UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
SEDE CONCEPCIÓN – REY BALDUINO DE BÉLGICA.



UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

Implementación de Plataforma Stewart didáctica con Control PID y Lógica Difusa

Memoria de titulación para optar al Título de Ingeniero de Ejecución en Control e Instrumentación Industrial.

Autor:

Pedro Manuel Sanhueza Méndez

Profesor guía:

Ricardo Vera González

2025

CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☒ Memoria o trabajo de título ☐ Tesis de Postgrado

Título del trabajo: Implementación de plataforma Stewart didáctica con control PID y lógica difusa.

Nombre del candidato(a): Pedro Manuel Sanhueza Méndez

Carrera / Grado: Ingeniería de ejecución en control e instrumentación industrial

Campus: Concepción **Departamento:** Electrónica e informática

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Ricardo Ignacio Vera González, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

☒ El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (embargo) por (marcar una opción):

☐ 6 meses ☐ 12 meses ☐ 2 años ☐ 3 años ☐ 5 años ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

_____

_____

_____

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 5/01/26 **Firma:** _____

Estudiante o Candidato(a):

Fecha: 05/01/2026 **Firma:** _____

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

Índice general

Agradecimientos **5**

Resumen **6**

Introducción **7**

Objetivo General **8**

Objetivos Específicos **9**

Alcance del Proyecto **10**

Restricciones **11**

1. Estado del Arte **12**

1.1. Antecedentes y Tipologías de Robots 12

1.1.1. Robots Serie vs. Robots Paralelos 12

1.1.2. Estructuras de Robots Paralelos 14

1.1.3. Cinemática de robots paralelos 15

1.2. La Plataforma Stewart 16

1.2.1. Qué es la Plataforma Stewart 16

1.2.2. Características de la plataforma Stewart 17

1.2.3. Tipos de configuraciones de la Plataforma Stewart 17

2. Descripción y Diseño de los componentes. **19**

2.1. Arquitectura General de Hardware y Piezas 3D 19

2.1.1. Componentes de Hardware 19

2.1.2. Piezas de Impresión 3D 24

2.1.3. Articulaciones 26

2.1.4. Sujeciones 28

2.1.5. Placas de Conexión 29

2.2. Herramientas de Software 30

2.2.1. Arduino IDE 30

2.2.2. LabVIEW 31
2.2.3. Autodesk Inventor 32
2.3. Protocolos de comunicación compatibles 32
2.3.1. Wi-fi 33
2.3.2. ESP-NOW 34
2.3.3. UART 34

3. Implementación de Plataforma Stewart 36
3.1. Ensamblado de la Plataforma 36
3.1.1. Base Fija 36
3.1.2. Soporte de cámara 37
3.1.3. Brazos de los servomotores 37
3.1.4. Base superior móvil 37
3.1.5. Articulaciones 38
3.2. Conexiones eléctricas 38
3.2.1. Alimentación de Servomotores 38
3.2.2. Alimentación de ESP32-Dev Kit y ESP32-CAM 39
3.2.3. Señales de control 39
3.2.4. Resumen de conexiones 40

4. Implementación de Control PID 42
4.1. Descripción del modelo de control 42
4.2. Formulación matemática del controlador PID 44
4.2.1. Ecuación PID en tiempo continuo 44
4.2.2. Ecuación PID en tiempo discreto (implementada en el ESP32) . . . 44
4.2.3. Rol de cada término 45
4.2.4. Control independiente de los ejes X e Y 45
4.3. Implementación del Control PID en el ESP32 46
4.3.1. Estructura general del lazo de control 46
4.3.2. Limitaciones mecánicas y prevención de windup 47
4.3.3. Conversión de inclinación a ángulos de servomotor 47
4.3.4. Proceso de calibración inicial 48
4.4. Integración PID + Cinemática Inversa (IK) 49

4.4.1. Modelo geométrico simplificado de la plataforma Stewart 50
4.4.2. Pose deseada a partir de la salida del controlador PID 51
4.4.3. Cálculo de las longitudes de las patas (cinemática inversa) 52
4.4.4. Conversión longitud–ángulo para cada servomotor 52
4.4.5. Flujo de integración PID–IK en tiempo real 53
4.5. Proceso de Sintonización del Control PID 54
4.5.1. Controlador base y primeras observaciones 54
4.5.2. Ajuste iterativo de las ganancias del PID 55
4.5.3. Resultados de la sintonización 56
4.5.4. Parámetros finales del controlador PID 56
4.6. Implementación del Interfaz de Control y Supervisión en LabVIEW 58
4.6.1. Objetivos de la interfaz en LabVIEW 59
4.6.2. Comunicación entre LabVIEW y ESP32 Dev Kit 59
4.6.3. Diseño del panel frontal 61
4.6.4. Procesamiento de datos dentro de LabVIEW 62
4.6.5. Rol de LabVIEW dentro del sistema de control 62
4.6.6. Ventajas del uso de LabVIEW en el proyecto 63

5. Implementación de Lógica Difusa 64
5.1. Introducción a la Lógica Difusa 64
5.1.1. De los conjuntos difusos al control difuso 64
5.1.2. Historia y aplicaciones en ingeniería 65
5.1.3. Ventajas y características del control difuso 67
5.2. Diseño del controlador difuso 67
5.2.1. Definición de variables lingüísticas y normalización 68
5.2.2. Funciones de membresía 69
5.2.3. Base de reglas del controlador difuso 71
5.2.4. Inferencia y desfusificación 72
5.2.5. Histéresis para reducción de vibraciones 73
5.2.6. Ganancia global del sistema 73
5.3. Implementación del controlador difuso en el ESP32 74
5.3.1. Estructura general del lazo difuso 74
5.3.2. Implementación de las funciones de membresía 75

5.3.3. Evaluación de reglas difusas 75
5.3.4. Desfusificación y ganancia global 76
5.4. Ajuste de Parámetros del Controlador Difuso 76
5.4.1. Parámetros regulables desde LabVIEW 77
5.4.2. Efecto de la ganancia global 79
5.4.3. Ajuste de las reglas Fuerte, Media y Suave Posicionales 80
5.4.4. Ajuste de las reglas Fuerte, Media y Suave de Frenado 80
5.4.5. Resultados del proceso de ajuste 81
5.5. Comparación entre Control PID y Control Fuzzy 81
5.5.1. Comportamiento dinámico 82
5.5.2. Estabilidad en el centro 82
5.5.3. Suavidad del control 82
5.5.4. Adecuación para fines didácticos 83

6. Resultados y Conclusiones **85**
6.1. Resultados experimentales 85
6.1.1. Desempeño del controlador PID 86
6.1.2. Desempeño del controlador difuso 86
6.1.3. Comportamiento del sistema de visión 87
6.2. Conclusiones generales 87
6.3. Limitaciones del sistema 88
6.4. Trabajo futuro 89

Agradecimientos

Deseo expresar mis más sinceros agradecimientos a aquellas personas que aportaron a la realización de esta memoria. Iniciando por aquellos profesores que ejercieron la labor de enseñar con paciencia y virtud, permitiendo las herramientas que respaldan esta memoria, y el deseo de buscar nuevos conocimientos.

Al **Profesor Ricardo Vera González** que con un gran compromiso, fue fundamental en la dirección de la memoria y su ejecución, aportando su cercanía y extensos conocimientos desde mi formación como Técnico Universitario en Electrónica, hasta ahora como Profesor Guía.

Al **Profesor Rodrigo Viza Zavala** quien tuvo una importantísima participación en la formulación de la idea de este proyecto y en la orientación de su realización.

Al **Profesor Jorge Pereira Vivanco**, docente durante mi periodo de enseñanza media en la especialidad Técnico Profesional en Telecomunicaciones, quien fue fundamental con su orientación para elegir el rumbo académico universitario, en el área de la electrónica y control.

Al apoyo docente del departamento, **Matías Cid Zapata**, por su excelente disposición en el asesoramiento para la impresión de piezas 3D, las cuales fueron de suma importancia para el proyecto.

Y finalmente, a mis padres, que siempre estuvieron junto a mí, entregándome su más sincero apoyo para continuar con mi formación académica, impartiendo la perseverancia y los valores y actitudes que hacen posible la realización de esta memoria. A todos ustedes, se los agradezco inmensamente.

Resumen

El presente proyecto de titulación, aborda la implementación física de un modelo del hexápodo, denominado Plataforma Stewart, como memoria para la Universidad Técnica Federico Santa María.

Su realización, traerá consigo la posibilidad de mayor visibilidad en centros de educación media de la zona, por medio de actividades de difusión de la Universidad y el Departamento de Electrónica e Informática, por medio de este modelo de maqueta didáctica, que permitirá acercar a los jóvenes al mundo de la tecnología y el control ejercido por medio de ella. Y a los estudiantes de carreras del departamento, brindarles una herramienta de comprensión con respecto a las dinámicas y tipos de control de una forma amigable.

Introducción

El control de sistemas multi-eje en entornos dinámicos es un pilar fundamental en la Ingeniería de Control e Instrumentación. El desafío de la **Estabilización de Bola** es un problema clásico que combina múltiples áreas del conocimiento: cinemática, teoría de control, visión artificial y automatización industrial.

Actualmente dentro del Departamento de Electrónica e Informática, se carece de una maqueta, o planta de un tamaño reducido y que visualmente pueda representar las Teorías de Control.

La **Plataforma Stewart** se selecciona como el mecanismo de actuación principal debido a sus propiedades únicas: alta rigidez, precisión en el posicionamiento y la capacidad de ofrecer 6 Grados de Libertad. En el contexto educativo, su implementación didáctica permite a los estudiantes comprender de manera tangible la complejidad de la **cinemática inversa** y la aplicación práctica de algoritmos de control avanzados en un manipulador paralelo. Y la integración del sistema con **LabVIEW** eleva la plataforma al estándar de un entorno real de variación de parámetros sobre una planta de control.

Las plantas que se tienen a disposición, tales como la Planta de Presión, Temperatura, Nivel y Flujo, son de un tamaño considerable, y de forma visual, difícilmente se pueden ver representadas las estrategias de control industrial.

Al momento futuro, de precisar realizar actividades de difusión de las carreras del Departamento en centros de educación media, será difícil el usar estas plantas, para atraer la atención de los jóvenes estudiantes.

De igual forma, al ser poco representativas visualmente, para los mismos estudiantes de carreras afines al Departamento, puede ser difícil el relacionar las teorías de control a algo palpable.

Objetivo General

"Implementar Plataforma Stewart didáctica con control PID y Lógica Difusa".

Esta implementación es capaz de solucionar la problemática contextualizada, al aportar con una maqueta de tamaño compacto llamativa visualmente, con la cual se puede interactuar directamente generando perturbaciones, donde el sistema automatizado se encargará de corregirlas.

Objetivos Específicos

I. Diseñar y armar la plataforma, considerando una ESP32-CAM como sistema de adquisición de datos comunicado con LabVIEW.

II. Desarrollar y aplicar la técnica de control PID para el sistema.

III. Desarrollar y aplicar la técnica de control de Lógica Difusa/Fuzzy Logic para el sistema.

Alcances del Proyecto

I. **Diseño y construcción de la plataforma Stewart didáctica.**

Incluye la fabricación de la estructura mediante impresión 3D, integración de servomotores y desarrollo del sistema mecánico completo que permitirá el movimiento controlado de la plataforma mediante articulaciones.

II. **Implementación del sistema de control PID.**

Se desarrollará y programará un algoritmo de control PID sobre el microcontrolador ESP32 Dev Kit, capaz de mantener la posición de una esfera de color rojo mediante la lectura de coordenadas obtenidas por la cámara ESP32-CAM.

III. **Implementación del sistema de control basado en Lógica Difusa.**

Se realizará un control por medio de Lógica Difusa (Fuzzy Logic), con el fin de aportar con una herramienta adicional a la formación académica de los estudiantes de las carreras del área de la tecnología.

IV. **Integración con interfaz HMI en LabVIEW.**

Se diseñará una interfaz que permita la visualización en tiempo real del sistema a través de lo capturado por la ESP32-CAM, además del ajuste de parámetros de control desde el software. Esto permitirá la interacción directa con la capacidad de control del sistema.

Restricciones

- **Actuadores:** Se usarán **Servomotores** como actuadores.
- **Microcontrolador:** Se usará el microcontrolador **ESP32 Dev Kit** para el control de los actuadores
- **Sensor: ESP32-CAM**, microcontrolador adaptado para conectarse a una cámara de 2MP.
- **Software Microcontrolador:** Se utilizará el Software **Arduino IDE** para la programación de los microcontroladores.
- **Software HMI:** Se utilizará el Software **LabVIEW** para generar la HMI.
- **Comunicación Microcontrolador-HMI-Cámara:** Se utilizará el protocolo de comunicación serial **UART**.
- **Tamaño Plataforma Superior:** Se usará una pieza de impresión 3D de forma cuadrada (20cm $\times$ 20cm).

1. Estado del Arte

Este capítulo establece el contexto del proyecto mediante una revisión de los tipos de manipuladores robóticos y sus propiedades cinemáticas. Además, se exponen los fundamentos teóricos esenciales que sustentan la implementación del control.

1.1. Antecedentes y Tipologías de Robots

1.1.1. Robots Serie vs. Robots Paralelos

Los manipuladores robóticos se clasifican comúnmente en dos grandes categorías estructurales: robots serie (o antropomórficos) y robots paralelos. Los robots serie poseen una estructura en cadena abierta, similar a la de las extremidades humanas debido a la disposición de sus actuadores. En contraste, los robots paralelos poseen una base fija y una plataforma móvil unidas por múltiples actuadores.

- **Robots Serie:** Utilizados ampliamente en líneas de producción (ej. industria automovilística). Su principal desventaja radica en que la inercia y el torque aplicado afectan significativamente el desempeño en el extremo de sus articulaciones.



Figura 1.1: Robot Serie

- **Robots Paralelos:** Tienen un excelente desempeño en términos de velocidad, fuerza y aceleración al manipular objetos con mayor precisión. Una característica crucial es que sus motores o actuadores están fijos a la base, lo que minimiza la afectación por inercias y disminuye la tasa de averías por sobrecarga en las articulaciones.



Figura 1.2: Robot Paralelo

Ventajas de los Robots Paralelos

La configuración estructural de los manipuladores paralelos ofrece las siguientes ventajas clave para el proyecto:

1. **Relación Carga-Potencia Favorable:** Los actuadores están conectados directamente a la base, proporcionando soporte estructural que favorece el control en cargas de elevada masa.

2. **Rigidez y Baja Inercia:** Las estructuras son rígidas y ligeras, lo que permite una mayor agilidad y la realización de movimientos más precisos.

3. **Velocidad de Respuesta:** La velocidad de respuesta de los actuadores suele traducirse en una mayor velocidad en el efector final.

Desafíos de los Robots Paralelos

A pesar de sus ventajas, los manipuladores paralelos presentan ciertas complejidades:

1. **Cinemática Inversa Compleja:** El cálculo de movimiento para estos robots es complicado, requiriendo sistemas matemáticos elaborados para el análisis de los lazos de control.

2. **Espacio de Trabajo Acotado:** Su correcto desenvolvimiento depende de un espacio de trabajo previamente calculado, siendo este típicamente más pequeño y acotado que el de los robots serie.

3. **Falta de Estandarización:** No existe una propuesta general de controladores, lo que hace que los controladores desarrollados sean a menudo únicos para cada robot.

1.1.2. Estructuras de Robots Paralelos

Las estructuras se clasifican según el movimiento que permiten en función de las restricciones de las articulaciones y los tipos de cadenas cinemáticas:

- **Planares:** Restringidos al plano cartesiano, con 2 o 3 Grados de Libertad (traslación y rotación).

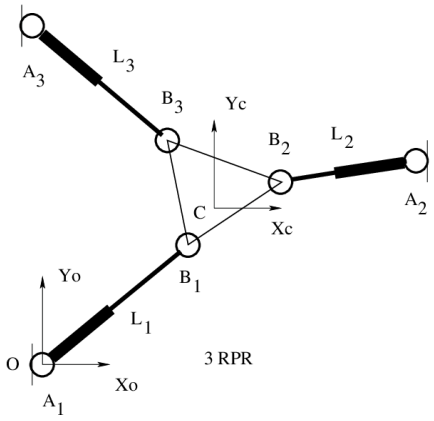


Figura 1.3: Robot paralelo tipo Planar

- **Espacial:** Poseen movimiento en el espacio tridimensional, usualmente de 3 a 6 Grados de Libertad. Las uniones con los efectores finales suelen usar juntas **universales (U)** o **esféricas (S)**.

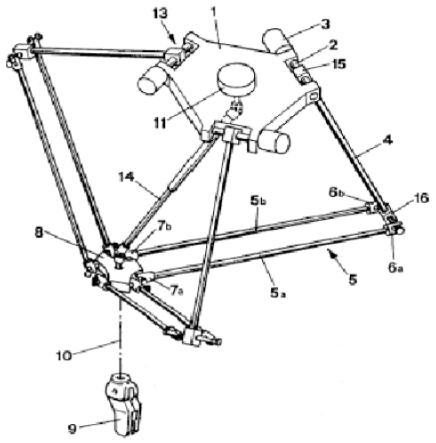


Figura 1.4: Robot paralelo tipo Espacial

1.1.3. Cinemática de robots paralelos

La cinemática estudia el movimiento de los robots con respecto a un sistema de referencia pero sin considerar las fuerzas que actúan sobre el sistema, sino únicamente la geometría del mismo.

Se divide en dos tipos, cinemática inversa y directa. El cálculo de la cinemática directa es complejo por los lazos cerrados de los sistemas. Sin embargo, su contraparte, la cinemática inversa es de una aplicación más simple.

- **Cinemática Inversa:** Determina las coordenadas articulares de cada actuador para una posición y orientación conocidas de la plataforma móvil. Existen varios métodos para el cálculo de ésta. Merlet propone dos métodos para calcularla, uno analítico y otro geométrico. En el caso de los robots paralelos el método geométrico facilita el análisis de la cinemática inversa, permitiendo determinar una solución para el ángulo que debe rotar el actuador dependiendo de la posición y rotación de la plataforma móvil con respecto a la plataforma fija.

- **Cinemática Directa:** Determina la posición y orientación final de la plataforma móvil, con respecto a un sistema de coordenadas que se toma como referencia, conociendo las coordenadas articulares del robot. Sus métodos de cálculo son los iterativos y polinomiales, que son métodos numéricos que requieren tiempo para ser realizados.

1.2. La Plataforma Stewart

1.2.1. Qué es la Plataforma Stewart

La Plataforma Stewart, o hexápodo, es un robot paralelo espacial que utiliza seis actuadores para controlar los 6 Grados de Libertad de la plataforma móvil.

La plataforma tiene sus inicios en 1965, donde Stewart propuso el uso de una estructura paralela de seis grados de libertad para generar movimiento en el espacio con una aplicación directa en simuladores de vuelo. El número de actuadores utilizados permitía la manipulación del peso que estaba sobre ella por el torque combinado distribuido, de tal modo que cada uno soporta 1/6 de los esfuerzos totales, proporcionando gran rigidez al mecanismo.

Inicialmente consistía en una plataforma triangular unida mediante articulaciones esféricas a tres actuadores lineales ajustables en longitud, conectados a una base fina mediante articulaciones universales.

La plataforma Stewart es un diseño de control de posición y movimiento. Utilizando este diseño se han realizado numerosas aplicaciones en la industria, tales como aeronáutica, automotriz, marina y maquinaria industrial. Y en aplicaciones fuera de la industria, se ha adaptado para sus usos en simuladores de vuelo, como originalmente fue diseñada, construcciones de puentes, mecanismos de brazos grúas, posicionamiento de antenas parabólicas, entre otras aplicaciones más.

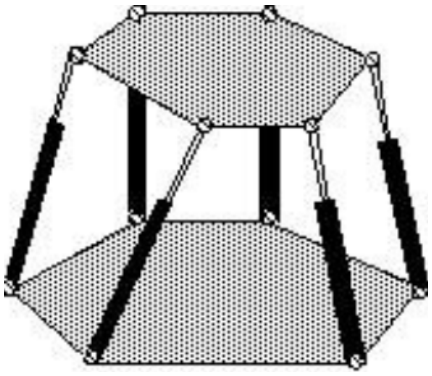


Figura 1.5: Representación Plataforma Stewart tipo 6-6

1.2.2. Características de la plataforma Stewart

La plataforma posee un amplio rango de movilidad y puede ser posicionada con una enorme precisión. Su arquitectura consiste de una base fija y una base superior móvil conectada en al menos tres puntos con los seis actuadores lineales.

Dado su diseño, permite seis grados de libertad con rotación sobre los ejes X,Y, y Z, además de traslación.

Generalmente como actuadores lineales, se suelen usar pistones hidráulicos que permiten un control preciso de la posición, combinado con la fuerza que naturalmente ejercen este tipo de pistones. Pueden ser vistos en los simuladores de vuelo clásicos, que al contemplar el peso de una cabina simulada para el entrenamiento de pilotos, más el mismo peso de las personas que se encuentran dentro, acumulan un peso a considerar, que los pistones son capaces de soportar sin mayor problema.

Para aplicaciones de menor escala, se pueden usar actuadores electromecánicos, tales como servomotores, motores lineales o motores paso a paso.

1.2.3. Tipos de configuraciones de la Plataforma Stewart

Inicialmente los diseños de la plataforma se enfocaron en dos configuraciones principalmente. Una con la base fija y la base móvil en forma triangular y los actuadores conectados de dos en dos coincidiendo con los vértices del triángulo. Otra con la plataforma móvil también en forma triangular, pero con la base fija en forma hexagonal, también con los actuadores ubicados de dos en dos, pero los seis puntos se encontrarían en los vértices de hexágono de la base fija.

Estas configuraciones se clasifican como 3-3 plataforma de Stewart y 6-3 plataforma de Stewart respectivamente. Prontamente se observó que la coincidencia de las articulaciones esféricas restringía la movilidad de la plataforma, por lo que se comenzó a pensar en otro modelo que pudiera compensar esta limitación.

A raíz de eso, empezó a ser muy popular una plataforma con hexágonos semiirregulares como puntos de conexión, tanto en la base como en la plataforma móvil, la misma que hoy se conoce como 6-6 plataforma de Stewart.

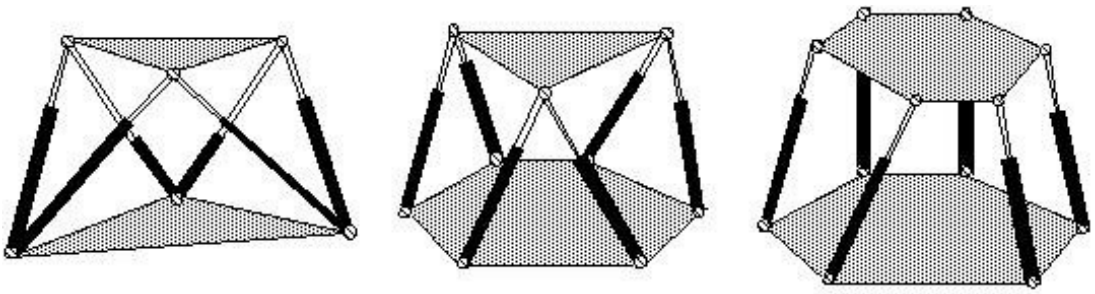


Figura 1.6: Plataforma Stewart (a) tipo 3-3m (b) 6-3, (c) tipo 6-6.

Existe una gran variedad de configuraciones mecánicas para este tipo de robots, según la estructura de las cadenas cinemáticas que unen la base fija con la plataforma móvil.

2. Descripción y Diseño de los componentes.

Este capítulo detalla la metodología y la arquitectura del sistema implementado, cubriendo desde la selección de componentes hasta el diseño cinemático.

2.1. Arquitectura General de Hardware y Piezas 3D

El sistema opera bajo una arquitectura de control de lazo cerrado que integra la actuación mecánica, la percepción por visión y un sistema de control distribuido mediante microcontroladores ESP32.

2.1.1. Componentes de Hardware

- **Actuadores:** Seis Servomotores **MG996R**. Su alta capacidad de torque es esencial para manejar la plataforma de manera dinámica y precisa.

  -Especificaciones:

  -Voltaje de Operación: 4.8 a 7.2[Vdc].

  -Grados de giro: 180°.

  -Corriente máxima: 2.5[A].

  -Corriente de trabajo: (500 – 900) [mA].

  -Torque de trabajo: 11kg/cm (6V).

  -Velocidad de operación: 0.15 seg/60°.

  -Control: PWM a 50[Hz].

  -Peso: 55[gr].



Figura 2.1: Servomotor MG996R.

- **Controlador Principal: ESP32 Dev Kit**. Centraliza la ejecución del algoritmo de control (PID/Fuzzy), y la cinemática inversa.

  -Especificaciones:

  -Voltaje de Alimentación (USB): 5 [Vdc].

  -Voltaje de Entradas/Salidas: 3.3 [Vdc].

  -Puerto: USB Tipo C.

  -Consumo de energía de 5[µA] en modo de suspensión.

  -Pines: Físicos: 30 /Digitales GPIO: 24.

  -Frecuencia de Reloj: hasta 240 [MHz].

  -Tipo: Módulo Wi-fi: 802.11 b/g/n/e/i (802.11n @ 2.4 GHz hasta 150 Mbit/s) + Bluetooth: 4.2 BR/EDR BLE Modo de control dual.

  -Peso: 9 [gr].

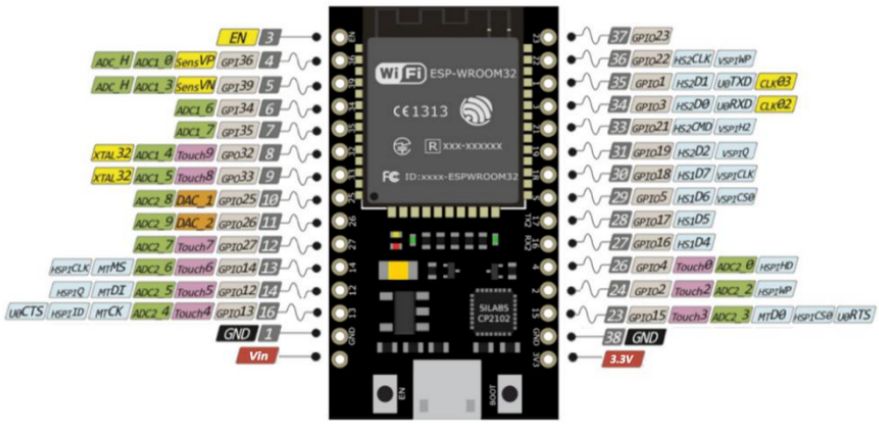


Figura 2.2: Distribución de pines de ESP32-Dev Kit.

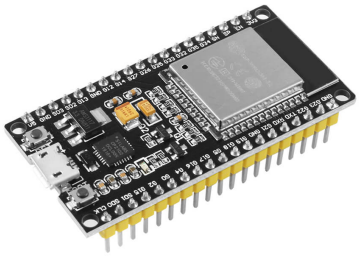


Figura 2.3: Microcontrolador ESP32-Dev Kit.

- **Sistema de Percepción: ESP32-CAM**. Encargado de la adquisición de la posición de la esfera mediante visión artificial, en conjunto con su base de programación y flash incluido, pero que no será usado para este proyecto.

  -Especificaciones:

  Voltaje de alimentación: 5V

  Corriente de trabajo con flash apagado: 180 [mA].

  Corriente en modo suspensión: 6 [mA].

  Modelo de cámara: OV2640.

  Resolución: 2 [MP].

  Transferencia de imagen: 15 a 60 [FPS]

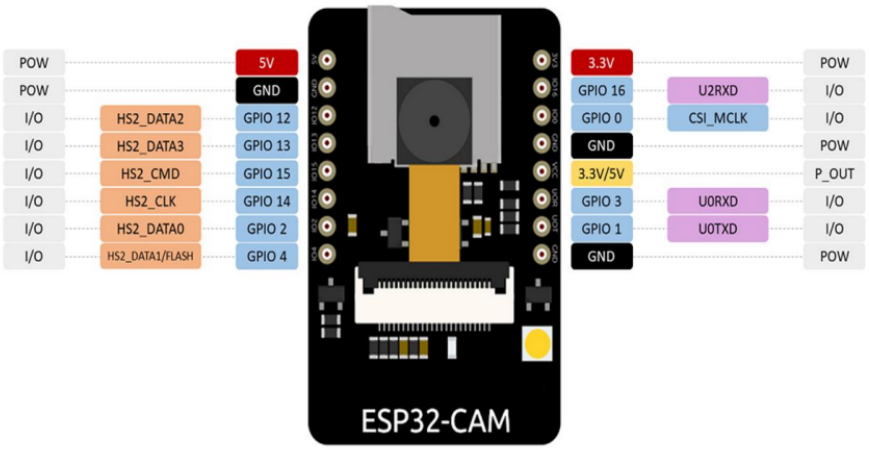


Figura 2.4: Distribución de pines de ESP32-CAM.

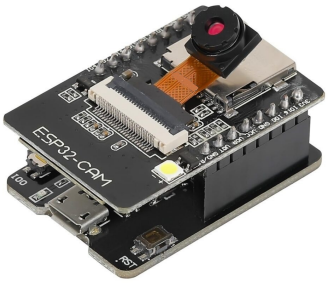


Figura 2.5: Módulo ESP32-CAM con base de programación.

- **Fuente de Poder:** Tendrá la labor de alimentación de los actuadores del sistema.

  -Especificaciones:

  -Voltaje de entrada: 115/230[V]

  -Voltaje de Salida: 12[Vdc]

  -Corriente máxima de Salida: 12.5[A]

  -Potencia: 150[W]



Figura 2.6: Fuente de Poder

- **Módulo Step Down:** Se encargará de bajar el voltaje proveniente de la Fuente de Poder, hasta 5[Vdc] para el correcto funcionamiento de los actuadores.

  -Especificaciones:

  -Voltaje de Entrada: 4.5 a 32[Vdc]

  -Voltaje de Salida: 0.8 a 30[Vdc] ajustable.

  -Corriente de Salida Máxima: 12[A]

  -Potencia de Salida: 200[W] max

  -Peso: 100 [g]



Figura 2.7: Módulo Step Down

2.1.2. Piezas de Impresión 3D

Para el armado físico, se precisaron de siete tipos de piezas 3D, las cuales contemplan el espacio asignado para equipos tales como los servomotores en la base inferior de la plataforma, el módulo ESP32-CAM con su espacio reservado a las dimensiones adecuadas para la monitorización de la plataforma superior, y finalmente la Plataforma Superior, sobre la cual se encontrará la bola.

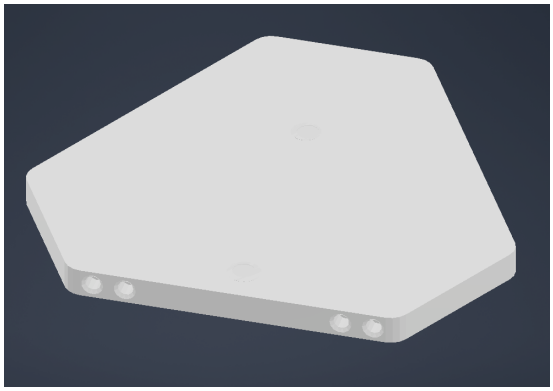


Figura 2.8: Base Inferior de los Servomotores

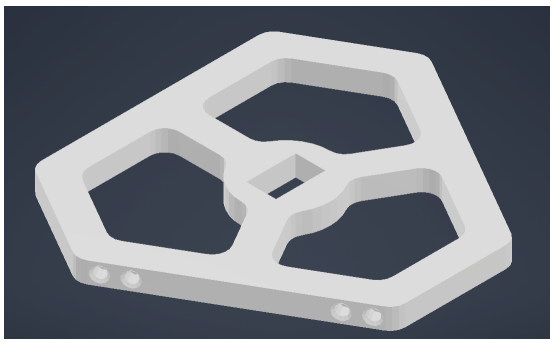


Figura 2.9: Base Superior de los Servomotores

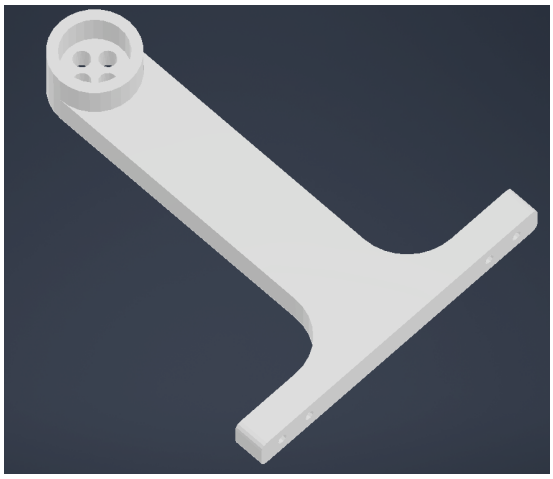


Figura 2.10: Soporte Inferior de ESP32-CAM

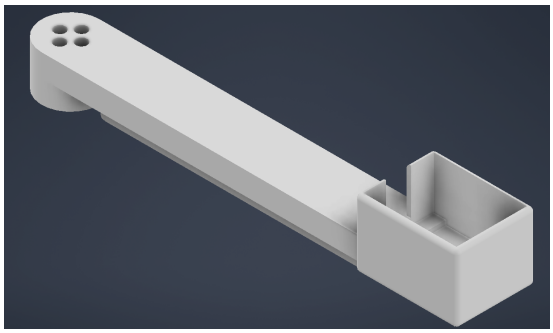


Figura 2.11: Soporte Superior de ESP32-CAM

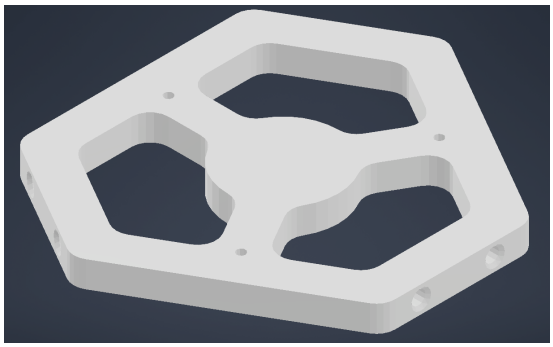


Figura 2.12: Sub-Base de Plataforma Superior

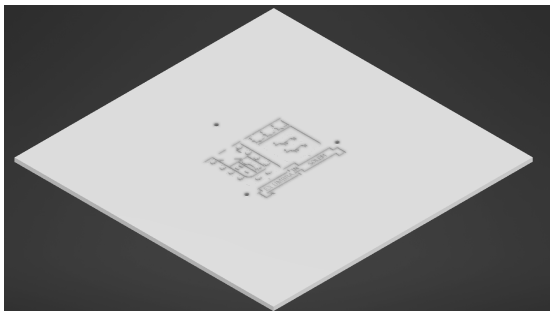


Figura 2.13: Plataforma Superior con logo UTFSM

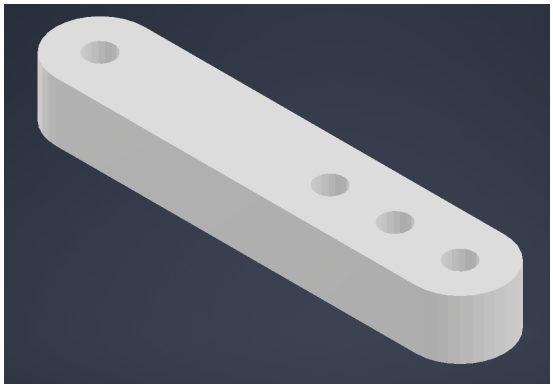


Figura 2.14: Brazo de Acople para los Servomotores

2.1.3. Articulaciones

Al trabajar con un sistema Hexápodo, se precisa de seis articulaciones que conecten físicamente los brazos de los servomotores hacia la Sub-Plataforma superior. Para esto se necesita de varillas con la suficiente firmeza y rótulas móviles capaces de entregar el movimiento que ejecutarán los servomotores de forma correcta hacia la Plataforma Superior. Adicionalmente como herramienta de soporte para los servomotores, se usarán discos metálicos para mayor solidez.



Figura 2.15: Varillas M3



Figura 2.16: Rótulas M3



Figura 2.17: Disco Metálico para Servomotores

2.1.4. Sujeciones

En cuanto al montaje, se usarán dos tipos de elementos para la firmeza de cada parte de la plataforma, en concordancia con las medidas y diseños realizados en impresión 3D. Tuercas de inserción para ser usadas en los orificios de las piezas de impresión 3D, sujetas con material adherente. Tornillos M3 de variadas longitudes para la fijación de las bases y articulaciones, además de tuercas zincadas para aportar el espacio de movimiento a las articulaciones.



Figura 2.18: Tuercas de inserción



Figura 2.19: Tornillo M3



Figura 2.20: Tuercas M3 zincadas

2.1.5. Placas de Conexión

Para una conexión ordenada y eficiente, se necesita de placas capaces de distribuir la energía de forma segura y eficaz. En este proyecto, se utiliza una fuente externa de alimentación para la energización de los servomotores, considerando que la corriente puede alcanzar niveles altos, por lo tanto se necesita una placa con numerosos canales para alimentar a los seis servomotores sin sufrir por el aumento de temperatura.

Y del mismo modo, placas en las cuales se pueda establecer de forma fija el ESP32 Dev Kit, para obtener del dispositivo las señales de control apropiadamente en conjunto a la tierra de referencia.

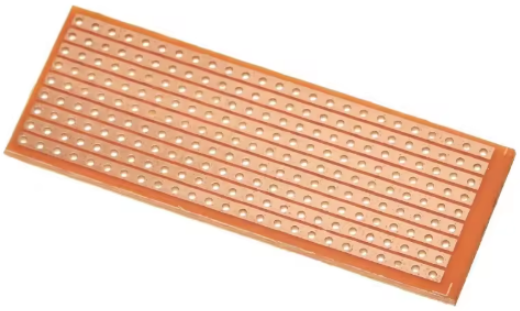


Figura 2.21: Placa de Circuito Impreso



Figura 2.22: Mini-protoboard

2.2. Herramientas de Software

Para el desarrollo del proyecto, se requerirá del uso de softwares compatibles con los microcontroladores, y lo suficientemente robustos como para establecer la comunicación entre los componentes del sistema. Además del diseño de piezas para impresión 3D mencionadas anteriormente.

2.2.1. Arduino IDE

Es un software de programación nativo de Arduino, permitiendo programar microcontroladores Arduino y también numerosos microcontroladores de procedencia distinta en C/C++. Para este proyecto es un eje central en la redacción de código de la maqueta y el desarrollo de los controladores ESP32 Dev Kit y ESP32-CAM, facilitando el acceso a librerías y documentación para la implementación de PID y Lógica Difusa. Se usarán las siguientes librerías para la ejecución de código.

- **ESP32Servo.h:** Permite que las placas ESP32 controlen servos, motores de tono y analógicos mediante la semántica de Arduino. Esta biblioteca puede controlar diversos tipos de servos. Utiliza los temporizadores PWM del ESP32: la biblioteca puede controlar hasta 16 servos en canales individuales.
- **math.h:** La librería ya está incluida por defecto en Arduino IDE, y proporciona funciones matemáticas avanzadas (como sqrt(), pow(), sin(), cos()) para trabajar

con números de punto flotante (float/double), útiles para cálculos trigonométricos, logaritmos y potencias que no están en las funciones básicas de Arduino, permitiendo un manejo preciso de operaciones complejas en microcontroladores.

2.2.2. LabVIEW

LabVIEW es un entorno de programación gráfica que ofrece aceleradores de productividad únicos para el desarrollo de sistemas de prueba, como un enfoque intuitivo para la programación y conectividad con cualquier instrumento e interfaces de usuario totalmente integradas. LabVIEW se utiliza en una amplia variedad de sectores industriales, incluyendo la automatización de procesos, la robótica, la monitorización de sistemas de energía, la instrumentación médica y la investigación científica. Su versatilidad lo hace adecuado para una amplia gama de aplicaciones. Sus principales funciones son las siguientes.

- **Adquisición y Control de Datos:** Se conecta a hardware (sensores, instrumentos) para adquirir datos y controlar dispositivos en tiempo real a través de buses como USB, Ethernet, GPIB, etc..
- **Automatización de Pruebas:** Diseña sistemas automatizados para pruebas de fabricación y validación de diseño, reduciendo el tiempo de desarrollo.
- **Monitoreo y Control de Procesos:** Permite supervisar y controlar máquinas y procesos industriales, incluso en aplicaciones de bajo volumen.
- **Análisis y Visualización:** Incluye funciones integradas para análisis matemático, procesamiento de señales (audio, vibraciones, imágenes) y presentación de datos en gráficos e indicadores interactivos.
- **Programación Gráfica:** Facilita la creación de aplicaciones mediante diagramas de bloques con iconos (funciones, controles, indicadores), que simulan el flujo de datos y la lógica de programación.

Para este proyecto se utilizará para generar la **Interfaz Hombre-Máquina (HMI)** para la lectura de datos de posición y la escritura de parámetros de control.

2.2.3. Autodesk Inventor

Es un software de diseño asistido por computadora (CAD) de modelado paramétrico en 3D para diseñadores e ingenieros. Se utiliza para crear modelos de productos, realizar simulaciones y generar documentación técnica detallada, integrando diseños 2D y 3D en un mismo entorno para la validación y optimización antes de la fabricación. Sus funciones principales son:

- **Modelado 3D paramétrico:**Permite crear y editar modelos 3D complejos utilizando un enfoque paramétrico, donde las relaciones definen la geometría.
- **Diseño mecánico:**Ofrece herramientas específicas para diseñar elementos como chapa, armazones, tuberías, cableado y ensamblajes.
- **Simulación y análisis:**Permite simular el movimiento, aplicar fuerzas y analizar el rendimiento del producto virtualmente.
- **Documentación técnica:**Genera automáticamente planos 2D, listas de materiales y otros documentos directamente desde el modelo 3D.
- **Diseño de formas libres:**Incluye herramientas para dar forma libre a los diseños mediante la manipulación de puntos, aristas y caras.
- **Integración:**Se integra con otras herramientas de Autodesk como AutoCAD y Fusion 360, y permite la colaboración en la nube para compartir diseños y recopilar comentarios.

2.3. Protocolos de comunicación compatibles

La comunicación es algo vital en la implementación de un proceso de control, la cual debe ser más robusta o menos, dependiendo de la intensidad de las lecturas requeridas. Para este caso, el uso de microcontroladores ESP32 abre la posibilidad a múltiples protocolos de comunicación, de los cuales se mencionarán algunos que se probaron a lo largo de la implementación.

2.3.1. Wi-fi

Es una tecnología de comunicación inalámbrica que permite a los dispositivos conectarse a una red local o a Internet sin necesidad de cables. Utiliza radiofrecuencias para transmitir datos entre dispositivos y un punto de acceso (router). El ESP32 es compatible con el estándar Wi-Fi 802.11 b/g/n y opera principalmente en la banda de frecuencia de 2,4 GHz. Esta capacidad le permite conectarse a redes Wi-Fi existentes, crear sus propias redes Wi-Fi o comunicarse directamente con otros dispositivos mediante Wi-Fi Direct. Tiene cuatro modos de operación.

- **Modo Estación (STA):** Donde el ESP32 actúa como un dispositivo cliente que se conecta a un punto de acceso (PA) Wi-Fi, como un router. Una vez conectado, puede comunicarse con otros dispositivos de la misma red y acceder a Internet. Este es el modo más común para dispositivos IoT que necesitan enviar datos a un servidor o recibir actualizaciones inalámbricas (OTA).

- **Modo Punto de Acceso a Software (SoftAP):** En el modo SoftAP, el ESP32 actúa como un punto de acceso, creando su propia red Wi-Fi a la que pueden conectarse otros dispositivos. Esto es útil para aplicaciones que requieren una red local sin un router externo, como la configuración de un dispositivo o la comunicación entre ellos pero que no necesiten un flujo de datos numeroso y constante, de lo contrario hay un riesgo de saturación.

- **Modo Estación + Punto de Acceso (STA + AP):** El ESP32 puede funcionar simultáneamente como cliente y punto de acceso, lo que permite la comunicación con un router Wi-Fi y, al mismo tiempo, proporciona una red local para otros dispositivos. Este modo es útil para aplicaciones que requieren comunicarse con una red externa y crear una red local para control o configuración.

- **Wi-Fi Direct:** Este modo permite la comunicación directa entre dispositivos Wi-Fi sin necesidad de un punto de acceso intermedio. Es útil para aplicaciones que requieren transferencias de datos rápidas y de corto alcance entre dispositivos.

2.3.2. ESP-NOW

ESP-NOW es un protocolo de comunicación inalámbrica desarrollado por Espressif Systems exclusivo para los microcontroladores ESP8266 y ESP32. Este protocolo está diseñado para la transmisión de datos de baja latencia entre dispositivos, sin necesidad de una conexión Wi-Fi tradicional ni un punto de acceso (AP).

ESP-NOW utiliza un modo de comunicación punto a punto que permite que los dispositivos se comuniquen directamente entre sí, lo que lo hace ideal para redes de sensores, aplicaciones de automatización y otras situaciones que requieren una comunicación rápida y eficiente, pero limitado a un flujo de datos no muy denso.

2.3.3. UART

Receptor-Transmisor Asíncrono Universal es uno de los protocolos de comunicación serie más comunes y utilizados en electrónica. Este protocolo permite la transmisión de datos entre dos dispositivos a través de una línea de comunicación serie. Su simplicidad y fiabilidad lo hacen ideal para una amplia gama de aplicaciones, desde microcontroladores hasta ordenadores.

Sus principales características son su **Comunicación asíncrona**, a diferencia de los protocolos síncronos, UART no requiere un reloj compartido entre los dispositivos transmisores y receptores. En su lugar, cada dispositivo utiliza su propio reloj interno para muestrear los datos, lo que hace que la comunicación sea más flexible y menos propensa a problemas de sincronización.

Los datos se transmiten en **Tramas** que incluyen un bit de inicio, un número configurable de bits de datos (normalmente 8), un bit de paridad opcional y uno o dos bits de parada. Esta estructura ayuda a garantizar la integridad de los datos transmitidos.

La velocidad de transmisión de datos (velocidad en baudios) puede variar ampliamente, normalmente desde 9600 baudios hasta **varios megabits por segundo**, dependiendo de las capacidades del hardware y las necesidades de la aplicación.

UART es **Bidireccional**, lo que permite la comunicación simultánea entre dos dispositivos, con una línea para la transmisión (TX) y otra para la recepción (RX).

El protocolo incluye mecanismos de **Comprobación de errores**, como el bit de paridad, para garantizar que los datos recibidos sean correctos. Sin embargo, la confiabilidad

también depende de la calidad de la conexión y de la configuración adecuada de la velocidad en baudios.

3. Implementación de Plataforma Stewart

En este capítulo se detallará el proceso de armado de la plataforma utilizando los elementos mencionados anteriormente, desde la conexión eléctrica, distribución de cables, ensamblaje físico y comunicación entre los equipos.

3.1. Ensamblado de la Plataforma

3.1.1. Base Fija

Haciendo uso de las piezas antes señaladas, el inicio del montaje es la Base Inferior de los servomotores de forma semi-hexagonal. Esta pieza está diseñada específicamente para los orificios de tornillos incorporados en los servomotores MG996R, por lo que el primer paso es atornillar fijando los servomotores a la base, usando los tornillos M3 en conjunto a las tuercas de inserción. Pero se debe tener en cuenta el lugar de montaje de los soportes de la cámara, ya que esta pieza requiere de tornillos de mayor longitud que pasen a través de ella y los dos servomotores del lado al que se encontrará montada hacia las tuercas.

Cada servomotor cuenta con dos pares de orificios para atornillar, en la zona superior e inferior. La base inferior de los servomotores está diseñada específicamente para dichos espacios.

Una vez fijados los servomotores a su base inferior, se acopló a continuación la base superior de los servomotores, que también cuenta con los orificios en los espacios diseñados en 3D. La base superior de los servomotores, cuenta con espacios intermedios para no estorbar en la salida de los cables de los motores, y el agujero del centro de la pieza, permite ordenarlos todos en un solo punto para evitar el desorden.

3.1.2. Soporte de cámara

El soporte de la cámara cuenta con dos piezas de impresión 3D y una vara de madera de 19 [mm] de diámetro y 45 [cm] de largo. Se fijarán las dos piezas a la vara de madera y el soporte inferior estará atornillado a la base inferior de los servomotores.

Sobre la pequeña sección del soporte superior, se ubicará la ESP32-CAM, esta pieza cuenta con el espacio necesario para almacenarla y dos aberturas, una para el lente de la cámara y otra para el flash que tiene incorporado.

3.1.3. Brazos de los servomotores

Una vez con la base lista, se atornillarán los discos metálicos a cada servomotor, en conjunto con los Brazos de acople para los servomotores. También producto de la creación de esta pieza, permite el acople perfecto sobre el motor. Atornillando directo hacia el servomotor desde el orificio central de los tres que se encuentran en conjunto. Y al menos uno de los orificios que están junto a este, deben de estar sujetos al disco metálico. Por otro lado, el orificio que se encuentra más alejado de los otros, es para la fijación a la rótula de la articulación.

Y antes de unir las articulaciones, se debe realizar una calibración del ángulo de los motores, con el objetivo de que los motores que comparten un lado del semi-hexágono, tengan los brazos directamente en frente el uno del otro. Es posible que en el momento de fijarlos, no queden completamente cercanos a esta posición, pero mientras estén en un rango de $\pm$ 15° pueden ser corregidos por programa para llegar a esta posición neutro, la cual teóricamente, debe ser de 90° para los motores en su posición neutro de funcionamiento, permitiendo el variar sus grados con la mayor cantidad de valores que se requiera en forma ascendente o descendente.

3.1.4. Base superior móvil

Para el armado de la base superior tenemos solo dos piezas para unir, que serían, la Sub-Base de la Plataforma Superior y la misma Plataforma. Estas piezas cuentan con tres orificios que las atraviesan, los cuales deben estar unidos por tornillos, pensando que el logo de la universidad debe quedar en perspectiva hacia arriba, la Sub-Base se debe posicionar correctamente abajo haciendo calzar los espacios. Y se debe proceder a atornillar de

abajo hacia arriba, con el fin de que los tornillos no estorben en el movimiento sobre la plataforma. También la Sub-Base cuenta con orificios donde deben de estar adheridas tuercas de inserción, para fijar las articulaciones.

3.1.5. Articulaciones

Una vez que se tenga la zona inferior, superior y la calibración de los servomotores a neutro lista , cada varilla debe estar unida en sus dos extremos a una rótula, la cual en la parte designada como inferior, debe estar fijada al orificio más alejado del brazo de acople para los servomotores por medio de un tornillo M3, y fijado con una tuerca zincada para aportar solidez a la unión, soportando el movimiento que debe realizar el sistema.

Mientras que para la parte superior, solo será necesario usar tornillos en dirección a las tuercas de inserción adheridas a la Sub-Base.

3.2. Conexiones eléctricas

En esta sección se explicará y detallará la conexión llevada a cabo para cada uno de los componentes que forman parte del sistema.

3.2.1. Alimentación de Servomotores

Dado el consumo considerable de corriente de los servomotores, se usará la fuente de poder y el módulo step-down mencionado previamente, con capacidad de entregar hasta 12 [A]. Permitiendo que para el accionamiento sea más que suficiente la alimentación para todos los actuadores, considerando que estos no están expuestos a un gran peso directamente, ni tendrán variaciones de grados extremas, sino que deben moverse en un umbral reducido.

La fuente de poder tiene una salida de 12[Vdc], por lo que es necesario reducir este voltaje para que pueda ser usada la energía por los motores.

El módulo step-down puede recibir en su entrada desde 4.5 a 32 [Vdc], y entregar salidas entre 0.8 a 30 [Vdc] con una corriente máxima de salida de 12[A]. El módulo puede ajustar su salida por medio de un potenciómetro de precisión para lograr los 5[Vdc] óptimos para el funcionamiento de los motores.

A partir de este voltaje, se distribuyó en la placa de circuito impreso con canales en serie, ordenando tres canales para 5[Vdc] y otros 3 canales a GND. Por cada uno de estos canales se conectaron dos servomotores, para así reducir riesgos por alzas de temperatura en el material conductor.

Saliendo los cables desde la placa, se ordenarán sus conectores para una mayor comodidad a la hora de energizar.

3.2.2. Alimentación de ESP32-Dev Kit y ESP32-CAM

La conexión del ESP32-Dev Kit es directa al puerto USB de un computador, alimentándose así también con 5 [Vdc], pero un reductor interno de voltaje, es capaz de bajarlo a 3.3 [Vdc]. La ESP32-CAM será conectada directamente al Dev Kit para ser energizada por él, de los pines VCC y GND de la cámara, a los pines 3V3 y GND del Dev Kit respectivamente.

3.2.3. Señales de control

Para este apartado, hay dos tipos de señales de control. Primeramente, las señales PWM que entregará el Dev kit hacia los servomotores que tienen una frecuencia establecida de 50 Hz (Periodo de 20 ms). La posición angular del servo (de 0° a 180°) es determinada por el Ciclo de Trabajo (Duty Cycle) de la señal PWM (es decir, el ancho del pulso).

El ESP32 Dev Kit utiliza sus timers y drivers PWM internos para generar las seis señales PWM simultáneamente. La resolución en bits del PWM del ESP32 determina la precisión con la que se puede controlar el ángulo del servo, lo cual impacta directamente en la precisión cinemática de la plataforma. Al ser seis, se necesitará de seis pines físicos para alimentar los servomotores, los cuales serán los pines: D16, D17, D18, D19, D21 y D22 ubicados a la zona derecha del Dev Kit, incluyendo una conexión a tierra común para una referencia fija de señales.

La asignación de los pines GPIO para las salidas PWM es crítica para el firmware, como se detalla en el Cuadro 3.1. Esta tabla relaciona el pin físico (el canal de PWM) con la etiqueta lógica de actuador utilizada en la Cinemática Inversa.

Cuadro 3.1: Asignación de Pines PWM del ESP32 Dev Kit a los Actuadores de la Plataforma Stewart.

Actuador	**Etiqueta Lógica**	**Pin Físico ESP32**
Servo 1	L_1	`GPIO 16 (D16)`
Servo 2	L_2	`GPIO 17 (D17)`
Servo 3	L_3	`GPIO 18 (D18)`
Servo 4	L_4	`GPIO 19 (D19)`
Servo 5	L_5	`GPIO 21 (D21)`
Servo 6	L_6	`GPIO 22 (D22)`

Por otro lado, los terminales pertenecientes a la comunicación serial UART, serán uno para la cámara y otro para el dev kit. El pin físico de la cámara es el pin VOT, usado como Tx. Mientras que el pin físico del dev kit será el D4, usado como Rx.

3.2.4. Resumen de conexiones

Las conexiones se pueden mencionar en dos grandes grupos, lo concerniente a la alimentación eléctrica de los dispositivos y la señales de control que hacen posible el funcionamiento del sistema. A continuación se encuentra un diagrama que contempla todas las conexiones eléctricas del sistema a partir de los 5[Vdc] del módulo step down, considerando que el pin 1 de los servomotores, se considera la señal PWM, el pin 2 se considera 5[Vdc], y el pin 3 se considera GND.

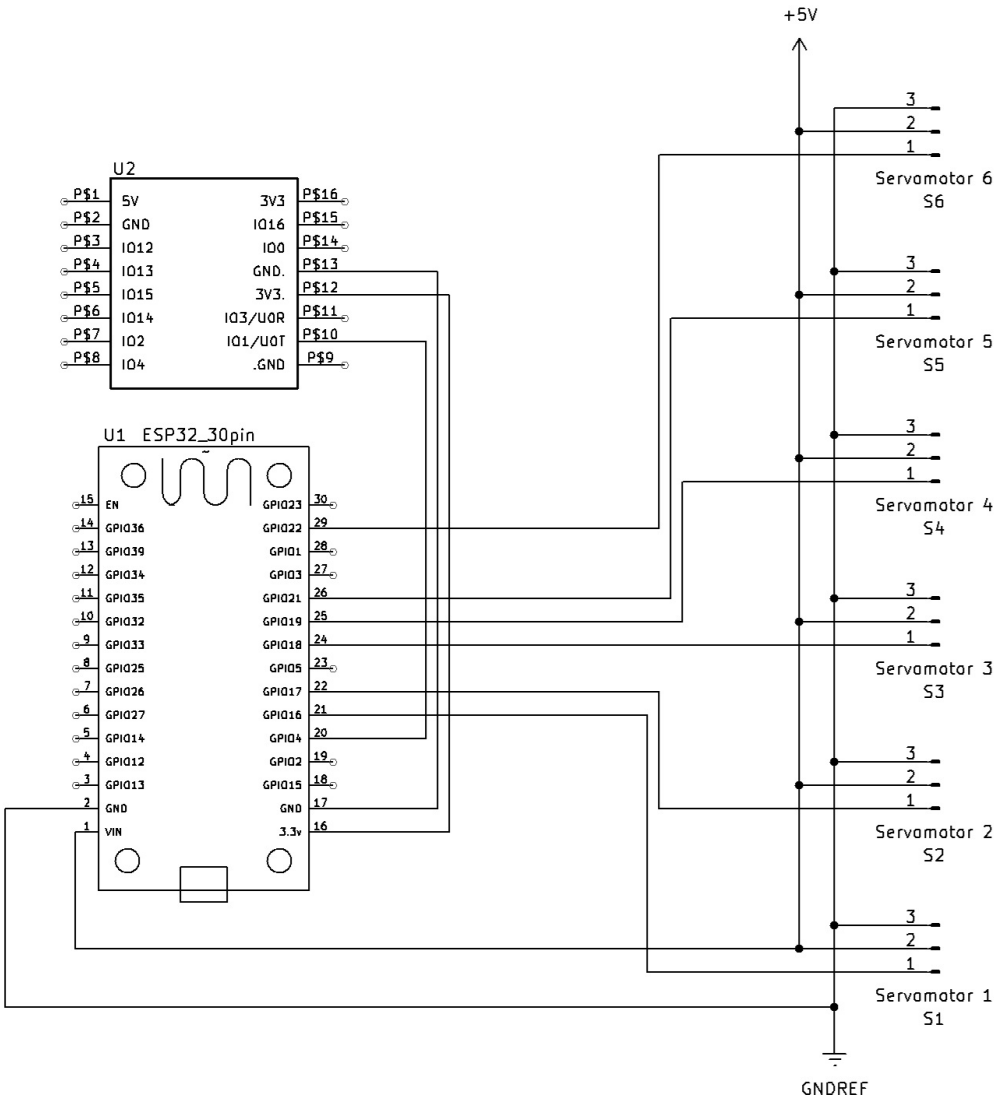


Figura 3.1: Diagrama de conexiones

4. Implementación de Control PID

4.1. Descripción del modelo de control

El control PID es una estrategia clásica de control que a través de numerosas décadas se ha encargado del control de una gran cantidad de plantas en el mundo. Su abreviación significa lo siguiente.

- **Proporcional (P):** Reacciona al error actual. Cuanto mayor es el error, mayor es la corrección. Reduce el error pero puede dejar uno residual en estado estacionario. Por sí mismo, siempre estará por sobre el setpoint o por debajo, haciendo que sea muy poco probable la obtención de un error igual a cero.
- **Integral (I):** Su misión es acumular error a lo largo del tiempo, eliminando el error residual de estado estacionario que puede generar el parámetro Proporcional. En algunos sistemas puede generar inestabilidad al usarse, mientras que en otros es un correctivo completamente eficaz.
- **Derivativa (D):** Predice errores futuros basándose en la tasa de cambio actual de la variable, amortiguando las oscilaciones y mejorando el tiempo de respuesta. Tal como su nombre lo indica, su funcionamiento radica fundamentalmente en la pendiente entre las señales de error detectadas, permitiendo que si el error aumenta con rapidez, el sistema también sea capaz de reaccionar a ella.

Estos parámetros reconocidos como Kp, Ki y Kd, actúan en función de una variable de procesos a medir, el setpoint y el error. Donde el setpoint es el valor deseado, la variable de procesos es el valor en tiempo real de la medición, y el error es la diferencia entre la variable de procesos y el setpoint, es decir, lo que se busca corregir.

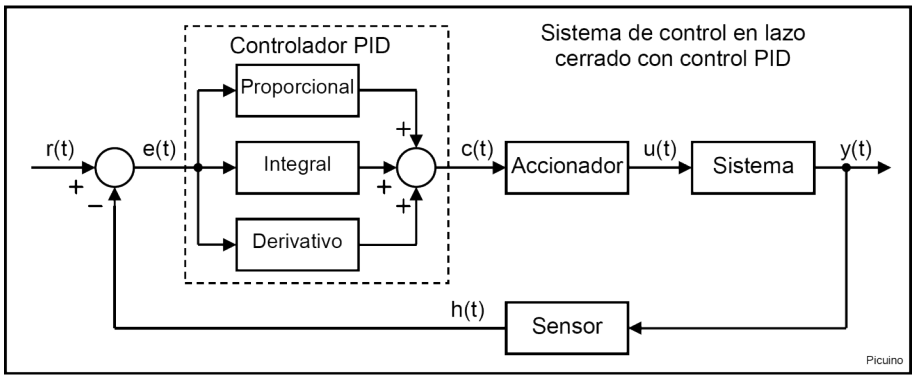


Figura 4.1: Diagrama de bloques de control PID

Para este sistema, la variable a medir, es la posición de la bola roja sobre la plataforma de 20 [cm] x 20 [cm]. El setpoint, o referencia, por defecto del sistema es el centro de la plataforma.

Para medir la posición se utilizará la ESP32-CAM, la cual hace un mapeo en pixeles, permitiendo así leer coordenadas físicas haciendo la detección de color sobre la superficie blanca. La resolución de la cámara es baja para priorizar una mayor frecuencia de datos (160x120p).

En base a los datos de la cámara, se trabajará en el controlador corrigiendo el error en este mapa de coordenadas basado en pixeles en cualquier dirección, sea izquierda, derecha, adelante o atrás.

Para esto, será necesario establecer un punto neutro para los actuadores, que se considere como el punto donde la plataforma está completamente horizontal.

Por lo tanto, la variable a medir, es la posición de la bola en todo momento; la reacción del sistema debe ser mayor o menor dependiendo de qué tan alejada esté del centro, o el setpoint seleccionado, y también debe reaccionar frente a la velocidad que la bola pueda alcanzar por medio de la inclinación generada en la plataforma superior.

Las perturbaciones del sistema, están ideadas para que sean la intervención de personas moviendo la bola con su mano, así generando una interacción directa con el sistema. Así, la plataforma frente a un movimiento en cualquier dirección, debe ser capaz de corregir su posición mediante movimientos que hagan a la bola tender a posicionarse en el setpoint. Según la variación de los parámetros de control, esto puede llegar a hacerse más rápido, más lento, o llegar a un punto de inestabilidad.

4.2. Formulación matemática del controlador PID

El controlador Proporcional–Integral–Derivativo (PID) utilizado en este proyecto tiene como objetivo generar la inclinación necesaria de la plataforma Stewart para mantener la posición de la bola en un punto deseado. El control se aplica de manera independiente en los ejes X e Y, formando dos lazos de control con realimentación obtenida desde el sistema de visión.

4.2.1. Ecuación PID en tiempo continuo

La versión continua del controlador PID se expresa como:

$$u(t) = K_p\, e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{de(t)}{dt} \tag{4.1}$$

donde:

- $e(t)$ es el error entre la posición deseada y la posición actual de la bola,
- K_p es la ganancia proporcional,
- K_i es la ganancia integral,
- K_d es la ganancia derivativa,
- $u(t)$ corresponde a la inclinación que debe realizar la plataforma en cada eje.

4.2.2. Ecuación PID en tiempo discreto (implementada en el ESP32)

Dado que el ESP32 opera de forma digital, el controlador se implementa utilizando una formulación discreta con un período de muestreo T_s. Así, el controlador PID en tiempo discreto queda expresado como:

$$u[k] = K_p\, e[k] + K_i\, T_s \sum_{i=0}^{k} e[i] + K_d \frac{e[k] - e[k-1]}{T_s} \tag{4.2}$$

Para su implementación práctica, cada componente del PID se calcula de la siguiente manera:

$$P[k] = K_p\, e[k] \tag{4.3}$$

$$I[k] = I[k-1] + K_i\, e[k]\, T_s \tag{4.4}$$

$$D[k] = K_d \frac{e[k] - e[k-1]}{T_s} \tag{4.5}$$

y la salida total del controlador es:

$$u[k] = P[k] + I[k] + D[k] \tag{4.6}$$

4.2.3. Rol de cada término

- **Acción proporcional (P):** entrega una corrección inmediata ante variaciones del error, permitiendo una respuesta rápida.
- **Acción integral (I):** elimina el error estacionario acumulando el error en el tiempo.
- **Acción derivativa (D):** suaviza el movimiento anticipándose a cambios bruscos del error, reduciendo oscilaciones.

4.2.4. Control independiente de los ejes X e Y

El sistema se modela con dos controladores PID independientes, ya que la diferencia de resolución entre ejes, ocasiona que los mismos valores puedan generar comportamientos distintos para los ejes. Los valores pueden ser cercanos pero no suelen ser iguales:

$$u_x[k] = PID(e_x[k]) \tag{4.7}$$

$$u_y[k] = PID(e_y[k]) \tag{4.8}$$

Cada salida se interpreta como una inclinación deseada:

$$\theta_x = u_x[k] \tag{4.9}$$

$$\theta_y = u_y[k] \tag{4.10}$$

Estas inclinaciones son utilizadas posteriormente como entrada para la función de cinemática inversa (IK), encargada de calcular los ángulos finales de los seis servomotores.

4.3. Implementación del Control PID en el ESP32

La ejecución del controlador PID se realiza directamente en el microcontrolador ESP32, encargado de procesar el error recibido desde el sistema de percepción, calcular las acciones de control en ambos ejes y generar las señales de comando para los seis servomotores de la plataforma Stewart. En esta sección se describe la estructura del lazo de control, las técnicas de limitación aplicadas y el proceso de calibración necesario para garantizar un funcionamiento estable.

4.3.1. Estructura general del lazo de control

El ESP32 ejecuta el controlador PID con un período de muestreo constante T_s, determinado empíricamente en función de la velocidad de adquisición y procesamiento de la imagen. En cada iteración del lazo, el microcontrolador realiza las siguientes tareas:

1. Recibe la posición de la bola (x, y) proveniente del módulo ESP32-CAM mediante comunicación serial.

2. Calcula los errores en cada eje:

$$e_x[k] = x_{\text{set}} - x[k] \tag{4.11}$$

$$e_y[k] = y_{\text{set}} - y[k] \tag{4.12}$$

3. Evalúa ambas señales de error en sus respectivos controladores PID.

4. Obtiene las inclinaciones requeridas en cada eje:

$$\theta_x[k] = PID(e_x[k]) \tag{4.13}$$

$$\theta_y[k] = PID(e_y[k]) \tag{4.14}$$

5. Aplica límites mecánicos y filtrado básico.

6. Utiliza la función de cinemática inversa (IK) para transformar θ_x y θ_y en los ángulos de servo:

$$\{\alpha_1, \alpha_2, \ldots, \alpha_6\} = IK(\theta_x, \theta_y) \tag{4.15}$$

7. Envía los comandos PWM a los seis servomotores.

4.3.2. Limitaciones mecánicas y prevención de windup

La plataforma Stewart utilizada posee un rango físico restringido de inclinaciones. Para evitar daños estructurales y comportamientos inestables, la salida del controlador se acota mediante:

$$u_{\text{sat}} = \text{máx}(\text{mín}(u[k],\, \theta_{\text{máx}}),\, -\theta_{\text{máx}}) \tag{4.16}$$

donde $\theta_{\text{máx}} = 14°$ representa el límite experimental seguro.

Para prevenir el fenómeno de *integral windup*, el acumulador integral se congela cuando la salida del PID alcanza saturación:

$$I[k] = \begin{cases} I[k-1] + K_i e[k] T_s, & \text{si } u[k] \text{ no está saturado} \\ I[k-1], & \text{si } u[k] \text{ está saturado} \end{cases} \tag{4.17}$$

Esto evita oscilaciones crecientes y mejora el tiempo de asentamiento.

4.3.3. Conversión de inclinación a ángulos de servomotor

La salida del PID no se envía directamente a los servomotores. En cambio, se interpreta como una inclinación deseada de la plataforma. Esta inclinación se transforma mediante la función de cinemática inversa:

$$\alpha_i[k] = f_i(\theta_x[k],\, \theta_y[k]) \qquad \text{para } i = 1, \ldots, 6 \tag{4.18}$$

Cada ángulo α_i se traduce posteriormente en un pulso PWM, respetando los límites de operación del servomotor MG996R:

$$PWM_i[k] = m \cdot \alpha_i[k] + b \tag{4.19}$$

donde m y b corresponden al mapeo lineal entre grados y microsegundos.

4.3.4. Proceso de calibración inicial

Antes de habilitar el lazo de control, la plataforma debe ser calibrada para garantizar una referencia geométrica consistente. Este proceso incluye:

1. **Nivelación de la plataforma**: se determina la posición en la cual la superficie superior queda completamente horizontal. Este punto define el estado $\theta_x = 0$ y $\theta_y = 0$.

2. **Corrección de offsets mecánicos**: debido a tolerancias de impresión 3D, variaciones en las rótulas y diferencias entre servos, cada actuador presenta un desalineamiento inicial. Se define un vector de offsets:

$$\text{offset}_i, \qquad i = 1, \ldots, 6 \tag{4.20}$$

   que se suma directamente a la salida de la cinemática inversa para cada servo.

3. **Validación de movimientos simétricos**: se comprueba que una inclinación positiva o negativa en X o Y produzca movimientos consistentes y libres de asimetrías.

El proceso de calibración permite que la plataforma responda de forma uniforme ante los comandos del PID y que el punto central de control coincida con el centro físico de la superficie, condición necesaria para lograr estabilidad prolongada durante el balanceo de la bola.

A continuación se encuentran los principales fragmentos del código inicial de calibración.

```cpp
#include <ESP32Servo.h>
const int servoPins[6] = {16, 17, 18, 19, 21, 22};
float offsetDeg[6] = {0,0,0,0,0,0};
const float CONTROL_MIN = -14.0f;
const float CONTROL_MAX =  14.0f;
const int SERVO_MIN_US = 1000;
const int SERVO_MAX_US = 2000;
Servo servos[6];
int mapAngleToPWM(float angleDeg, int idx)
{float corrected = angleDeg + offsetDeg[idx];
  float span = SERVO_MAX_US - SERVO_MIN_US;
  float pwm;
  if (idx % 2 == 0)
  {
    pwm = (corrected - CONTROL_MIN) * span / (CONTROL_MAX - CONTROL_MIN) + SERVO_MIN_US;
  }
  else
  {
    pwm = (corrected - CONTROL_MIN) * (-span) / (CONTROL_MAX - CONTROL_MIN) + SERVO_MAX_US;
  }

  return constrain((int)lroundf(pwm), SERVO_MIN_US, SERVO_MAX_US);
}
```

Figura 4.2: Fragmento de definición de servomotores

```cpp
for (int i = 0; i < 6; i++)
{
  servos[i].attach(servoPins[i], SERVO_MIN_US, SERVO_MAX_US);
}
for (int i = 0; i < 6; i++)
{
  int us = mapAngleToPWM(0.0f, i);
  servos[i].writeMicroseconds(us);
}
```

Figura 4.3: Fragmento de escritura de ángulos

A partir de este código, se logró la posición completamente horizontal de la plataforma en este llamado "punto neutro", variando los offsets en un rango máximo de ± 15°.

4.4. Integración PID + Cinemática Inversa (IK)

El controlador PID opera a nivel de *plataforma*, generando como salida una inclinación deseada en los ejes X e Y. Sin embargo, la plataforma Stewart está accionada por seis servomotores dispuestos en una configuración paralela, por lo que resulta necesario transformar dicha inclinación en los ángulos individuales de cada servo. Esta transformación se realiza mediante la *cinemática inversa* (Inverse Kinematics, IK), la cual relaciona la orientación deseada de la plataforma con las longitudes (o ángulos equivalentes) de las seis patas actuadas.

En esta sección se describe el modelo geométrico simplificado utilizado, la forma en que se traduce la salida del PID a una pose deseada de la plataforma, y cómo se obtienen finalmente los ángulos de comando para cada servomotor.

4.4.1. Modelo geométrico simplificado de la plataforma Stewart

Para el desarrollo de la cinemática inversa se considera un modelo geométrico de la plataforma Stewart con las siguientes hipótesis:

- La plataforma superior se modela como un sólido rígido.
- La altura de operación z_0 se mantiene aproximadamente constante durante el proceso de balanceo.
- La orientación de la plataforma se describe mediante dos ángulos pequeños: inclinación en X (balanceo) e inclinación en Y (cabeceo). La rotación en torno al eje Z (guiñada) se mantiene nula.

Se definen dos sistemas de referencia:

- Un sistema fijo en la base, con origen en el centro de la base inferior, denotado como marco $\{B\}$.
- Un sistema móvil asociado a la plataforma superior, con origen en el centro de la plataforma, denotado como marco $\{P\}$.

Las posiciones de las rótulas de la plataforma en su propio marco se representan como:

$$\mathbf{p}_i^P = \begin{bmatrix} x_i^P \\ y_i^P \\ z_i^P \end{bmatrix}, \qquad i = 1, \ldots, 6 \tag{4.21}$$

Mientras que las posiciones de las rótulas fijas en la base, en el marco $\{B\}$, se representan como:

$$\mathbf{b}_i^B = \begin{bmatrix} x_i^B \\ y_i^B \\ z_i^B \end{bmatrix}, \qquad i = 1, \ldots, 6 \tag{4.22}$$

Las coordenadas $\mathbf{p}_i^P$ y $\mathbf{b}_i^B$ se determinan a partir del diseño mecánico (diámetro de la base, separación angular de las rótulas, etc.) y permanecen constantes durante la operación.

4.4.2. Pose deseada a partir de la salida del controlador PID

El controlador PID, aplicado de forma independiente en los ejes X e Y, entrega como salidas las inclinaciones deseadas θ_x y θ_y de la plataforma, tal como se definió en la sección anterior:

$$\theta_x[k] = u_x[k] \tag{4.23}$$

$$\theta_y[k] = u_y[k] \tag{4.24}$$

La orientación de la plataforma respecto a la base se describe mediante una matriz de rotación $\mathbf{R}_{BP}$ que, bajo la hipótesis de ausencia de guiñada, puede aproximarse como la combinación de dos rotaciones sucesivas:

$$\mathbf{R}_{BP} = \mathbf{R}_x(\theta_x)\,\mathbf{R}_y(\theta_y) \tag{4.25}$$

donde:

$$\mathbf{R}_x(\theta_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix} \tag{4.26}$$

$$\mathbf{R}_y(\theta_y) = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \tag{4.27}$$

La traslación del centro de la plataforma respecto al centro de la base se modela como:

$$\mathbf{t} = \begin{bmatrix} 0 \\ 0 \\ z_0 \end{bmatrix} \tag{4.28}$$

Donde z_0 es la altura media fija de operación. La posición de cada rótula de la plataforma expresada en el marco de la base se obtiene entonces como:

$$\mathbf{p}_i^B = \mathbf{R}_{BP}\,\mathbf{p}_i^P + \mathbf{t} \tag{4.29}$$

4.4.3. Cálculo de las longitudes de las patas (cinemática inversa)

Cada pata de la plataforma Stewart conecta un punto fijo en la base con un punto móvil en la plataforma. El vector que une ambos puntos se calcula como:

$$\mathbf{l}_i = \mathbf{p}_i^B - \mathbf{b}_i^B \tag{4.30}$$

La longitud requerida de la pata i resulta de la norma de dicho vector:

$$L_i = \|\mathbf{l}_i\| = \sqrt{(x_i^B - x_i^{P\prime})^2 + (y_i^B - y_i^{P\prime})^2 + (z_i^B - z_i^{P\prime})^2} \tag{4.31}$$

donde $(x_i^{P\prime}, y_i^{P\prime}, z_i^{P\prime})$ corresponden a las componentes de $\mathbf{p}_i^B$.

En el diseño particular de este proyecto, cada servomotor acciona una biela de longitud conocida, unida mediante rótulas a la base y a la plataforma. La longitud L_i se relaciona geométricamente con el ángulo del brazo del servo, de manera que la cinemática inversa entrega finalmente un valor angular deseado para cada actuador.

4.4.4. Conversión longitud–ángulo para cada servomotor

Para cada pata se define un triángulo mecánico equivalente, compuesto por:

- El brazo del servomotor de longitud r.
- La biela de conexión de longitud d.
- La distancia fija entre el eje del servo y la rótula de la base.

A partir de esta geometría se obtiene una relación trigonométrica entre la longitud L_i y el ángulo de giro del servomotor α_i. De forma general, puede expresarse:

$$\alpha_i = g_i(L_i) \tag{4.32}$$

Donde $g_i(\cdot)$ representa la función geométrica específica de la pata i, derivada de las dimensiones reales de la plataforma. En la implementación en firmware, esta relación se codifica como una función que, a partir de la inclinación deseada (θ_x, θ_y), entrega directamente los ángulos de comando α_i.

Además, para compensar desalineamientos mecánicos y garantizar que la plataforma se encuentre horizontal cuando el controlador se encuentra en reposo, se considera un vector de offsets de calibración:

$$\alpha_i^{\text{cmd}} = \alpha_i + \text{offset}_i, \qquad i = 1, \ldots, 6 \tag{4.33}$$

donde offset_i corresponde al ángulo adicional necesario para que, con la plataforma nivelada, el servomotor i quede correctamente referenciado.

4.4.5. Flujo de integración PID–IK en tiempo real

El proceso completo de integración entre el controlador PID y la cinemática inversa puede resumirse en los siguientes pasos ejecutados en cada ciclo de control:

1. El sistema de visión entrega la posición actual de la bola (x, y).
2. Se calculan los errores e_x y e_y y se evalúan los controladores PID correspondientes, obteniendo θ_x y θ_y.
3. La pareja de inclinaciones (θ_x, θ_y) define la orientación deseada de la plataforma, que se traduce en posiciones $\mathbf{p}_i^B$ para cada rótula superior.
4. A partir de $\mathbf{p}_i^B$ y $\mathbf{b}_i^B$ se calculan las longitudes L_i de cada pata.
5. Las longitudes L_i se convierten en ángulos de servomotor α_i mediante la geometría mecánica definida.
6. Se aplican los offsets de calibración offset_i para obtener los ángulos finales de comando α_i^{cmd}.
7. Los ángulos α_i^{cmd} se mapean a señales PWM y se envían a los seis servomotores.

Esta integración permite que la lógica de control (PID) permanezca desacoplada de los detalles mecánicos de la plataforma, mientras que la cinemática inversa se encarga de traducir la inclinación requerida en los movimientos coordinados de los seis actuadores, logrando así el balanceo de la bola sobre la superficie de la plataforma Stewart.

4.5. Proceso de Sintonización del Control PID

La sintonización del controlador PID constituye una etapa crítica en el diseño del sistema de control de la plataforma Stewart. Debido a la naturaleza altamente no lineal del mecanismo, la presencia de múltiples actuadores acoplados, la variabilidad mecánica asociada a la fabricación mediante impresión 3D y el retardo inherente al procesamiento de imagen de la ESP32-CAM, los métodos clásicos de sintonización (como Ziegler–Nichols) no entregaron resultados satisfactorios. Este tipo de métodos supone sistemas lineales, monovariables y con retrasos mínimos, condiciones que no se cumplen en la plataforma desarrollada.

Por estas razones, se optó por un proceso de sintonización *empírico* basado en la observación directa del comportamiento dinámico del sistema, también conocido como *tuning por respuesta*. Este enfoque resulta apropiado para sistemas no lineales donde un modelo matemático preciso es difícil de obtener y donde el controlador debe adaptarse a condiciones físicas reales como fricción, elasticidad estructural y variaciones entre servomotores.

4.5.1. Controlador base y primeras observaciones

Se partió desde un controlador PID funcional que, si bien permitía evitar que la bola cayera de la plataforma, presentaba oscilaciones significativas y movimientos bruscos. Este controlador inicial se caracterizaba por:

- un valor de K_p lo suficientemente alto para lograr correcciones rápidas,
- un valor considerable de K_d que generaba una acción derivativa fuerte y un frenado predictivo,
- un término integral reducido para evitar acumulación excesiva en presencia de retardos.

Este punto de partida permitió establecer un marco inicial para evaluar cómo cada ganancia influía en la estabilidad y suavidad del sistema.

Durante esta fase se observaron los siguientes comportamientos típicos:

- Para valores altos de K_p, la respuesta era rápida pero generaba oscilaciones crecientes.

- Al aumentar K_d, la plataforma lograba frenar más eficazmente la bola, aunque a costa de introducir vibraciones mecánicas de mayor magnitud mientras más grande fuera el K_d.
- Para valores bajos de K_p y K_d, el sistema reaccionaba con demasiada lentitud, permitiendo que la bola escapara del área de control.

Estas observaciones permitieron identificar el régimen dinámico del sistema, distinguiendo entre estados sub-amortiguados, sobre-amortiguados e inestables.

4.5.2. Ajuste iterativo de las ganancias del PID

Una vez caracterizado el comportamiento inicial, se aplicó un proceso de ajuste iterativo guiado por las siguientes directrices:

1. **Ajuste del término derivativo K_d:** dado que la vibración observada estaba estrechamente relacionada con una acción derivativa excesiva, se procedió a reducir gradualmente su valor. Este ajuste permitió atenuar oscilaciones de alta frecuencia y suavizar el movimiento de la plataforma, manteniendo al mismo tiempo la capacidad de frenar el desplazamiento de la bola.
2. **Ajuste proporcional K_p:** establecido un valor adecuado de K_d, se ajustó K_p con el objetivo de obtener una corrección suficientemente ágil sin inducir oscilaciones. Un K_p demasiado bajo generaba una respuesta lenta, mientras que un valor alto volvía el sistema sub-amortiguado.
3. **Ajuste del término integral K_i:** el término integral se mantuvo pequeño debido al retardo introducido por el sistema de visión. Valores altos de K_i producían acumulación de error (windup) y deterioraban la estabilidad. Se seleccionó un valor que permitiera corregir errores en estado estacionario sin comprometer la estabilidad del lazo.

Este proceso permitió refinar la respuesta del controlador, logrando un equilibrio adecuado entre rapidez, suavidad y estabilidad.

4.5.3. Resultados de la sintonización

El proceso descrito condujo a un conjunto de ganancias que permitió mantener la bola en equilibrio, reducir las oscilaciones y lograr la convergencia hacia el centro de la plataforma. La combinación final de parámetros obtuvo:

- Un movimiento suave y estable,
- Una disminución significativa de vibraciones,
- Capacidad de corrección rápida ante perturbaciones,
- estabilidad sostenida incluso ante movimientos amplios de la plataforma.

En particular, la reducción progresiva del término derivativo tuvo un efecto crítico en la mejora del desempeño, mientras que el ajuste proporcional permitió moldear la agresividad del control sin inducir inestabilidad. La contribución integral, aunque pequeña, ayudó a centrar la bola una vez eliminadas las oscilaciones transitorias.

Este enfoque empírico de sintonización, fundamentado en principios de estabilidad y observación dinámica, resultó adecuado para un sistema complejo y no lineal como la plataforma Stewart utilizada en este proyecto.

4.5.4. Parámetros finales del controlador PID

Tras completar el proceso iterativo de sintonización descrito en las secciones anteriores, se obtuvo un conjunto de parámetros que entregó el mejor compromiso entre rapidez de corrección, estabilidad dinámica y suavidad en los movimientos de la plataforma. Estos valores corresponden a la configuración final utilizada en todas las pruebas experimentales y durante la validación del sistema.

Los parámetros definitivos del controlador PID fueron los siguientes:

- **Ganancias proporcionales por eje**:

$$K_{pX} = 0{,}40, \qquad K_{pY} = 0{,}40$$

Estas ganancias constituyen el nivel base de acción proporcional. Además, el sistema utiliza una estrategia de *gain–scheduling* mediante un incremento fijo de:

$$\Delta K_p = 0{,}40,$$

aplicado cuando la bola se encuentra lejos del centro, aumentando la agresividad del controlador en regiones de error elevado.

- **Ganancias integrales**:

$$K_{iX} = 0{,}0, \qquad K_{iY} = 0{,}0$$

El término integral fue descartado debido a los retardos asociados a la adquisición de imágenes y a la dinámica rápida de la pelota. La inclusión de K_i provocaba inestabilidad y acumulación de error, por lo que se fijó en cero.

- **Ganancias derivativas reales**:

$$K_{dX} = 0{,}11, \qquad K_{dY} = 0{,}125$$

Estas ganancias actúan sobre la velocidad del error normalizado, permitiendo amortiguar la dinámica sin producir vibraciones excesivas. Su ajuste fino fue determinante para lograr un movimiento estable y suave.

- **Filtro pasabajos aplicado a la medición**:

$$\alpha = 0{,}55$$

Este coeficiente controla el suavizado de la señal proveniente de la cámara. Valores mayores generaban ruido perceptible en el control, mientras que valores menores ralentizaban la respuesta. El valor seleccionado logró un equilibrio adecuado.

- **Zona muerta en torno al centro**:

$$\text{deadzone} = 0{,}04$$

Esta zona evita correcciones innecesarias cuando la bola está cercana al equilibrio,

reduciendo oscilaciones residuales.

- **Límite máximo de inclinación**:

$$\theta_{\text{máx}} = 6{,}5^\circ$$

  Este límite fue definido experimentalmente para evitar saturación mecánica y garantizar un movimiento suave y seguro.

- **Límite anti-windup del integrador**:

$$I_{\text{máx}} = 2{,}0,$$

  valor que se mantiene por seguridad, pese a que el término integral fue desactivado en la versión final del controlador.

En conjunto, estos parámetros permitieron obtener un desempeño estable, sin vibraciones excesivas, con buena capacidad de recuperación ante perturbaciones externas y con movimientos coherentes con el diseño mecánico de la plataforma. Estos valores representan la sintonización óptima alcanzada durante el proceso experimental.

4.6. Implementación del Interfaz de Control y Supervisión en LabVIEW

Además del controlador PID implementado en el ESP32, se desarrolló una interfaz de supervisión en LabVIEW destinada a facilitar el monitoreo del sistema, la visualización del movimiento de la bola y el ajuste dinámico de los parámetros del controlador. Esta interfaz cumple un rol fundamental en la experimentación, la validación del desempeño del sistema y la presentación didáctica del proyecto, permitiendo a estudiantes y observadores comprender en tiempo real el efecto de cada parámetro del controlador.

LabVIEW no forma parte del lazo de control principal; su función es la de una *HMI* (Human–Machine Interface) que opera en un nivel superior, proporcionando visualización, registro de datos y ajuste de parámetros, mientras el ESP32 ejecuta el control local en tiempo real.

4.6.1. Objetivos de la interfaz en LabVIEW

El uso de LabVIEW dentro del proyecto tuvo los siguientes propósitos:

- Visualizar en tiempo real la posición de la bola en un plano cartesiano;
- Representar el vector de error entre el setpoint y la posición actual;
- Permitir la modificación interactiva de los parámetros K_p, K_i y K_d;
- Monitorear la respuesta dinámica del sistema durante la sintonización;
- Ofrecer una plataforma didáctica para la demostración del control PID.

El entorno gráfico de LabVIEW, ampliamente utilizado en laboratorios de control y automatización, resultó adecuado para esta finalidad.

4.6.2. Comunicación entre LabVIEW y ESP32 Dev Kit

La comunicación entre LabVIEW y el microcontrolador se implementó mediante el protocolo UART. Para ello, el ESP32 envía periódicamente la posición detectada de la bola en formato de cadena de texto, mientras que LabVIEW interpreta, separa y transforma estos datos en valores numéricos.

Asimismo, cuando el usuario modifica un parámetro del controlador, LabVIEW envía un mensaje de retorno al ESP32 con el nuevo valor, empleando el mismo canal serial. De esta manera, la interfaz permite ajustar los parámetros del PID sin necesidad de reprogramar el microcontrolador.

El flujo de comunicación puede resumirse en los siguientes pasos:

1. La ESP32-CAM detecta la posición (x, y) de la bola y la envía mediante UART.
2. LabVIEW recibe la cadena, la interpreta y actualiza los indicadores gráficos.
3. El usuario ajusta los parámetros del PID directamente en la interfaz.
4. LabVIEW envía los nuevos valores al ESP32 para actualizar el controlador.

Este mecanismo de doble sentido permite una integración fluida entre supervisión y control.

Este intercambio de datos se implementa en el diagrama de bloques de LabVIEW mediante un conjunto reducido de funciones VISA, dispuestas en un bucle de ejecución continua (*while loop*). A continuación se detallan los bloques más relevantes y su función dentro del sistema:

- **VISA Configure Serial Port**: Es el bloque encargado de inicializar el puerto serial. En esta función se selecciona el puerto COM correspondiente al ESP32 (por ejemplo, COM3, COM4, etc.) y se configuran los parámetros de comunicación: velocidad en baudios (por ejemplo 115200 baudios), número de bits de datos, paridad, bits de parada y tiempo de espera (timeout) para las operaciones de lectura. Esta configuración se ejecuta una sola vez al inicio del programa, garantizando que tanto LabVIEW como el ESP32 utilicen los mismos parámetros de enlace.

- **VISA Write**: Este bloque se utiliza para enviar datos desde LabVIEW hacia el ESP32. A partir de los valores ingresados por el usuario en el panel frontal (por ejemplo, nuevas ganancias K_p, K_i y K_d o comandos de operación), se construye una cadena de texto con un formato previamente definido (por ejemplo, valores separados por comas y finalizados en un carácter de fin de línea). El bloque VISA Write toma esta cadena y la transmite por el puerto serial configurado.

- **VISA Read**: este bloque permite recibir datos enviados por el ESP32. En cada iteración del bucle principal, VISA Read intenta leer una cantidad determinada de bytes o hasta que se cumpla el tiempo de espera configurado. La cadena recibida suele contener la posición de la bola y/o otras variables de interés. Posteriormente, se aplica procesamiento en LabVIEW para convertir la cadena en valores numéricos (por ejemplo, separando por delimitadores y utilizando funciones de conversión de texto a número), que luego se utilizan para actualizar los gráficos y cálculos de la interfaz.

- **VISA Close**: al finalizar la ejecución del programa, este bloque cierra el puerto serial y libera los recursos asociados. Su uso es importante para evitar que el puerto quede bloqueado por la aplicación, lo cual impediría que otros programas o futuras ejecuciones de LabVIEW accedan correctamente al mismo puerto.

Adicionalmente, las funciones VISA se conectan mediante el error cluster estándar de LabVIEW, lo que permite propagar información de error a lo largo del diagrama de

bloques. Esto facilita la detección de problemas de comunicación, como la desconexión del dispositivo o la selección incorrecta del puerto COM.

En conjunto, el uso de estos bloques VISA proporciona una comunicación serial robusta y flexible entre LabVIEW y el ESP32, sin necesidad de bibliotecas externas. Gracias a esta integración, la interfaz puede operar como una herramienta de supervisión y ajuste en tiempo real, mientras que el control PID y la cinemática inversa se ejecutan localmente en el microcontrolador.

4.6.3. Diseño del panel frontal

El panel frontal de LabVIEW fue diseñado con énfasis en claridad visual y carácter didáctico. Sus elementos principales son:

- Un gráfico cartesiano donde se representa la posición actual de la bola;
- Un marcador del setpoint en el centro del gráfico;
- Un vector que une la posición real con la posición deseada, indicando la dirección y magnitud del error;
- Controles deslizantes y numéricos para ajustar K_p, K_i y K_d;
- Indicadores adicionales para mostrar magnitudes relevantes del sistema.

El vector de error, en particular, permite visualizar intuitivamente la acción del controlador: un vector grande indica un error significativo, mientras que un vector corto refleja proximidad al equilibrio.

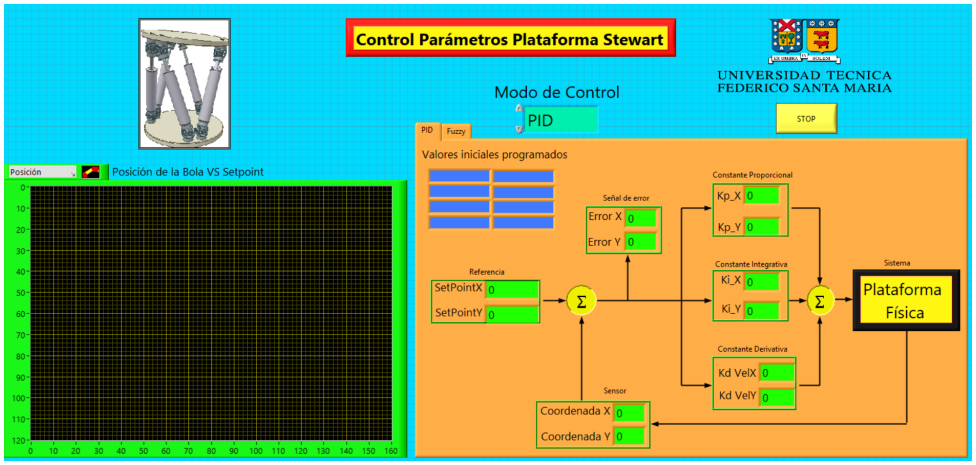


Figura 4.4: Panel en LabVIEW para control PID

4.6.4. Procesamiento de datos dentro de LabVIEW

Para transformar los datos recibidos desde el ESP32 en señales visuales coherentes, LabVIEW realiza las siguientes operaciones internas:

1. Conversión de la cadena de texto en un arreglo numérico;
2. Normalización de los valores para que coincidan con las escalas del gráfico;
3. Construcción de un punto (x, y) mostrado en tiempo real;
4. Cálculo del vector de error como:

$$\mathbf{e} = \begin{bmatrix} x_{\text{set}} - x \\ y_{\text{set}} - y \end{bmatrix}; \tag{4.34}$$

5. Dibujo del vector sobre el plano cartesiano mediante un segmento orientado.

Esta visualización permitió evaluar rápidamente el desempeño del controlador durante la fase de sintonización.

4.6.5. Rol de LabVIEW dentro del sistema de control

Es importante destacar que LabVIEW no interviene directamente en el lazo de control de la plataforma Stewart. El cálculo del PID, la cinemática inversa y la generación de los pulsos PWM se ejecutan exclusivamente en el ESP32. Por lo tanto, LabVIEW actúa como una capa de supervisión externa con las siguientes funciones:

- Monitoreo continuo del estado del sistema.
- Modificación de parámetros del controlador en tiempo real.
- Visualización didáctica del comportamiento de la plataforma.
- Apoyo a la experimentación y análisis del control.

Esta separación entre control y supervisión garantiza que el desempeño del sistema no dependa del computador anfitrión ni del entorno gráfico, manteniendo así la robustez del lazo de control.

4.6.6. Ventajas del uso de LabVIEW en el proyecto

La integración de LabVIEW proporcionó varios beneficios relevantes:

- Permitió observar en tiempo real la dinámica del sistema durante la sintonización.
- Facilitó la comprensión del funcionamiento del PID mediante gráficos.
- Habilitó la demostración del proyecto a estudiantes y público general.
- Permitió ajustar parámetros sin interrupciones y sin recompilar firmware, que es un proceso demoroso al intentar actualizar parámetros volviendo a subir el código.
- Mejoró la documentación experimental y la validación del control.

En conjunto, estas funcionalidades hicieron de LabVIEW una herramienta clave para la interacción humana con el sistema, complementando el trabajo del controlador y fortaleciendo el carácter educativo y didáctico del proyecto.

5. Implementación de Lógica Difusa

5.1. Introducción a la Lógica Difusa

La lógica difusa o también llamada Fuzzy logic, surge como una extensión de la lógica booleana clásica, cuyo propósito es representar matemáticamente conceptos imprecisos o vagos. Mientras que en la lógica tradicional una proposición sólo puede asumir los valores 0 o 1, en lógica difusa la verdad se modela como un valor continuo en el intervalo $[0, 1]$, permitiendo expresar grados intermedios de pertenencia.

El origen formal de esta teoría se remonta al trabajo pionero de Lotfi A. Zadeh, quien en 1965 introduce el concepto de *conjuntos difusos* (*fuzzy sets*) como una generalización de la teoría de conjuntos clásica. En un conjunto difuso, a cada elemento del universo se le asigna un grado de pertenencia $\mu_A(x) \in [0, 1]$, en contraste con la pertenencia estrictamente binaria de los conjuntos tradicionales. Esta formulación permitió modelar de manera matemática conceptos lingüísticos tales como "alto", "cálido" o "cerca del centro", los cuales no poseen límites nítidos.

Posteriormente, Zadeh amplió estos fundamentos mediante la introducción de las llamadas variables lingüísticas, y los sistemas difusos basados en reglas, orientados a aproximar el razonamiento humano en problemas de decisión y control. De manera formal, un conjunto difuso A, definido sobre un universo X, se expresa como:

$$A = \{(x, \mu_A(x)) \mid x \in X,\ 0 \leq \mu_A(x) \leq 1\}, \tag{5.1}$$

Donde $\mu_A(x)$ es la función de pertenencia. Esta estructura permite extender operaciones clásicas como unión, intersección y complemento, redefiniéndolas mediante operadores difusos (t-normas y t-conormas).

5.1.1. De los conjuntos difusos al control difuso

La transición desde los fundamentos matemáticos hacia su aplicación en sistemas de control se consolidó durante la década de 1970. Un hito fundamental fue la investigación

de E. H. Mamdani, quien en 1974 propuso el primer controlador difuso basado en reglas lingüísticas del tipo "SI–ENTONCES", aplicado inicialmente al control de un motor de vapor.

Esta propuesta sentó las bases del *control difuso* contemporáneo, donde la ley de control no se define mediante una ecuación analítica explícita (como ocurre en un controlador PID), sino mediante un conjunto de reglas heurísticas de la forma:

- *SI la bola está a la derecha Y la velocidad es positiva, ENTONCES aplicar corrección fuerte hacia la izquierda.*
- *SI el error es pequeño Y la velocidad es cercana a cero, ENTONCES aplicar una corrección suave o nula.*

Un sistema de control difuso típico está compuesto por tres etapas:

1. **Fuzzificación:** Conversión de variables reales (error, velocidad, temperatura, etc.) en grados de pertenencia a conjuntos difusos tales como "izquierda", "centro" o "derecha".
2. **Inferencia difusa:** Evaluación de las reglas utilizando operadores lógicos difusos, combinando los grados de pertinencia para obtener una salida difusa.
3. **Defuzzificación:** Conversión de la salida difusa en un valor real empleando métodos como el centroide o el promedio ponderado.

5.1.2. Historia y aplicaciones en ingeniería

Tras los avances de Zadeh y Mamdani, el control difuso comenzó a consolidarse a finales de los años 70 e inicios de los 80, especialmente en Japón. Entre las aplicaciones más tempranas y destacadas se encuentran:

- Control de hornos industriales.
- Sistemas de climatización inteligentes.
- Control del Metro de Sendai, uno de los primeros casos de aplicación en transporte.
- Sistemas de navegación y control para vehículos móviles.

Durante las décadas siguientes, los sistemas difusos se incorporaron masivamente a productos de consumo como:

- Lavadoras con ciclos adaptativos basados en sensores.
- Cámaras fotográficas con autoenfoque inteligente.
- Aires acondicionados con regulación continua del confort térmico.
- Transmisiones automáticas optimizadas para consumo de combustible.

En ingeniería y automatización moderna, la lógica difusa se utiliza ampliamente en:

- Control de robots móviles y manipuladores.
- Control de motores y convertidores de potencia.
- Sistemas de climatización y eficiencia energética.
- Procesos químicos y metalúrgicos con alta incertidumbre.
- Vehículos autónomos y sistemas avanzados de asistencia al conductor.

La literatura técnica destaca que los controladores difusos presentan ventajas específicas cuando el sistema:

- Es altamente no lineal,
- Posee incertidumbres estructurales o ruido significativo,
- Carece de un modelo matemático preciso,
- Debe operar con transiciones suaves entre modos de funcionamiento.

Estos atributos hacen del control difuso una herramienta idónea para sistemas como la plataforma Stewart utilizada en este proyecto, la cual exhibe un comportamiento complejo, saturaciones mecánicas y mediciones ruidosas. Además, su naturaleza cualitativa lo convierte en un recurso pedagógico adecuado para integrar conceptos de razonamiento humano y control automático.

5.1.3. Ventajas y características del control difuso

Entre las ventajas más relevantes del control difuso, ampliamente documentadas en la literatura especializada, se encuentran:

- **Manejo nativo de la incertidumbre:** La noción de grados de pertenencia permite trabajar directamente con información imprecisa.
- **No requiere un modelo matemático exacto:** A diferencia de técnicas basadas en ecuaciones de estado, la lógica difusa se basa en reglas y funciones de membresía.
- **Transiciones suaves entre acciones:** Las funciones de pertenencia contínuas evitan saltos bruscos.
- **Apropiado para sistemas no lineales:** Las reglas permiten aproximar comportamientos complejos sin necesidad de linealizaciones.
- **Incorporación directa de conocimiento experto:** Las reglas pueden diseñarse a partir de la experiencia de operadores o del análisis experimental.

No obstante, también se reconocen desafíos inherentes al enfoque:

- Sintonización empírica de las funciones de membresía.
- Análisis formal de estabilidad más complejo que en control lineal clásico.
- Necesidad de validación experimental extensa en sistemas críticos.

En el contexto de este proyecto, la lógica difusa no solo ofrece un desempeño adecuado para estabilizar la bola en la plataforma Stewart, sino que además aporta un enfoque didáctico para demostrar cómo reglas lingüísticas simples pueden implementarse en un microcontrolador para resolver un problema de control en tiempo real.

5.2. Diseño del controlador difuso

El controlador difuso implementado en este proyecto corresponde a un sistema de inferencia tipo Mamdani, aplicado de forma independiente sobre los ejes X e Y de la plataforma Stewart. El objetivo del controlador es generar una acción de corrección capaz

de estabilizar la bola mediante reglas lingüísticas, emulando el razonamiento humano ante la dinámica observada del sistema. Para ello, se emplean como variables de entrada la posición normalizada de la bola y su velocidad normalizada, y como salida una señal difusa de corrección posteriormente convertida en ángulos de servomotor mediante el módulo de cinemática inversa.

El diseño del controlador puede dividirse en cuatro etapas fundamentales: definición de las variables lingüísticas, creación de funciones de membresía, construcción de la base de reglas y selección del método de inferencia y desfusificación.

5.2.1. Definición de variables lingüísticas y normalización

Dado que la cámara entrega posiciones en píxeles, la primera etapa consiste en transformar dichas coordenadas a un espacio adimensional y acotado. Esto se realiza mediante una normalización lineal en el intervalo aproximado $[-1, 1]$, donde valores cercanos a ± 1 representan posiciones próximas al borde del área útil, mientras que valores cercanos a 0 indican cercanía al centro geométrico.

La velocidad se obtiene mediante derivada discreta de la posición normalizada, considerando el intervalo temporal real entre muestras (dt), y posteriormente se reescala mediante el parámetro `VEL_SCALE` para mantenerla en el intervalo $[-1, 1]$. En la versión final del controlador, esta velocidad normalizada se utiliza principalmente en un bloque de **frenado difuso** independiente, cuya finalidad es amortiguar desplazamientos rápidos sin afectar la interpretabilidad del término posicional.

Se definen dos variables lingüísticas de entrada:

- **Posición**: {Izquierda/Abajo, Centro, Derecha/Arriba}
- **Velocidad**: {Negativa, Cero, Positiva}

La variable de salida corresponde a la acción de control normalizada $u \in [-1, 1]$, la cual representa la magnitud de inclinación requerida sobre cada eje. Para cada eje, la salida final se obtiene como la suma de un término posicional (orientado a recentrar la bola) y un término de frenado (orientado a oponerse al movimiento).

5.2.2. Funciones de membresía

Las funciones de membresía adoptan un diseño que busca simplicidad computacional y transiciones suaves. Para la variable **posición**, se definieron conjuntos por eje:

- **Izquierda / Centro / Derecha**: Izquierda y Derecha como funciones lineales saturadas, proporcionales a la distancia al centro en el eje X, y Centro como una función tipo campana recortada, con máximo en el centro y decrecimiento lineal hacia los extremos.

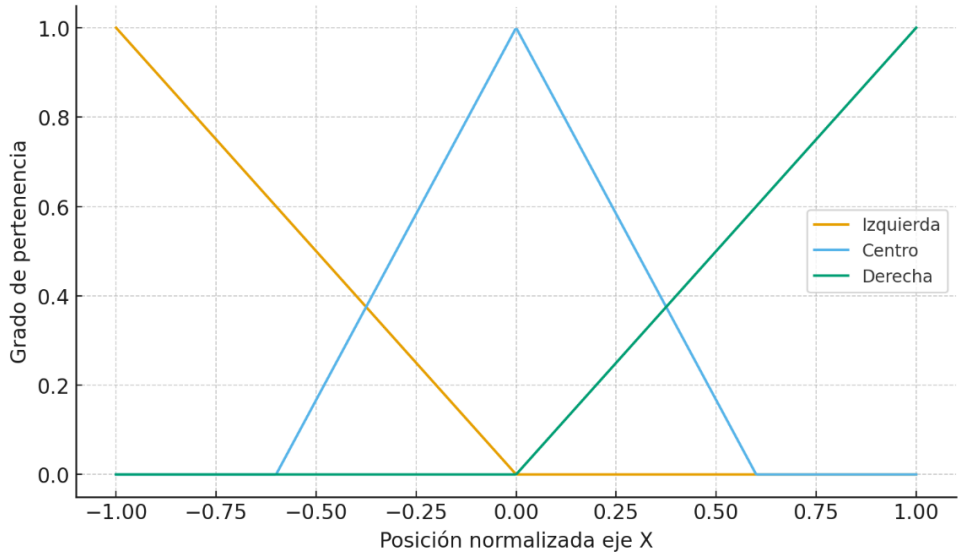


Figura 5.1: Funciones de Pertenencia eje X

- **Abajo / Centro / Arriba**: Abajo y Arriba como funciones lineales saturadas, proporcionales a la distancia al centro en el eje Y, y Centro como una función tipo campana recortada, con máximo en el centro y decrecimiento lineal hacia los extremos.

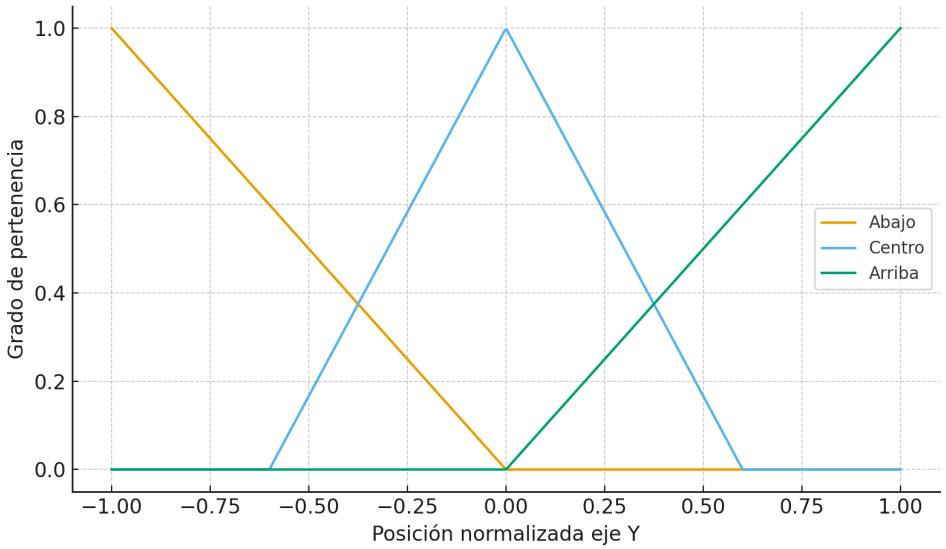


Figura 5.2: Funciones de Pertenencia eje Y

Para la **velocidad**, se definieron las funciones:

- **Negativa**: activa cuando la bola se desplaza hacia el lado izquierdo/abajo.
- **Positiva**: activa cuando la bola se desplaza hacia el lado derecho/arriba.
- **Cero**: función angosta que permite detectar estados cuasi–estáticos y reducir acciones innecesarias.

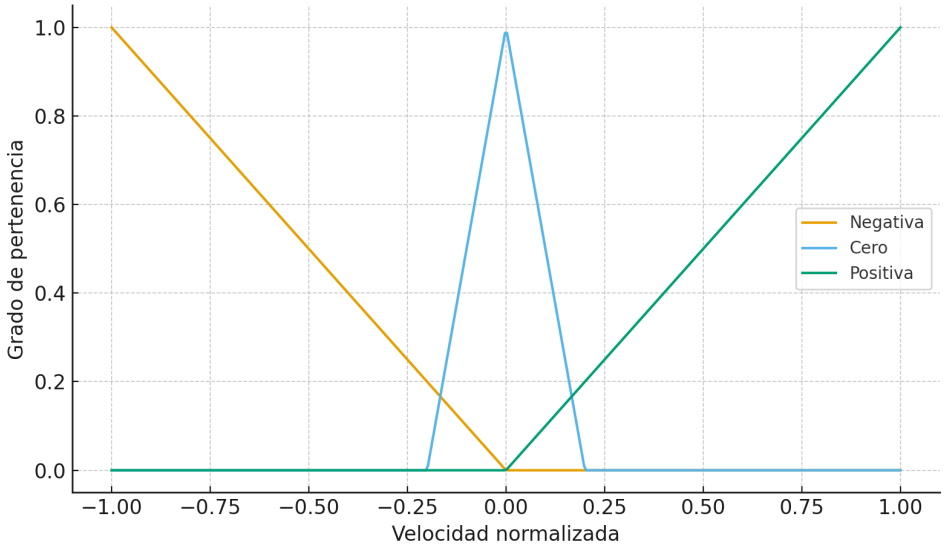


Figura 5.3: Funciones de Pertenencia Velocidad

Este conjunto de funciones permite discriminar entre desplazamientos notorios, suaves variaciones y estados cuasi–estáticos. En particular, en la implementación final la velocidad se utiliza para generar un término de frenado cuyo signo es opuesto al movimiento, y cuya contribución se atenúa automáticamente cuando la rapidez es pequeña.

5.2.3. Base de reglas del controlador difuso

La base de reglas constituye el núcleo del sistema y se diseñó mediante criterios intuitivos de corrección. En la versión final del proyecto, el control difuso se implementa como la suma de dos contribuciones por eje:

1. **Término posicional:** Orientado a generar una inclinación base que recupere la bola hacia el centro, encargándose solo de la lectura con respecto la posición en tiempo real.

2. **Término de frenado:** Orientado a oponerse al movimiento para amortiguar desplazamientos, sean rápidos o lentos. De este modo se complementa a la corrección posicional amortiguando los movimientos.

Reglas representativas del término posicional:

- Si la bola está a la derecha y la velocidad es positiva, entonces la corrección debe ser fuerte hacia la izquierda.

- Si la bola está a la derecha y la velocidad es cero, entonces la corrección debe ser media hacia la izquierda.

- Si la bola está a la derecha y la velocidad es negativa, entonces la corrección debe ser suave hacia la izquierda.

Para la zona central se definieron reglas orientadas a minimizar vibraciones, tales como:

- Si la posición es centro y la velocidad es cero, entonces la salida posicional es cero.

- Si la posición es centro y la velocidad es distinta de cero, se aplica una corrección posicional suave en sentido opuesto.

Reglas representativas del término de frenado (solo velocidad):

- Si la velocidad es positiva, entonces el frenado aplica una corrección de signo negativo.
- Si la velocidad es negativa, entonces el frenado aplica una corrección de signo positivo.
- Si la velocidad es cero, entonces el frenado es cero.

Las reglas se aplican de forma independiente para cada eje, respetando el desacoplamiento cinemático y simplificando el diseño del controlador. La separación explícita entre el término posicional y el término de frenado permite ajustar el amortiguamiento de la dinámica sin modificar directamente la autoridad direccional asociada a la corrección por posición.

5.2.4. Inferencia y desfusificación

El método de inferencia seleccionado corresponde al esquema Mamdani clásico.

El proceso concluye con desfusificación mediante un *centroide simplificado* (promedio ponderado), el cual produce un valor único en el rango $[-1, 1]$. En la implementación final se generan dos salidas por eje:

- u_{pos}: salida difusa asociada al término posicional.
- u_{brk}: salida asociada al término de frenado, de signo opuesto a la velocidad y con magnitud proporcional a la rapidez normalizada.

La acción de control normalizada final por eje se obtiene como:

$$u_{\text{mix}} = \underbrace{\text{GLOBAL\_GAIN} \cdot u_{\text{pos}}}_{\text{corrección posicional}} + \underbrace{\text{GLOBAL\_GAIN} \cdot \text{IMPACT\_BRAKE\_GAIN} \cdot \text{brkG}(v) \cdot w_B(v) \cdot u_{\text{brk}}}_{\text{frenado por velocidad}} \tag{5.2}$$

Donde IMPACT_BRAKE_GAIN es una ganancia fija del firmware, $brkG(v)$ corresponde a un nivel de frenado (suave/medio/fuerte) seleccionado en función de la rapidez, y $w_B(v)$ es un factor de activación que suprime el frenado cerca de velocidad cero para evitar respuestas por ruido.

Finalmente, la salida se multiplica por el ángulo máximo permitido de inclinación:

$$\theta = u_{\text{mix}} \cdot \theta_{\text{máx}}$$

Este enfoque mantiene interpretabilidad y robustez en un sistema no lineal, y permite un ajuste didáctico separado entre corrección posicional y amortiguamiento por velocidad.

5.2.5. Histéresis para reducción de vibraciones

Cercano al centro geométrico pueden aparecer vibraciones debido al cruce repetido por regiones de transición de funciones de membresía. Para mitigar este efecto, se implementó un mecanismo de histéresis por eje, que fija la posición y velocidad a cero cuando se entra en una zona reducida en torno al centro, liberándose al superar un umbral mayor.

Este mecanismo evita oscilaciones innecesarias, mejora la estabilidad local y permite que la bola permanezca en reposo sin movimientos pulsantes. Adicionalmente, al anular la velocidad dentro de la zona de histéresis, el término de frenado se desactiva naturalmente en régimen estacionario.

5.2.6. Ganancia global del sistema

A diferencia del controlador PID, en el cual las ganancias poseen interdependencias, el controlador difuso utiliza un parámetro de ajuste maestro denominado **Ganancia Global**. Este factor escala la magnitud final de la acción de control y afecta tanto al término posicional como al término de frenado:

- Ganancias bajas producen un comportamiento suave y adecuado para demostraciones didácticas.
- Ganancias muy bajas ejecutan una falta de fuerza y corrección total en el sistema.
- Ganancias altas generan correcciones rápidas y enérgicas.
- Ganancias muy altas generan inestabilidad.

Además, el sistema permite ajustar de forma independiente la intensidad del frenado mediante tres niveles configurables desde LabVIEW (BRK-SOFT, BRK-MEDIUM, BRK-

STRONG). Esto habilita una demostración más clara del rol de la velocidad sobre la estabilidad sin modificar la estructura de la base posicional.

5.3. Implementación del controlador difuso en el ESP32

La implementación del controlador difuso en el ESP32 se estructuró considerando las restricciones temporales propias del sistema, la necesidad de mantener una frecuencia de control estable y la obligatoriedad de minimizar la carga computacional para evitar retrasos en la respuesta de la plataforma Stewart. El controlador difuso fue programado en lenguaje C++ en la Arduino IDE, integrándose directamente en el lazo principal de control junto con los módulos de filtrado, normalización, cinemática inversa y accionamiento de servomotores.

El desarrollo siguió un enfoque modular, donde cada fase del proceso difuso (membresías, reglas, inferencia y desfusificación) se implementó mediante funciones optimizadas y de tamaño reducido, permitiendo la ejecución en tiempos muy por debajo del período operacional del sistema.

5.3.1. Estructura general del lazo difuso

El lazo de control se ejecuta únicamente al recibir un nuevo paquete de coordenadas provenientes del módulo ESP32-CAM. Para ello, se utiliza una marca temporal asociada a la llegada del último cuadro. Si no hay actualización de datos, el controlador no realiza cálculos, evitando inconsistencias en la derivada y artefactos debidos a retardos.

El flujo general del controlador difuso implementado es:

1. Lectura del paquete de posición (x, y) enviado por la cámara.
2. Aplicación de un filtro pasabajos para reducir ruido.
3. Normalización de la posición al rango $[-1, 1]$.
4. Cálculo de la velocidad normalizada mediante derivada discreta y escalado VEL-SCALE.
5. Aplicación de histéresis en torno al centro.
6. Evaluación de funciones de pertenencia para posición y velocidad.

7. Cálculo del término posicional u_{pos} por eje.

8. Cálculo del término de frenado u_{brk} por eje a partir de la velocidad.

9. Combinación de ambos términos y escalamiento mediante GLOBAL-GAIN.

10. Conversión a ángulos de inclinación (roll y pitch).

11. Envío al módulo de cinemática inversa (IK).

En la versión final del proyecto, el amortiguamiento asociado a la velocidad se concentra en el término de frenado difuso, facilitando el ajuste desde LabVIEW y mejorando la estabilidad ante movimientos rápidos.

5.3.2. Implementación de las funciones de membresía

Cada función de pertenencia se implementó como una operación algebraica directa, evitando el uso de librerías externas que podrían incrementar el tiempo de ejecución. Las funciones son lineales a tramos, lo que facilita que la evaluación consista únicamente en comparaciones y multiplicaciones simples.

Por ejemplo, la función central de posición se implementa como:

$$\mu_C(x) = \text{máx}\left(0,\ \text{mín}\left(1,\ 1 - \frac{|x|}{B}\right)\right)$$

Donde B representa el ancho efectivo de la región central definido por diseño. Este enfoque permite evaluaciones determinísticas, sin operaciones costosas como exponenciales.

5.3.3. Evaluación de reglas difusas

La base de reglas se implementó utilizando una estructura condensada mediante un acumulador numérico y un peso total. Para el término posicional, cada regla se activa con:

$$\mu_{\text{regla}} = \text{mín}(\mu_{\text{pos}})$$

Y aporta una contribución ponderada a la salida:

$$\text{num} \leftarrow \text{num} + \mu_{\text{regla}} \cdot u_{\text{salida}}$$

Para el término de frenado, la activación se realiza exclusivamente a partir de membresías de velocidad, generando una salida de signo opuesto al movimiento y cuya magnitud crece con la rapidez normalizada. Adicionalmente, el frenado se pondera mediante un factor $w_B(v)$ que anula su acción cerca de $v \approx 0$, y mediante un nivel $brkG(v)$ (suave/medio/fuerte) seleccionado según la rapidez.

5.3.4. Desfusificación y ganancia global

Una vez acumuladas las salidas de las reglas activas, el sistema realiza una desfusificación por *promedio ponderado*:

$$u = \frac{\text{num}}{\text{den}}$$

Donde "den" es la suma de los grados de activación relevantes. En la implementación final, la salida por eje se construye como:

$$u_{\text{mix}} = \text{GLOBAL\_GAIN} \cdot u_{\text{pos}} + \text{GLOBAL\_GAIN} \cdot \text{IMPACT\_BRAKE\_GAIN} \cdot brkG(v) \cdot w_B(v) \cdot u_{\text{brk}}$$

Finalmente, la salida se convierte en inclinación real multiplicando por el ángulo máximo permitido:

$$\theta = u_{\text{mix}} \cdot \theta_{\text{máx}}$$

Estas señales corresponden a los comandos de roll y pitch enviados al módulo de cinemática inversa.

5.4. Ajuste de Parámetros del Controlador Difuso

El desempeño del controlador difuso implementado depende directamente de un conjunto acotado de parámetros ajustables que permiten moldear su respuesta dinámica y su comportamiento ante perturbaciones. A diferencia del controlador PID, cuya sintonización se basa en ganancias lineales asociadas a términos matemáticos específicos, el control difuso permite modificar la intensidad relativa de reglas y bloques funcionales, manteniendo una interpretación intuitiva de cada parámetro adaptada a lenguaje comprensible con

mucha más facilidad.

El proceso de ajuste descrito en esta sección se realizó de manera experimental sobre la plataforma Stewart física, utilizando la interfaz desarrollada en LabVIEW para modificar los parámetros en tiempo real y observar su impacto inmediato sobre la dinámica del sistema. Este enfoque permitió refinar el comportamiento del controlador sin recurrir a modelos matemáticos explícitos, apoyándose en la observación directa del sistema.

5.4.1. Parámetros regulables desde LabVIEW

El firmware desarrollado para el ESP32 permite modificar en tiempo real los siguientes parámetros asociados al controlador difuso:

- **Ganancia global (GLOBAL-GAIN)**: escala común que afecta tanto al término posicional como al término de frenado. Define la magnitud total de la inclinación aplicada por la plataforma.
- **Parámetros posicionales (FUZZY-SOFT, FUZZY-MEDIUM, FUZZY-STRONG)**: Pesos asociados a la base de reglas posicionales. Controlan la autoridad direccional del sistema para recentrar la bola según su distancia al centro.
- **Parámetros de frenado (BRAKE-SOFT, BRAKE-MEDIUM, BRAKE-STRONG)**: Ganancias asociadas al bloque de frenado difuso, que actúa exclusivamente en función de la velocidad normalizada.

La interfaz LabVIEW permite modificar estos parámetros mediante campos numéricos, facilitando la experimentación controlada y el análisis comparativo entre distintas configuraciones sin necesidad de recompilar el código una y otra vez en Arduino IDE. Por defecto al alimentar todo el sistema, el código inicia ejecutando los valores programados por defecto, que están ideados para una respuesta inicial suave y sin descontrol antes del ingreso de valores nuevos de forma manual en el panel de LabVIEW. Al dar inicio al programa de LabVIEW, por defecto los campos de entrada figurarán en 0, pero esto no quiere decir que se envíe este valor a los parámetros, sino que es simplemente algo visual.

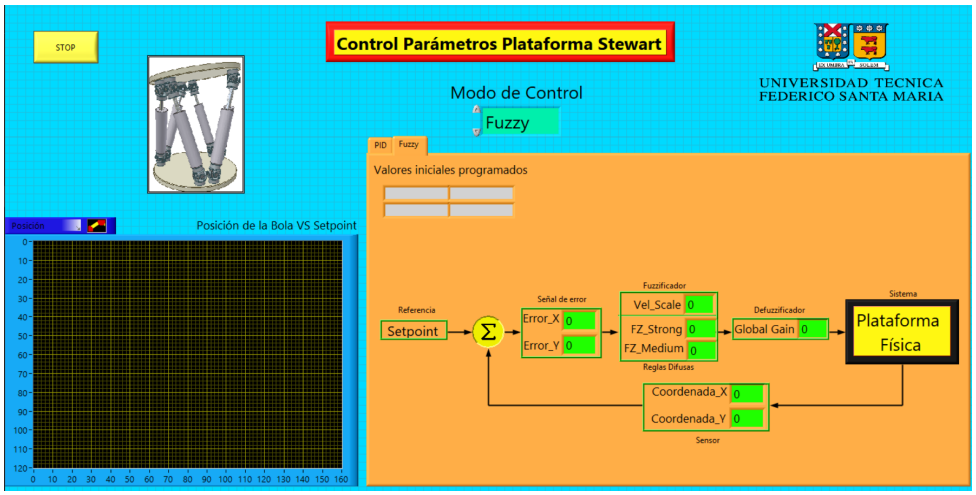


Figura 5.4: Panel en LabVIEW para Fuzzy

El código completo confeccionado en LabVIEW es el siguiente:

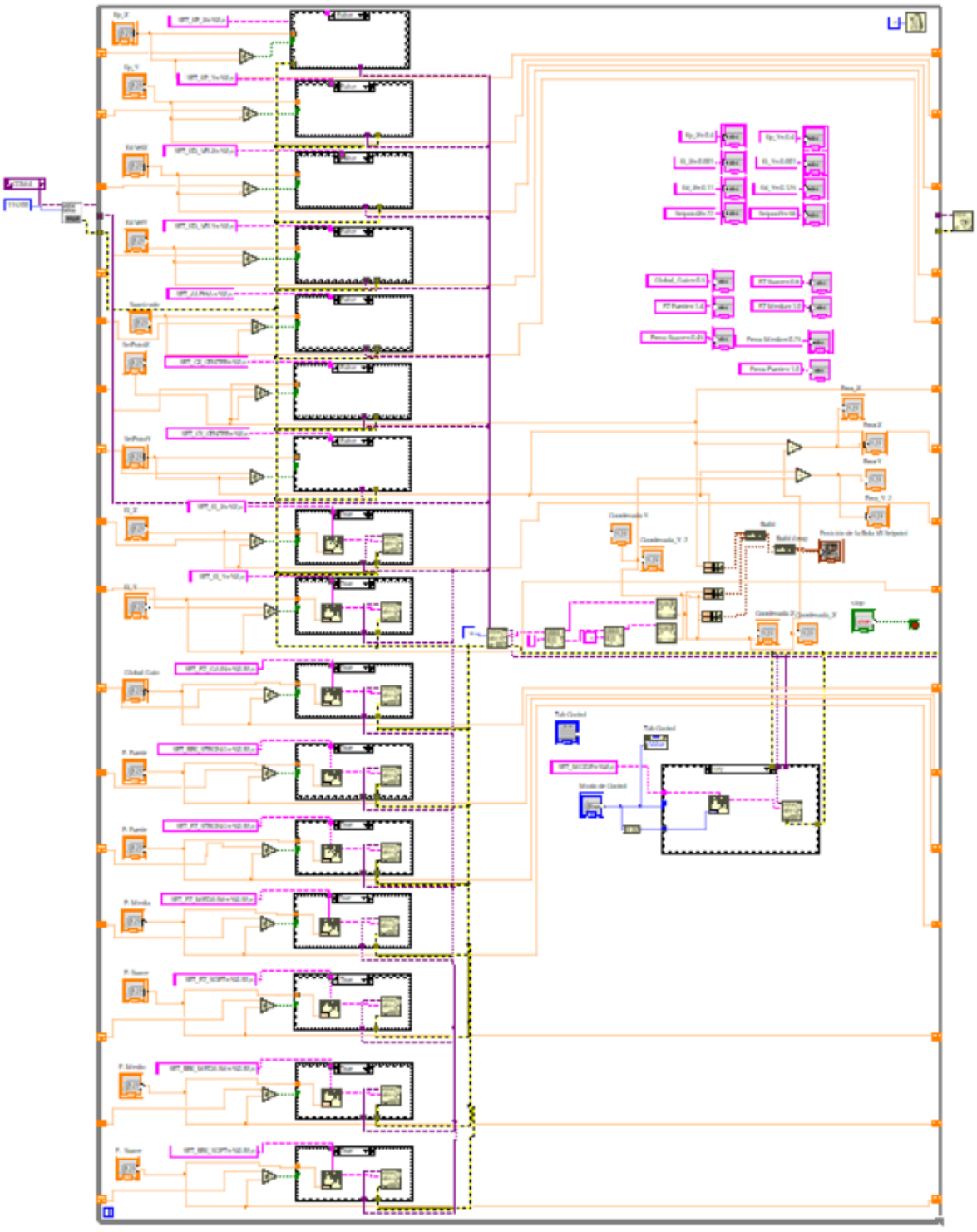


Figura 5.5: Código de LabVIEW

5.4.2. Efecto de la ganancia global

La ganancia global (GLOBAL-GAIN) tiene un impacto directo sobre la amplitud de las inclinaciones aplicadas por la plataforma. Valores bajos producen una respuesta

suave y visualmente estable, adecuada para demostraciones didácticas, aunque con menor capacidad para corregir perturbaciones importantes.

Al incrementar la ganancia global, la plataforma responde con mayor intensidad, logrando correcciones rápidas pero aumentando el riesgo de vibraciones o sobreimpulsos, especialmente en las cercanías del equilibrio. Durante las pruebas experimentales se determinó que valores comprendidos entre 0.45 y 0.65 ofrecían un compromiso adecuado entre estabilidad y capacidad de corrección.

Dado que este parámetro actúa como un factor común en la combinación de los términos posicional y de frenado, permite ajustar la "energía" total del controlador sin alterar la proporción relativa entre ambos efectos, por lo que un valor bajo de esta ganancia, genera una reducción en la respuesta del control difuso posicional y de frenado que es visible físicamente.

5.4.3. Ajuste de las reglas Fuerte, Media y Suave Posicionales

Las reglas difusas de tipo **Fuerte**, **Media** y **Suave** definen la intensidad de la corrección posicional aplicada por la plataforma. La regla Fuerte se activa principalmente cuando la bola se encuentra lejos del centro, mientras que la regla Media opera en regiones intermedias del espacio de estados, y la regla Suave tiene respuestas en la zona cercana al centro, donde el sistema tiende a reducir su respuesta en favor del equilibrio.

El ajuste se realizó asegurando que la regla Fuerte tuviera suficiente magnitud para recuperar la bola desde posiciones extremas sin generar saturación mecánica, mientras que la regla Media suavizara la transición entre regiones de corrección suave y agresiva y que la regla Suave tuviera una participación menor al realizar solo pequeñas correcciones.

Los valores finales seleccionados fueron:

- **Fuerte**: 1.4
- **Media**: 1.0
- **Suave**: 0.6

5.4.4. Ajuste de las reglas Fuerte, Media y Suave de Frenado

Las reglas perteneciantes a Frenado que también son Fuerte, Media y Suave, regulan la reacción del sistema ante la velocidad que la bola puede llegar a tomar al moverse sobre

la plataforma. Activándose cada una de ellas de acuerdo a la velocidad de la bola en cada momento, comparando los datos de entrada de coordenadas y analizando la variabilidad de ellas. La regla Fuerte actuará cuando la bola tenga una alta velocidad, tal como puede ser que se arroje la bola desde un borde de la plataforma con fuerza. Con valores altos de la regla fuerte, la plataforma es capaz de frenar mayores velocidades. La regla media actúa con velocidades que no llegan a ser muy grandes, como la pequeña oscilación del sistema para llevar la bola al setpoint. Mientras que la regla suave aplica el pequeño correctivo para velocidades de baja escala. Los valores seleccionados por defecto en el código son los siguientes:

- **Fuerte**: 1.05
- **Media**: 0.75
- **Suave**: 0.45

Esta configuración mostró un comportamiento que cumple el objetivo de llevar la bola al centro posicionándola directamente en uno de los bordes sin ejercer velocidad sobre ella. Pero por medio de escritura de valores en LabVIEW, es posible hacer que el sistema tenga respuestas más sólidas y rápidas fortaleciendo parámetros.

5.4.5. Resultados del proceso de ajuste

El proceso de ajuste permitió obtener un controlador difuso capaz de mantener la bola centrada durante largos períodos, con movimientos suaves y sin la vibración característica observada en algunas configuraciones del controlador PID.

Se destaca especialmente la separación explícita entre el término posicional y el término de frenado, lo que permitió ajustar el amortiguamiento dinámico sin afectar la autoridad direccional del controlador. En conjunto con la histéresis implementada en la zona central, el sistema logró un comportamiento robusto, estable y visualmente atractivo, cumpliendo plenamente los objetivos técnicos y didácticos del proyecto.

5.5. Comparación entre Control PID y Control Fuzzy

La plataforma Stewart desarrollada permite implementar y comparar de manera directa dos estrategias de control de naturaleza distinta: un controlador PID clásico y un

controlador basado en lógica difusa. Ambos fueron evaluados bajo condiciones experimentales equivalentes, utilizando la misma arquitectura de sensado, filtrado, cinemática inversa y accionamiento.

5.5.1. Comportamiento dinámico

El controlador PID presenta un comportamiento dinámico predecible y altamente dependiente de la sintonización de sus ganancias. Incrementos en la ganancia proporcional reducen el tiempo de respuesta, pero pueden inducir vibraciones si no se equilibran adecuadamente con el término derivativo.

El controlador difuso, en cambio, presenta una respuesta no lineal cuya intensidad depende de las reglas activadas y del grado de pertenencia a los conjuntos difusos. La incorporación de un bloque de frenado difuso permite amortiguar desplazamientos rápidos sin requerir un modelo dinámico explícito, logrando un comportamiento dinámico más tolerante a variaciones y perturbaciones.

5.5.2. Estabilidad en el centro

Con respecto al equilibrio se observaron diferencias significativas entre ambos enfoques:

- **PID**: Pequeñas variaciones en la detección de la posición o ruido de medición activan continuamente los términos proporcional y derivativo, generando microoscilaciones persistentes.
- **Control difuso**: La combinación de conjuntos difusos centrados y un mecanismo de histéresis permite que la plataforma permanezca completamente inmóvil cuando la bola se encuentra suficientemente cerca del centro. Adicionalmente, al anular la velocidad en esta región, el término de frenado se desactiva naturalmente.

Esta diferencia se traduce en una estabilidad visual y física notablemente superior para el controlador difuso.

5.5.3. Suavidad del control

El controlador PID puede generar movimientos bruscos ante pequeñas variaciones del error, especialmente bajo ganancias elevadas. Aunque esto favorece tiempos de respuesta

rápidos, compromete la suavidad del movimiento.

El controlador difuso genera acciones graduales gracias a la superposición de conjuntos de pertenencia y a la interpolación continua entre reglas. El frenado difuso contribuye a amortiguar movimientos rápidos sin introducir oscilaciones, produciendo trayectorias suaves y continuas, especialmente apreciables en las cercanías del equilibrio.

5.5.4. Adecuación para fines didácticos

Desde una perspectiva pedagógica, ambos controladores aportan valor, pero en dimensiones distintas:

- **PID**: Resulta ideal para ilustrar conceptos clásicos de control, como proporcionalidad, amortiguamiento y sensibilidad a la sintonización y funcionalidad. Demostrativamente sigue siendo útil para cumplir el objetivo del sistema de llevar la bola al setpoint, y académicamente como una herramienta de introducción al control puede funcionar excelente para alumnos de las carreras del Departamento de Electrónica e Informática, al entender las nociones de control por medio de un sistema seguro y estudiar la influencia de los parámetros sobre una variable de procesos.
- **Lógica difusa**: Permite la alternativa de introducir un paradigma basado en razonamiento lingüístico, facilitando la comprensión intuitiva del control en públicos no especializados en el área de control, tales como alumnos de centros de educación media, pasando de conceptos matemáticos más elaborados como la Proporcionalidad, Integral y la Derivada, a términos mucho más amigables como una reacción Fuerte, Media o Suave. Teóricamente, la separación entre corrección posicional y frenado por velocidad refuerza la comprensión del fenómeno físico y del rol de cada componente del controlador.

En el contexto del proyecto, ambos controladores han logrado el objetivo de lograr el control de posición de la bola y enviarla al setpoint. Permitiendo el visualizar dos caminos para llegar al mismo destino.

El uso de esta maqueta puede ser de gran ayuda, al llevar a pensar a los usuarios que aunque el sistema tenga por objetivo ser un material didáctico, imita el funcionamiento de sistemas industriales donde podrían desempeñarse en el futuro.

Para esta maqueta el objetivo es el llevar una bola al setpoint y que se mantenga ahí, mientras que en el futuro es posible que deban monitorear plantas de control de diversas variables de procesos, tales como Presión, Temperatura, Nivel, Flujo, Viscosidad, entre muchas otras. Donde también deberán preocuparse de que la variable de procesos se mantenga en los valores adecuados. De no ser así en el futuro eso podría provocar problemas, pero al experimentar con esta maqueta lo peor que puede pasar es que la bola caiga al suelo, dando la posibilidad de volverlo a intentar.

6. Resultados y Conclusiones

Este capítulo presenta los resultados obtenidos tras la implementación integral del sistema de control de la plataforma Stewart, así como las conclusiones principales derivadas del desarrollo de este proyecto. Se evalúa el desempeño del controlador PID, del controlador basado en lógica difusa, de la integración con la plataforma mecánica y del sistema de percepción visual. Finalmente, se discuten las limitaciones del sistema y se proponen líneas de trabajo futuro.

6.1. Resultados experimentales

Durante la etapa de validación se realizaron múltiples pruebas con ambos controladores implementados en el ESP32 Dev Kit: PID y lógica difusa. Las pruebas se centraron en cuatro aspectos fundamentales: estabilidad, capacidad de corrección ante perturbaciones, suavidad de movimiento y comportamiento en el centro de la plataforma.

Para esto fue vital el protocolo de comunicación del sistema, que en etapa preliminar, se usó SoftAP para el envío de datos de las coordenadas, y también para la escritura de datos desde LabVIEW, todo de forma inalámbrica, pero no se llegó a buenos resultados debido a lo crítico que es la frecuencia de muestreo del proyecto a la hora de obtener datos de posición y corregir. Generó pequeñas demoras, que son más amplias de lo que la plataforma puede soportar para funcionar correctamente.

Se aplicó posteriormente el uso de ESP-NOW para la comunicación de los microcontroladores, pero se logró un resultado similar que con Soft-AP, generando pequeñas demoras incorregibles, que para otro tipo de sistemas automatizados, o proyectos de IoT pueden funcionar perfectamente, pero para la implementación de una Plataforma Stewart se hizo imposible usar estos protocolos.

En este punto se dio paso a ocupar el protocolo UART el cual terminó siendo la herramienta definitiva de comunicación.

6.1.1. Desempeño del controlador PID

El controlador PID, una vez sintonizado empíricamente, logró mantener la bola en equilibrio y responder de manera adecuada a perturbaciones moderadas. Los resultados experimentales mostraron:

- **Estabilidad satisfactoria**: La plataforma pudo corregir la posición de la bola incluso desde regiones alejadas del centro.
- **Correcciones rápidas**: El término proporcional, junto con la acción derivativa, permitió reaccionar con agilidad ante movimientos repentinos.
- **Oscilaciones residuales**: A pesar de la correcta sintonización, se observaron pequeñas vibraciones en la zona central, propias de la naturaleza lineal del controlador y del ruido del sistema de visión que es parte de las limitaciones del proyecto.

En general, el PID presentó un comportamiento robusto y de fácil interpretación, siendo adecuado para explicar conceptos fundamentales de control.

6.1.2. Desempeño del controlador difuso

El controlador basado en lógica difusa presentó características distintivas respecto al PID. Las pruebas experimentales evidenciaron lo siguiente:

- **Suavidad en los movimientos**: Gracias a la interpolación continua de las reglas difusas, la plataforma produjo movimientos fluidos y progresivos.
- **Estabilidad sobresaliente en el centro**: La combinación de conjuntos centrados y la histéresis implementada permitió que la plataforma se mantuviera completamente inmóvil cuando la bola se encontraba cerca del equilibrio, eliminando prácticamente todas las microoscilaciones.
- **Corrección efectiva desde posiciones lejanas**: Las reglas fuertes (*strong*) permitieron aplicar acciones contundentes cuando la bola se alejaba significativamente del centro, garantizando su retorno.

El desempeño del controlador difuso fue de efectividad similar al PID, pero desde una forma distinta de entender el control y ejecutarlo mediante las reglas antes mencionadas en este proyecto.

6.1.3. Comportamiento del sistema de visión

El sistema de percepción basado en la ESP32-CAM permitió obtener mediciones en tiempo real de la posición de la bola. Su desempeño incluyó:

- **Frecuencia de lectura estable**: Aproximadamente 40–50 Hz, suficiente para un control responsivo.
- **Detección robusta**: La identificación cromática de la bola fue consistente en la detección del color rojo.
- **Iluminación**: La ESP32-CAM posee un led que puede ser activado por programa. Con este led activado se asegura una luminosidad de un nivel constante sobre la plataforma, asegurando su efectividad en lugares de luz reducida.

Si bien no alcanza la precisión de sistemas de visión industriales, su rendimiento resultó adecuado para los requerimientos del proyecto.

6.2. Conclusiones generales

El proyecto desarrollado permitió implementar con éxito una plataforma Stewart capaz de equilibrar una bola en tiempo real utilizando dos estrategias de control distintas: un controlador PID clásico y un controlador basado en lógica difusa. Las conclusiones más relevantes son las siguientes:

- **La plataforma Stewart demostró ser un excelente sistema de prueba para controladores avanzados**, su naturaleza multivariables, sensibilidad estructural y fuerte acoplamiento mecánico la convierten en un desafío ideal para explorar técnicas de control.
- **El controlador PID logró un desempeño sólido**, con capacidad de corrección rápida y estabilidad suficiente para mantener la bola equilibrada. Sin embargo, su tendencia a generar micromovimientos cercanos al equilibrio evidenció las limitaciones de los controladores lineales en sistemas con ruido y retardo. Esta limitación es perfectamente solucionable robusteciendo los dispositivos que forman parte del sistema, como la mejora en el sistema de adquisición de datos, o un microcontrolador que tenga capacidades mayores.

- **El controlador difuso es una herramienta útil e intuitiva**, la plataforma bajo control difuso fue capaz de mantener la bola inmóvil durante largos períodos, eliminando vibraciones y mejorando la interacción visual del sistema y la posibilidad de entender con palabras simples el funcionamiento del sistema, con reglas tales como Fuerte, Media y Suave.
- **La integración con LabVIEW enriqueció el carácter educativo del proyecto**, el monitoreo gráfico de posiciones, el ajuste dinámico de parámetros y la visualización del vector de error permitieron comprender y comparar de forma intuitiva el comportamiento de ambos controladores, además de la gran ayuda de escribir los parámetros en tiempo real sin la necesidad de cambiar valores en un código extenso y volver a cargar el programa en el ESP32 Dev Kit.
- **El sistema de visión demostró ser suficiente para el control en tiempo real**, a pesar de sus limitaciones inherentes en resolución y ruido.

En conjunto, estos resultados validan la implementación realizada y confirman que la plataforma desarrollada cumple plenamente con los objetivos planteados, tanto en términos funcionales como didácticos.

6.3. Limitaciones del sistema

A pesar de los resultados positivos, el sistema presenta ciertas limitaciones que deben considerarse:

- **Precisión limitada de los servomotores**: Los MG996R poseen un pequeño juego mecánico (*backlash*) que afecta la exactitud del movimiento, especialmente en pequeñas correcciones.
- **Sensibilidad a la iluminación**: El sistema de detección de color puede degradarse bajo condiciones lumínicas muy cambiantes, aunque para este proyecto la solución fue usar el flash integrado de la ESP32-CAM para obtener un alto grado de iluminación concentrado, garantizando una constancia en qué tan iluminada está la plataforma.

Estas limitaciones no impiden el funcionamiento correcto del sistema, pero constituyen oportunidades claras de mejora.

6.4. Trabajo futuro

El presente proyecto abre múltiples líneas de desarrollo que podrían explorarse en trabajos posteriores:

- **Implementación de un controlador difuso adaptativo**, capaz de modificar sus reglas en tiempo real según el comportamiento observado.
- **Integración de una cámara de mayor resolución y menor latencia**, lo cual permitiría mejorar la precisión y estabilidad del sistema.
- **Control predictivo basado en modelos (MPC)**, aprovechando la estructura multivariable de la plataforma Stewart.
- **Uso de algoritmos de aprendizaje de máquina** para ajustar automáticamente los parámetros del controlador.
- **Diseño de una interfaz web o aplicación móvil** para ampliar el carácter interactivo y didáctico del proyecto, aplicándolo al área de redes.
- **Añadir un control manual** puede ser una alternativa aún más interactiva. Existen diversas formas de control manual, tales como Joysticks o Sliders en programas como el mismo LabVIEW, o Sliders remotos retomando el punto anterior, a través de aplicaciones o páginas Web, comparando en control manual con respecto a los controles ya implementados en el sistema.

Estas propuestas permitirían ampliar la robustez, precisión y alcance educativo del sistema, transformando esta plataforma en una herramienta aún más completa para el estudio de técnicas modernas de control.

Bibliografía

[1] "Application of Fuzzy Algorithms for Control of Simple Dynamic Plant", Ebrahim H. Mamdani, Proceedings of the Institution of Electrical Engineers, Vol. 121, No. 12, 1974.

[2] "Fuzzy Logic with Engineering Applications", Timothy J. Ross, Third Edition, Wiley, 2010.

[3] "Fuzzy Control", Kevin M. Passino y Stephen Yurkovich, Addison-Wesley, 1998.

[4] "A Survey of Fuzzy Control: Fundamental Theory and Applications", Wen Yu y Humberto Steven Salazar, Applied Soft Computing, 2005.

[5] "A Survey of Industrial Applications of Fuzzy Control", Terano, T., Asai, K., Sugeno, M., Elsevier, 1992.

[6] "Estudio, diseño y construcción de una plataforma robótica didáctica tipo Stewart aplicada al estudio de Controladores Difusos", Natalia González Álvarez, Efren Reinoso Mendoza., Año 2011.

[7] "ESP32 Series Datasheet, Espressif Systems", Documento técnico oficial, 2023

[8] "ESP32-CAM Product Specification", Espressif Systems, Especificación técnica del módulo ESP32-CAM, 2022.