño

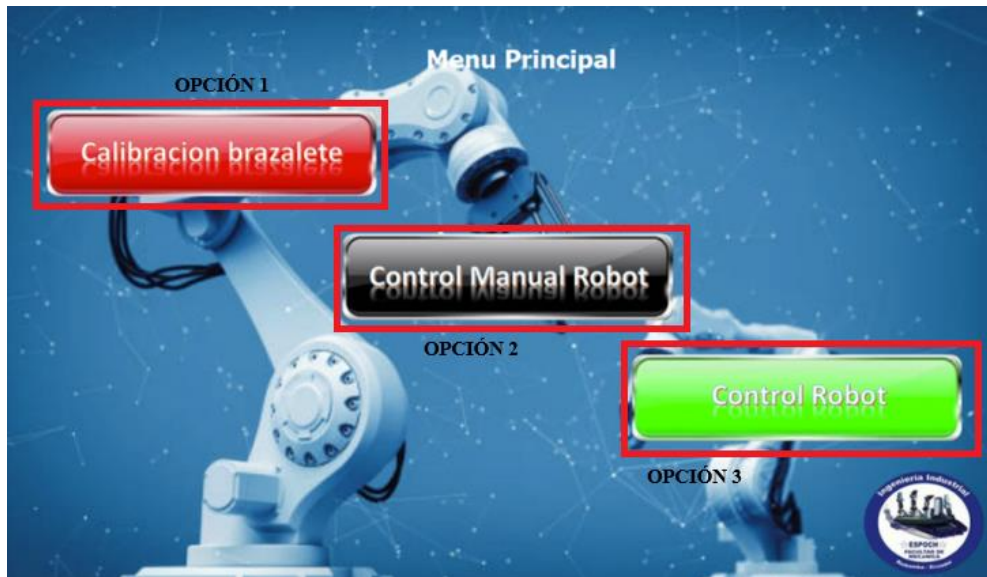


Ilustración 4-4: Selección de las opciones de control

Realizado por: Bautista A., & Oña D., 2023

3. Calibración del brazaletes MYO

El guide proporciona un total de 9 opciones, iniciando con la calibración del primer gesto mediante el almacenamiento de sus valores en la base de datos, cada gesto asociado en el guide representa el movimiento de un grado de libertad del robot y comparten las características previamente descritas, adicionalmente, incluye una sección para la calibración del gripper.



Ilustración 4-5: Entrenamiento de gestos para cada grado de libertad

Realizado por: Bautista A., & Oña D., 2023

La guide está basada en un código que se encarga de leer o almacenar los nuevos datos, procesándolos con la librería **KNN classifier**, principalmente se emplea el comando **train**, que tiene como función el entrenamiento del brazaletes, recopilando los datos de todos los sensores del MYO. Otro de los comandos empleados es el **restart** que sirve para borrar la base de datos y por último un temporizador que inicia el conteo del tiempo determinado para guardar los gestos, una vez finalizada la fase de almacenamiento, se visualiza la palabra "False".

4. Control Manual

La interfaz de control manual dispone de controles individuales para cada eslabón del robot, incluso para el gripper. Estos controles comprenden dos botones destinados a incrementar o disminuir el ángulo, otro botón para restablecer o "Home" que permite que el robot vuelva a su posición inicial, y un botón de retorno al menú principal.

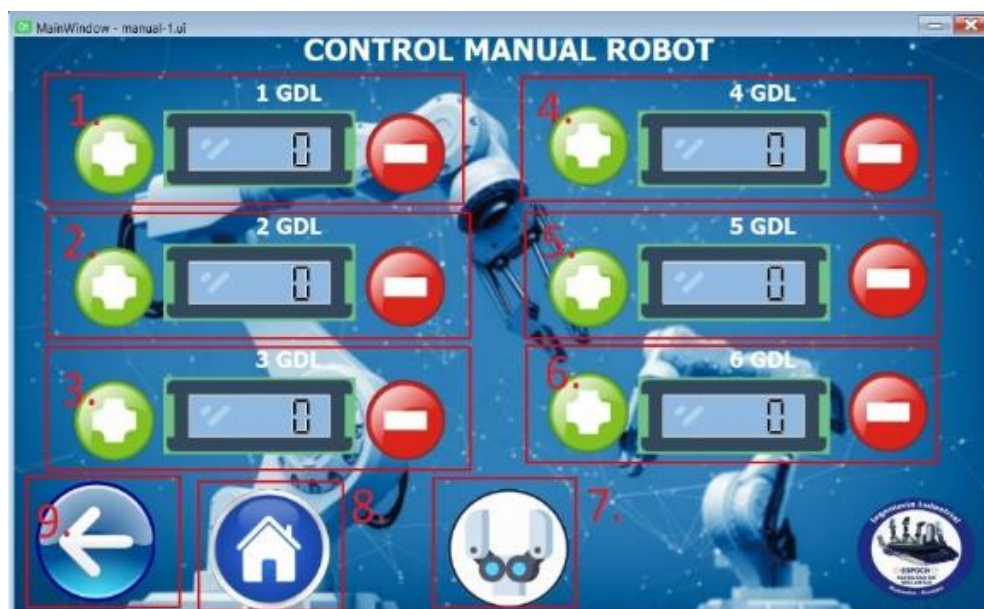


Ilustración 4-6: Interfaz de control manual del brazo

Realizado por: Bautista A., & Oña D., 2023

5. Control Automático

Esta guide desempeña una función fundamental en el sistema, al ser la receptora de las señales emitidas por los sensores y, mediante la interpretación de un algoritmo, exhibe en la pantalla el gesto correspondiente. Posteriormente, ejecuta una transformación que le permite identificar el gesto y llevar a cabo el ajuste angular almacenado en la base de datos. Además de estas funciones, consta de dos botones "start": uno para iniciar la comunicación entre los brazaletes y otro para establecer la conexión con el robot, de esta manera, es factible desactivar únicamente el

funcionamiento del robot sin afectar la operatividad de los brazaletes, así también incluye la visualización de la posición final del gripper en los ejes x, y, z, junto con los ángulos en A, B y C.

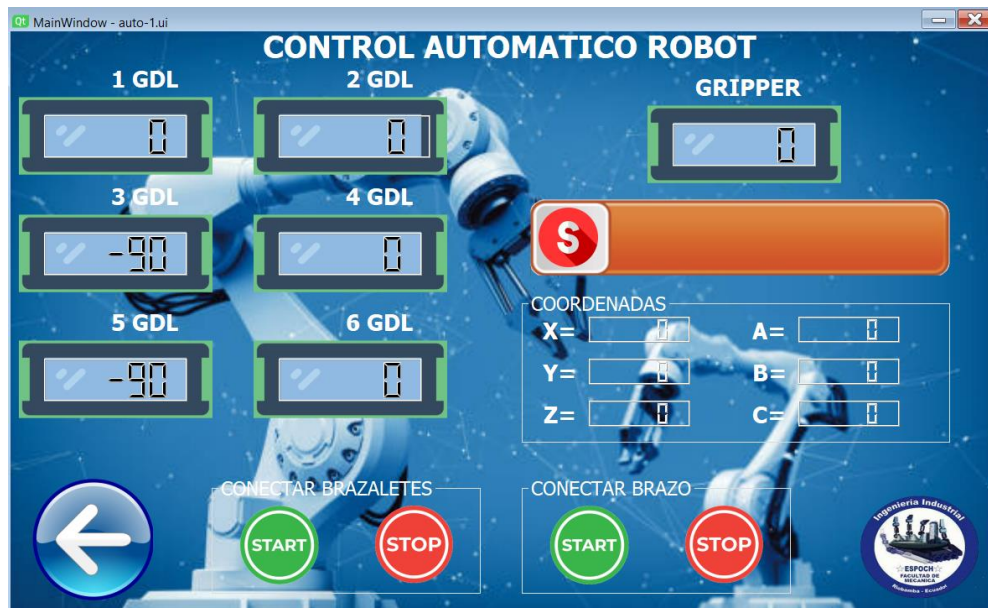


Ilustración 4-7: Interfaz para el control automático

Realizado por: Bautista A., & Oña D., 2023

Este modelo fue seleccionado debido a su operatividad intuitiva, asegurando que el usuario final no encuentre obstáculos al utilizarlo. Todas estas pantallas, junto con las imágenes empleadas, se almacenan en la misma carpeta donde se encuentra el programa desarrollado en Python, como se presenta en la ilustración adjunta. En esta ubicación, es posible observar los archivos con extensión .UI, que corresponde al formato de almacenamiento de las guides, así como las imágenes destinadas a su empleo.

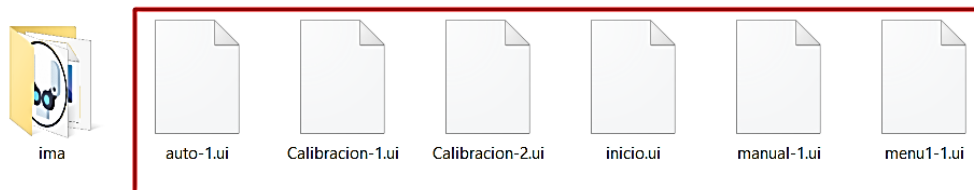


Ilustración 4-8: Tipos de archivos de Python

Realizado por: Bautista A., & Oña D., 2023





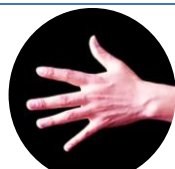
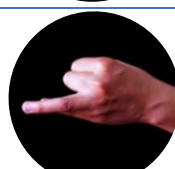
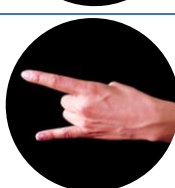
4.2. Pruebas de operación



Para realizar las pruebas operativas se partió de la calibración para los brazaletes, que implica el entrenamiento de los 6° de libertad y el gripper del robot, con este propósito se definió un período de calibración fijado en 42 segundos, durante este lapso, se registró un promedio de 1845 a 2214 muestras y se debe considerar que mientras mayor sea el tiempo mayor precisión tendrá.

4.2.1. Entrenamiento de brazaletes MYO

Durante la fase de entrenamiento, se definió un gesto específico y la codificación empelada para cada uno de los grados de libertad como se observa en la Tabla 4-1. El brazaletes principal situado en el brazo izquierdo controla los seis grados de libertad mientras que el brazaletes secundario ubicado en el brazo derecho se encarga de enviar señales destinadas a realizar movimientos hacia arriba y abajo para la segunda, tercera y quinta articulación, así como movimientos hacia la izquierda y derecha para la primera, cuarta y sexta articulación.

Tabla 4-1: Descripción de gestos codificados para el brazo robótico

MYO	GESTOS	DESCRIPCIÓN	GESTO CODIFICADO
MYO PRINCIPAL (Mano Izquierda)		Controla el primer grado de libertad.	BO-01
		Controla el segundo grado de libertad.	BO-02
		Controla el tercer grado de libertad	BO-03
		Controla el cuarto grado de libertad	BO-04
		Controla el quinto grado de libertad	BO-05
		Controla el sexto grado de libertad	BO-06
		Controla el gripper	BO-07

MYO SECUNDARIO (Mano Derecha)		Control del eslabón (Izquierda, cierre, giro izquierdo)	BO-08
		Control del eslabón (Derecha, abierto, giro derecho)	BO-09

Realizado por: Bautista A., & Oña D., 2023

4.2.2. Pruebas de control

En la Ilustración 4-9 se muestra la posición inicial del brazo robótico donde cada articulación o junta toma los valores de: $J1=0$, $J2=0$, $J3=-90$, $J4=0$, $J5=-90$ y $J6=0$.



Ilustración 4-9: Posición inicial de robot

Realizado por: Bautista A., & Oña D., 2023

En la Ilustración 4-10 se observa el movimiento del primer grado de libertad del robot, para lo cual se combinó los gestos BO-01 y BO-08, por ende, el movimiento del brazo robótico es hacia la izquierda.



Ilustración 4-10: Movimiento de la junta 1

Realizado por: Bautista A., & Oña D., 2023

Tabla 4-2: Coordenadas de la posición final de la junta 1

Posición Inicial		Posición Final	
x=355	A=0	x=413,8	A=57,03
y=209,5	B=0	y=-717	B=1,5
z=433	C=0	z=514,6	C=2,6

Realizado por: Bautista y Oña, 2023

En la Ilustración 4-10 se observa el movimiento del segundo grado de libertad del robot, para lo cual se combinó los gestos BO-02 y BO-09, por esto el movimiento que realiza el brazo robótico es hacia atrás.



Ilustración 4-11: Movimiento de la junta 2

Realizado por: Bautista A., & Oña D., 2023

Tabla 4-3: Coordenadas de la posición final de la junta 2

Posición Inicial		Posición Final	
x=413,8	A=57,03	x=538,2	A=45,04
y=-717	B=1,5	y=-598	B=2,01
z=433514,6	C=2,6	z=389,2	C=2,23

Realizado por: Bautista A., & Oña D., 2023

En la Ilustración 4-12 se observa el control del tercer grado de libertad del robot, para lo cual se combinó los gestos BO-03 y BO-09, por ende, el brazo robótico se mueve hacia la arriba.



Ilustración 4-12: Movimiento de la junta 3

Realizado por: Bautista A., & Oña D., 2023

Tabla 4-4: Coordenadas de la posición final de la junta 3

Posición Inicial		Posición Final	
x=538,2	A=45,04	x=573,5	A=42,35
y=-598	B=2,01	y=-637	B=-6,01
z=389,2	C=2,23	z=301,2	C=-6,71

Realizado por: Bautista A., & Oña D., 2023

En la Ilustración 4-13 se observa el control del cuarto grado de libertad del robot, para lo cual se combinó los gestos BO-04 y BO-08, por ende, el brazo robótico gira hacia la izquierda.



Ilustración 4-13: Movimiento de la junta 4

Realizado por: Bautista A., & Oña D., 2023

Tabla 4-5: Coordenadas de la posición final de la junta 4

Posición Inicial		Posición Final	
x=573,5	A=42,35	x=618	A=25,28
y=-637	B=-6,01	y=-397	B=8,48
z=301,2	C=-6,71	z=262	C=31,3

Realizado por: Bautista A., & Oña D., 2023

En la Ilustración 4-14 se observa el control del quinto grado de libertad del robot, para lo cual se combinó los gestos BO-05 y BO-08, por esto, el eslabón se mueve hacia abajo.



Ilustración 4-14: Movimiento de la junta 5

Realizado por: Bautista A., & Oña D., 2023

Tabla 4-6: Coordenadas de la posición final de la junta 5

Posición Inicial		Posición Final	
x=618	A=25,28	x=90,48	A=24,33
y=-397	B=8,48	y=-698	B=-19,2
z=262	C=31,3	z=225,4	C=23,49

Realizado por: Bautista A., & Oña D., 2023

En la Ilustración 4-15 se observa el control del sexto grado de libertad del robot, para lo cual se combinó los gestos BO-06 y BO-09, por ende, el gripper gira hacia la derecha.



Ilustración 4-15: Movimiento de la junta 6

Realizado por: Bautista A., & Oña D., 2023

Tabla 4-7: Coordenadas de la posición final de la junta 6

Posición Inicial		Posición Final	
x=90,48	A=24,33	x=458,4	A=82,01
y=-698	B=-19,2	y=-578	B=-12,8
z=225,4	C=23,49	z=206,8	C=24,34

Realizado por: Bautista A., & Oña D., 2023

En la Ilustración 4-16 se observa el control del gripper, para el cierre del gripper se combinó los gestos BO-07 y BO-08, mientras que para su apertura se combinó el gesto BO-07 con el BO-09.

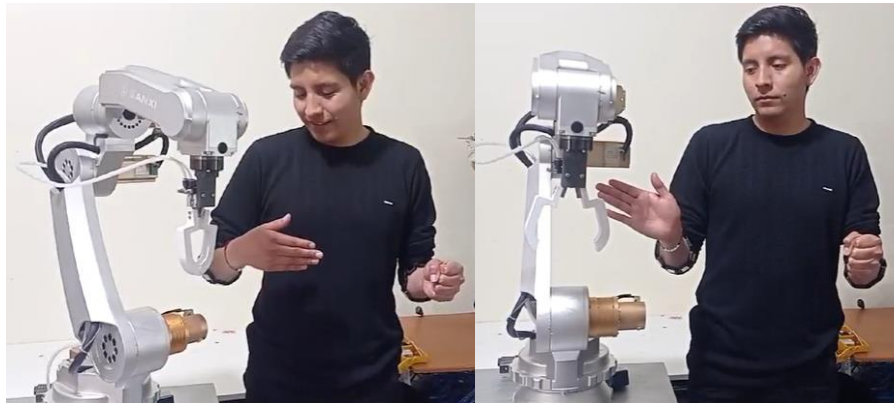


Ilustración 4-16: Control del gripper

Realizado por: Bautista A., & Oña D., 2023

4.3. Análisis de reconocimiento de gesto - Brazalete principal

Para obtener los resultados del reconocimiento de gestos entrenados en el brazalete principal se empleó una base de datos, donde el algoritmo analizó el número óptimo de muestras requeridas para lograr un reconocimiento eficiente, en este análisis se contempló tres conjuntos de muestras, específicamente con tamaños de 800, 1600 y más de 2400 registros.

Los valores obtenidos de este estudio se presentan en la Tabla 4-8, donde se observa que, con 800 muestras, la tasa de eficacia del clasificador es del 64%, un porcentaje insatisfactoriamente bajo para su aplicación. Por otra parte, al aumentar las muestras a 1600, se logró una eficacia del 92%, un valor que resulta adecuado para su implementación práctica, sin embargo, el mayor nivel de rendimiento se alcanza con 2400 muestras, donde la eficacia asciende al 96%, lo cual lo convierte en ideal para su uso efectivo.

Tabla 4-8: Reconocimiento de gesto del brazalete principal

# pruebas	n=800	n=1600	n=2400
1	0	1	1

2	1	1	1
3	1	1	1
4	1	0	1
5	0	1	1
6	1	1	1
7	0	1	1
8	0	1	1
9	1	1	1
10	1	1	0
11	1	1	1
12	0	1	1
13	1	1	1
14	0	1	1
15	1	0	1
16	1	1	1
17	0	1	1
18	1	1	1
19	0	1	1
20	1	1	1
21	1	1	1
22	0	1	1
23	1	1	1
24	1	1	1
25	1	1	1
Eficacia %	64	92	96

Realizado por: Bautista A., & Oña D., 2023

4.3.1. Latencia de tiempo de respuesta - Brazalete principal

Se llevó a cabo múltiples pruebas, dividiéndolas en tres grupos que compartían el mismo número de muestras, en cada caso, se midió el tiempo de respuesta necesario para determinar la clasificación, lo que arrojó los siguientes resultados: con 800 muestras, el tiempo promedio de clasificación o reconocimiento fue de 615,76 ms. Para un conjunto de 1600 muestras, el tiempo requerido aumentó a 1396,36 ms, y finalmente, en el caso de 2400 muestras, el tiempo se extendió a 2224,2 ms.

Tabla 4-9: Tiempo de respuesta del brazalete principal

N° de pruebas	n=800	n=1600	n=2400
1	600	1330	2368
2	700	1253	1968
3	500	1506	2054
4	651	1485	1963
5	485	1695	1875
6	695	1485	2569

7	700	1200	2121
8	651	126	2695
9	784	148	2148
10	524	1964	1986
11	620	1300	1994
12	652	1752	1896
13	752	1375	2014
14	601	1269	2368
15	547	1498	2254
16	400	1574	2485
17	586	1685	2148
18	634	1965	2136
19	586	1452	2896
20	598	1945	2474
21	604	1369	2301
22	702	1358	2485
23	685	1395	2105
24	532	1378	2163
25	605	1402	2139
T. Promedio (ms)	615,76	1396,36	2224,2

Realizado por: Bautista A., & Oña D., 2023

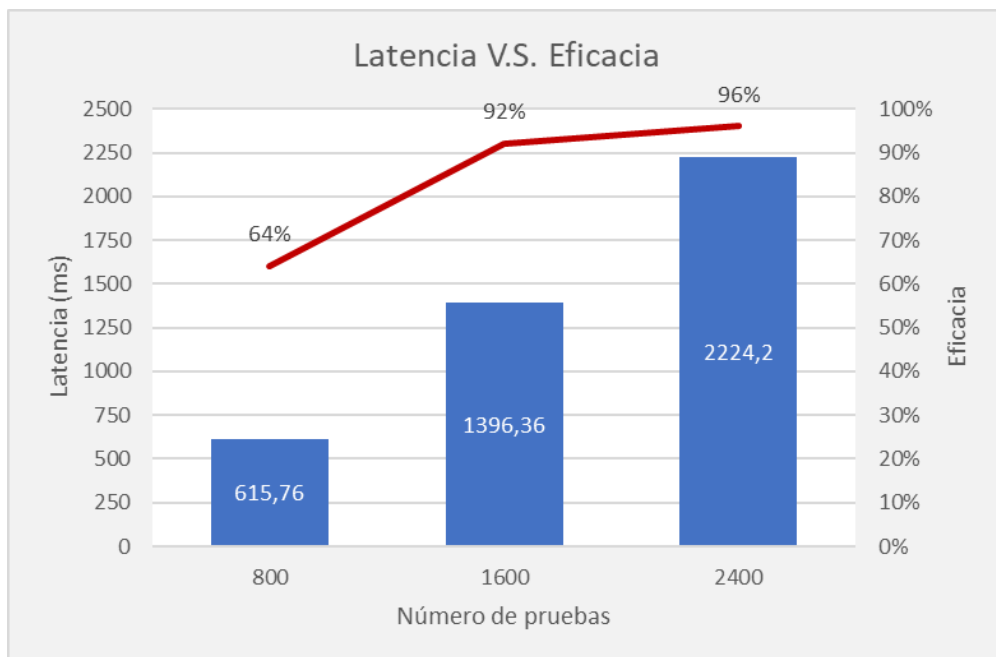


Ilustración 4-17: Eficacia-Brazalete secundario

Realizado por: Bautista A., & Oña D., 2023

La Ilustración 4-17 muestra que la elección apuntaría hacia las 2400 muestras como el punto ideal ya que, las pruebas demostraron que el uso de 800 muestras no alcanzaba los niveles requeridos de eficacia, por lo tanto, como resultado, se concluye que un número de muestras adecuado con un tiempo de respuesta aceptable se encuentra alrededor de las 1600 a 2400 muestras, generando

un tiempo promedio que oscila de 1396.36 a 2224.2 ms, respectivamente, obteniendo una eficacia mayor al 90%.

4.4. Análisis de reconocimiento de gestos - Brazalete secundario y Arduino

Durante las pruebas de reconocimiento de gestos del brazalete secundario, el Arduino se configuró para identificar tres gestos específicos: el gesto de movimiento hacia la derecha, izquierda y el stop, por lo tanto, estos gestos se traducen en un sentido positivo, negativo y cero, respectivamente.

Tabla 4-10: Reconocimiento de gestos de brazalete secundario

N° de pruebas	Gesto (+)	Gesto (S)	Gesto (-)
1	1	1	0
2	1	1	1
3	1	1	1
4	1	1	1
5	1	0	1
6	1	1	1
7	1	1	1
8	1	1	1
9	1	1	0
10	1	1	1
11	1	1	1
12	1	1	1
13	1	0	1
14	1	1	1
15	1	1	1
16	1	1	1
17	1	1	1
18	0	1	1
19	1	1	1
20	1	1	1
21	1	1	1
22	1	0	1
23	1	1	1
24	1	1	1
25	1	1	1
Aciertos	24	22	23

Realizado por: Bautista A., & Oña D., 2023

En este proceso, se llevó a cabo un total de 25 pruebas, los resultados de estas demostraron una eficacia del 96% en el reconocimiento del movimiento positivo, mientras que, para el gesto de movimiento negativo y el stop, se obtuvo una tasa de aciertos del 92% y 88%, como se muestra

en la Ilustración 4-18. Ambos porcentajes se sitúan dentro del rango de aceptación establecido para el prototipo en cuestión, cabe recalcar que el valor de 1 se da cuando se detecta el gesto y cero cuando no se lo detecta.

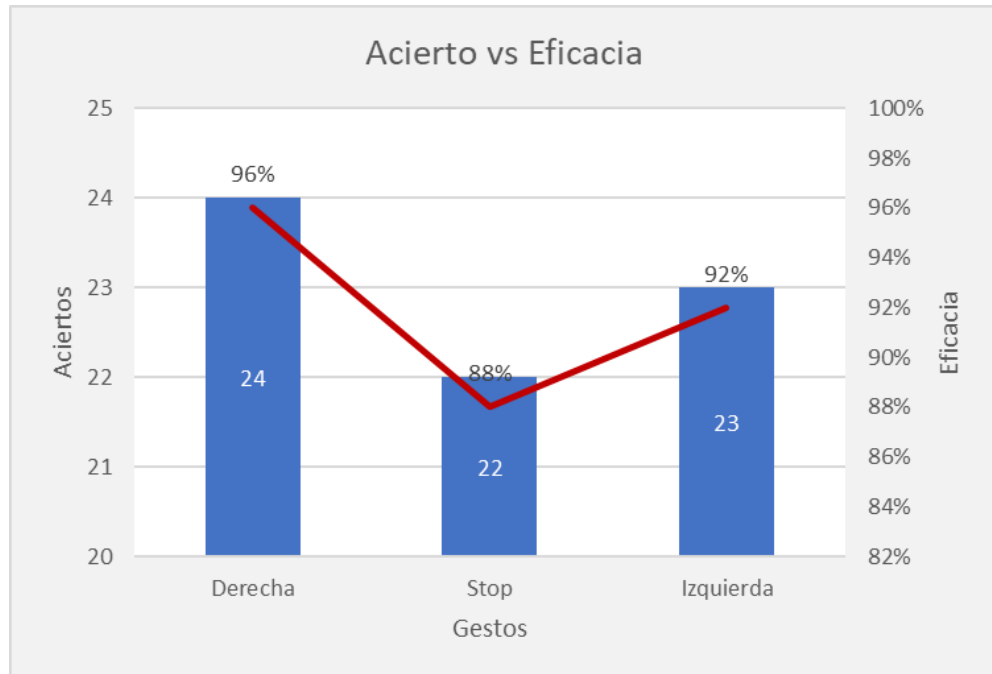


Ilustración 4-18: Eficacia-Brazaletes secundario

Realizado por: Bautista A., & Oña D., 2023

4.4.1. Latencia del tiempo de respuesta - Brazaletes secundario

Para esta prueba se llevaron a cabo pruebas utilizando 25 muestras para determinar el tiempo de respuesta del sistema de detección del brazaletes. Los resultados revelaron que, para el gesto hacia la derecha, se registró un tiempo promedio de latencia de 400 ms, mientras que para el gesto hacia la izquierda se obtuvo un tiempo de latencia de 300 ms. Por otro lado, el gesto de "stop" exhibió una latencia significativamente mayor, alcanzando los 601 ms. Este análisis indica que el reconocimiento del gesto de "stop" presenta una demora superior en comparación con el gesto hacia la derecha, que exhibe una latencia más breve.

Tabla 4-11: Tiempo de reconocimiento de brazaletes secundario

GESTOS	TIEMPO PROMEDIO (ms)
Izquierda	400
Stop	601
Derecha	300

Realizado por: Bautista A., & Oña D., 2023

4.5. Tiempo de respuesta a diferentes velocidades

Las pruebas ejecutadas en el brazo robótico se centraron en el análisis del tiempo de respuesta, es decir, el intervalo requerido para alcanzar el ángulo deseado. Estos resultados se muestran en la Tabla 4-12, donde se consideran diversos ajustes de velocidad propios del robot y “seteando” un ángulo de 90°, es evidente en los datos que a medida que se incrementa la velocidad, el robot es capaz de alcanzar el ángulo deseado en un lapso menor de tiempo.

Tabla 4-12: Tiempo de respuesta de eslabones acorde a modificación de velocidad

Velocidad	Eslabón 1	Eslabón 2	Eslabón 3	Eslabón 4	Eslabón 5	Eslabón 6
5%	10,36	10,28	12,52	19,9875	16,3025	32,43
20%	2,5	2,41	2,63	4,87	3,91	8,05
50%	0,88	0,88	0,65	1,88	1,52	3,09
75%	0,56	0,20	0,54	1,23	1,00	2,04
100%	0,37	0,38	0,03	0,9	0,78	1,47

Realizado por: Bautista A., & Oña D., 2023

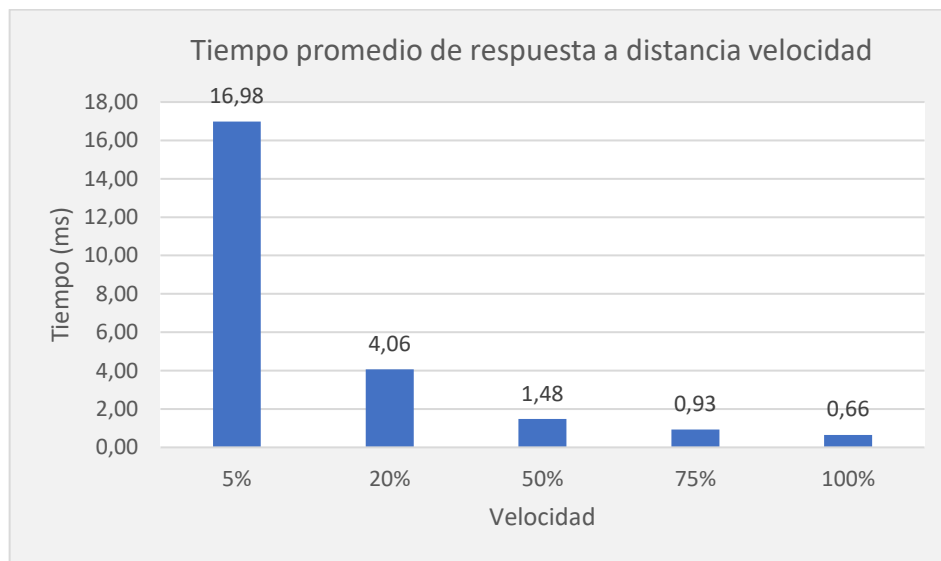


Ilustración 4-19: Tiempo respuesta a distinta velocidad

Realizado por: Bautista A., & Oña D., 2023

En base a la Ilustración 4-19 se puede determinar que el menor tiempo de respuesta promedio empleado por el robot es de 0,66 ms a una velocidad del 100%, sin embargo, para la implementación del control remoto basado en señales EMG, se ha elegido una velocidad del 75%, considerando diversos factores que afectan la seguridad y entorno del sistema.

Trabajar a una velocidad del 75% minimiza los riesgos asociados con movimientos rápidos, especialmente en entornos donde hay interacción con humanos u otros objetos sensibles, también

permite realizar movimientos más controlados y precisos, lo cual es crítico en aplicaciones que requieren alta precisión, como ensamblaje o manipulación de objetos delicados. Por otra parte, operar a una velocidad reducida puede disminuir el desgaste mecánico en los componentes del brazo robótico, prolongando la vida útil del sistema y reduciendo los costos de mantenimiento.

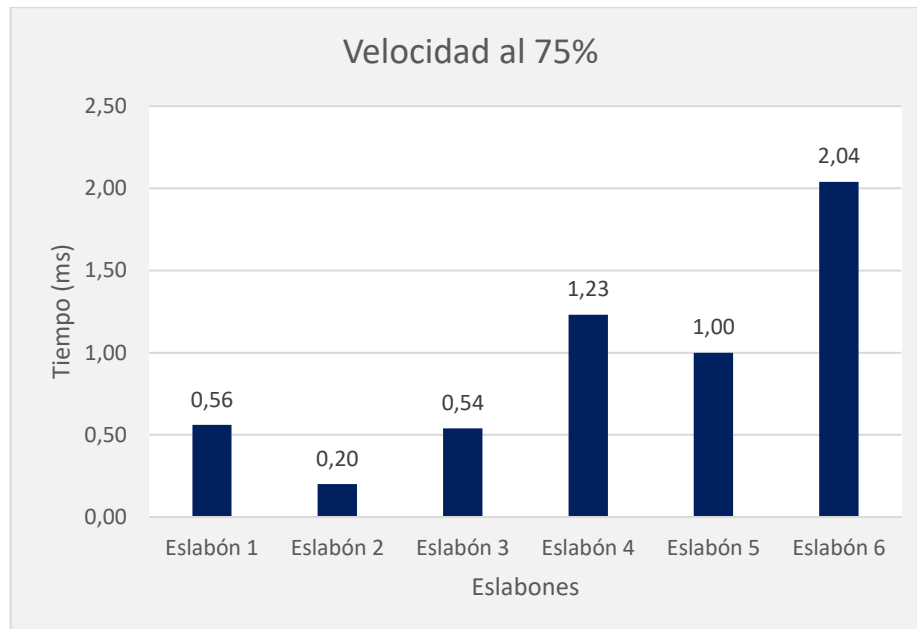


Ilustración 4-20: Tiempo respuesta a una velocidad del 75%

Realizado por: Bautista A., & Oña D., 2023

En la Ilustración 4-20, se ha realizado un análisis del tiempo de latencia de respuesta para cada eslabón del brazo robótico. Se observa que los tiempos de latencia se encuentran dentro de un rango de 0,20 a 2.04 ms, siendo el segundo eslabón el que presenta menor tiempo y el sexto eslabón el que tiene mayor tiempo. Es importante destacar que estos resultados se basan en una velocidad del 75%, ya que esta fue la velocidad de referencia utilizada durante la implementación. El tiempo respuesta por eslabón a una velocidad del 5, 20, 50 y 100% se puede observar en los anexos adjuntos.

4.6. Evaluación de rendimiento del sistema de control

Con un rango óptimo de muestras, de 1600 a 2400 y una velocidad del 75%, se llevó a cabo un total de 50 pruebas para evaluar la eficacia de cada grado de libertad, como se detalla en la Tabla 4-13. Los resultados obtenidos de estas pruebas presentan un alto nivel de éxito en la clasificación, alcanzando una tasa de reconocimiento del 100% en ciertos gestos, y no bajando de un 96% como valor mínimo, se consolida un sistema de clasificación o reconocimiento de gestos con un promedio general del 98.86%.

Tabla 4-13: Reconocimiento de gestos

GESTO	# PRUEBAS	# ACIERTOS	EFICACIA
1 er GDL	50	50	100%
2 do GDL	50	50	100%
3 er GDL	50	50	100%
4 to GDL	50	49	98%
5 to GDL	50	49	98%
6 to GDL	50	48	96%
Gripper	50	50	100%
Valor total promedio			98,86%

Realizado por: Bautista A., & Oña D., 2023

4.6.1. Tiempos de respuesta

Se ejecutó un total de 50 pruebas por cada eslabón, durante estas pruebas, se midió el tiempo desde el momento en que el gesto fue reconocido hasta que se realiza la acción correspondiente, contemplando una velocidad del 75% de la capacidad total del robot y con un incremento de ángulo de 3, evitando aumentarla más allá de este límite recomendado.

Tabla 4-14: Tiempo de respuesta

Gesto	Número de pruebas	Tiempo Promedio (ms)
1 er GDL	50	2369
2 do GDL	50	2147
3 er GDL	50	2369
4 to GDL	50	2478
5 to GDL	50	2361
6 to GDL	50	2510
Gripper	50	2145

Realizado por: Bautista A., & Oña D., 2023

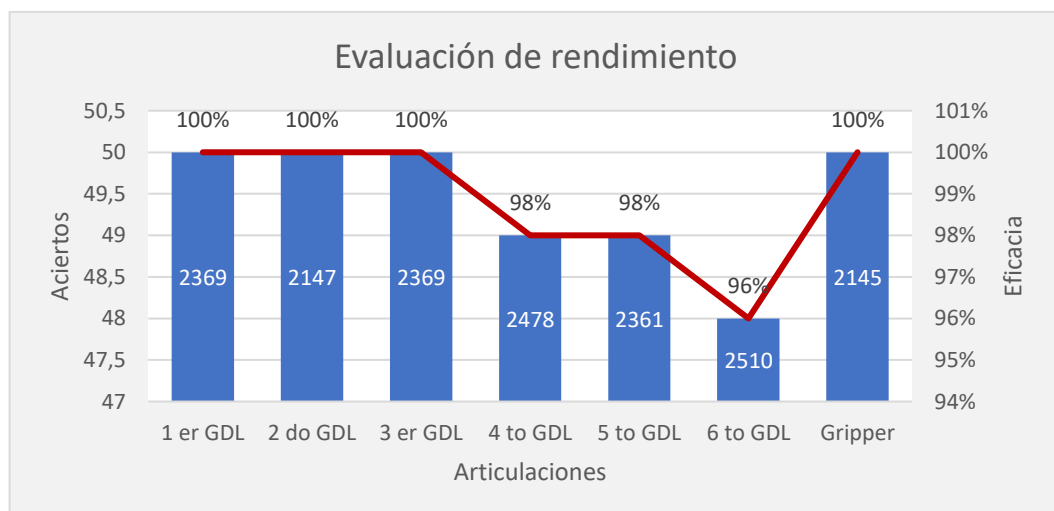


Ilustración 4-21: Evaluación de rendimiento

Realizado por: Bautista A., & Oña D., 2023

La Ilustración 4-21, muestra el tiempo que transcurre entre el inicio de la acción y la observación de su efecto, este lapso está influenciado por la velocidad con la cual una persona puede alternar entre diferentes gestos para cambiar de movimiento.

4.7. Gestión del proyecto

4.7.1. Cronograma

La duración de las tareas durante la ejecución del proyecto de titulación se ajustó conforme se especifica en la tabla adjunta. El plan de actividades se estructuró según el cronograma presentado a continuación:

Tabla 4-15: Cronograma de las actividades

Actividades	Abril				Mayo				Junio				Julio			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Análisis de Software Actual																
Análisis bibliográfico y recolección de información referente al estudio																
Creación de la interfaz para el control del robot ANNO RV264																
Evaluación del comportamiento del Robot mediante señales EMG																
Obtención de datos para el análisis de resultados																
Etapa de redacción del informe final																

Realizado por: Bautista A., & Oña D., 2023

4.7.2. Costos

Los costos para la implementación del sistema de control de movimiento remoto del brazo ANNO RV624 basado en señales electromiográficas, se detalla en la **Tabla 4-16**.

Tabla 4-16: Costos totales para la implementación del proyecto

N°	Detalle	Cantidad	Valor/unidad	Valor Total
1	Soporte metálico	1	\$ 130,00	\$ 130,00
2	Juego de hexagonales milimétricos	1	\$ 12,35	\$ 12,35
3	Arduino uno	1	\$ 16,00	\$ 16,00
4	Módulo HM11	1	\$ 10,30	\$ 10,30
5	Pernos milimétricos	6	\$ 0,80	\$ 4,80
6	Impresión 3D – Soporte del Gripper	1	\$ 25,00	\$ 25,00
7	Impresión 3D – Gripper	1	\$ 30,00	\$ 30,00
7	MYO Armband	2	\$ 150,00	\$ 300,00
	Impresión de placa electrónica	1	\$ 20,00	\$ 20,00
8	Streten HUB USB de cuatro puertos	1	\$ 10,00	\$ 10,00
9	Extensión de tomacorriente	1	\$ 14,60	\$ 14,60
10	Juego de desarmadores	1	\$ 3,45	\$ 3,45
TOTAL				\$ 576,50

Realizado por: Bautista A., & Oña D., 2023

4.7.3. *Recurso humano*

En la implementación del presente proyecto, el principal recurso humano fueron sus autores Adriana Maribel Bautista Guanin y Ermenson David Oña Chanaluisa, autores del proyecto. Su participación, respaldada por la aplicación de una metodología y un compromiso, posibilitó el logro satisfactorio de los objetivos propuestos.

4.7.4. *Recurso material*

Las herramientas de software y la plataforma mencionadas previamente ofrecen diversas opciones para el control del brazo robótico, permitiendo a los usuarios seleccionar aquellas que mejor se ajusten a sus preferencias y necesidades. Los costos asociados fueron asumidos por los estudiantes autores de este trabajo, con el propósito de proporcionar una alternativa para desarrollar una interfaz destinada al control remoto del brazo robótico ANNO RV624 basada en señales electromiográficas. Además, esta iniciativa busca servir como cimiento para futuras investigaciones, con el objetivo de contribuir al avance en el campo de la investigación.

CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

El principio de funcionamiento del sistema del robot ANNO RV624 se determinó mediante un análisis de sus componentes, este robot consta de seis grados de libertad y utiliza un controlador servo con seis motores paso a paso e incorpora un bucle semicerrado por pulsos y un interruptor de límite positivo y negativo de seis vías. En términos de parámetros de control, posee entrada y salida de 24 puntos y una interfaz de punto de entrada/salida de transmisión en serie de 256 puntos. Su funcionalidad incluye el soporte directo para operaciones de código G, además es importante destacar que es compatible con diversos sistemas operativos, como Windows, WinCE, Linux, Android, entre otros.

A través del proceso de selección de hardware para la adquisición de señales electromiográficas, se estableció el uso del MYO Armband, un dispositivo que tiene la capacidad para procesar y capturar señales EMG de manera inalámbrica, además que se destaca por su diseño ergonómico. Así también, el MYO Armband facilitó la integración fluida con Python, gracias a la librería PYOMYO que permite la tarea de capturar, analizar y utilizar las señales electromiográficas de manera eficaz.

La aplicación informática desarrollada para la calibración del sistema de señales electromiográficas, se implementó en Python con el apoyo de Pycharm, Python como software de programación mediante la librería Classifier y Pycharm facilitó el desarrollo al ofrecer soporte para frameworks y herramientas de inspección y corrección del código. También se utilizó Qt Designer donde se diseñó la interfaz de usuario, que abarca funciones clave como la calibración de gestos, el control manual y automático del brazo robótico.

Se generó un sistema de control que posibilita la transmisión de comandos de movimiento a partir de las señales musculares del usuario. En relación con el brazalete principal, este sistema presentó una latencia de tiempo de respuesta que oscila entre 1875 y 2896 ms, considerando un rango ideal de muestras entre 1600 y 2400, mientras que para el brazalete secundario se generó una latencia promedio de 433,66 ms, alcanzando así ambos una eficacia del 96%, sin embargo, se puede evidenciar un margen de error del 4% en la vinculación entre el sistema de control de los brazaletes MYO y el brazo robótico ANNO RV624.

Las pruebas de funcionalidad evidenciaron la capacidad del sistema implementado para controlar el brazo robótico ANNO RV624, registrando tiempos de respuesta con una latencia comprendida entre 2145 y 2510 ms con una velocidad del 75%, siendo el gripper el componente más rápido con 2145 ms, seguido por el segundo GDL con 2147 ms y el sexto GDL con la mayor latencia de 2510 ms. Además, al considerar un conjunto de 1600 muestras, se alcanzó una eficacia del 98,86%.

En relación con la cinemática del brazo robótico se optó un enfoque basado en la cinemática directa, utilizando el algoritmo de Denavit-Hartenberg, se seleccionó este método debido a la disponibilidad de las longitudes de los eslabones y los ángulos del brazo en su posición inicial, por lo tanto, cada vez que se ejecutan movimientos, los ángulos de las articulaciones tomaran valores variables, que permiten determinar en tiempo real las coordenadas x , y , z y del efector del robot.

5.2. Recomendaciones

Para mantener la eficacia y relevancia del sistema implementado, se recomienda mantenerse al tanto de los avances tecnológicos en hardware, software y técnicas de procesamiento de señales EMG, por ello, se debe realizar investigaciones continuas que permita identificar oportunidades para mejorar la precisión, la velocidad de respuesta y la experiencia general del usuario.

Previo al uso del brazo robótico, es importante que el usuario verifique la correcta vinculación de ambos brazaletes con el ordenador. Para evitar posibles inconvenientes durante este proceso de aconseja iniciar vinculando el brazaletes principal (izquierdo) y, posteriormente, el brazaletes secundario.

Se recomienda evitar presionar el botón “Start” ubicado encima del paro de emergencia, con el fin de prevenir movimientos involuntarios o bruscos por parte del robot que podría causar daños en los componentes del brazo.

Realizar pruebas con distintos usuarios y recabar retroalimentación directa de ellos; esto no solo permitirá identificar posibles limitaciones del sistema, sino que también posibilita mejoras tanto la interfaz de usuario como la aplicación informática de calibración optimizando así su adaptabilidad.

BIBLIOGRAFÍA

1. **AGUIRRE, Gustavo; et al.** “Control de Señales EMG para el movimiento de un brazo robótico de tres grados de libertad”. *Culcyt Mecatrónica* [en línea], 2015, (México) 55(1), pp.8-13. [Consulta: 4 junio 2023]. ISSN 2007-0411. Disponible en: <https://erevistas.uacj.mx/ojs/index.php/culcyt/article/view/741>
2. **ASNALEMA, A.** Desarrollo de un sistema de visión artificial a través de una tarjeta de desarrollo para controlar el brazo robótico Mitsubishi. [En línea] (Trabajo de titulación). (Pregrado) Universidad Politécnica Salesiana. Quito-Ecuador. 2017. [Consulta: 5 junio 2023]. Disponible en: <https://dspace.ups.edu.ec/bitstream/123456789/14537/1/UPS%20-%20ST003185.pdf>
3. **AVILA, J. & FAJARDO, F.** Análisis entre señales electromiográficas en los músculos cervicales y el ángulo de inclinación de la cabeza en niños con PCI entre 5 y 10 años aplicada a terapia cervical. [En línea] (Trabajo de titulación). (Pregrado) Universidad Politécnica Salesiana. Cuenca-Ecuador. 2019. p. 16. [Consulta: 6 junio 2023]. Disponible en: <https://dspace.ups.edu.ec/bitstream/123456789/17997/1/UPS-CT008551.pdf>
4. **BARRIENTOS, Antonio; et al.** *Fundamentos de robótica*. [en línea]. 2. Madrid-España, 2010. 2da edición. [Consulta: 4 junio 2023]. Disponible en: https://web.instipp.edu.ec/Libreria/libro/fundamentos_de_robotica%20Segunda%20Edicion.pdf
5. **CALVOPIÑA, Z.** Almacén inteligente mediante robot omnidireccional y visión artificial. [En línea] (Trabajo de titulación). (Pregrado) Universidad Politécnica Salesiana. Quito, Ecuador. 2018. [Consulta: 10 junio 2023]. Disponible en: <https://dspace.ups.edu.ec/bitstream/123456789/16495/1/UPS-ST003840.pdf>
6. **CAPARROSO, Iván; et al.** “Robótica Industrial”. Facultad de Ingeniería [en línea], 1999. [Consulta: 20 junio 2023]. Disponible en: <https://dialnet.unirioja.es/servlet/articulo?codigo=5313913>
7. **CEDEÑO, R.** Diseño de un sistema de control de movimientos de agarre para una prótesis de mano robótica con brazalete MYO. [En línea] (Trabajo de titulación). (Pregrado) Escuela Superior Politécnica del Litoral, Electricidad y Computación, Electrónica y Automatización

- Industrial. Guayaquil-Ecuador. 2017. [Consulta: 20 junio 2023]. Disponible en: <https://www.dspace.espol.edu.ec/xmlui/bitstream/handle/123456789/42546/D-106304.pdf?sequence=1&isAllowed=y>
8. **CE. ENTEL.** “Impacto de la Robótica en las distintas Industrias”. CE Comunidad Empresas [en línea], 2022, [Consulta: 6 junio 2023]. Disponible en: <https://ce.entel.cl/articulos/impacto-de-la-robotica/#:~:text=La%20rob%C3%B3tica%20est%C3%A1%20permitiendo%20que,en%20otras%20tareas%20m%C3%A1s%20creativas>
 9. **COPACI, Dorin; et al.** “Evaluación y desempeño de los sensores Myo Armband y MindRove”. Bioingeniería [en línea], 2022, (España), [Consulta: 5 julio 2023]. Disponible en: <https://doi.org/10.17979/spudc.9788497498418.0058>
 10. **DATASCIENTEST.** *PyCharm*. [blog]. [Consulta: 10 noviembre 2023]. Disponible en: <https://datascientest.com/es/pycharm>.
 11. **DETTA, F.** Aplicación de IoT con diversas tecnologías inalámbricas. [En línea] (Trabajo de titulación). (Pregrado) Universidad de la República Uruguay, Montevideo-Uruguay. 2019. [Consulta: 24 octubre 2023]. Disponible en: <https://www.innovaciondigital360.com/i-a/robotica-industrial-que-es-como-funciona-y-areas-de-aplicacion/>
 12. **DIGITAL 360.** *Robótica industrial*. [blog], [Consulta: 13 agosto 2023]. Disponible en: <https://www.innovaciondigital360.com/i-a/robotica-industrial-que-es-como-funciona-y-areas-de-aplicacion/>
 13. **DING, F. & LIU, C.** “Aplicación del método de coordenadas fijas Denavit-Hartenberg para resolver el espacio de trabajo del brazo robótico de perforación”. *International journal of advanced robotic systems* [en línea], 2018, (España) 15(4). [Consulta: 10 septiembre 2023]. ISSN 1729-8814. Disponible en: <https://journals.sagepub.com/doi/epub/10.1177/1729881418793283>
 14. **ELECTRONICS, B.** *Sensor muscular MyoWare* [en línea]. 2023. [Consulta: 22 agosto 2023]. Disponible en: <https://littlebirdelectronics.com.au/products/myoware-muscle-sensor-41cff0c5-8b09-4dbb-b43c-d95df4b42853>.

15. **ELECTROSTORE, G.** *Sensor muscular ECG EMG 3,3V AD8832* [en línea]. 2019a. [Consulta: 22 agosto 2023]. Disponible en: <https://grupoelectrostore.com/shop/sensores/senales-bioelectricas/sensor-muscular-ecg-emg-33v-ad8832-arduino-raspberry/>
16. **ELECTROSTORE, G.** *Sensor muscular MYOWARE EMG* [en línea]. 2019b. [Consulta: 22 agosto 2023]. Disponible en: <https://grupoelectrostore.com/shop/sensores/senales-bioelectricas/sensor-muscular-myoware-emg/>
17. **ESCANDÓN, R.** Interfaz de control de un brazo robótico articulado basado en software de desarrollo integral. [En línea] (Trabajo de titulación). (Pregrado) Universidad Politécnica Salesiana, Cuenca-Ecuador. 2013. [Consulta: 6 junio 2023]. Disponible en: <https://dspace.ups.edu.ec/bitstream/123456789/4646/1/UPS-CT002637.pdf>
18. **GARCÍA, F.** “Utilidad de la electromiografía de superficie en rehabilitación”. Researchgate [en línea], 2017. [Consulta: 25 junio 2023]. Disponible en: <https://www.researchgate.net/publication/316588275>
19. **GOMEZ, L. & TRETO, H.** “Electromiografía de fibra única en el diagnóstico de la miastenia gravis”. *Revista de Neurología* [en línea], 2013, (EE. UU) 30(7). [Consulta: 8 agosto 2023]. Disponible en: DOI 10.33588/rn.3007.99564
20. **GUZMÁN, E. & MÉNDEZ, G.** “Electromiografía en las Ciencias de la Rehabilitación”. *Salud Uninorte* [en línea], 2018, (Colombia) 34(3), pp.753-765. [Consulta: 25 junio 2023]. Disponible en: <https://www.redalyc.org/journal/817/81759607022/html/>
21. **HERRMANN, M.** *Qt Designer Download for Windows and Mac*. [blog], [Consulta: 8 agosto 2023]. Disponible en: <https://build-system.fman.io/qt-designer-download>
22. **KUMAR, S.** *Introducción a la robótica* [en línea]. Monterrey-México, McGRAW-HILL/INTERAMERICANA, 2010. [Consulta: 4 junio 2023]. Disponible en: <https://web.instipp.edu.ec/Libreria/libro/Introduccion%20a%20la%20robotica.pdf>
23. **MANUAL ANNO RV624.** *YZ-AIMManual_v2.53*. [en línea], 2018. [Consulta: 9 noviembre 2023]. Disponible en https://www.robotanno.com/web/userfiles/download/ACCESSORIES/IntegratedServoMotor/YZ-AIMManual_v2_55.pdf

24. **MERA, Jhon; & QUICALIQUIN, Cristian.** Desarrollo de una arquitectura software para teleoperación y monitoreo del robot ANNO RV624 (Trabajo de titulación) (Pregrado). [En línea] Escuela Superior Politécnica de Chimborazo. Riobamba-Ecuador. 2021. pp. 7-75 [Consulta: 2023-06-06]. Disponible en: <http://dspace.esPOCH.edu.ec/bitstream/123456789/15788/1/85T00624.pdf>
25. **POSSO, María Fernanda Villoria; et al.** “Sistema automático para el control de un brazo robótico utilizando una banda de control por gestos y técnicas de aprendizaje de máquina”. *Asociación Colombiana de Facultades de Ingeniería* [en línea], 2020, (Colombia), pp. 1-3. [Consulta: 20 marzo 2023]. Disponible en: <https://acofipapers.org/index.php/eiei/article/view/844/848>
26. **MORENO SANZ, Álvaro.** Procesado avanzado de señal EMG. (Trabajo de titulación) (Pregrado). [En línea] Universidad Carlos III de Madrid. Leganés-España. 2017. pp. 17-19 [Consulta: 2023-07-20]. Disponible en: <http://hdl.handle.net/10016/28105>
27. **FLOREZ, J.; et al.** “Aplicaciones de las señales mioeléctricas para el control de interfaces hombre-máquina”. *Universidad Nacional de Educación a Distancia* [en línea], 2004, (España), pp. 2-3. [Consulta: 20 agosto 2023]. Disponible en: <http://e-spacio.uned.es/fez/eserv/taee:congreso-2004-1088/S3C01.pdf>
28. **ORTEGA GARCÍA, Aidán Alejandro; et al.** “Aplicación del análisis de señales EMGS para el control de una prótesis de mano”. *Sociedad Mexicana de Ingeniería Mecánica*. [en línea], 2018, (México) pp. 1. [Consulta: 05 julio 2023]. ISSN 2448-5551. Disponible en: https://somim.org.mx/memorias/memorias2018/articulos/A1_152.pdf
29. **PAKA.** *El impacto de los robots en la industria*. [en línea]. 2019. [Consulta: 5 junio 2023]. Disponible en: <https://paka.pt/el-impacto-de-los-robots-en-la-industria/?lang=es>
30. **PÁSTOR JÁCOME, Erick Samir.** Implementación de un sistema inclusivo basado en el uso del brazo robótico ANNO RV624 para alimentar a personas con discapacidad motora (Trabajo de titulación) (Pregrado). [En línea] Escuela Superior Politécnica de Chimborazo. Riobamba-Ecuador. 2022. pp. 18-73. [Consulta: 2023-07-23]. Disponible en: <http://dspace.esPOCH.edu.ec/bitstream/123456789/18172/1/85T00719.pdf>
31. **PÉREZ CISNEROS, Marco Antonio; et al.** *Fundamentos de robotica y mecatronica con Matlab y Simulink*. [en línea]. ISBN 978-84-9964-451-6. Madrid-España: RA-MA Editorial

- y Publicaciones, 2014. pp. 111-112 [Consulta: 20 marzo 2023]. Disponible en: <https://elibro.net/es/lc/epoch/titulos/106568?prev=as>
32. **PROAÑO, Daniel.** Sistema de adquisición de señales EMG de superficie multicanal para prótesis de miembro superior (Trabajo de titulación) (Pregrado). [En línea] Universidad Politécnica Salesiana. Cuenca-Ecuador. 2019. pp. 4-20 [Consulta: 2023-04-05]. Disponible en: <https://dspace.ups.edu.ec/bitstream/123456789/17636/1/UPS-CT008386.pdf>
 33. **REVISTA DE ROBOTS.** *¿Qué es la robótica y para qué sirve?* [blog]. 2023. [Consulta: 5 junio 2023]. Disponible en: <https://revistaderobots.com/robots-y-robotica/que-es-la-robotica/?cn-reloaded=1>
 34. **REYNOSO, Julinho.** Diseño y control de una mano robótica para función motora de personas discapacitadas (Trabajo de titulación) (Pregrado). [En línea] Universidad Continental. Huancayo-Perú. 2021. pp. 28-29 [Consulta: 2023-04-25]. Disponible en: https://repositorio.continental.edu.pe/bitstream/20.500.12394/10242/1/IV_FIN_111_TE_Reynoso_Villaverde_2021.pdf
 35. **ROBOTANNO.** Robot Anno Controller Specific. [en línea], 2021. [Consulta: 9 noviembre 2023]. Disponible en: <http://www.robotanno.com/web/userfiles/download/STM32Controller/RobotAnnoControllerSpecific.pdf>
 36. **RUBIO, Rafael.** “Aplicaciones de las señales electromiográficas”. *Encuentros en la Biología* [en línea], 1999, (España), n°. 53, pp. 1. [Consulta: 25 junio 2023]. ISSN 1134-8496. Disponible en: <https://www.uma.es/estudios/centros/Ciencias/publicaciones/encuentros/ENCUENTROS53/aplicaciones.html>
 37. **SEO.** *Guía de Redes Inalámbricas* [blog]. 2020. [Consulta: 2 agosto 2023]. Disponible en: <https://termired.com/redes-inalambricas-que-es/>
 38. **SOLANO, Juan.** Control del movimiento remoto del brazo robótico Mitsubishi usando el sistema operativo Ros a través de la Web (Trabajo de titulación) (Tesis de pregrado). [En línea] Universidad Politécnica Salesiana. Cuenca-Ecuador. 2023. pp. 51-64 [Consulta: 2023-03-29]. Disponible en: <https://dspace.ups.edu.ec/handle/123456789/24461>

39. **SUBERVIOLA, Aaron.** Control de un exoesqueleto mediante señales EMG (electromiográficas) (Trabajo de titulación) (Tesis doctoral). [En línea] Universidad del País Vasco. País Vasco-España. 2019. pp. 53-99 [Consulta: 2023-04-02]. Disponible en: <https://addi.ehu.es/handle/10810/41463>
40. **TURIBOT.** *Modulo Bluetooth HM-11.* [en línea]. 2023. [Consulta: 10 noviembre 2023]. Disponible en: <https://www.turibot.es/modulo-bluetooth-hm-11-compatible-cc2541>
41. **VERA TUMBACO, Edwin Alberto.** Desarrollo de un sistema de visión artificial asociado con el robot ANNO RV 624 para reconocimiento y clasificación de piezas (Trabajo de titulación) (Pregrado). [En línea] Escuela Superior Politécnica de Chimborazo. Riobamba-Ecuador. 2021. pp. 74-89 [Consulta: 2023-04-02]. Disponible en: <http://dspace.espace.edu.ec/bitstream/123456789/15952/1/85T00634.pdf>
42. **YIZHI.** *Servo-Motores.* [en línea]. 2020. [Consulta: 9 noviembre 2023]. Disponible en: http://www.yizhi.info/products_show.asp?id=54
43. **ZE, Jonathan.** Implementación de un sistema de clasificación de gestos del brazo humano utilizando MYO ARMBAND para mando a distancia de un brazo robótico de 3GDL (Trabajo de titulación) (Tesis de pregrado). [En línea] Escuela Politécnica Nacional. Quito-Ecuador. 2017. pp. 16-76 [Consulta: 2023-04-08]. Disponible en: <https://bibdigital.epn.edu.ec/bitstream/15000/19190/1/CD-8571.pdf>

ANEXOS

ANEXO A: TIEMPO DE RECONOCIMIENTO DE BRAZALETE SECUNDARIO

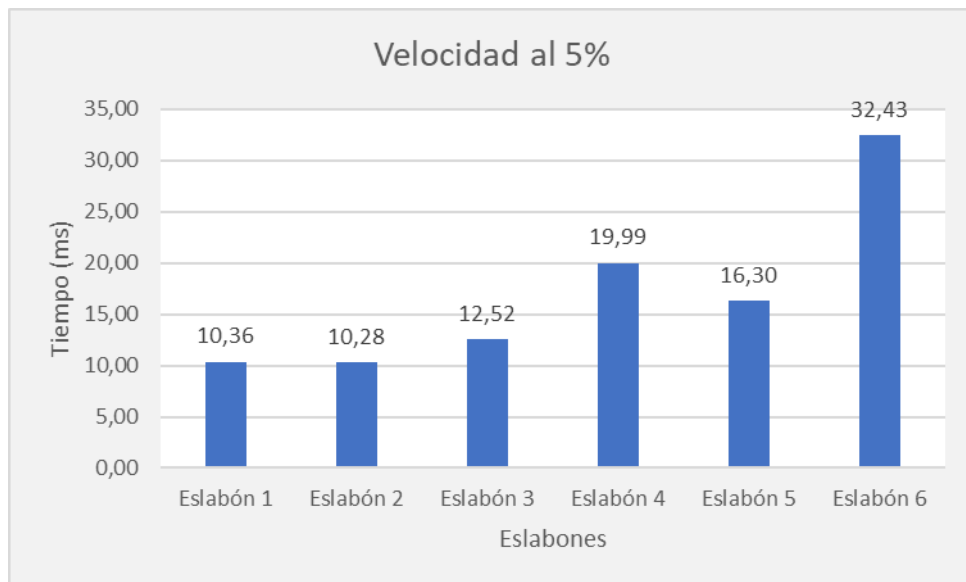
Izquierda	Stop	Derecha
391,70	593,60	287,68
392,15	600,35	299,65
400,27	593,87	306,75
401,05	598,51	296,04
409,24	602,54	291,51
401,99	604,46	297,33
406,23	593,72	295,36
397,58	595,29	292,48
410,44	589,33	300,94
400,51	597,21	305,30
403,05	596,83	309,53
398,60	589,36	313,09
401,31	610,74	291,67
407,06	595,66	299,62
400,58	601,51	303,60
403,07	608,09	300,24
402,10	595,11	297,59
395,49	608,98	299,99
393,53	597,35	304,78
402,90	592,01	301,98
395,14	597,05	291,78
398,69	599,81	300,53
394,99	609,34	309,04
392,79	595,86	299,15
398,08	603,53	306,39

ANEXO B: TIEMPOS DE LATENCIA PARA UNA VELOCIDAD DEL 5%

Velocidad	Eslabón 1	Eslabón 2	Eslabón 3	Eslabón 4	Eslabón 5	Eslabón 6
5%	10,26	9,53	15,89	19,41	14,40	31,50
	10,01	10,34	14,43	19,27	15,94	31,11
	9,17	9,79	13,91	20,54	15,99	32,74
	11,04	10,49	16,75	18,80	16,93	32,01
	9,43	10,27	14,73	20,84	16,64	33,06
	11,08	9,88	17,28	19,07	16,76	33,20
	9,27	10,66	16,96	19,95	16,12	32,17
	10,85	11,07	17,14	18,06	15,65	32,68
	11,56	9,79	15,52	20,91	16,88	33,64
	9,98	11,84	15,61	20,86	13,50	31,43

	10,54	9,43	15,48	20,03	17,05	32,33
	10,59	9,79	13,89	19,23	17,96	33,81
	10,07	11,16	16,47	18,88	14,81	31,57
	11,45	10,21	14,91	18,85	16,53	31,45
	10,41	8,80	15,31	19,20	16,97	33,55
	9,82	8,89	16,49	18,87	16,09	32,93
	11,69	10,34	15,85	18,65	15,69	32,50
	8,43	7,77	15,58	19,54	15,60	33,45
	10,87	8,26	16,24	19,10	16,61	32,87
	10,83	10,35	16,41	21,13	18,11	33,29
	10,94	9,25	16,17	19,04	16,53	31,66
	10,07	11,01	14,96	19,59	16,88	34,44
	11,46	11,59	16,59	20,03	17,00	31,92
	11,01	10,83	14,74	19,72	17,03	34,15
	11,67	11,56	13,94	21,83	16,99	31,82

ANEXO C: GRÁFICAS DE TIEMPO DE RESPUESTA A UNA VELOCIDAD DEL 5%

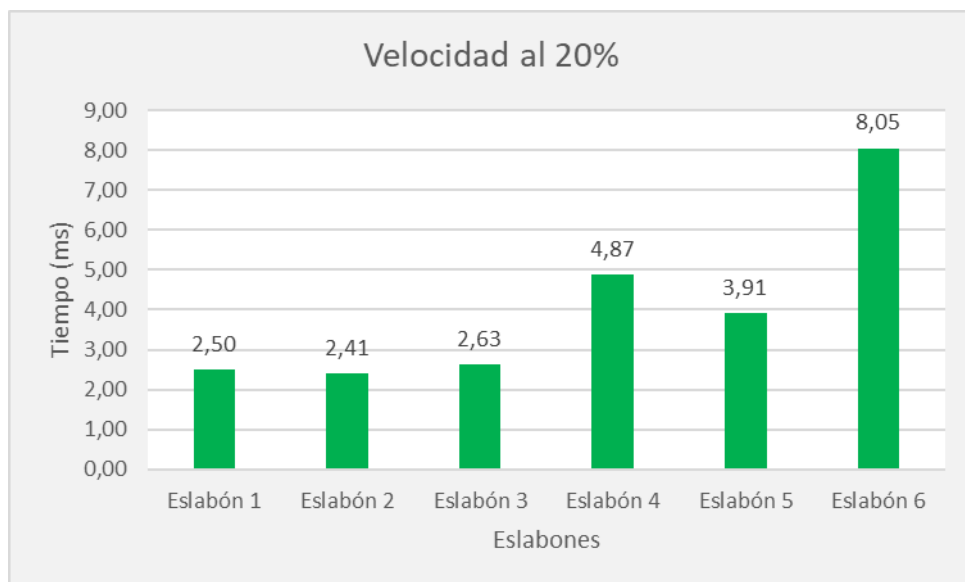


ANEXO D: TIEMPOS DE LATENCIA PARA UNA VELOCIDAD DEL 20%

Velocidad	Eslabón 1	Eslabón 2	Eslabón 3	Eslabón 4	Eslabón 5	Eslabón 6
20%	3,14	2,13	1,57	5,85	2,55	6,29
	3,42	3,10	2,38	5,74	3,95	8,44
	1,99	3,04	3,51	2,30	5,23	6,82
	3,67	1,99	2,93	3,70	2,89	8,08
	1,10	3,89	2,86	4,29	4,21	8,57
	3,49	2,44	3,47	5,38	3,92	8,36
	1,27	1,93	3,32	3,83	5,17	8,77
	2,12	3,00	3,56	3,42	3,75	8,57
	3,58	4,18	1,40	4,65	3,63	8,19
	2,29	3,19	1,83	3,64	3,64	6,22

	2,87	0,99	2,53	3,89	3,18	7,70
	3,56	1,64	0,46	4,32	4,52	7,20
	3,74	0,46	2,15	4,61	2,36	8,08
	2,71	3,68	1,49	3,55	3,89	7,64
	3,28	3,62	3,01	5,79	4,50	10,00
	2,99	3,10	2,21	5,02	5,29	6,67
	3,65	1,86	2,28	4,22	4,04	10,07
	2,77	2,70	2,91	5,35	4,36	8,23
	2,25	2,99	2,05	5,91	3,15	8,33
	3,78	3,99	3,18	4,26	3,50	7,69
	2,23	3,04	2,40	5,21	6,63	6,29
	1,44	2,66	2,35	3,92	6,25	7,34
	3,02	1,81	1,30	5,72	4,26	9,14
	2,52	4,60	2,53	5,62	2,56	8,69
	2,99	1,99	1,20	5,85	4,98	7,79

ANEXO E: GRÁFICAS DE TIEMPO DE RESPUESTA A UNA VELOCIDAD DEL 20%

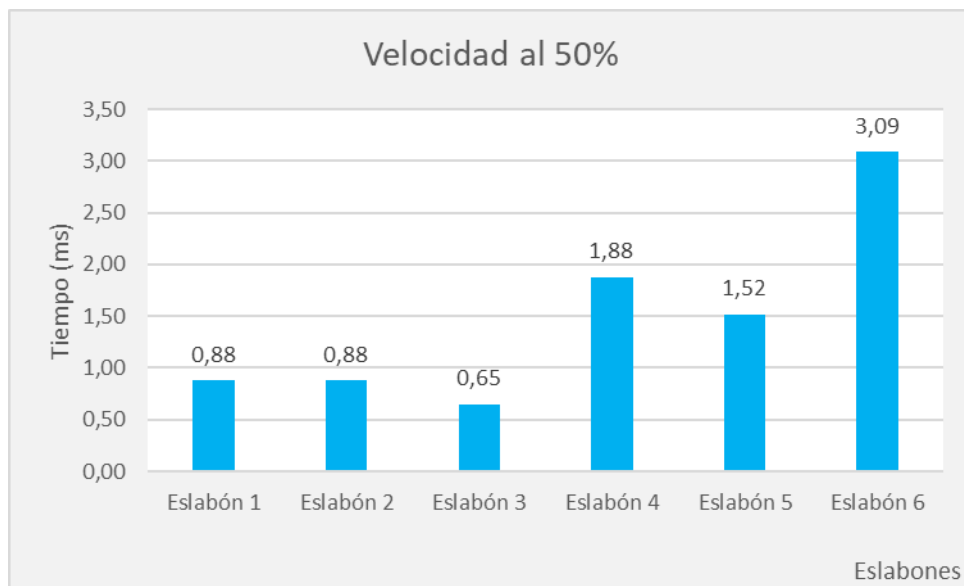


ANEXO F: TIEMPOS DE LATENCIA PARA UNA VELOCIDAD DEL 50%

Velocidad	Eslabón 1	Eslabón 2	Eslabón 3	Eslabón 4	Eslabón 5	Eslabón 6
50%	0,18	0,73	0,06	1,29	0,36	4,11
	0,38	1,86	0,96	1,64	0,66	4,40
	0,49	1,45	1,96	1,71	1,02	4,97
	0,93	1,11	0,14	2,24	1,54	4,21
	0,29	2,15	1,20	3,54	2,09	1,22
	1,81	1,03	2,26	2,62	0,52	2,68
	0,84	1,08	0,62	2,23	1,18	3,74
	1,32	0,49	0,18	1,98	1,08	3,37
	1,80	1,01	1,33	0,37	-0,04	3,39
	0,01	0,33	0,43	1,95	2,14	3,40

	0,13	1,25	0,28	0,74	1,03	2,26
	0,05	0,24	0,83	1,55	2,18	4,80
	1,02	0,82	1,10	2,76	3,04	4,65
	0,05	0,22	0,50	1,61	1,66	3,77
	0,11	1,35	0,27	2,57	1,37	4,77
	0,43	2,83	0,88	1,49	2,15	3,93
	1,35	1,31	0,10	2,79	1,77	2,52
	1,01	1,44	2,42	1,24	2,14	1,56
	-0,36	1,04	1,08	1,90	1,37	2,20
	0,92	1,70	1,01	2,21	2,89	3,68
	0,50	1,18	1,60	0,98	2,01	2,11
	0,05	1,31	0,85	0,78	2,15	3,59
	0,21	2,89	0,15	1,08	0,12	4,32
	1,41	3,04	1,33	2,67	2,37	4,10
	3,72	1,38	1,64	1,88	3,37	2,79

ANEXO G: GRÁFICAS DE TIEMPO DE RESPUESTA A UNA VELOCIDAD DEL 50%

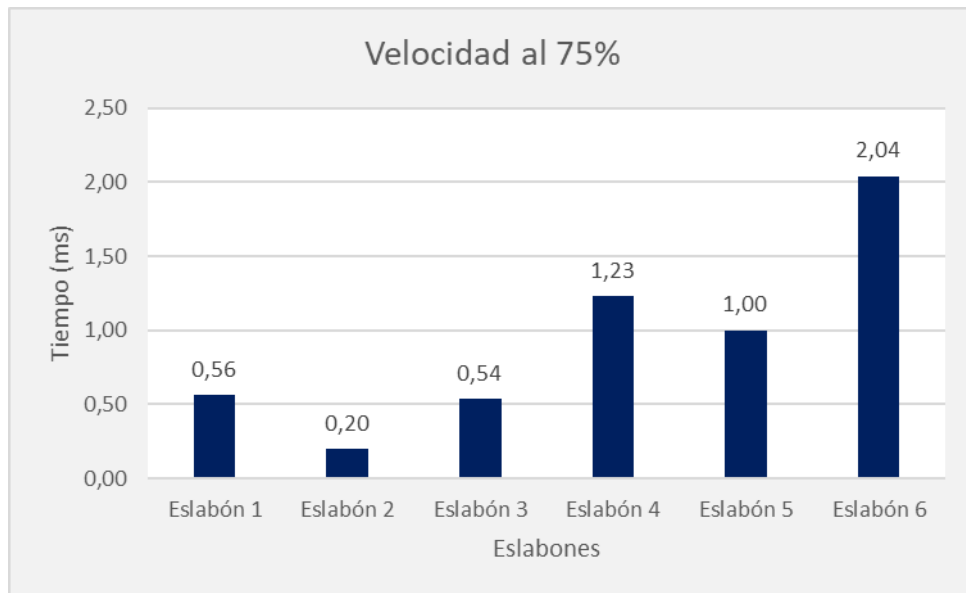


ANEXO H: TIEMPOS DE LATENCIA PARA UNA VELOCIDAD DEL 75%

Velocidad	Eslabón 1	Eslabón 2	Eslabón 3	Eslabón 4	Eslabón 5	Eslabón 6
75%	0,36	3,19	1,45	0,48	0,54	2,04
	3,03	1,32	0,15	0,79	0,79	2,37
	1,50	0,54	0,62	1,47	0,45	0,85
	0,48	0,02	0,13	0,87	1,78	2,55
	1,06	1,43	0,12	2,81	0,41	2,64
	2,25	0,29	0,05	0,73	1,70	2,61
	1,08	0,25	1,10	2,16	0,71	3,84
	2,27	0,11	0,38	0,77	1,38	2,18
	1,59	1,07	0,04	2,13	0,49	1,74
	0,31	0,05	0,23	0,64	0,54	3,01

	0,78	0,87	1,94	2,29	0,63	2,50
	0,31	1,48	2,05	0,59	0,57	3,35
	1,24	0,30	2,37	1,45	1,41	1,62
	1,33	2,20	0,41	1,51	0,15	4,04
	0,10	0,25	0,98	1,19	0,81	2,87
	0,20	0,22	0,30	0,81	0,61	2,44
	1,42	1,30	0,05	1,17	0,05	0,65
	1,49	0,73	0,86	-0,22	0,61	1,92
	2,48	0,40	0,52	1,30	0,49	2,09
	1,90	1,02	1,18	2,18	1,31	2,79
	1,70	0,70	1,39	0,90	1,47	1,88
	1,05	2,37	0,62	1,48	0,45	2,14
	0,94	1,04	2,16	0,67	0,86	1,01
	1,36	1,43	0,80	1,79	0,05	0,72
	0,71	0,68	0,10	3,20	0,24	1,65

ANEXO I: GRÁFICAS DE TIEMPO DE RESPUESTA A UNA VELOCIDAD DEL 75%

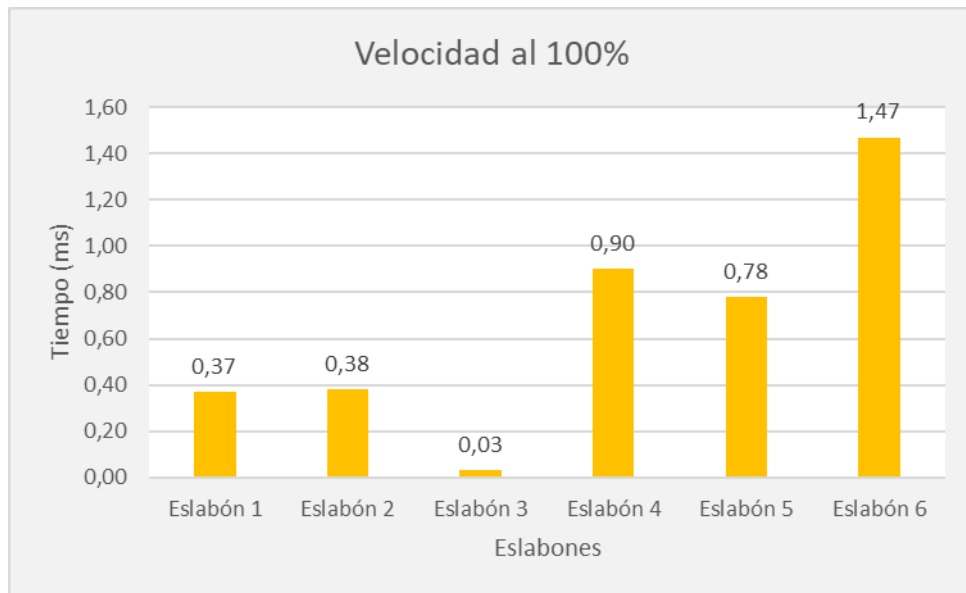


ANEXO J: TIEMPOS DE LATENCIA PARA UNA VELOCIDAD DEL 100%

Velocidad	Eslabón 1	Eslabón 2	Eslabón 3	Eslabón 4	Eslabón 5	Eslabón 6
100%	0,40	0,31	0,06	0,88	0,76	1,41
	0,34	0,38	0,06	1,04	0,76	1,52
	0,38	0,43	0,04	0,92	0,83	1,39
	0,43	0,48	0,04	0,88	0,74	1,36
	0,36	0,39	0,10	0,82	0,76	1,39
	0,32	0,41	0,11	1,00	0,83	1,51
	0,41	0,35	0,07	0,96	0,71	1,36
	0,42	0,51	0,02	0,90	0,73	1,48
	0,40	0,38	0,06	1,01	0,78	1,38
	0,29	0,32	0,07	0,83	0,70	1,47

	0,40	0,31	0,13	0,82	0,78	1,44
	0,41	0,30	0,07	0,89	0,78	1,51
	0,40	0,47	0,11	0,84	0,79	1,47
	0,36	0,38	0,06	1,13	0,78	1,38
	0,40	0,25	0,06	0,82	0,73	1,39
	0,36	0,37	0,08	0,92	0,75	1,44
	0,36	0,49	0,02	0,95	0,74	1,38
	0,38	0,37	0,00	0,90	0,79	1,37
	0,31	0,53	0,09	0,99	0,77	1,42
	0,46	0,34	0,09	0,91	0,78	1,48
	0,32	0,48	0,13	0,89	0,71	1,44
	0,32	0,39	0,02	0,85	0,81	1,45
	0,41	0,40	0,02	0,97	0,67	1,36
	0,36	0,26	0,02	0,90	0,74	1,44
	0,34	0,41	0,15	0,98	0,80	1,47

ANEXO K: GRÁFICAS DE TIEMPO DE RESPUESTA A UNA VELOCIDAD DEL 100%



ANEXO L: RECONOCIMIENTO DE GESTOS

N.º	1er GLD	2do GLD	3er GLD	4to GLD	5to GLD	6to GLD	Gripper
1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1

11	1	1	1	1	1	1	1
12	1	1	1	1	1	1	1
13	1	1	1	1	1	1	1
14	1	1	1	1	1	1	1
15	1	1	1	1	1	1	1
16	1	1	1	0	1	1	1
17	1	1	1	1	1	1	1
18	1	1	1	1	1	1	1
19	1	1	1	1	1	1	1
20	1	1	1	1	1	1	1
21	1	1	1	1	0	1	1
22	1	1	1	1	1	1	1
23	1	1	1	1	1	1	1
24	1	1	1	1	1	1	1
25	1	1	1	1	1	0	1
26	1	1	1	1	1	1	1
27	1	1	1	1	1	1	1
28	1	1	1	1	1	1	1
29	1	1	1	1	1	1	1
30	1	1	1	1	1	1	1
31	1	1	1	1	1	1	1
32	1	1	1	1	1	0	1
33	1	1	1	1	1	1	1
34	1	1	1	1	1	1	1
35	1	1	1	1	1	1	1
36	1	1	1	1	1	1	1
37	1	1	1	1	1	1	1
38	1	1	1	1	1	1	1
39	1	1	1	1	1	1	1
40	1	1	1	1	1	1	1
41	1	1	1	1	1	1	1
42	1	1	1	1	1	1	1
43	1	1	1	1	1	1	1
44	1	1	1	1	1	1	1
45	1	1	1	1	1	1	1
46	1	1	1	1	1	1	1
47	1	1	1	1	1	1	1
48	1	1	1	1	1	1	1
49	1	1	1	1	1	1	1
50	1	1	1	1	1	1	1

ANEXO M: TIEMPO DE RESPUESTA DE CADA ESLABÓN

N.º	1er GLD	2do GLD	3er GLD	4to GLD	5to GLD	6to GLD	Gripper
1	2505,92	2137,05	2222,04	2444,32	2298,02	2518,56	2065,59

2	2533,18	2207,85	2542,29	2561,24	2336,85	2410,34	2075,58
3	2260,23	2061,68	2579,01	2613,88	2360,37	2554,84	2098,94
4	2543,86	2142,00	2260,56	2480,59	2392,10	2419,93	2207,70
5	2241,67	2125,47	2551,63	2459,71	2276,47	2444,77	2204,31
6	2283,85	2291,82	2083,12	2635,06	2339,50	2613,00	2241,75
7	2289,05	2275,74	2275,26	2358,29	2338,56	2528,06	2145,25
8	2423,80	2039,87	2259,54	2488,26	2277,59	2405,69	2127,45
9	2266,52	2152,91	2398,50	2302,18	2234,86	2668,23	2422,89
10	2225,31	2260,22	2350,64	2478,95	2364,44	2381,55	2092,83
11	2557,06	2161,98	2181,52	2299,82	2320,09	2514,38	1928,13
12	2283,69	2141,34	2453,13	2517,38	2392,29	2467,35	2129,72
13	2472,82	2039,01	2122,17	2532,57	2270,10	2475,84	2154,39
14	2326,54	2176,43	2555,82	2488,35	2262,02	2537,54	2268,22
15	2370,31	2238,57	2317,02	2533,03	2504,54	2588,38	2026,64
16	2473,80	2077,22	2500,63	2661,58	2451,89	2493,12	2250,39
17	2216,86	2230,06	2259,78	2500,78	2379,69	2741,54	2129,61
18	2402,19	2198,42	2302,31	2512,04	2250,68	2444,98	1992,45
19	2298,00	2050,02	2526,06	2390,48	2412,47	2413,20	2086,06
20	2371,30	2084,27	2210,86	2614,54	2142,35	2329,75	2091,90
21	2372,93	2203,21	2211,82	2410,30	2357,79	2680,14	2322,89
22	2426,95	2154,33	2218,06	2477,60	2412,79	2718,49	2096,35
23	2350,33	2128,29	2539,90	2469,90	2324,32	2589,90	2131,53
24	2371,71	1985,28	2321,61	2597,34	2224,47	2338,65	2282,38
25	2184,15	2167,50	2443,99	2216,83	2238,78	2522,24	2132,84
26	2301,25	1933,59	2250,22	2479,35	2464,95	2412,24	2156,25
27	2574,62	2317,56	2481,39	2369,42	2333,50	2357,14	2053,89
28	2422,22	2044,43	2432,09	2487,03	2218,39	2584,96	2051,80
29	2315,80	2152,66	2413,79	2537,02	2322,11	2499,00	2124,31
30	2354,65	2081,87	2337,94	2473,30	2413,58	2441,54	2263,42
31	2233,80	2113,46	2403,91	2621,19	2397,65	2349,96	2216,78
32	2313,35	2129,10	2487,11	2504,76	2354,23	2513,01	2182,00
33	2396,90	2139,74	2307,92	2485,22	2302,90	2503,90	2128,54
34	2307,10	2185,72	2599,51	2502,44	2355,49	2385,60	2291,90
35	2195,61	2195,70	2693,53	2495,47	2454,10	2647,27	2391,29
36	2439,96	2170,05	2457,50	2561,57	2282,23	2492,88	2229,44
37	2376,62	2210,34	2322,52	2562,84	2457,64	2467,99	2028,54
38	2558,65	2166,62	2404,97	2401,49	2551,29	2278,72	2014,33
39	2328,92	2239,85	2411,17	2503,80	2329,85	2810,37	2149,30
40	2435,73	2162,08	2420,56	2418,37	2208,15	2595,86	2216,36
41	2408,91	2015,53	2625,82	2421,39	2427,98	2661,29	2075,89
42	2208,56	2141,83	2153,93	2426,40	2382,46	2599,04	2160,27
43	2368,32	2075,82	2077,41	2524,55	2161,52	2522,37	2171,59
44	2446,62	2095,51	2750,56	2478,93	2247,86	2493,91	1982,96
45	2337,42	2176,98	2358,97	2491,44	2328,65	2738,08	2046,46
46	2290,45	2225,43	2469,18	2431,49	2362,45	2386,52	2272,54

47	2421,88	2110,55	2300,92	2414,83	2365,69	2525,08	2142,68
48	2371,07	2248,81	2431,51	2496,11	2386,44	2379,87	2261,79
49	2261,45	2121,34	2195,99	2509,29	2193,05	2568,00	2098,82
50	2524,80	2031,39	2398,62	2608,20	2442,55	2477,61	1853,67

ANEXO N: EXTRACTO DE LA PROGRAMACIÓN EN PYTHON

```

import sys
import cv2
import time
from PyQt5.QtGui import *
from PyQt5.QtWidgets import *
from PyQt5.uic import *
from PyQt5.QtCore import QTimer
import numpy as np
from collections import Counter, deque
import struct
import pygame
from pygame.locals import *
from sklearn import neighbors
from pyomyo import Myo, emg_mode
from pyomyo.Classifier import Classifier, MyoClassifier, EMGHandler
import serial
import math
import threading

def ejecutar():
    while 1:
        m.run()
        print('aqui')

ser = serial.Serial('COM4', 115200, timeout=.1)
print(ser)

myo_arduino = serial.Serial('COM5', 115200, timeout=.1)
print(myo_arduino)

SUBSAMPLE = 3
K = 15
lim1_1 = -90
lim1_2 = 90
lim2_1 = -80
lim2_2 = 50
lim3_1 = -150
lim3_2 = 0
lim4_1 = -90

```



```
lim4_2 = 90
lim5_1 = -135
lim5_2 = -45
lim6_1 = -90
lim6_2 = 90
```

```
angJ1 = 0
angJ2 = 0
angJ3 = -90
angJ4 = 0
angJ5 = -90
angJ6 = 0
```

```
estado = 0
predA = 0
```

```
class KNN_Classifier(Classifier):
```

```
    def __init__(self):
```

```
        Classifier.__init__(self)
```

```
    def train(self, X, Y):
```

```
        self.X = X
```

```
        self.Y = Y
```

```
        self.model = None
```

```
        if self.X.shape[0] >= K * SUBSAMPLE:
```

```
            self.model = neighbors.KNeighborsClassifier(n_neighbors=K, algorithm='kd_tree')
```

```
            self.model.fit(self.X[::SUBSAMPLE], self.Y[::SUBSAMPLE])
```

```
    def classify(self, emg):
```

```
        global predA
```

```
        x = np.array(emg).reshape(1, -1)
```

```
        if self.X.shape[0] < K * SUBSAMPLE:
```

```
            predA = 0
```

```
            return 0
```

```
        pried = self.model.predict(x)
```

```
        predA = pried.copy()
```

```
        return int(pried[0])
```

```
# ***** message box *****
```

```
class CustomMessageBox(QMessageBox):
```

```
    def __init__(self, *_args):
```

```
        QMessageBox.__init__(self)
```

```
        self.timeout = 0
```

```
        self.autoclose = False
```

```

self.currentTime = 0

def showEvent(self, QShowEvent):
    self.currentTime = 0
    if self.autoclose:
        self.startTimer(1000)
# *****
class menu(QMainWindow):

    def __init__(self):
        super(menu, self).__init__()
        loadUi('guide/menu1-1.ui', self)
        self.calibra1.clicked.connect(self.c1)
        self.control_M.clicked.connect(self.m1)
        self.control_A.clicked.connect(self.a1)
        ser.write(str.encode("G06 O=P1.0\r\n"))
        hilo1 = threading.Thread(target=ejecutar)
        hilo1.start()

    def c1(self):
        self.hide()
        self.var1 = cali1()
        self.var1.setWindowTitle('calibracion Brazalete')
        self.var1.show()
# *****
class manual1(QMainWindow):

    def __init__(self):
        super(manual1, self).__init__()
        loadUi('guide/manual-1.ui', self)
        self.regresar.clicked.connect(self.back)
        self.home.clicked.connect(self.zero)
        self.mas_1.clicked.connect(self.mas1)
        self.mas_2.clicked.connect(self.mas2)
        self.mas_3.clicked.connect(self.mas3)
        self.mas_4.clicked.connect(self.mas4)
        self.mas_5.clicked.connect(self.mas5)
        self.mas_6.clicked.connect(self.mas6)
        self.menos_1.clicked.connect(self.menos1)
        self.menos_2.clicked.connect(self.menos2)
        self.menos_3.clicked.connect(self.menos3)
        self.menos_4.clicked.connect(self.menos4)
        self.menos_5.clicked.connect(self.menos5)
        self.menos_6.clicked.connect(self.menos6)
        self.garra.clicked.connect(self.herra)

```

```

self.ang1 = 0
self.ang2 = 0
self.ang3 = -90
self.ang4 = 0
self.ang5 = -90
self.ang6 = 0
self.he = 0
ser.write(str.encode("G00 J1=0 J2=0 J3=-90 J4=0 J5=-90 J6=0\r\n"))
time.sleep(0.5)
ser.write(str.encode("G06 O=P1.1\r\n"))
time.sleep(0.5)
ser.write(str.encode("G07 VP=75\r\n"))
self.timer = QTimer(self)
self.timer.timeout.connect(self.actualizar)
self.timer.start(100)
class auto1(QMainWindow):

```

```

def __init__(self):
    super(auto1, self).__init__()
    loadUi('guide/auto-1.ui', self)
    self.regresar.clicked.connect(self.back)
    self.start.clicked.connect(self.clinician)
    self.stop.clicked.connect(self.para)
    self.start_2.clicked.connect(self.I_rob)
    self.stop_2.clicked.connect(self.S_rob)
    self.timer = QTimer(self)
    self.timer.timeout.connect(self.actualize)
    self.timer.start(100)
    self.ang1 = 0
    self.ang2 = 0
    self.ang3 = -90
    self.ang4 = 0
    self.ang5 = -90
    self.ang6 = 0
    self.he = 0
    self.est = cv2.imread('guide/ima/base.png')
    self.J_1 = cv2.imread('guide/ima/J1.png')
    self.J_2 = cv2.imread('guide/ima/J2.png')
    self.J_3 = cv2.imread('guide/ima/J3.png')
    self.J_4 = cv2.imread('guide/ima/J4.png')
    self.J_5 = cv2.imread('guide/ima/J5.png')
    self.J_6 = cv2.imread('guide/ima/J6.png')
    self.H_1 = cv2.imread('guide/ima/HE.png')
    self.izq = cv2.imread('guide/ima/IZQ.png')
    self.der = cv2.imread('guide/ima/DER.png')

```

```

self.stp = cv2.imread('guide/ima/ST.png')
self.actT = False
self.actR = False
ser.write(str.encode("G00 J1=0 J2=0 J3=-90 J4=0 J5=-90 J6=0\r\n"))
time.sleep(0.5)
ser.write(str.encode("G06 O=P1.1\r\n"))
time.sleep(0.5)
ser.write(str.encode("G07 VP=75\r\n"))
self.val_M2='S'

```

```
def cinematica(self):
```

```
    L1 = 240
```

```
    L2 = 350
```

```
    L3 = 120
```

```
    L4 = 372
```

```
    L5 = 72
```

```
    L6 = 203
```

```
    pi = 3.1416
```

```
    Q1 = self.ang1
```

```
    Q2 = self.ang2
```

```
    Q3 = self.ang3
```

```
    Q4 = self.ang4
```

```
    Q5 = self.ang5
```

```
    Q6 = self.ang6
```

```
    q1 = Q1 * pi / 180
```

```
    q2 = Q2 * pi / 180
```

```
    q3 = Q3 * pi / 180
```

```
    q4 = Q4 * pi / 180
```

```
    q5 = Q5 * pi / 180
```

```
    q6 = Q6 * pi / 180
```

```
    X = L2 * math.cos(q1) * math.cos(q2) - (
```

```
        L3 + L4 + (math.pow(math.cos(pi / 2), 2) - math.pow(math.sin(pi / 2), 2) * math.cos(q5)) * (
```

```
            L5 + L6)) * (
```

```
            math.sin(pi / 2) * math.cos(q3) * (
```

```
                math.cos(q1) * math.sin(q2) + math.cos(pi / 2) * math.cos(q2) * math.sin(q1)) + math.sin(
```

```
                pi / 2) * math.sin(q3) * (
```

```
                    math.cos(q1) * math.cos(q2) - math.cos(pi / 2) * math.sin(q1) * math.sin(q2)) - math.cos(
```

```
                pi / 2) * math.sin(pi / 2) * math.sin(q1)) - math.sin(pi / 2) * (
```

```
                    math.sin(q3) * (
```

```
                        math.cos(q1) * math.sin(q2) + math.cos(pi / 2) * math.cos(q2) * math.sin(q1)) - math.cos(
```

```
                    q3) * (math.cos(q1) * math.cos(q2) - math.cos(pi / 2) * math.sin(q1) * math.sin(q2))) * (L5 + L6) * (
```

```
                        math.cos(q4) * math.sin(q5) + math.cos(pi / 2) * math.sin(q4) + math.cos(pi / 2) * math.cos(
```

```
                            q5) * math.sin(
```

```
                            q4)) - L2 * math.cos(pi / 2) * math.sin(q1) * math.sin(q2) + math.sin(pi / 2) * (L5 + L6) * (
```

$$\begin{aligned} & \text{math.cos}(\pi / 2) * \text{math.cos}(q4) - \text{math.sin}(q4) * \text{math.sin}(q5) + \text{math.cos}(\pi / 2) * \text{math.cos}(\\ & q4) * \text{math.cos}(\\ & q5)) * (\text{math.cos}(\pi / 2) * \text{math.cos}(q3) * (\\ & \text{math.cos}(q1) * \text{math.sin}(q2) + \text{math.cos}(\pi / 2) * \text{math.cos}(q2) * \text{math.sin}(q1)) - \text{math.sin}(q1) * (\\ & \text{math.pow}(\text{math.cos}(\pi / 2), 2) - 1) + \text{math.cos}(\pi / 2) * \text{math.sin}(q3) * (\\ & \text{math.cos}(q1) * \text{math.cos}(q2) - \text{math.cos}(\pi / 2) * \text{math.sin}(q1) * \text{math.sin}(q2))) \end{aligned}$$

El código de programación completo se encuentra en el CD proporcionados a biblioteca

ANEXO O: MATRICES DE DENAVIT-HARTENBERG

$$\mathbf{H30} = \mathbf{H1} * \mathbf{H2} * \mathbf{H3};$$

H30 =

$$\begin{bmatrix} \cos(\pi/2), & -\sin(\pi/2), & 1/2 - \cos(\pi/2), & 470*\cos(\pi/2) \\ \sin(\pi/2), & \cos(\pi/2)^2 + \cos(\pi/2)^3 - 1, & -(\sin(\pi/2)*(\cos(\pi/2) + 1))/2, & 235*\sin(\pi) \\ 1/2 - \cos(\pi/2), & (\sin(\pi/2)*(\cos(\pi/2) + 1))/2, & \cos(\pi/2)*(\cos(\pi/2) + \cos(\pi/2)^2 - 1), & 475 - 235*\cos(\pi) \\ 0, & 0, & 0, & 1 \end{bmatrix}$$

$$\mathbf{H63} = \mathbf{H4} * \mathbf{H5} * \mathbf{H6};$$

H63 =

$$\begin{bmatrix} \cos(\pi/2), & \sin(\pi/2), & \cos(\pi/2) - 1/2, & (267*\cos(\pi))/2 - 267/2 \\ -\sin(\pi/2), & \cos(\pi/2)^3 - \cos(\pi/2)^2 + 1, & -(\sin(\pi/2)*(\cos(\pi/2) - 1))/2, & -(267*\sin(\pi)*(\cos(\pi/2) - 1))/2 \\ 1/2 - \cos(\pi/2), & -(\sin(\pi/2)*(\cos(\pi/2) - 1))/2, & \cos(\pi/2)*(\cos(\pi/2) - \cos(\pi/2)^2 + 1), & 267*\cos(\pi/2) + 267*\cos(\pi/2)^2 \\ - 267*\cos(\pi/2)^3 + 355 \\ 0, & 0, & 0, & 1 \end{bmatrix}$$

$$\mathbf{H60} = \mathbf{H30} * \mathbf{H63};$$

H60 =

$$\begin{bmatrix} 1, & 0, & 0, & 470*\cos(\pi/2) - 355*\cos(\pi/2)^2 + 355 \\ 0, & \cos(\pi), & -\sin(\pi), & -(\sin(\pi/2)*(355*\cos(\pi/2) + 419))/2 \\ 0, & \sin(\pi), & \cos(\pi), & 419*\cos(\pi/2)^2 - 355*\cos(\pi/2) + 355*\cos(\pi/2)^3 + 443 \\ 0, & 0, & 0, & 1 \end{bmatrix}$$

$$\mathbf{H60t} = \mathbf{H1} * \mathbf{H2} * \mathbf{H3} * \mathbf{H4} * \mathbf{H5} * \mathbf{H6};$$

H60t =

$$\begin{bmatrix} 1, & 0, & 0, & 470*\cos(\pi/2) - 355*\cos(\pi/2)^2 + 355 \\ 0, & \cos(\pi), & -\sin(\pi), & -(\sin(\pi/2)*(355*\cos(\pi/2) + 419))/2 \\ 0, & \sin(\pi), & \cos(\pi), & 419*\cos(\pi/2)^2 - 355*\cos(\pi/2) + 355*\cos(\pi/2)^3 + 443 \\ 0, & 0, & 0, & 1 \end{bmatrix}$$

$$\mathbf{X} = 470*\cos(\pi/2) - 355*\cos(\pi/2)^2 + 355$$

$$\mathbf{Z} = 419*\cos(\pi/2)^2 - 355*\cos(\pi/2) + 355*\cos(\pi/2)^3 + 443$$

$$\mathbf{Y} = -(\sin(\pi/2)*(355*\cos(\pi/2) + 419))/2$$

$$\mathbf{A}=0$$

$$\mathbf{C}=0$$

$$\mathbf{B}=0$$