



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL
CARRERA DE MECATRÓNICA

**DESARROLLO DE UN CONTROLADOR PID PARA CUBO
AUTOEQUILIBRANTE EN UNA ARISTA**

Trabajo de titulación previo a la obtención del
Título de Ingeniero en Mecatrónica

AUTORES: Dario Manuel Carpio Holguin
Francisco Alejandro Fernández Morales
TUTOR: Franklin Illich Kuonquí Gaínza

Guayaquil - Ecuador
2024 - 2025

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, **Dario Manuel Carpio Holguin** con documento de identificación N° **0953445590** y **Francisco Alejandro Fernández Morales** con documento de identificación N° **0928434240**; manifestamos que:

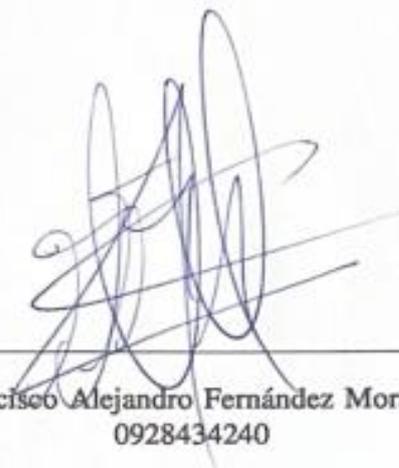
Somos los autores y responsables del presente trabajo; y, autorizamos a que, sin fines de lucro, la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo.

Guayaquil, 13 de Febrero del año 2025

Atentamente,



Dario Manuel Carpio Holguin
0953445590



Francisco Alejandro Fernández Morales
0928434240

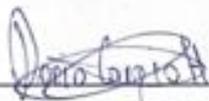
**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA
UNIVERSIDAD POLITÉCNICA SALESIANA**

Nosotros, **Dario Manuel Carpio Holguin** con documento de identificación N° **0953445590** y **Francisco Alejandro Fernandez Morales** con documento de identificación N° **0928434240**, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del **Dispositivo Tecnológico: DESARROLLO DE UN CONTROLADOR PID PARA CUBO AUTOEQUILIBRANTE EN UNA ARISTA**, el cual ha sido desarrollado para optar por el título de: Ingeniero en Mecatrónica, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 13 de Febrero del año 2025

Atentamente,



Dario Manuel Carpio Holguin
0953445590



Francisco Alejandro Fernández Morales
0928434240

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, **Franklin Illich Kuonquí Gaínza**, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **DESARROLLO DE UN CONTROLADOR PID PARA CUBO AUTOEQUILIBRANTE EN UNA ARISTA**, realizado por **Dario Manuel Carpio Holguin** con documento de identificación N° 0953445590 y por **Francisco Alejandro Fernandez Morales** con documento de identificación N° 0928434240, obteniendo como resultado final el trabajo de titulación bajo la opción **Dispositivo Tecnológico** que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 13 de Febrero del año 2025

Atentamente,



Franklin Illich Kuonquí Gaínza, Mg.
0909627432

DEDICATORIA

Jehová, por ser mi guía, darme fortaleza y permitir que cada decisión en mi camino sea la correcta.

A mis padres, Francisco Rafael Segundo Fernández Ruiz y Dolores Guadalupe Morales Guerrero, por su amor, consejos y motivación inquebrantable, que han sido el motor de mi vida y la base de mi perseverancia.

A esa persona especial, cuya guía y amor han moldeado mi carácter y me han enseñado a luchar, incluso en los momentos difíciles.

A mi amigo y compañero Darío Manuel Carpio Holguín, por su paciencia y esfuerzo, y por compartir conmigo este último tramo del camino.

Francisco Alejandro Fernández Morales

Primero a mis padres por su gran apoyo, Lorena y Guido, todo esto se los debo a ellos y a mis hermanos que aparte de molestar, también aportaron.

A mis compañeros que me ayudaron a pasar algunas materias explicándome cosas que no entendía.

A docentes muy importantes en este procesoso

Dario Manuel Carpio Holguin

AGRADECIMIENTO

En primer lugar, agradezco a Jehová, cuya guía y fortaleza han sido fundamentales en cada paso de este camino, permitiéndome avanzar con determinación y confianza.

A mis padres, Francisco Rafael Segundo Fernández Ruiz y Dolores Guadalupe Morales Guerrero, así como a esa persona especial que ha dejado una huella profunda en mi vida. Gracias por su amor incondicional, sus sabios consejos y el impulso constante que me ha llevado a superar cada obstáculo. Su apoyo ha sido mi mayor fuente de inspiración y fortaleza para seguir adelante.

Expreso mi más sincero agradecimiento a nuestro tutor de tesis, el Ingeniero Franklin Illich Kuonqui Gainza, por su invaluable guía y enseñanza. Su sabiduría y compromiso no solo nos han ayudado a llevar a cabo este proyecto con excelencia, sino que también nos han motivado a superar nuestros propios límites.

Asimismo, agradezco a todos los ingenieros, amigos y compañeros que, de una u otra forma, contribuyeron a que esta etapa de mi vida se cumpliera con éxito. Su apoyo y colaboración han sido fundamentales en este logro.

Francisco Alejandro Fernández Morales

Quiero agradecer a mi familia, la cual fue un apoyo muy valioso para mi carrera. Cada uno sabe cómo aportó en este proceso y quiero darles un muy grande reconocimiento.

A los compañeros que me guiaron y me ayudaron a terminar este largo camino el cual tuvo muchos obstáculos, pero gracias a la perseverancia fueron superados.

A todos esos docentes que se volvieron parte fundamental en este proceso, se esmeraron en compartir sus conocimientos para poder lograr este título.

Dario Manuel Carpio Holguin

RESUMEN

En el presente trabajo se desarrolla un sistema de control basado en técnicas de estabilización para un cubo autoequilibrante, el cual implementa un controlador PID discreto para mantener su equilibrio en una arista y ante perturbaciones externas. El sistema integra un motor brushless con un volante de inercia, sensores inerciales de alta precisión y un microcontrolador ESP32-WROOM-32 para la ejecución del control en tiempo real.

El modelo matemático del sistema se basa en la representación dinámica del péndulo invertido con volante de inercia, permitiendo establecer las ecuaciones diferenciales que describen su comportamiento. Mediante simulaciones en MATLAB y validaciones experimentales, se ajustan los parámetros del controlador PID para mejorar la respuesta del sistema y reducir errores en la estabilización. El diseño estructural se realizó mediante modelado CAD en SolidWorks, utilizando impresión 3D en PLA para la fabricación de las piezas, asegurando ligereza y robustez en la construcción del cubo.

La electrónica del sistema comprende la integración del ESP32-WROOM-32 con el sensor inercial MPU-6000, encargado de proporcionar datos de inclinación y velocidad angular en tiempo real. La señal de control generada por el PID es aplicada al motor mediante un controlador VESC, permitiendo una regulación precisa del torque y la velocidad del volante de inercia.

Los resultados experimentales demuestran la efectividad del controlador en la estabilización del cubo autoequilibrante, validando la precisión del modelo matemático y la implementación del sistema de control. La propuesta presentada es una plataforma versátil y escalable para el estudio de técnicas avanzadas de control, con aplicaciones potenciales en robótica, sistemas de estabilización y educación en ingeniería.

Palabras clave: Control PID, cubo autoequilibrante, ESP32-WROOM-32, motor brushless, volante de inercia, MPU-6000, VESC, estabilización dinámica, impresión 3D, simulación en MATLAB.

ABSTRACT

This work develops a control system based on stabilization techniques for a self-balancing cube, implementing a discrete PID controller to maintain its equilibrium on different surfaces and under external disturbances. The system integrates a brushless motor with a flywheel, high-precision inertial sensors, and an ESP32-WROOM-32 microcontroller for real-time control execution.

The mathematical model of the system is based on the dynamic representation of the inverted pendulum with a flywheel, establishing differential equations that describe its behavior. Through MATLAB simulations and experimental validations, the PID controller parameters are adjusted to optimize system response and minimize stabilization errors. The structural design was carried out using CAD modeling in SolidWorks, with 3D printing in PLA for manufacturing the components, ensuring a lightweight yet robust cube construction.

The system's electronics include the integration of the ESP32-WROOM-32 with the MPU-6000 inertial sensor, responsible for providing real-time inclination and angular velocity data. The control signal generated by the PID is applied to the motor through a VESC controller, enabling precise regulation of the torque and speed of the flywheel. Additionally, a wireless communication interface is implemented for real-time system calibration and monitoring via Bluetooth.

Experimental results demonstrate the effectiveness of the controller in stabilizing the self-balancing cube, validating the accuracy of the mathematical model and the control system implementation. The proposed solution serves as a versatile and scalable platform for studying advanced control techniques, with potential applications in robotics, stabilization systems, and engineering education.

Keywords: PID control, self-balancing cube, ESP32-WROOM-32, brushless motor, flywheel, MPU-6000, VESC, dynamic stabilization, 3D printing, MATLAB simulation.

ÍNDICE

I.	PROBLEMA	1
II.	INTRODUCCIÓN	2
III.	OBJETIVOS	3
III-A.	Objetivo general	3
III-B.	Objetivos específicos	3
IV.	FUNDAMENTOS TEÓRICOS	4
IV-A.	Modelo Matemático del Sistema de Control de Balance	4
IV-A1.	Modelo Dinámico del Sistema con Motor Brushless	4
IV-A2.	Ecuaciones Dinámicas del Sistema	4
IV-A3.	Controlador PID	5
IV-B.	Materiales y diseño mecánico	6
IV-B1.	Giroscopio	6
IV-B2.	Motores Brushless (BLDC)	7
IV-B3.	Microcontrolador ESP32	9
IV-C.	Controlador PID Discreto	10
IV-D.	Integración Numérica en el Control PID	12
IV-D1.	Formulación del Término Integral	12
IV-D2.	Propósito del Término Integral	12
IV-D3.	Consideraciones Prácticas	12
IV-E.	Softwares Utilizados	13
V.	MARCO METODOLÓGICO	15
V-A.	Selección de Componentes	15
V-B.	Microcontrolador ESP32	15
V-C.	Motor	15
V-D.	Giroscopio	17
V-E.	Diseño de Estructura	17
V-F.	Diseño de PCB y Electrónica	23
V-F1.	Consideraciones del Microcontrolador	24
V-F2.	Consideraciones del Sensor	24
V-F3.	Consideraciones de la Batería	24
V-F4.	Consideraciones de los Motores	24
V-G.	Modelo matemático	25
V-G1.	Cálculo de las Ganancias del Controlador PID	25
V-G2.	Modelo Dinámico del Sistema	25
V-G3.	Discretización del Sistema	26
V-G4.	Diseño del Controlador PID	26
V-G5.	Simulación de la Respuesta del Sistema	26
V-H.	Sistema de Volante de Inercia	27
V-I.	Energía y Momento Angular en un Volante de Inercia	27
V-J.	Dinámica del Volante de Inercia con Motor Brushless	28
V-K.	Diseño del Controlador	30
V-L.	Programación	33
V-L1.	Declaraciones y configuraciones en ESP32.h	33
V-L2.	Archivo principal ESP32_cube.ino	33
V-L3.	Inicialización de periféricos	35

V-L4.	Ciclo principal	35
V-L5.	Funciones auxiliares en functions.ino	35
V-M.	Comunicación y ajuste en tiempo real	35
VI.	RESULTADOS	37
VI-A.	Diseño e Implementación de la Tarjeta PCB para el Control del Sistema	37
VI-B.	Implementación del Control de Lazo Cerrado en el Microcontrolador	39
VI-C.	Validación del Funcionamiento y Evaluación del Tiempo de Respuesta	39
VII.	CRONOGRAMA	41
VIII.	PRESUPUESTO	42
IX.	CONCLUSIONES	43
X.	RECOMENDACIONES	44
	Anexo A: Prototipo Inicial	47
	Anexo B: Prototipo Final	48

ÍNDICE DE FIGURAS

1.	Cubo Autoequilibrante [5].	1
2.	Sistema de circuito cerrado [10].	6
3.	Sensor MPU-6000 [11].	7
4.	Motor Brushless Nidec 24H [16].	8
5.	Microcontrolador ESP32 [23].	10
6.	Sistema de control PID [10].	11
7.	Plano acotado de motor [39].	16
8.	Base del controlador [Fuente: Autores].	18
9.	Pared del Motor [Fuente: Autores].	18
10.	Pared del Motor [Fuente: Autores].	19
11.	Pared del Motor [Fuente: Autores].	19
12.	Pared secundaria [Fuente: Autores].	19
13.	Pared del Motor [Fuente: Autores].	20
14.	Superficie de apoyo del controlador [Fuente: Autores].	20
15.	Vértice para unión de paredes [Fuente: Autores].	21
16.	Disco de motor [Fuente: Autores].	21
17.	Disco de motor [Fuente: Autores].	22
18.	Soporte de motor [Fuente: Autores].	22
19.	Base del controlador [Fuente: Autores].	22
20.	Diagrama de conexión electrónica [Fuente: Autores].	24
21.	Placa electrónica PCB [Fuente: Autores].	25
22.	Simulación en MATLAB con función de transferencia discreta y ajuste de ganancias Parte 1. [Fuente: Autores]	31
23.	Simulación en MATLAB con función de transferencia discreta y ajuste de ganancias Parte 2. [Fuente: Autores]	32
24.	Diagrama electrónico de conexiones. [Fuente: Autores]	32
25.	Declaraciones y configuración de pines en ESP32 Parte 1.[Fuente: Autores]	34
26.	Declaraciones y configuración de pines en ESP32 Parte 2.[Fuente: Autores]	35
27.	ESP32 Cube: Configuración inicial y motores. [Fuente: Autores]	36
28.	ESP32 Cube: Final del bucle principal y control de batería.[Fuente: Autores]	37
29.	Placa electrónica PCB superior [Fuente: Autores].	37
30.	Placa electrónica PCB inferior [Fuente: Autores].	38
31.	Placa electrónica PCB [Fuente: Autores].	38
32.	Microcontrolador ESP32 [23].	40
33.	Prototipo inicial para próximas mejoras [Fuentes: Autores]	47
34.	Prototipo final operativo Frontal [Fuentes: Autores]	48
35.	Prototipo final operativo lateral[Fuentes: Autores]	49
36.	Prototipo final operativo parado[Fuentes: Autores]	50
37.	Disco de inercia [Fuentes: Autores]	51
38.	Pared de prototipo [Fuentes: Autores]	52
39.	Pared con soporte de motor. [Fuentes: Autores]	53

ÍNDICE DE TABLAS

I.	Cronograma	41
II.	Presupuesto	42

I. PROBLEMA

En las últimas décadas, a pesar de los avances tecnológicos y el desarrollo de nuevas herramientas y metodologías, persisten importantes limitaciones en la enseñanza de la ingeniería y la automatización. Los sistemas de control y simulación, aunque prometen ser soluciones efectivas, a menudo no logran mejorar de manera significativa la enseñanza y el aprendizaje de habilidades técnicas y prácticas en instituciones educativas que carecen de infraestructura adecuada. [1]. En este contexto, la ausencia de dispositivos accesibles y prácticos, como el cubo autoequilibrante, restringe la capacidad de los estudiantes para comprender conceptos avanzados de sistemas retroalimentados de manera aplicada, debilitando su formación en el diseño e implementación de sistemas de control.

Actualmente, los proyectos educativos relacionados con herramientas prácticas como el cubo autoequilibrante enfrentan múltiples desafíos en su desarrollo e implementación. Las características esenciales, como la selección de componentes adecuados (microcontroladores y sensores giroscópicos), suelen quedar subestimadas o desatendidas, lo que compromete la estabilidad y funcionalidad del sistema. Además, los procesos de diseño estructural y la integración de interfaces de usuario carecen de un enfoque sistemático, lo que dificulta la retroalimentación efectiva entre el operador y el sistema. Estas carencias no solo limitan la precisión del control, sino que también generan un riesgo de que los parámetros de estabilidad no sean ajustados de manera eficiente en condiciones variadas, exponiendo el sistema a fallos y limitando su confiabilidad [2].

Por otro lado, las interfaces de usuario en dispositivos educativos, lejos de ser intuitivas y accesibles, suelen ser complejas y poco prácticas. Esto representa una barrera significativa para los estudiantes, quienes se ven obligados a enfrentar complicaciones técnicas que distraen del aprendizaje de los principios de control y retroalimentación. La falta de simplicidad en las herramientas educativas agrava las dificultades de uso, especialmente para principiantes, lo que desmotiva y limita el desarrollo de competencias prácticas en los alumnos [3].

La limitada repetibilidad y robustez de muchos sistemas diseñados para la educación restringe su utilidad en entornos académicos. Los estudiantes, a menudo, no pueden realizar múltiples pruebas y ajustes sin temor a dañar el equipo o comprometer su funcionalidad. Esto impide que adquieran experiencia práctica y dificulta su comprensión profunda de los conceptos de control. La falta de seguridad en la manipulación de dispositivos experimentales merma la experimentación y, por ende, frena el proceso de aprendizaje y la formación integral de los futuros profesionales en el campo de la ingeniería y la automatización [4].

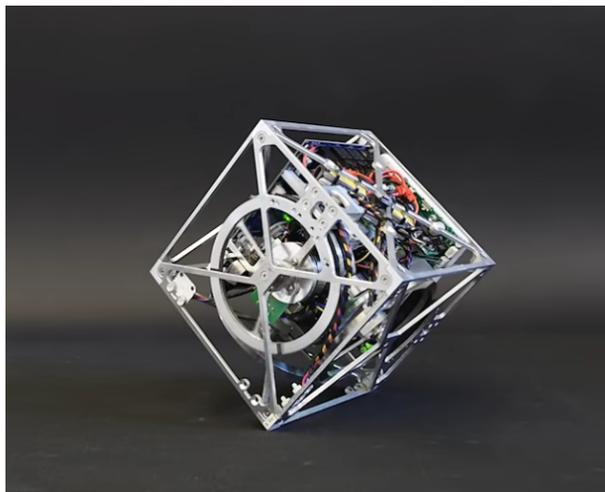


Figura 1: Cubo Autoequilibrante [5].

II. INTRODUCCIÓN

El estudio y la enseñanza de los sistemas de control y estabilidad en Ingeniería en Mecatrónica requieren herramientas que permitan una comprensión práctica y aplicada de conceptos teóricos complejos. En este contexto, el desarrollo de un cubo autoequilibrante se presenta como una solución innovadora para reforzar la formación en estas áreas. Este dispositivo permite a los estudiantes interactuar con sistemas de retroalimentación y control en tiempo real, facilitando la experimentación con principios fundamentales como el equilibrio dinámico y la estabilidad en sistemas mecánicos. Al integrar tecnologías avanzadas como el microcontrolador ESP32 y sensores giroscópicos, el cubo no solo mejora la experiencia de aprendizaje, sino que también fortalece habilidades esenciales en el diseño e implementación de sistemas de control modernos.

Una de las ventajas clave de este sistema es su interfaz intuitiva y accesible, diseñada para eliminar barreras técnicas y facilitar su uso tanto para principiantes como para usuarios avanzados. A diferencia de otras plataformas tecnológicas que pueden requerir un conocimiento previo extenso, el cubo autoequilibrante ofrece una interacción sencilla que permite monitorear y ajustar parámetros de estabilidad en tiempo real. Esta característica fomenta un entorno de aprendizaje dinámico en el que los estudiantes pueden centrarse en la comprensión y aplicación de los principios de control sin distracciones innecesarias.

Desde una perspectiva práctica, la incorporación del cubo autoequilibrante en el laboratorio de Ingeniería en Mecatrónica representa una mejora significativa en la accesibilidad a sistemas de control avanzados. Su diseño de bajo costo permite que un mayor número de estudiantes tenga acceso a experimentación práctica sin una inversión excesiva en infraestructura. Además, su robustez y capacidad de repetición lo convierten en una herramienta sostenible que puede soportar múltiples ajustes y pruebas sin comprometer su funcionalidad, garantizando una formación más efectiva y duradera.

En términos educativos, la implementación de este sistema representa un avance estratégico en la enseñanza de control y estabilidad, mejorando los recursos del laboratorio y proporcionando una herramienta accesible y de alto valor académico. Su introducción en el currículo de la universidad contribuirá al desarrollo de habilidades esenciales en los futuros ingenieros, preparándolos para enfrentar los desafíos del sector industrial con una comprensión sólida de los sistemas de control. Este proyecto, por tanto, no solo refuerza la calidad de la formación, sino que también posiciona a la universidad en la vanguardia de la enseñanza práctica en mecatrónica.

III. OBJETIVOS

III-A. Objetivo general

Desarrollar un controlador PID aplicado a un cubo autoequilibrante utilizando un microcontrolador y un sensor giroscópico para su equilibrio en una arista.

III-B. Objetivos específicos

1. Diseñar una tarjeta PCB que permita el control y estabilidad de su sistema de autoequilibrio en una arista.
2. Implementar un control de lazo cerrado en el microcontrolador procesando la información obtenida del giroscopio.
3. Validar el correcto funcionamiento del sistema, evaluando la estabilidad en relación al tiempo de respuesta y desempeño.

IV. FUNDAMENTOS TEÓRICOS

IV-A. Modelo Matemático del Sistema de Control de Balance

El modelado matemático de sistemas dinámicos es una herramienta clave en la ingeniería de control, ya que permite describir el comportamiento de sistemas físicos mediante ecuaciones que relacionan entradas, estados y salidas. Estas representaciones matemáticas sirven como base para analizar y diseñar estrategias de control que aseguren la estabilidad y el desempeño deseado del sistema. En sistemas continuos, las ecuaciones diferenciales son ampliamente utilizadas para modelar fenómenos como la dinámica de masas, resortes y amortiguadores, mientras que en sistemas discretos se emplean ecuaciones en diferencias y transformaciones para simplificar el análisis en el dominio del tiempo [6].

Un enfoque común en el modelado de sistemas dinámicos es la representación en el dominio de la frecuencia mediante la transformada de Laplace, la cual simplifica las ecuaciones diferenciales al convertirlas en expresiones algebraicas. Sin embargo, en sistemas no lineales, esta técnica puede resultar insuficiente, por lo que es necesario recurrir a aproximaciones linealizadas alrededor de puntos de operación. Adicionalmente, el uso de ecuaciones de estado permite representar sistemas multivariantes de manera más general, facilitando su análisis y control [7].

El modelado matemático también tiene aplicaciones importantes en sistemas eléctricos y térmicos. Por ejemplo, los circuitos eléctricos con inductores y capacitores pueden representarse mediante analogías mecánicas, permitiendo un análisis más intuitivo de su comportamiento dinámico. Este enfoque es esencial en el diseño de convertidores de potencia y sistemas de almacenamiento de energía, donde se requiere un control preciso de las interacciones entre variables [8].

En el caso de sistemas dinámicos complejos, como robots industriales o vehículos autónomos, los modelos en espacio de estados permiten describir un sistema de manera compacta y eficiente. Estos modelos utilizan matrices para relacionar entradas, salidas y estados internos, facilitando el análisis de estabilidad y la implementación de controladores avanzados como el LQR (Control Óptimo Cuadrático Lineal) y el MPC (Control Predictivo Basado en Modelos) [9].

El sistema de control de balance se modela como un péndulo invertido en dos dimensiones, donde el objetivo es mantener el equilibrio mediante el ajuste de las señales de control enviadas a los actuadores (motores). A continuación, se describe el modelo matemático del sistema y su implementación en MATLAB.[2].

IV-A1. Modelo Dinámico del Sistema con Motor Brushless: El sistema de balanceo se modela como un volante de inercia en dos dimensiones, donde el control del equilibrio se logra aplicando un torque generado por un motor brushless (BLDC) controlado mediante una señal PWM (modulación por ancho de pulso). A continuación, se presenta el modelo matemático del sistema y su implementación en el controlador.

IV-A2. Ecuaciones Dinámicas del Sistema: El comportamiento dinámico del volante de inercia está gobernado por las ecuaciones 1 y 2:

$$\frac{d\theta}{dt} = \omega \quad (1)$$

$$\frac{d\omega}{dt} = \frac{\tau - mgL \sin(\theta)}{J} \quad (2)$$

Donde:

- θ : Ángulo de inclinación del sistema (rad).
- ω : Velocidad angular del volante de inercia (rad/s).
- m : Masa del sistema ($m = 1,0$ kg).
- g : Aceleración gravitatoria ($g = 9,81$ m/s²).

- L : Longitud equivalente del centro de masa respecto al eje de rotación ($L = 0,1$ m).
- J : Momento de inercia del volante ($J = m \cdot L^2 = 0,01$ kg · m²).
- τ : Torque aplicado por el motor brushless (N · m).

Generación del Torque mediante PWM en Motor Brushless: El torque τ es generado por un motor brushless (BLDC), el cual requiere una modulación de la señal PWM para regular su velocidad y torque mediante un controlador electrónico (ESC). A diferencia de los motores de corriente continua con escobillas, en los BLDC la corriente no se aplica directamente a un único devanado, sino que se distribuye en un sistema trifásico mediante el ESC, que gestiona la conmutación electrónica.

La relación entre el torque y la corriente del motor se expresa en la ecuación 3:

$$\tau = K_m \cdot I_{motor} \quad (3)$$

Donde:

- K_m : Constante de torque del motor (N · m/A).
- I_{motor} : Corriente aplicada al motor en función de la señal PWM.

En los motores BLDC, el ESC ajusta la corriente suministrada a las bobinas del motor según el ciclo de trabajo de la señal PWM. Para garantizar un control preciso, es necesario conocer el valor máximo de corriente que puede circular por las fases del motor, denotado como I_{max} . Este valor representa la corriente nominal que el motor puede manejar de manera segura sin riesgo de sobrecalentamiento ni saturación magnética de los devanados.

La relación entre la corriente de fase y la señal PWM está determinada por la ecuación 4:

$$I_{motor} = \frac{PWM}{255} I_{max} \quad (4)$$

Donde:

- PWM : Señal de control con valores en el rango de 0 a 255.
- I_{max} : Corriente máxima de operación del motor, determinada por su diseño eléctrico y térmico.

El valor de I_{max} depende de varios factores, como la resistencia de los devanados del motor, la inductancia de fase y la capacidad del ESC para suministrar corriente continua y de pico. En aplicaciones de alta precisión, el monitoreo de este parámetro es esencial para evitar pérdidas de eficiencia y garantizar una operación estable del sistema.

En los motores brushless, la señal PWM controla el ancho de pulso enviado al ESC, que a su vez ajusta la corriente enviada a las bobinas del motor. La relación entre la corriente y la señal PWM está dada por la ecuación 4.

Donde:

- PWM : Señal de control con valores en el rango de 0 a 255.
- I_{max} : Corriente máxima de operación del motor.

IV-A3. Controlador PID: Para mantener el equilibrio del volante de inercia, se implementa un controlador PID, el cual genera la señal de control PWM en función del error angular gobernado por la ecuación 5:

$$\tau = K_p \theta + K_d \omega + K_i \int \theta dt \quad (5)$$

Donde:

- K_p : Ganancia proporcional, que responde a la desviación angular.
- K_d : Ganancia derivativa, que responde a la velocidad angular.
- K_i : Ganancia integral, que corrige errores acumulados en el tiempo.

La salida del PID, que representa la señal PWM deseada, se convierte en una corriente aplicada al motor, generando un torque como se plantea en la ecuación 6:

$$PWM = \frac{\tau}{K_m I_{\max}} \times 255 \quad (6)$$

Este esquema permite controlar de manera precisa el torque aplicado al volante de inercia, manteniendo la estabilidad del sistema y corrigiendo perturbaciones en tiempo real, como se detalla en la figura 2.

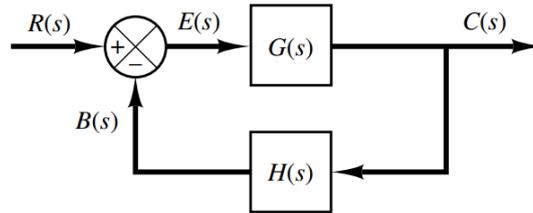


Figura 2: Sistema de circuito cerrado [10].

IV-B. Materiales y diseño mecánico

En los sistemas de control y estabilización, los sensores inerciales juegan un papel fundamental en la medición de variables como la velocidad angular y la aceleración lineal. Dentro de esta categoría, los giroscopios y acelerómetros permiten obtener datos críticos sobre la orientación y el movimiento de un objeto en el espacio, lo que es esencial en aplicaciones de navegación, estabilización y control de equilibrio [11].

IV-B1. Giroscopio: Un giroscopio es un sensor que mide la velocidad angular de un objeto en uno o más ejes, lo que permite determinar cambios en la orientación con alta precisión. Estos dispositivos funcionan con principios de conservación del momento angular y utilizan tecnologías basadas en sistemas microelectromecánicos (MEMS) para detectar y cuantificar el movimiento [11].

El funcionamiento del giroscopio se basa en la medición de la velocidad angular en los ejes X , Y y Z , lo que permite calcular la inclinación y la orientación del sistema mediante técnicas de integración numérica. Además, su uso aumenta la estabilidad en tiempo real, ya que proporciona datos esenciales para el ajuste del sistema de control. En aplicaciones de autoequilibrio, la precisión del giroscopio es crítica, ya que cualquier error en la medición de la velocidad angular puede comprometer la respuesta del controlador y, por ende, la estabilidad del dispositivo [11].

Los giroscopios MEMS han sido ampliamente utilizados en la industria debido a su bajo costo, reducido tamaño y alta eficiencia energética. Estos sensores son clave en aplicaciones como drones, sistemas de navegación inercial, dispositivos de autoequilibrio y estabilización de cámaras, ya que permiten compensar perturbaciones externas y corregir la inclinación en tiempo real [11].

Uno de los giroscopios más utilizados en la industria de estabilización es el MPU-6000, un sensor inercial desarrollado por TDK InvenSense. Este dispositivo integra un giroscopio y un acelerómetro triaxial en un encapsulado compacto de 4 mm, permitiendo medir simultáneamente la velocidad angular y la aceleración en los ejes X , Y y Z [11].

Su diseño mejora facilita la integración en sistemas de control de estabilidad, proporcionando datos de alta precisión en tiempo real. Gracias a su reducido tamaño y bajo consumo energético, el MPU-6000 se convierte en una opción eficiente para aplicaciones de autoequilibrio y estabilización dinámica.

Características Técnicas El MPU-6000 posee las siguientes especificaciones:

- Rango de medición del giroscopio: ± 250 , ± 500 , ± 1000 y ± 2000 °/s.
- Rango de medición del acelerómetro: $\pm 2g$, $\pm 4g$, $\pm 8g$ y $\pm 16g$.
- Procesador de Movimiento Digital (DMP) integrado para procesamiento en tiempo real.
- Interfaz de comunicación: Compatible con I²C y PI, facilitando su integración con microcontroladores.
- Consumo de energía: Re para aplicaciones de bajo consumo con modos de suspensión y espera.
- Auto prueba integrada para verificar su correcto funcionamiento.

Aplicaciones: Gracias a sus características, el MPU-6000 es ampliamente utilizado en:

- Dispositivos de autoequilibrio.
- Robots bípedos.
- Sistemas de estabilización de drones.
- Sistemas de navegación inercial.

Este sensor proporciona datos de movimiento con alta precisión y baja latencia, lo que lo hace ideal para aplicaciones donde la estabilidad y el control preciso son fundamentales [12].

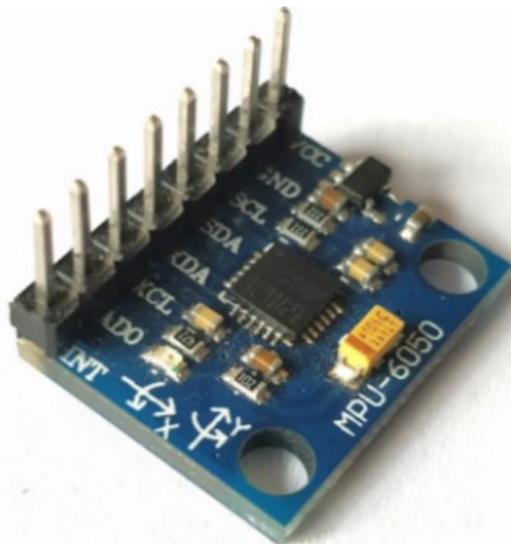


Figura 3: Sensor MPU-6000 [11].

IV-B2. Motores Brushless (BLDC): Los motores brushless, también conocidos como motores de corriente continua sin escobillas (BLDC), son dispositivos electromecánicos que convierten la energía eléctrica en energía mecánica mediante la conmutación electrónica de corriente en bobinas fijas, eliminando la necesidad de escobillas y conmutadores mecánicos [13]. Estos motores son ampliamente utilizados en aplicaciones donde se requiere alta eficiencia, bajo mantenimiento y un control preciso de velocidad y torque, como en robótica, automatización industrial y vehículos eléctricos [14].

El funcionamiento de los motores BLDC se basa en la conmutación electrónica de las fases del motor a través de un controlador, generalmente un ESC (Electronic Speed Controller), el cual genera la secuencia de conmutación

necesaria para inducir un campo magnético rotatorio que impulsa el rotor. La ausencia de escobillas reduce la fricción y el desgaste mecánico, aumentando la eficiencia y prolongando la vida útil del motor [15].

El comportamiento dinámico de un motor BLDC se modela a través de ecuaciones eléctricas y mecánicas que describen la relación entre la tensión aplicada, la corriente, la velocidad angular y el torque generado. La ecuación de la ley de Kirchhoff para el circuito del estator del motor se expresa en la ecuación 7:

$$V = R_s I + L_s \frac{dI}{dt} + e_b \quad (7)$$

donde:

- V es el voltaje aplicado al motor.
- R_s es la resistencia del devanado del estator.
- L_s es la inductancia del devanado del estator.
- I es la corriente suministrada a las fases del motor.
- e_b es la fuerza contraelectromotriz generada por el motor.

La fuerza contraelectromotriz e_b en los motores BLDC se define como la ecuación 8:

$$e_b = K_e \omega \quad (8)$$

donde:

- K_e es la constante de velocidad del motor.
- ω es la velocidad angular del rotor.

Esta ecuación indica que a medida que la velocidad del motor aumenta, la fuerza contraelectromotriz también crece, lo que reduce la corriente suministrada al motor y, por lo tanto, limita el torque generado. Esto es crucial para el control de velocidad en aplicaciones de precisión [14].

Los motores BLDC han ganado popularidad en sistemas de control avanzados debido a su eficiencia, durabilidad y capacidad de respuesta rápida. A diferencia de los motores DC con escobillas, los BLDC ofrecen un mayor rendimiento energético y una mejor disipación térmica, lo que los hace ideales para aplicaciones que requieren operación continua y altas velocidades [13]. Su diseño sin escobillas permite una mayor vida útil y una menor necesidad de mantenimiento, lo que los convierte en una solución óptima en robótica, drones, vehículos eléctricos y automatización industrial como se muestra en la figura 4.

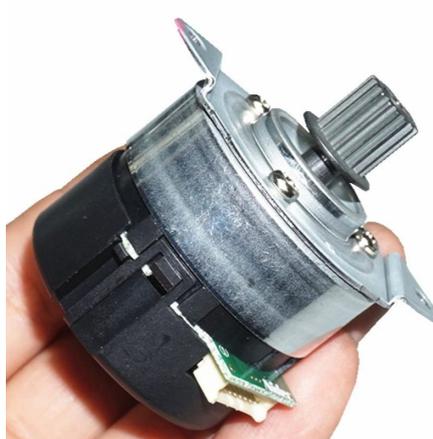


Figura 4: Motor Brushless Nidec 24H [16].

IV-B3. Microcontrolador ESP32: Su diseño reducido permite ejecutar múltiples tareas de procesamiento en tiempo real, lo que lo hace ideal para aplicaciones de Internet de las Cosas (IoT), automatización y control industrial. Su arquitectura combina un potente procesador con capacidades avanzadas de conectividad y bajo consumo energético, facilitando su integración en entornos embebidos [17].

El funcionamiento del ESP32-WROOM-32 se basa en una arquitectura eficiente que permite la ejecución de múltiples tareas simultáneamente. Sus principales componentes incluyen:

- Procesador Dual-Core Xtensa LX6: Opera a una frecuencia de hasta 240 MHz, permitiendo la ejecución de algoritmos en tiempo real.
- Memoria integrada: Cuenta con 520 KB de SRAM y 4 MB de memoria flash, asegurando suficiente almacenamiento para programas complejos [18].
- Manejo de interrupciones: Dispone de un sistema avanzado de interrupciones para mejorar la ejecución de tareas prioritarias sin afectar el rendimiento del sistema.
- Módulo de reloj y temporización: Incorpora temporizadores internos que permiten la programación precisa de eventos y la sincronización con periféricos.

Las características del ESP32-WROOM-32 se distinguen por varias cualidades que lo hacen ideal para sistemas embebidos de alto rendimiento:

- Bajo consumo energético: Soporta múltiples modos de ahorro de energía, como el modo deep sleep, reduciendo el consumo a menos de 10 μ A, lo cual es ideal para dispositivos alimentados por batería [19].
- Alta capacidad de procesamiento: Su arquitectura permite manejar tareas concurrentes sin degradación del rendimiento, mejorando la ejecución de algoritmos de control.
- Soporte para comunicación inalámbrica: Incorpora conectividad Wi-Fi y Bluetooth, facilitando la integración con sistemas de monitoreo remoto [20].
- Interfaces de comunicación versátiles: Cuenta con buses de comunicación como SPI, I²C, UART, ADC, DAC, PWM y CAN, permitiendo la conexión con una amplia variedad de sensores y actuadores.

Para la implementación en el proyecto, es importante considerar los siguientes parámetros técnicos del ESP32-WROOM-32:

- Voltaje de operación: 3.3V (requiere un regulador si se usa con fuentes de 5V o superiores).
- Corriente máxima de operación: Hasta 500 mA en plena carga (puede requerir una fuente externa dependiendo de los periféricos conectados).
- Tiempo de arranque: Aproximadamente 200 ms desde el encendido hasta la inicialización completa.
- Frecuencia del bus SPI: Hasta 80 MHz, lo que permite transferencias rápidas de datos con sensores y pantallas [21].

El ESP32-WROOM-32 es particularmente útil en este proyecto debido a su capacidad para manejar múltiples sensores y actuadores con eficiencia. Gracias a su conectividad inalámbrica, permite la implementación de un sistema de monitoreo remoto en tiempo real, mientras que su arquitectura de procesamiento en paralelo facilita la ejecución de algoritmos de control con baja latencia. Estas características lo convierten en una solución óptima para sistemas embebidos con requerimientos de estabilidad y bajo consumo energético [22].

El módulo ESP32-WROOM-32 se ha diseñado para facilitar la integración en entornos industriales y de consumo, ofreciendo una alta capacidad de procesamiento con conectividad inalámbrica integrada.

Uno de los aspectos clave del ESP32-WROOM-32 es su capacidad de comunicación, ya que soporta múltiples protocolos que permiten la transmisión de datos eficiente en sistemas de control y monitoreo. Entre las tecnologías más relevantes se encuentran:

- Wi-Fi: El ESP32-WROOM-32 permite conexiones de hasta 150 Mbps en la banda de 2.4 GHz, lo que lo hace ideal para aplicaciones IoT y sistemas de automatización que requieren transferencia de datos a alta velocidad [21].
- Bluetooth 4.2 LE: El módulo incorpora Bluetooth Low Energy (BLE), lo que permite la comunicación de corto alcance con bajo consumo de energía, facilitando la integración con sensores y dispositivos móviles.
- Interfaces de comunicación por hardware: Además de la conectividad inalámbrica, el ESP32-WROOM-32 como se detalla en la figura 5, cuenta con múltiples interfaces seriales que permiten la integración con sensores y periféricos [19], entre ellas:
 - SPI (Serial Peripheral Interface): Ideal para la comunicación de alta velocidad con memorias y pantallas.
 - I²C (Inter-Integrated Circuit): Protocolo de comunicación en serie ampliamente usado en sensores y dispositivos de bajo consumo.
 - UART (Universal Asynchronous Receiver-Transmitter): Interfaz de comunicación en serie utilizada para depuración y control de periféricos.

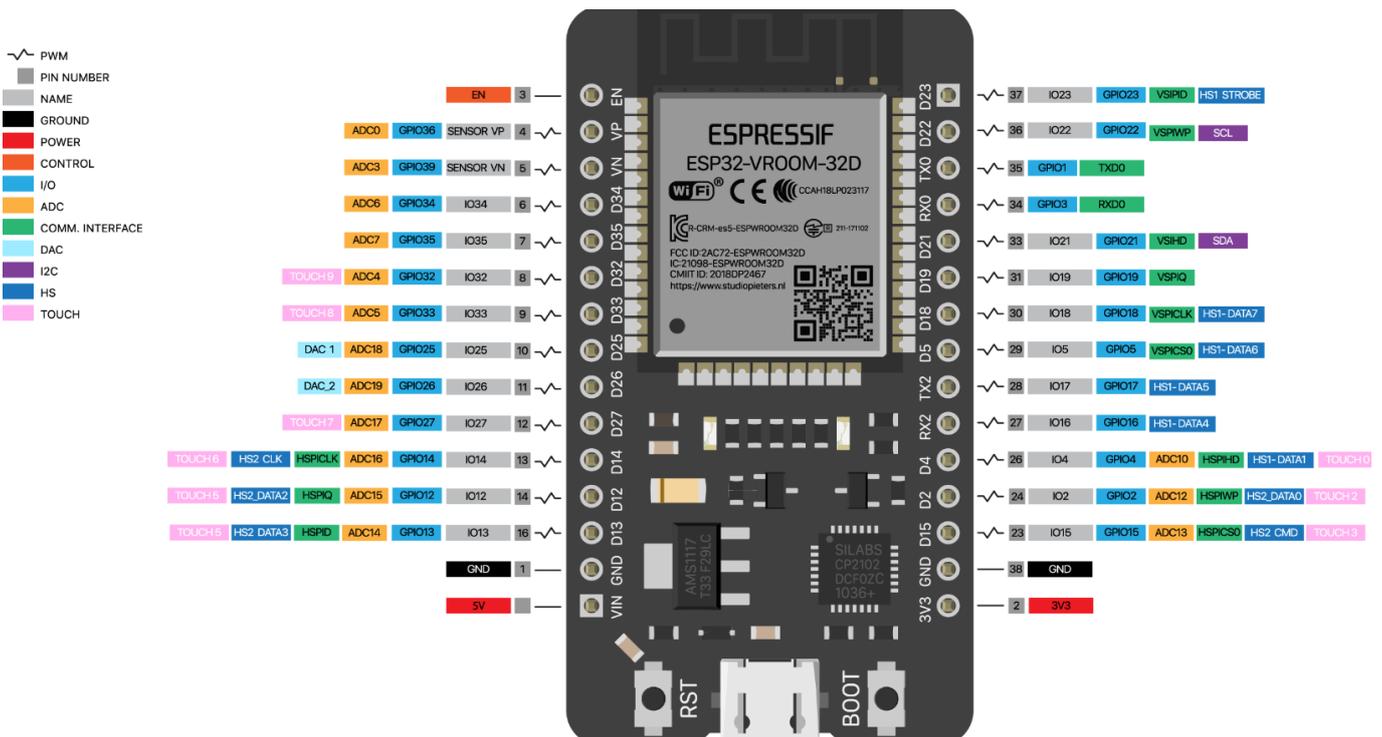


Figura 5: Microcontrolador ESP32 [23].

IV-C. Controlador PID Discreto

El controlador PID discreto es una implementación digital del control proporcional-integral-derivativo (PID), utilizado en sistemas de control de lazo cerrado como se evidencia en la figura 6. A diferencia de su versión continua, el PID discreto opera sobre señales digitalizadas y se implementa en microcontroladores, DSPs o PLCs [15].

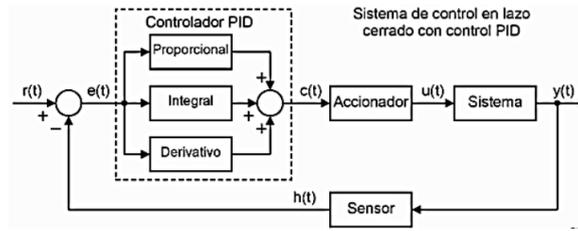


Figura 6: Sistema de control PID [10].

El control PID se basa en la combinación de tres términos:

- Proporcional (P): Genera una acción de control proporcional al error.
- Integral (I): Acumula el error en el tiempo para eliminar el error estacionario.
- Derivativo (D): Predice el error futuro con base en la tasa de cambio del error.

La ecuación en el dominio del tiempo discreto se expresa como se evidencia en la ecuación 9:

$$u(k) = K_p e(k) + K_i \sum_{i=0}^k e(i) T_s + K_d \frac{e(k) - e(k-1)}{T_s} \quad (9)$$

donde:

- $u(k)$ es la señal de control en el instante k .
- $e(k)$ es el error en el instante k .
- T_s es el tiempo de muestreo.
- K_p , K_i y K_d son las ganancias del controlador PID.

Implementación en Diferencias Atrasadas, para evitar la acumulación de errores y mejorar la estabilidad numérica, la ecuación en diferencias atrasadas del PID discreto se expresa en la ecuación 10:

$$u(k) = u(k-1) + K_p [e(k) - e(k-1)] + K_i e(k) T_s + K_d \frac{e(k) - 2e(k-1) + e(k-2)}{T_s} \quad (10)$$

Esta formulación permite calcular la señal de control de manera recursiva, aumentando la implementación en sistemas embebidos [24].

El ajuste de los parámetros K_p , K_i y K_d es fundamental para garantizar la estabilidad y el rendimiento del sistema de control. Existen diversos métodos para su ajuste, dependiendo de la complejidad del sistema y del nivel de precisión requerido [2].

Algunos de los enfoques más utilizados son:

- Método de Ziegler-Nichols: Se basa en el análisis de la respuesta del sistema a perturbaciones y en la obtención de la ganancia crítica para determinar los parámetros óptimos [25].
- Ajuste por ensayo y error: Aplicado en sistemas donde no se dispone de un modelo matemático preciso, permitiendo una calibración basada en la observación de la respuesta del sistema [26].
- Reducción mediante algoritmos evolutivos: Utilizado en sistemas no lineales para obtener parámetros óptimos a través de estrategias computacionales avanzadas [27].

El control PID discreto se implementa en microcontroladores para regular sistemas dinámicos en tiempo real. Su aplicación en un cubo autoequilibrante requiere de:

- Sensor giroscópico: Proporciona información sobre la inclinación y velocidad angular del sistema, permitiendo calcular el error de referencia.
- Motor: Modifica la salida del sistema en función de la señal de control generada por el PID.
- Algoritmo de control: Ejecuta la ecuación PID en cada intervalo de muestreo T_s , ajustando la respuesta del sistema en tiempo real.

IV-D. Integración Numérica en el Control PID

En sistemas digitales, como los microcontroladores, el término integral del controlador PID se implementa mediante integración numérica. Este método aproxima la integral continua del error utilizando una suma discreta de los errores en cada ciclo de muestreo.

IV-D1. Formulación del Término Integral: El término integral $I(k)$ en el ciclo k se calcula como en la ecuación 11:

$$I(k) = I(k - 1) + e(k) \cdot T_s \quad (11)$$

Donde:

- $I(k)$: Valor actual de la integral.
- $I(k - 1)$: Valor anterior de la integral.
- $e(k)$: Error en el ciclo actual ($e(k) = r(k) - y(k)$).
- T_s : Tiempo de muestreo (intervalo entre ciclos).

IV-D2. Propósito del Término Integral: El término integral acumula los errores pasados para eliminar el error en estado estacionario. Su contribución a la señal de control como se muestra en la ecuación 12:

$$u_I(k) = K_i \cdot I(k) \quad (12)$$

Donde K_i es la ganancia integral. Este término asegura que el sistema corrija desviaciones persistentes, mejorando la precisión.

IV-D3. Consideraciones Prácticas:

- Anti-windup: Para evitar que el término integral crezca excesivamente (windup), se limita su valor máximo y mínimo.
- Tiempo de muestreo (T_s): Debe ser lo suficientemente pequeño para capturar la dinámica del sistema, pero no tan pequeño que sobrecargue el microcontrolador.

La implementación del control PID en un microcontrolador sigue los siguientes pasos para este proyecto:

1. Adquisición de datos: Se lee la señal del sensor y se calcula el error $e(k)$, definido como la diferencia entre el valor de referencia y la medición actual.
2. Cálculo de la señal de control: Se evalúa la ecuación discreta del PID para obtener $u(k)$.
3. Aplicación de la señal al actuador: Se envía la señal de control al motor mediante una modulación PWM o una señal analógica.
4. Ejecución en lazo cerrado: Se repite el proceso en cada ciclo de muestreo T_s para mantener la estabilidad y precisión del sistema.

Entre las ventajas más importantes del uso de un PID discreto se tiene:

- Simplicidad de implementación: Basado en ecuaciones matemáticas bien definidas y de fácil integración en hardware digital.
- Estabilidad y respuesta rápida: Capaz de regular la mayoría de sistemas lineales con ajustes adecuados.
- Aplicabilidad en múltiples entornos: Utilizado en robótica, automatización industrial y sistemas embebidos.

El control de velocidad en un volante de inercia es fundamental en sistemas que requieren almacenamiento y regulación de energía mecánica. La velocidad angular del volante determina la cantidad de energía cinética que puede

acumular y liberar, lo que lo convierte en un componente clave en aplicaciones como sistemas de estabilización, almacenamiento de energía cinética y control de precisión en robots autoequilibrantes [28]. La variación de la velocidad se logra mediante la aplicación de un torque, que puede ser generado por un motor de corriente continua o sin escobillas, y modulado mediante técnicas de control automático [15].

La relación entre el torque aplicado y la velocidad del volante se rige por la ecuación dinámica del sistema, donde la aceleración angular es proporcional al torque neto aplicado y al momento de inercia del volante. En la práctica, este torque se genera a partir de una señal PWM (modulación por ancho de pulso), que regula la tensión aplicada al motor y, por lo tanto, la velocidad de giro. Un mayor ciclo de trabajo en la señal PWM incrementa el voltaje del motor, aumentando el torque y la aceleración angular del volante [29].

Para garantizar un control preciso de la velocidad, se pueden emplear estrategias como el control PID, que ajusta el torque en función de la diferencia entre la velocidad deseada y la velocidad real [26]. Además, la incorporación de sensores como encoders ópticos o giroscopios permite la realimentación del sistema, asegurando que las correcciones se apliquen en tiempo real y reduciendo errores en la regulación de la velocidad. En algunos casos, se utilizan técnicas avanzadas como control por modos deslizantes o control adaptativo, que mejoran la estabilidad frente a perturbaciones externas [30].

El control de velocidad en volantes de inercia es ampliamente utilizado en vehículos eléctricos, sistemas de estabilización satelital y robots de balanceo [31]. En estas aplicaciones, la capacidad de ajustar la velocidad del volante permite mejorar la eficiencia energética, reducir vibraciones no deseadas y mantener la estabilidad del sistema. La combinación de técnicas de control avanzadas con sensores de alta precisión ha permitido desarrollar sistemas más eficientes y confiables en el uso de volantes de inercia para diversas aplicaciones tecnológicas [32].

Los sistemas de autoequilibrio representan un desafío único en el ámbito del control, ya que requieren mantener un estado de equilibrio dinámico frente a perturbaciones constantes. El control de estos sistemas se basa en el uso de sensores como giroscopios y acelerómetros, los cuales miden la inclinación y la aceleración angular, proporcionando datos en tiempo real al controlador. A partir de esta información, el controlador ajusta los actuadores, como motores o servomecanismos, generando una respuesta inmediata para contrarrestar el desequilibrio. Aplicaciones como el cubo autoequilibrante y los Segways ilustran la eficacia de estos sistemas, destacando la importancia de algoritmos como el PID y el control por estados para mantener la estabilidad en condiciones dinámicas [33].

Sin embargo, los sistemas de autoequilibrio no solo son relevantes en dispositivos recreativos, sino también en la robótica y la industria. Por ejemplo, los robots móviles que operan en terrenos irregulares y los sistemas de estabilización de cámaras utilizan principios similares para mantener el equilibrio y realizar tareas precisas. Los avances en control adaptativo y predictivo han mejorado la capacidad de estos sistemas para adaptarse a perturbaciones no modeladas, incrementando su robustez y precisión en aplicaciones reales. La combinación de control en tiempo real y retroalimentación sensorial precisa ha establecido a los sistemas de autoequilibrio como una plataforma ideal para el desarrollo y la enseñanza de teoría de control avanzada [34].

IV-E. Softwares Utilizados

El diseño y validación del sistema de control requieren herramientas avanzadas de simulación y modelado. En este contexto, MATLAB y SolidWorks han sido seleccionados como las principales plataformas de desarrollo. MATLAB, desarrollado por MathWorks, es un entorno de programación y computación numérica ampliamente utilizado en ingeniería para el modelado de sistemas dinámicos, procesamiento de señales y análisis de datos. Adicionalmente, MATLAB ofrece compatibilidad con hardware embebido como el ESP32-WROOM-32, permitiendo la implementación directa de algoritmos de control en sistemas físicos [35]. Su capacidad para resolver ecuaciones diferenciales y analizar respuestas dinámicas lo convierte en una herramienta esencial en el diseño y validación del

sistema autoequilibrante [10].

Por otro lado, SolidWorks, desarrollado por Dassault Systèmes, es una plataforma de diseño asistido por computadora (Computer-Aided Design, CAD) utilizada en la ingeniería mecánica para la creación de modelos tridimensionales, simulación de ensamblajes y análisis estructurales. Su módulo de análisis por elementos finitos (Finite Element Analysis, FEA) permite evaluar la resistencia de los componentes del sistema, asegurando su viabilidad antes de la fabricación [36]. En conjunto, MATLAB y SolidWorks proporcionan un enfoque integral para el desarrollo del sistema, donde MATLAB se enfoca en la simulación del modelo matemático y SolidWorks permite la validación de su implementación estructural [37].

V. MARCO METODOLÓGICO

V-A. Selección de Componentes

La selección de los componentes en el sistema de autoequilibrio se realizó considerando criterios de eficiencia, compatibilidad y, especialmente, el peso de cada elemento. Dado que el diseño del cubo autoequilibrante depende de una distribución de masas precisa para garantizar estabilidad y control dinámico, se priorizaron componentes con una relación óptima entre rendimiento y peso. Un peso excesivo en ciertas partes del sistema podría desplazar el centro de masa de manera no deseada, afectando la capacidad del controlador para realizar correcciones en tiempo real.

V-B. Microcontrolador ESP32

El ESP32-WROOM-32 es un microcontrolador ampliamente utilizado en sistemas embebidos debido a su alto rendimiento, versatilidad y capacidades de conectividad. Desarrollado por Espressif Systems, este módulo integra un procesador Xtensa LX6 de doble núcleo con una frecuencia de operación de hasta 240 MHz, lo que permite la ejecución eficiente de algoritmos de control en tiempo real. Su arquitectura avanzada y su capacidad de procesamiento lo convierten en una opción ideal para aplicaciones que requieren respuestas rápidas y precisas, como el control de equilibrio en un sistema de volante de inercia [38].

Una de las principales ventajas del ESP32-WROOM-32 es su capacidad para manejar múltiples interfaces de comunicación, incluyendo SPI, I²C y UART, lo que facilita la integración de sensores y actuadores esenciales para el control del sistema. 5 [38].

Adicionalmente, un aspecto clave del ESP32-WROOM-32 es su conectividad inalámbrica integrada, que incluye Wi-Fi y Bluetooth 4.2 LE. Esta capacidad es especialmente útil en este proyecto, ya que permite la supervisión remota del sistema y la configuración de parámetros sin necesidad de conexión física. La posibilidad de ajustar las constantes del controlador PID y monitorear el comportamiento del sistema en tiempo real mejora significativamente el rendimiento y la capacidad de ajuste del sistema de autoequilibrio [38].

El motor BLDC, el giroscopio y la unidad de control fueron seleccionados no solo por su capacidad de respuesta y precisión, sino también por su peso reducido, lo que facilita un diseño más equilibrado. El motor BLDC Nidec 24H, con un peso de 250 gramos, ofrece una excelente relación entre torque y eficiencia sin generar una carga innecesaria sobre la estructura del cubo [39].

El giroscopio MPU-6000, con su diseño compacto y ligero, permite la adquisición de datos de orientación sin afectar la distribución del peso. Finalmente, el ESP32-WROOM-32, como unidad de procesamiento, proporciona la capacidad de control necesaria con un consumo energético reducido y un peso insignificante en el balance total del sistema [40].

El criterio de selección basado en el peso permitió mejorar la ubicación del centro de masa, facilitando la implementación de estrategias de control más eficientes. Al mantener la masa distribuida de manera uniforme y dentro de los límites del diseño mecánico, se reduce la demanda sobre el actuador principal, lo que mejora la capacidad del sistema para corregir inclinaciones y mantener la estabilidad en tiempo real [15].

V-C. Motor

Para el diseño del sistema de autoequilibrio, la selección del motor es un factor crítico, ya que debe proporcionar el torque y la velocidad angular necesarios para mantener la estabilidad del volante de inercia. En este proyecto, se ha optado por el motor BLDC Nidec 24H, debido a sus características mecánicas y eléctricas óptimas para esta aplicación. Su diseño sin escobillas permite una respuesta rápida y precisa, facilitando el control del volante de

inercia mediante aceleraciones y desaceleraciones controladas [39].

El motor seleccionado tiene un diámetro de 54 mm y una longitud de 30 mm, lo que lo hace compacto y fácil de integrar en la estructura del sistema sin afectar la distribución de masas del dispositivo [16]. Su peso total es de 250 g, lo que permite reducir la carga sobre la estructura y evitar efectos negativos en la estabilidad del sistema. El eje de salida tiene un diámetro de 4 mm y una longitud de 20 mm, lo que proporciona una conexión estable con el volante de inercia y facilita la transmisión del torque requerido para el equilibrio dinámico del cubo como se evidencia en la figura 7 .

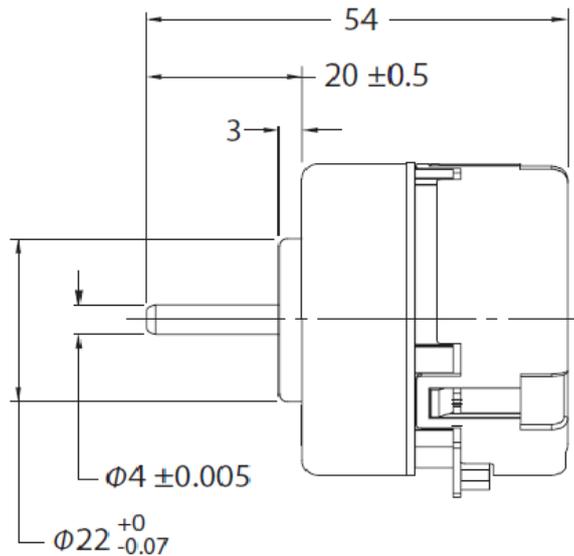


Figura 7: Plano acotado de motor [39].

La elección del motor BLDC Nidec 24H se fundamenta en varios factores clave. En primer lugar, su alta eficiencia energética permite reducir el consumo eléctrico, ya que al ser un motor sin escobillas reduce las pérdidas por fricción y mejora la conversión de energía en movimiento útil [13].

En segundo lugar, su rápida respuesta al control lo hace ideal para sistemas de estabilización en tiempo real, ya que permite cambios de velocidad y dirección, esenciales para corregir el equilibrio del volante de inercia. Además, la ausencia de escobillas en su diseño contribuye a un bajo mantenimiento, reduciendo el desgaste mecánico y prolongando la vida útil del sistema [41].

Adicionando otro aspecto clave en su selección es su compatibilidad con el controlador VESC, lo que permite una modulación precisa del torque y la velocidad mediante señales PWM generadas por el ESP32-WROOM-32. Esta integración facilita la implementación de algoritmos de control avanzados y garantiza un funcionamiento estable y eficiente del motor dentro del sistema de autoequilibrio [42].

Finalmente, su peso ligero y dimensiones compactas permiten mejorar el centro de masa del sistema, favoreciendo la estabilidad general y mejorando la respuesta del control ante perturbaciones externas [33].

V-D. Giroscopio

El giroscopio es un sensor fundamental en el sistema de autoequilibrio, ya que permite medir la velocidad angular y estimar la orientación en tiempo real. Su incorporación en el sistema garantiza la estabilidad mediante la retroalimentación de datos precisos al controlador, lo que permite realizar correcciones dinámicas y mantener el equilibrio. Para su selección, es fundamental considerar aspectos como la precisión, la frecuencia de muestreo, el consumo energético y la compatibilidad con la plataforma de control, en este caso, el ESP32-WROOM-32.

Para la selección del giroscopio adecuado, se analizaron diversos factores técnicos. En primer lugar, la precisión y estabilidad del sensor debe ser alta para reducir el ruido en la medición. Un bajo nivel de ruido permite mejorar la estimación de la orientación del sistema y evita la acumulación de errores en la integración de la señal. En segundo lugar, la frecuencia de muestreo debe ser suficiente para captar variaciones rápidas en la dinámica del sistema, por lo que se establece como criterio un mínimo de 1 kHz. Otro aspecto clave es el consumo energético, que debe mantenerse por debajo de los 5 mA para evitar una carga excesiva en la alimentación del sistema. Finalmente, la compatibilidad con el ESP32-WROOM-32 es un factor determinante, por lo que se priorizan sensores con interfaces SPI o I²C, que garantizan una comunicación eficiente con el microcontrolador.

Tras un análisis comparativo de sensores giroscópicos disponibles en el mercado, se seleccionó el MPU-6000 debido a sus características avanzadas. Este sensor ofrece una alta precisión con una frecuencia de muestreo de hasta 8 kHz, lo que permite una respuesta rápida y una integración numérica más precisa. Además, presenta un bajo consumo energético de 3.6 mA, lo que lo hace eficiente en términos de consumo energético. Su compatibilidad con el ESP32-WROOM-32, mediante interfaces SPI e I²C, facilita su integración con el sistema de control. Asimismo, el MPU-6000 cuenta con un filtro digital integrado, lo que mejora la estabilidad de la señal y reduce el ruido en las mediciones, contribuyendo a una estimación más precisa de la orientación.

En el proyecto, el MPU-6000 se utilizará para medir la velocidad angular del sistema y, en conjunto con su acelerómetro interno, permitirá la estimación de la orientación en tiempo real. Su integración con el ESP32-WROOM-32 facilitará la ejecución de algoritmos avanzados de control, reduciendo perturbaciones externas ante la respuesta del sistema. La implementación de este sensor garantiza un monitoreo preciso del estado del sistema, permitiendo realizar ajustes en tiempo real para mantener la estabilidad del cubo autoequilibrante.

V-E. Diseño de Estructura

El diseño estructural del cubo autoequilibrante fue desarrollado considerando tanto la resistencia mecánica como la reducción del peso total del sistema. Para lograr una estructura ligera pero resistente, se utilizó PLA como material de fabricación, debido a su facilidad de impresión en 3D, su estabilidad dimensional y su adecuado balance entre rigidez y peso [43]. El diseño fue segmentado en diferentes tipos de piezas con funciones específicas, garantizando una distribución eficiente de la masa y una estructura lo suficientemente robusta para soportar los esfuerzos generados por el motor y el volante de inercia, como se muestra en la figura.

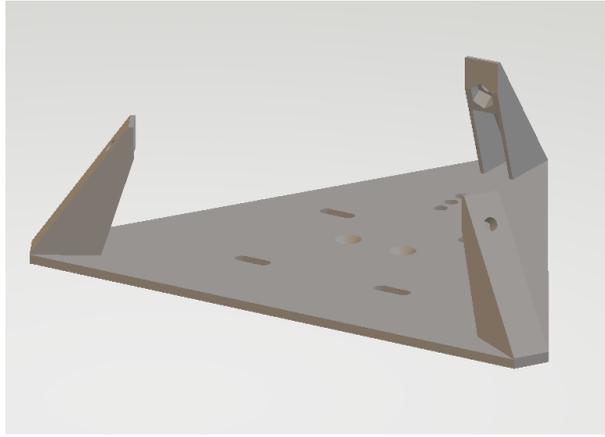


Figura 8: Base del controlador [Fuente: Autores].

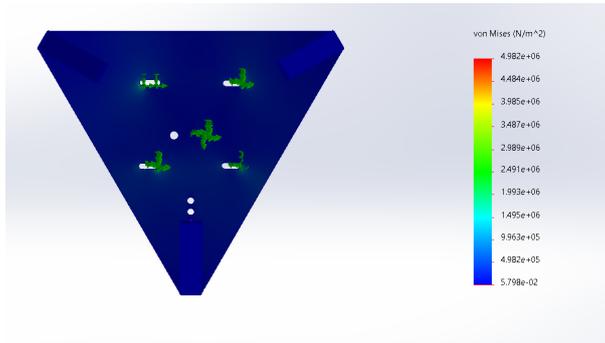


Figura 9: Pared del Motor [Fuente: Autores].

Para la construcción del cubo, se fabricaron dos tipos de paredes. Dos de ellas fueron diseñadas con refuerzos estructurales internos, aumentando su resistencia como se muestra en la figura 11 y proporcionando un soporte firme al motor como se muestra en la figura 10. Estas piezas están estratégicamente ubicadas para reducir la flexión, soportar los esfuerzos y absorber la vibración generada por la aceleración y desaceleración del motor BLDC [44]. Las otras tres paredes fueron diseñadas con un espesor menor y sin refuerzos adicionales, con el objetivo de reducir el peso total del sistema sin comprometer la rigidez global de la estructura.

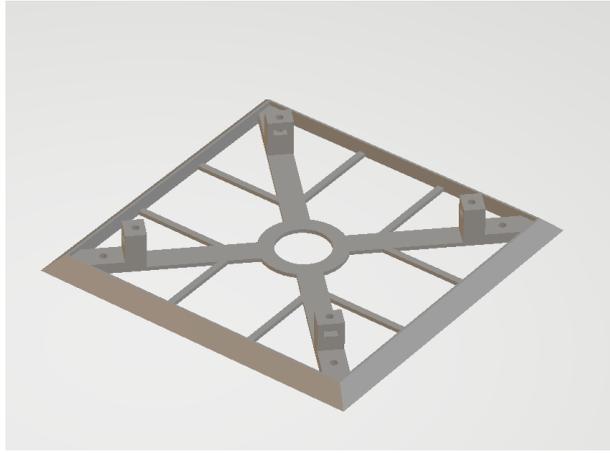


Figura 10: Pared del Motor [Fuente: Autores].

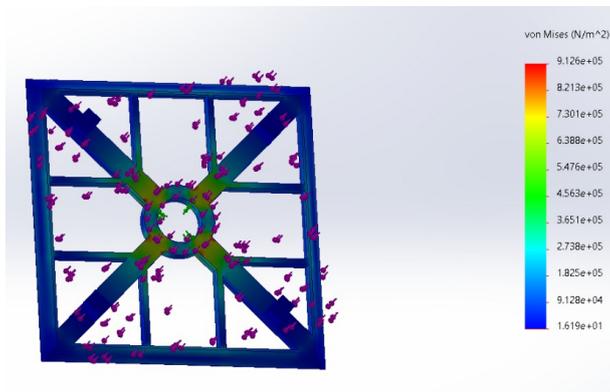


Figura 11: Pared del Motor [Fuente: Autores].

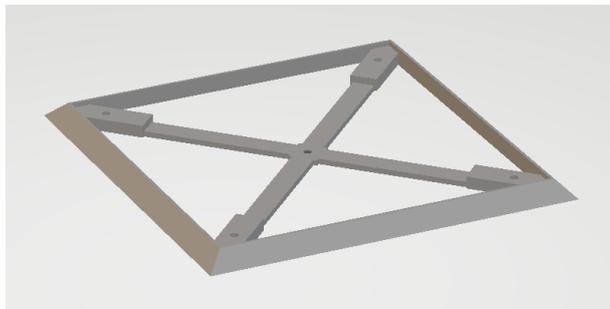


Figura 12: Pared secundaria [Fuente: Autores].

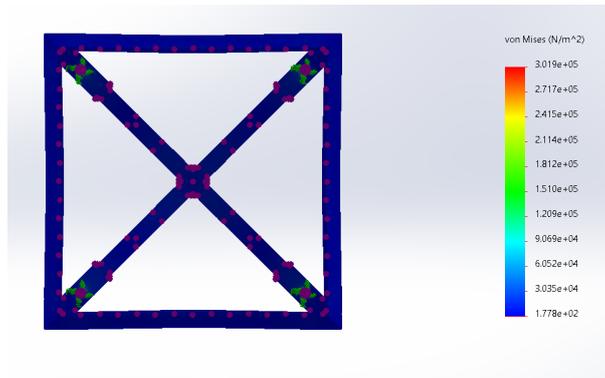


Figura 13: Pared del Motor [Fuente: Autores].

En la parte interna del cubo se integró una superficie rectangular destinada a la fijación de la placa de circuito impreso (PCB) como se muestra en la figura 14. Esta superficie permite atornillar de manera segura el controlador, soportar los esfuerzos, los puertos del motor y la conexión de la batería. Además, esto permite una conexión ordenada del cableado evitando movimientos indeseados durante la operación del sistema asegurando una distribución ordenada del cableado y evitando movimientos indeseados durante la operación del sistema. La ubicación del controlador fue seleccionada de manera que se reduzcan interferencias electromagnéticas y se facilite el acceso a los puertos de comunicación y alimentación [28].

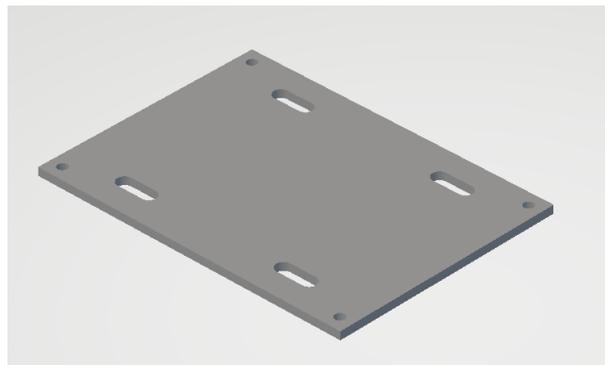


Figura 14: Superficie de apoyo del controlador [Fuente: Autores].

Para garantizar la resistencia mecánica del cubo, se diseñó una unión de soporte como se evidencia en la figura 15, de la cual se fabricaron 8 unidades encargadas de conectar las intersecciones de las seis caras. Estas uniones aseguran un ensamblaje firme de las paredes, soportan los esfuerzos, y evitan la deformación de la estructura ante impactos o vibraciones. Su diseño en forma de bloques de fijación permite un ensamblaje preciso, asegurando que todas las caras se mantengan alineadas correctamente y que la distribución de cargas sea uniforme [31].

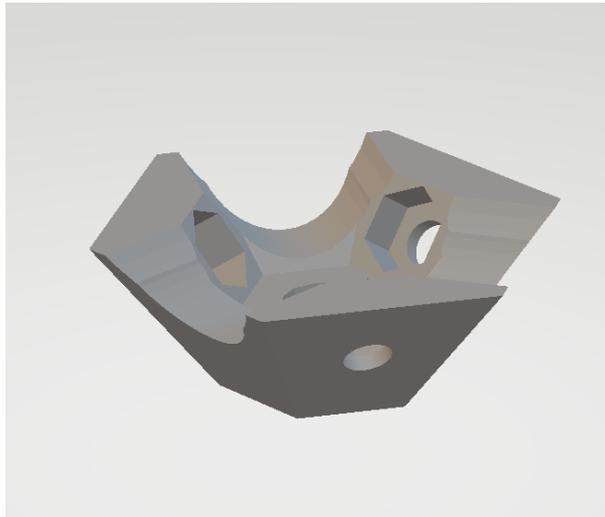


Figura 15: Vértice para unión de paredes [Fuente: Autores].

En la figura 16, se muestra el disco de inercia diseñado, el cual juega un papel fundamental en la generación del momento angular necesario para el autoequilibrio del cubo. Este disco fue diseñado con una distribución de masa para aumentar su momento inercia sin agregar un peso excesivo al sistema. Soporta los esfuerzos generados durante su rotación. La selección de su diámetro y grosor se realizó en función de las especificaciones del motor BLDC, garantizando un equilibrio adecuado entre resistencia a los esfuerzos generados, como se muestra en la figura 17 capacidad de reacción y eficiencia energética [28].

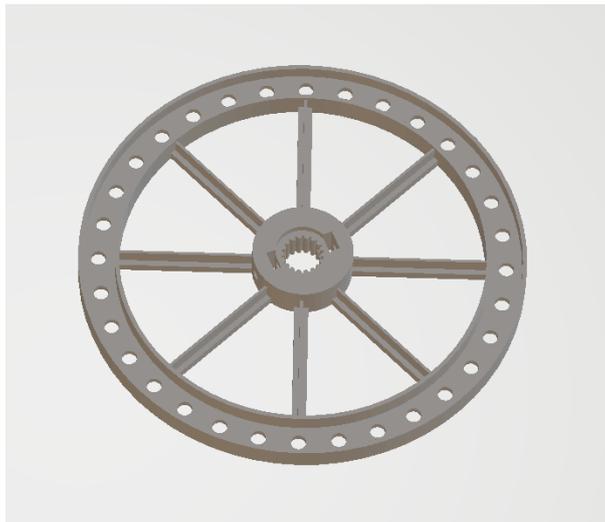


Figura 16: Disco de motor [Fuente: Autores].

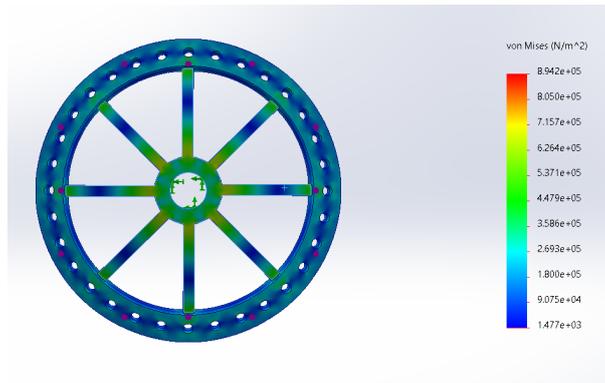


Figura 17: Disco de motor [Fuente: Autores].

Por otro lado, se incorporaron soportes específicos para el motor, como se evidencia en la figura 18, con el fin de reducir vibraciones no deseadas, soportar los esfuerzos y mejorar la transferencia del torque al disco de inercia. Estos soportes fueron diseñados con una estructura reforzada y orificios precisos para la fijación del motor con tornillos [44], asegurando que el eje de salida del motor permanezca en la alineación óptima durante la operación, como se evidencia en la figura 19. Posteriormente, fueron fabricados en PLA para garantizar durabilidad y precisión dimensional.

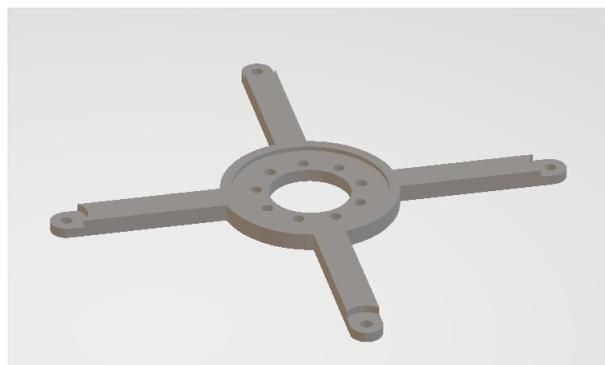


Figura 18: Soporte de motor [Fuente: Autores].

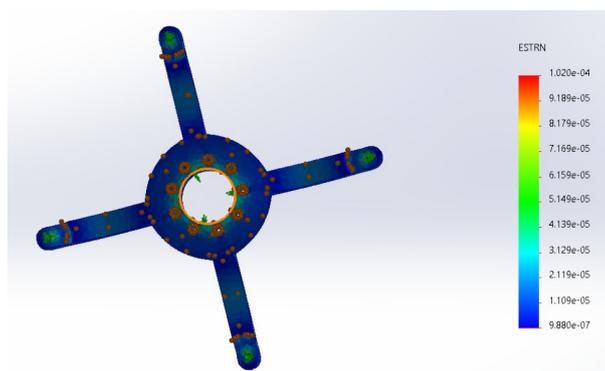


Figura 19: Base del controlador [Fuente: Autores].

El proceso de diseño e impresión de estas piezas se llevó a cabo en software de modelado 3D, permitiendo realizar simulaciones estructurales previas a la fabricación. Se evaluó la capacidad de las piezas para soportar los

esfuerzos mediante pruebas con diferentes espesores y configuraciones para reducir la relación peso-rigidez de cada componente, asegurando así un rendimiento adecuado en términos de estabilidad y respuesta dinámica del sistema [32].

El diseño estructural del cubo autoequilibrante fue desarrollado considerando tanto la resistencia mecánica como la reducción del peso total del sistema. Para lograr una estructura ligera pero resistente, se utilizó PLA como material de fabricación, debido a su facilidad de impresión en 3D, su estabilidad dimensional y su adecuado balance entre rigidez y peso [43]. El diseño fue segmentado en diferentes tipos de piezas con funciones específicas, garantizando una distribución eficiente de la masa y una estructura lo suficientemente robusta para soportar los esfuerzos generados por el motor y el volante de inercia.

Para la construcción del cubo, se fabricaron dos tipos de paredes. Dos de ellas fueron diseñadas con refuerzos estructurales internos, aumentando su resistencia y proporcionando un soporte firme al motor. Estas piezas están estratégicamente ubicadas para reducir la flexión y absorber la vibración generada por la aceleración y desaceleración del motor BLDC [44]. Las otras tres paredes fueron diseñadas con un espesor menor y sin refuerzos adicionales, con el objetivo de reducir el peso total del sistema sin comprometer la rigidez global de la estructura.

En la parte interna del cubo se integró una superficie rectangular destinada a la fijación de la placa de circuito impreso (PCB). Esta superficie permite atornillar de manera segura el controlador, los puertos del motor y la conexión de la batería, asegurando una distribución ordenada del cableado y evitando movimientos indeseados durante la operación del sistema. La ubicación del controlador fue seleccionada de manera que se reducen interferencias electromagnéticas y se facilite el acceso a los puertos de comunicación y alimentación [28].

Para garantizar la integridad mecánica del cubo, se diseñaron y fabricaron ocho uniones de soporte, las cuales se encargan de conectar las intersecciones de las seis caras. Estas uniones aseguran un ensamblaje firme de las paredes y evitan la deformación de la estructura ante impactos o vibraciones. Su diseño en forma de bloques de fijación permite un ensamblaje preciso, asegurando que todas las caras se mantengan alineadas correctamente y que la distribución de cargas sea uniforme [31].

Un elemento crítico en el diseño es el disco de inercia, el cual juega un papel fundamental en la generación del momento angular necesario para el autoequilibrio del cubo. Este disco fue diseñado con una distribución de masa para aumentar su inercia sin agregar un peso excesivo al sistema. La selección de su diámetro y grosor se realizó en función de las especificaciones del motor BLDC, garantizando un equilibrio adecuado entre capacidad de reacción y eficiencia energética [28].

Sin embargo, se incorporaron soportes específicos para el motor con el fin de reducir vibraciones no deseadas y mejorar la transferencia de torque al disco de inercia. Estos soportes, fabricados también en PLA, fueron diseñados con una estructura reforzada y orificios precisos para la fijación del motor con tornillos, asegurando que el eje de salida del motor permanezca en la alineación óptima durante la operación [44].

El proceso de diseño e impresión de estas piezas se llevó a cabo en software SolidWorks de modelado 3D, permitiendo realizar simulaciones estructurales previas a la fabricación. Se realizaron pruebas con diferentes espesores y configuraciones para mejorar la relación peso-rigidez de cada componente, asegurando así un rendimiento adecuado en términos de estabilidad y respuesta dinámica del sistema [32].

V-F. Diseño de PCB y Electrónica

El diseño de la PCB fue desarrollado con el objetivo de integrar todos los componentes electrónicos necesarios para el control eficiente del sistema de autoequilibrio. La tarjeta cumple la función de interconectar el microcontrolador ESP32-WROOM-32, los motores brushless Nidec 24H, el sensor inercial y los módulos de potencia. Su

diseño compacto permite reducir la cantidad de cableado externo, reduciendo interferencias electromagnéticas y asegurando una conexión fiable entre los distintos módulos [43].

V-F1. Consideraciones del Microcontrolador: El ESP32-WROOM-32 es el núcleo del sistema, encargado de recibir la información del giroscopio y generar las señales de control para los motores [17]. Se seleccionó por su capacidad de procesamiento dual-core, conectividad Wi-Fi/Bluetooth integrada y compatibilidad con el protocolo I2C para el sensor MPU-6050. Su bajo consumo energético y facilidad de programación lo hacen adecuado para este sistema.

V-F2. Consideraciones del Sensor: El sistema emplea un sensor MPU-6050, proporcionando datos en tiempo real sobre inclinación y velocidad angular [11]. Este sensor combina acelerómetro y giroscopio, ofreciendo lecturas precisas mediante I2C [45].

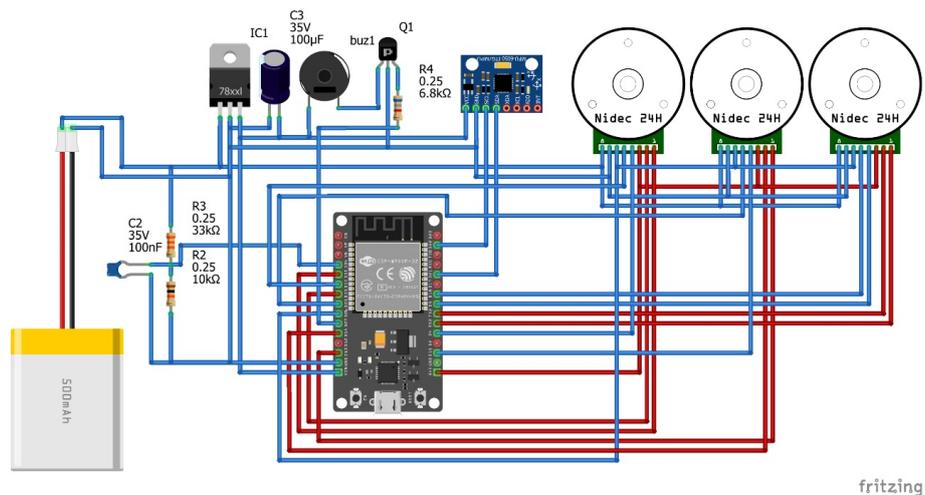


Figura 20: Diagrama de conexión electrónica [Fuente: Autores].

V-F3. Consideraciones de la Batería: El sistema se alimenta mediante una batería LiPo de 500mAh, destacada por su alta densidad energética y capacidad de descarga estable. Un regulador 7805 suministra 5V a los componentes lógicos y un convertidor DC-DC proporciona 12V a los motores [21].

V-F4. Consideraciones de los Motores: Se utilizan tres motores brushless Nidec 24H, elegidos por su eficiencia y rápida respuesta [16]. La etapa de potencia, controlada por señales PWM desde el ESP32, ajusta velocidad y dirección de giro [33].

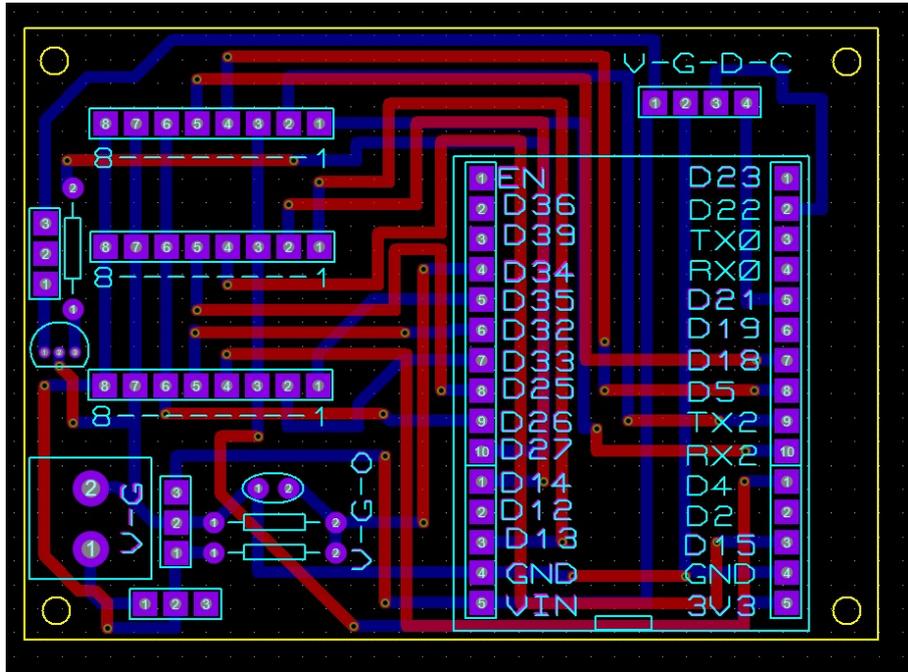


Figura 21: Placa electrónica PCB [Fuente: Autores].

V-G. Modelo matemático

V-G1. *Cálculo de las Ganancias del Controlador PID*: El diseño del controlador PID para el sistema autoequilibrante se basa en el modelo dinámico del volante de inercia y su interacción con el motor brushless. Se requiere obtener valores óptimos para K_1 , K_2 y K_3 , asegurando estabilidad y respuesta rápida del sistema.

Las ganancias han sido determinadas empíricamente mediante un método de prueba y error, observando la respuesta del sistema en simulaciones y pruebas físicas. Este enfoque permitió ajustar los valores de forma iterativa hasta obtener un desempeño adecuado, asegurando que el sistema se estabilice en un tiempo razonable y sin oscilaciones excesivas. Para esta tarea, se tomaron como referencia metodologías documentadas en [46], donde se implementó un esquema similar de control PID en un cubo autoequilibrante, cuyos valores referenciales del proyecto eran $K_1 = 135$, $K_2 = 11,0$ y $K_3 = 0,03$.

V-G2. *Modelo Dinámico del Sistema*: La ecuación de movimiento del sistema es:

$$J \frac{d\omega}{dt} = \tau - mgL \sin(\theta) \quad (13)$$

Para ángulos pequeños, $\sin(\theta) \approx \theta$, por lo que la ecuación linealizada es:

$$J \frac{d\omega}{dt} = \tau - mgL\theta \quad (14)$$

Donde los parámetros del sistema son:

- $J = 0,005 \text{ kg}\cdot\text{m}^2$ (momento de inercia del volante de inercia).
- $m = 1,2 \text{ kg}$ (masa del sistema).
- $g = 9,81 \text{ m/s}^2$ (aceleración de la gravedad).
- $L = 0,08 \text{ m}$ (distancia al centro de masa).
- $K_m = 0,02 \text{ N}\cdot\text{m/A}$ (constante de torque del motor).
- $I_{\text{máx}} = 4,5 \text{ A}$ (corriente máxima del motor).

Aplicando la transformada de Laplace:

$$Js\Omega(s) = \tau(s) - mgL\Theta(s) \quad (15)$$

Despejando la función de transferencia del sistema:

$$G(s) = \frac{\Theta(s)}{\tau(s)} = \frac{1}{Js^2 + mgL} \quad (16)$$

Sustituyendo valores:

$$G(s) = \frac{1}{0,005s^2 + (1,2)(9,81)(0,08)} \quad (17)$$

$$G(s) = \frac{1}{0,005s^2 + 0,94} \quad (18)$$

V-G3. *Discretización del Sistema:* Se emplea un tiempo de muestreo $T_s = 0,01$ s, utilizando la aproximación:

$$s \approx \frac{z-1}{T_s} \quad (19)$$

Sustituyendo en 18:

$$G(z) = \frac{1}{0,005 \left(\frac{(z-1)^2}{0,0001} \right) + 0,94} \quad (20)$$

$$G(z) = \frac{1}{z^2 - 2z + 1 + 94} \quad (21)$$

V-G4. *Diseño del Controlador PID:* El controlador PID discreto se define como:

$$PWM(k) = K_1\theta(k) + K_2\omega(k) + K_3 \sum \theta(k) \quad (22)$$

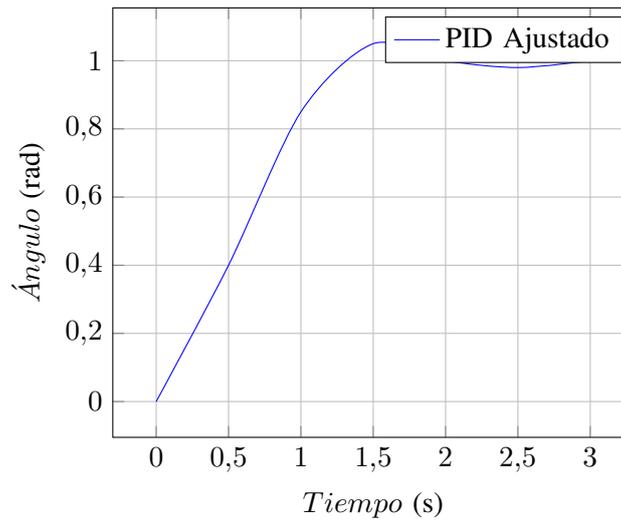
Donde:

- $K_1 = 140$ (acción proporcional).
- $K_2 = 12,0$ (acción derivativa).
- $K_3 = 0,02$ (acción integral).

Estos valores fueron determinados empíricamente mediante un método de prueba y error, evaluando la respuesta del sistema en simulaciones y ajustándolos hasta lograr un equilibrio entre estabilidad y rapidez de respuesta. Esta técnica fue basada en la metodología documentada en [46], donde se realizó un ajuste similar en un sistema de control para un cubo autoequilibrante.

V-G5. *Simulación de la Respuesta del Sistema:* A continuación, se muestra la simulación del sistema con las ganancias seleccionadas. Se observa que el sistema alcanza la estabilidad en aproximadamente 2 segundos, con un sobrepaso moderado y una respuesta amortiguada.

Respuesta al Escalón del Sistema PID



V-H. Sistema de Volante de Inercia

El volante de inercia es un componente clave en el sistema de estabilización, ya que permite generar un torque de reacción al variar su velocidad angular. Se emplea en diversas aplicaciones, incluyendo sistemas de estabilización de robots, almacenamiento de energía en sistemas renovables y estabilización de vehículos aeroespaciales.

V-I. Energía y Momento Angular en un Volante de Inercia

La energía acumulada en el volante de inercia depende de su momento de inercia y su velocidad angular, dada por la ecuación 23:

$$E = \frac{1}{2}I\omega^2 \quad (23)$$

Para resolver la ecuación usando valores reales, consideramos:

$$I = \frac{1}{2}mr^2 = \frac{1}{2}(0,5 \text{ kg})(0,1 \text{ m})^2 = 0,0025 \text{ kg} \cdot \text{m}^2$$

$$\omega = 314,16 \text{ rad/s} \quad (\text{equivalente a } 3000 \text{ rpm})$$

Sustituimos los valores en la ecuación de energía:

$$E = \frac{1}{2}(0,0025 \text{ kg} \cdot \text{m}^2)(314,16 \text{ rad/s})^2$$

$$E = \frac{1}{2}(0,0025)(98696,45) = 123,37 \text{ Joules}$$

La energía almacenada en el volante de inercia es $E = 123,37 \text{ Joules}$.

El volante de inercia conserva su momento angular L , expresado en la ecuación 24:

$$L = I\omega \quad (24)$$

Sustituimos los valores:

$$L = (0,0025 \text{ kg} \cdot \text{m}^2)(314,16 \text{ rad/s}) = 0,7854 \text{ kg} \cdot \text{m}^2/\text{s}$$

El momento angular es $L = 0,7854 \text{ kg} \cdot \text{m}^2/\text{s}$.

V-J. Dinámica del Volante de Inercia con Motor Brushless

La dinámica del volante de inercia en presencia del motor brushless se describe mediante la ecuación diferencial 25:

$$\tau = I \frac{d\omega}{dt} \quad (25)$$

Usando valores reales, si la aceleración angular es:

$$\frac{d\omega}{dt} = 50 \text{ rad/s}^2$$

Sustituimos en la ecuación:

$$\tau = (0,0025 \text{ kg} \cdot \text{m}^2)(50 \text{ rad/s}^2) = 0,125 \text{ N} \cdot \text{m}$$

El torque generado es $\tau = 0,125 \text{ N} \cdot \text{m}$.

Este análisis muestra cómo se relacionan la energía, el momento angular y el torque del volante de inercia, proporcionando una base sólida para el control del sistema.

Este modelo matemático proporciona la base para la implementación del sistema de control, asegurando que el motor brushless actúe de manera eficiente para estabilizar el sistema de autoequilibrio mediante el control del torque aplicado al volante de inercia.

Para analizar el comportamiento del cubo autoequilibrante, se emplea un modelo en diferencias que describe la dinámica del control digital. La idea fundamental es expresar la relación entre la inclinación del sistema y la señal de control (pwm) que se envía a los motores. A continuación se detalla el proceso de obtención de dichas ecuaciones y su interpretación:

■ Definición de variables y estados.

Se asume que el cubo puede inclinarse en un eje (por ejemplo, el eje X), y que el firmware del ESP32 mide tanto el ángulo de inclinación como la velocidad angular mediante un sensor inercial (MPU6000 o similar).

Para el controlador, se manejan las siguientes variables:

- $\theta(k)$: inclinación del cubo en el instante de muestreo k .
- $\dot{\theta}(k)$: velocidad angular estimada en el instante k .
- $m(k)$: estado interno asociado al control, que se actualizará en cada ciclo.
- $\text{pwm}(k)$: señal de salida que se envía a los motores para corregir la inclinación.

■ Ecuación de salida (ley de control).

En un control proporcional-derivativo con integración discreta (o un control PD+I no estándar), puede escribirse la señal de salida como la combinación lineal de:

- $\theta(k)$ para la acción proporcional,
- $\dot{\theta}(k)$ para la acción derivativa,
- $m(k)$ para la acción integradora discreta.

Así surge la primera ecuación, donde K_1 , K_2 y K_3 son las ganancias ajustables como se muestra en la ecuación 26:

$$\text{pwm}(k) = K_1 \theta(k) + K_2 \dot{\theta}(k) + K_3 m(k). \quad (26)$$

■ Ecuación de actualización del estado interno.

Se introduce la variable $m(k)$ para acumular, a lo largo del tiempo, la acción de la señal $\text{pwm}(k)$. De esta forma, en cada ciclo de muestreo como se muestra en la ecuación 27:

$$m(k+1) = m(k) + \text{pwm}(k). \quad (27)$$

Esta segunda ecuación describe un integrador discreto, cuyo efecto es similar al de una acción integral en un controlador PID, pero en lugar de integrar el error, se integra la propia señal de pwm . De esta manera, el sistema puede eliminar errores residuales o compensar perturbaciones constantes. El controlador implementado en el ESP32-WROOM-32 queda descrito por el sistema:

$$\begin{cases} \text{pwm}(k) = K_1 \theta(k) + K_2 \dot{\theta}(k) + K_3 m(k), \\ m(k+1) = m(k) + \text{pwm}(k). \end{cases} \quad (28)$$

La primera ecuación de la expresión 28 calcula $\text{pwm}(k)$ (la acción de control), mientras que la segunda describe la forma en que dicha acción se integra en la variable $m(k)$.

■ Interpretación física.

- K_1 (proporcional a θ) incrementa o reduce $\text{pwm}(k)$ en función de la inclinación actual del cubo. Una ganancia K_1 alta hace al sistema más rígido ante desviaciones.
- K_2 (proporcional a $\dot{\theta}$) introduce amortiguación al medir la velocidad angular: una inclinación que crece rápido genera una corrección más enérgica.
- K_3 relaciona $\text{pwm}(k)$ con el estado acumulado $m(k)$. Al integrarse sucesivas correcciones, se contrarrestan de manera sostenida errores o perturbaciones de larga duración.

■ Función de transferencia en el dominio Z .

Para profundizar en el análisis, se pasa al dominio Z , donde $\text{pwm}(k)$ se expresa como $P(z)$, $\theta(k)$ como $\Theta(z)$ y $m(k)$ como $M(z)$. La relación de $P(z)$ respecto a $\Theta(z)$ queda representada en la ecuación 29:

$$\frac{P(z)}{\Theta(z)} = \frac{K_1 + K_2 \frac{z-1}{z}}{1 - K_3 \frac{z}{z-1}} \quad (29)$$

donde $P(z)$ y $\Theta(z)$ representan las transformadas Z de $\text{pwm}(k)$ y $\theta(k)$, respectivamente. Este modelo ayuda a comprender cómo, al variar K_1 , K_2 y K_3 , se ajusta la dinámica total del sistema.

■ Ajuste de ganancias.

En la práctica, para estabilizar el cubo, las ganancias K_1 , K_2 y K_3 se ajustan de acuerdo a la ecuación 29:

- K_1 (proporcional al ángulo, ver ecuación 29) provee la rigidez necesaria para contrarrestar desviaciones grandes de θ . Un valor alto de K_1 aumenta la rapidez de la respuesta, pero también puede inducir oscilaciones si no se ajusta correctamente.
- K_2 (proporcional a la velocidad angular, ver ecuación 29) contribuye a la amortiguación y evita oscilaciones, actuando como un freno dinámico frente a perturbaciones rápidas.
- K_3 (componente de integración discreta, ecuación 29) permite cancelar errores residuales o perturbaciones constantes a largo plazo. Sin embargo, un valor excesivo de K_3 puede desplazar el polo demasiado cerca del origen, generando saturación o inestabilidad.

El ajuste de estas ganancias se realiza iterativamente, observando la respuesta del cubo y utilizando herramientas analíticas como el análisis de polos y ceros en el plano Z . La ubicación de los polos determina la estabilidad, y los ceros afectan la rapidez y precisión de la respuesta. En general, se busca un balance donde K_1 y K_2 proporcionen una respuesta rápida y estable, mientras que K_3 elimine errores a largo plazo sin comprometer la estabilidad.

Estos parámetros se pueden afinar tanto experimentalmente (observando la respuesta del cubo) como mediante métodos de control digital (por ejemplo, a partir de un modelo linealizado y técnicas de diseño por ubicación de polos en el plano Z).

En esta simulación se observa cómo la variable $\text{pwm}(k)$ y el estado $m(k)$ interactúan para regular el ángulo θ . Al aumentar K_1 , el control se vuelve más enérgico frente a cambios de θ . Al incrementar K_2 , se introduce mayor amortiguamiento (dependencia de la velocidad angular). Y al modificar K_3 , se ajusta la compensación de errores residuales a largo plazo. Estas consideraciones concuerdan con las técnicas de sintonía de control digital mencionadas en [47]-[49].

V-K. Diseño del Controlador

El diseño del controlador PID para el sistema autoequilibrante se fundamenta en la regulación de la inclinación del cubo mediante la acción de un volante de inercia, controlado por un motor BLDC de alta velocidad. La elección del esquema PID responde a la necesidad de un controlador de acción continua que proporcione estabilidad y robustez ante perturbaciones externas. Para su implementación, se adoptó una formulación en diferencias finitas, asegurando estabilidad numérica en la discretización del sistema [15].

```

1 %% MODELO MATEMATICO Y AJUSTE DE GANANCIAS
2 % Parametros del controlador (iniciales)
3 K1 = 160;
4 K2 = 10.5;
5 K3 = 0.03;
6
7 % Tiempo de muestreo
8 Ts = 0.01;
9
10 % Duracion
11 N = 3000;
12 t = (0:N-1)*Ts;
13
14 % Variables:
15 % theta(k): angulo [grados]
16 % w(k): velocidad angular [grados/s]
17 % m(k): estado "motor_speed" discreto
18 % pwm(k): senal de control
19
20 theta = zeros(1,N);
21 w = zeros(1,N);
22 m = zeros(1,N);
23 pwm = zeros(1,N);
24
25 % Condiciones iniciales
26 theta(1) = 15; % 15 grados de inclinacion inicial
27 w(1) = 0;
28 m(1) = 0;
29
30 % Parametros simplificados de la "planta":
31 A = 0.1; % Factor "gravedad" simplificado
32 B = 0.05; % Rozamiento
33 Kmotor = 0.8; % Eficacia en el torque por pwm
34
35 for k = 1:N-1
36
37 % -- Controlador discreto (ver ecuacion (1)) --
38 pwm(k) = K1*theta(k) + K2*w(k) + K3*m(k);
39
40 % m(k+1) = m(k) + pwm(k)
41 m(k+1) = m(k) + pwm(k);
42
43 % -- Dinamica de la planta --
44 % w(k+1) = w(k) + Ts*( -A*sin(theta) -B*w(k) + Kmotor*pwm(k) )
45 rad_k = deg2rad(theta(k));
46 torque_total = -A*sin(rad_k) - B*w(k) + Kmotor*pwm(k);
47 w(k+1) = w(k) + Ts*torque_total;
48
49 % theta(k+1) = theta(k) + Ts*w(k+1)
50 theta(k+1) = theta(k) + Ts*w(k+1);
51 end
52
53 % Ultimo valor de pwm
54 pwm(end) = pwm(end-1);
55
56 %% GRAFICAS
57 figure('Name','Simulacion_Control_Discreto','Color','white');

```

Figura 22: Simulación en MATLAB con función de transferencia discreta y ajuste de ganancias Parte 1. [Fuente: Autores]

```

1 subplot(3,1,1);
2 plot(t, theta, 'LineWidth', 1.5); grid on;
3 ylabel('\theta_[grados]');
4 title('Angulo_del_sistema');
5
6 subplot(3,1,2);
7 plot(t, w, 'LineWidth', 1.5); grid on;
8 ylabel('\omega_[grados/s]');
9 title('Velocidad_angular');
10
11 subplot(3,1,3);
12 plot(t, pwm, 'LineWidth', 1.5); hold on;
13 plot(t, m, 'LineWidth', 1.2);
14 grid on;
15 xlabel('Tiempo_[s]');
16 ylabel('Valor');
17 legend('pwm(k)', 'm(k)', 'Location', 'best');
18 title('Senales_de_control_y_estado_interno');
19
20 sgtitle(sprintf('Simulacion_PD+Integracion_Discreta:_K1=%.1f,_K2=%.2f,_K3=%.3f', ...
21               K1, K2, K3));

```

Figura 23: Simulación en MATLAB con función de transferencia discreta y ajuste de ganancias Parte 2. [Fuente: Autores]

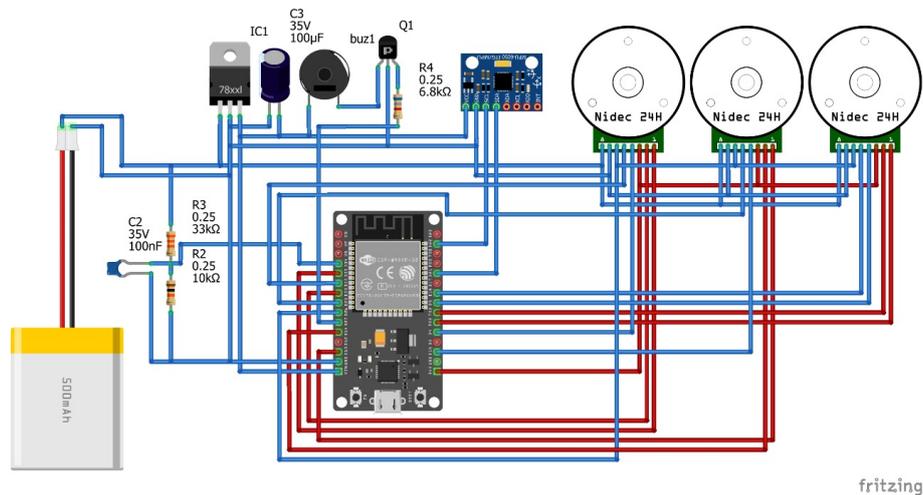


Figura 24: Diagrama electrónico de conexiones. [Fuente: Autores]

Dado que el modelo matemático del sistema responde a la dinámica de un péndulo invertido, su ecuación de estado es inherentemente inestable, requiriendo una estrategia de control que compense la desviación angular en tiempo real. La formulación en espacio de estados se define como:

$$\dot{x} = Ax + Bu \quad (30)$$

Donde x representa el vector de estados compuesto por el ángulo θ y la velocidad angular ω , mientras que u es la señal de control aplicada por el motor. La linealización alrededor del punto de equilibrio proporciona una representación aproximada que facilita el análisis y diseño del controlador [2].

Para garantizar una respuesta rápida y precisa, se emplea una versión discreta del controlador PID, dada por la ecuación en diferencias 1.

La implementación en el microcontrolador ESP32-WROOM-32 requiere una integración eficiente del controlador PID con los módulos de adquisición de datos y control del motor. La estructura del software se divide en:

- Adquisición de datos del sensor MPU-6000 para estimar la inclinación y velocidad angular.
- Cálculo del error $e(k)$ y evaluación de la ecuación discreta del PID en cada ciclo de muestreo.
- Generación de una señal PWM para modular la velocidad del motor BLDC en función de la salida $u(k)$.
- Supervisión de la estabilidad del sistema mediante detección de valores extremos en θ y ω .

El código de control se diseñó con una arquitectura de ejecución en lazo cerrado, asegurando una actualización periódica de la señal de control con una frecuencia de muestreo de 1 kHz, suficiente para capturar la dinámica del sistema sin incurrir en aliasing [34].

V-L. Programación

El proyecto se basa en un sistema de cubo autoequilibrante, implementado en un microcontrolador ESP32-WROOM-32. El objetivo es mantener estable la posición del cubo en sus diferentes vértices, compensando inclinaciones en tiempo real. Para ello se integran:

- Sensores inerciales (MPU6000) para medir aceleración y velocidad angular.
- Un algoritmo de control basado en lógica PD con un término de integración interna.
- Accionadores (motores y drivers) manipulados mediante señales PWM.
- Comunicación inalámbrica por Bluetooth.

A continuación se describe la estructura de código distribuida en tres ficheros principales:

- ESP32.h
- ESP32_cube.ino
- functions.ino

Se explica cómo se inicializan los periféricos, se adquiere información de los sensores y se aplican las estrategias de control.

V-L1. Declaraciones y configuraciones en ESP32.h: En este fichero se definen constantes, pines y parámetros para el manejo de los motores y el sensor MPU6000, así como variables globales utilizadas en el control.

- Pines de control: Se asignan pines para dirección, PWM y freno de los tres motores.
- Parámetros del sensor: Dirección I2C del MPU6000 y registros para configurar acelerómetro y giróscopo.
- Variables globales:
 - Gyro_amount, alpha, utilizadas en el filtro complementario.
 - balancing_point, calibrated y vertical para la lógica de equilibrio.
 - K1, K2, K3, que conforman la estrategia de control sobre ángulo y velocidad.
 - motor_speed_X, motor_speed_Y para acumular la consigna de control (integración discreta).
- EEPROM_SIZE: Tamaño de la EEPROM para almacenamiento de calibraciones.

V-L2. Archivo principal ESP32_cube.ino: Aquí se definen las funciones setup y loop, donde:

- Se inicializan periféricos (Serial, Bluetooth, motores, EEPROM).
- Se configura el sensor MPU6000 (con angle_setup).
- Se desarrolla el ciclo de control: lectura de datos, cálculo de ángulo y generación de las señales PWM para los motores.

```

1  #define BUZZER      27
2  #define VBAT       34
3  #define BRAKE      26
4
5  #define DIR1       4
6  #define PWM1       32
7  #define PWM1_CH    1
8
9  #define DIR2       15
10 #define PWM2       25
11 #define PWM2_CH    0
12
13 #define DIR3       5
14 #define PWM3       18
15 #define PWM3_CH    2
16
17 #define TIMER_BIT   8
18 #define BASE_FREQ  20000
19
20 // Direcciones y registros especificos para MPU6000
21 #define MPU6000     0x68
22 #define ACCEL_CONFIG 0x1C
23 #define GYRO_CONFIG 0x1B
24 #define PWR_MGMT_1  0x6B
25 #define PWR_MGMT_2  0x6C
26
27 #define EEPROM_SIZE 64
28
29 float Gyro_amount = 0.1;
30
31 bool vertical = false;
32 bool calibrating = false;
33 bool calibrated = false;
34 int balancing_point = 0;
35
36 float K1 = 160;
37 float K2 = 10.50;
38 float K3 = 0.03;
39 int loop_time = 10;
40 int ID1;
41     float X1;
42     float Y1;
43     int ID2;
44     float X2;
45     float Y2;
46     int ID3;
47     float X3;
48     float Y3;
49     int ID4;
50     float X4;
51     float Y4;

```

Figura 25: Declaraciones y configuración de pines en ESP32 Parte 1.[Fuente: Autores]

```

1  struct OffsetsObj {
2  OffsetsObj offsets;
3
4  float alpha = 0.74;
5
6  int16_t AcX, AcY, AcZ, GyX, GyY, GyZ, gyroX, gyroY, gyroZ, gyroYfilt, gyroZfilt;
7  int16_t GyZ_offset = 0;
8  int16_t GyY_offset = 0;
9  int16_t GyX_offset = 0;
10
11 float robot_angleX, robot_angleY, angleX, angleY;
12 float Acc_angleX, Acc_angleY;
13 int32_t motor_speed_X;
14 int32_t motor_speed_Y;
15
16 long currentT, previousT_1, previousT_2 = 0;

```

Figura 26: Declaraciones y configuración de pines en ESP32 Parte 2.[Fuente: Autores]

V-L3. *Inicialización de periféricos:* Para inicializar los periféricos se realizan las siguientes acciones:

- Se inicia la comunicación Bluetooth con un nombre de dispositivo.
- Se habilita la EEPROM para leer y escribir offsets de calibración.
- Se configuran motores y freno, generando PWM a 20 kHz.

V-L4. *Ciclo principal:* En el ciclo principal se realizan las siguientes acciones:

- Cada loop_time milisegundos se efectúa la rutina de control.
- Tuning permite ajustar K1, K2, K3 mediante Bluetooth.
- angle_calc actualiza los ángulos usando datos de acelerómetro y giroscopio.
- Se determina el vértice actual y se corrige con los offsets adecuados.
- Se calcula la salida PWM, se integra en motor_speed_X e Y y se aplican las señales a los motores.

V-L5. *Funciones auxiliares en functions.ino:* El archivo functions.ino agrupa la implementación de funciones que abarcan lectura de sensores, control de motores y utilidades de calibración.

- Adquisición de datos del sensor:
 - angle_setup configura y calibra el MPU6000, calculando offsets del giroscopio.
 - angle_calc combina lecturas de giroscopio y acelerómetro, usando un filtro complementario para obtener ángulos precisos.
- Control de motores
 - XY_to_threeWay traduce la consigna de dos ejes a tres motores.
 - MotorX_control asigna dirección (HIGH/LOW) y PWM limitado a 255.

V-M. *Comunicación y ajuste en tiempo real*

En la función loop se invoca Tuning, que procesa bytes recibidos vía BluetoothSerial. Según los comandos, se ajustan K1, K2 y K3 o se inicia la calibración de cada vértice. Así se logra:

- Ajustar parámetros sin recompilar el firmware.
- Calibrar posición para distintos apoyos.
- Mostrar mensajes de depuración (por ejemplo, si falta calibrar).

```

1  #include "ESP32.h"
2  #include <Wire.h>
3  #include <EEPROM.h>
4  #include "BluetoothSerial.h"
5
6  BluetoothSerial SerialBT;
7
8  void setup() {
9      Serial.begin(115200);
10     SerialBT.begin("ESP32-Cube-blue");
11     EEPROM.begin(EEPROM_SIZE);
12     pinMode(BUZZER, OUTPUT);
13     pinMode(BRAKE, OUTPUT);
14     digitalWrite(BRAKE, HIGH);
15     pinMode(DIR1, OUTPUT);
16     ledcSetup(PWM1_CH, BASE_FREQ, TIMER_BIT);
17     ledcAttachPin(PWM1, PWM1_CH);
18     Motor1_control(0);
19     pinMode(DIR2, OUTPUT);
20     ledcSetup(PWM2_CH, BASE_FREQ, TIMER_BIT);
21     ledcAttachPin(PWM2, PWM2_CH);
22     Motor2_control(0);
23     pinMode(DIR3, OUTPUT);
24     ledcSetup(PWM3_CH, BASE_FREQ, TIMER_BIT);
25     ledcAttachPin(PWM3, PWM3_CH);
26     Motor3_control(0);
27     EEPROM.get(0, offsets);
28     if (offsets.ID1 == 99 && offsets.ID2 == 99 && offsets.ID3 == 99 && offsets.ID4 == 99)
29         calibrated = true;
30     else calibrated = false;
31     delay(2000);
32     digitalWrite(BRAKE, HIGH);
33     delay(70);
34     digitalWrite(BRAKE, LOW);
35     angle_setup();
36 }
37
38     Tuning();
39     angle_calc();
40     if (balancing_point == 1) { angleX -= offsets.X1; angleY -= offsets.Y1; if (abs(angleX)
41         > 8 || abs(angleY) > 8) vertical = false; }
42     else if (balancing_point == 2) { angleX -= offsets.X2; angleY -= offsets.Y2; if (abs(
43         angleY) > 5) vertical = false; }
44     else if (balancing_point == 3) { angleX -= offsets.X3; angleY -= offsets.Y3; if (abs(
45         angleY) > 5) vertical = false; }
46     else if (balancing_point == 4) { angleX -= offsets.X4; angleY -= offsets.Y4; if (abs(
47         angleX) > 5) vertical = false; }
48     if (abs(angleX) < 8 || abs(angleY) < 8) { Gyro_amount = 0.996; } else { Gyro_amount =
49         0.1; }
50     if (vertical && calibrated && !calibrating) {
51         digitalWrite(BRAKE, HIGH);
52         gyroZ = GyZ / 131.0; gyroY = GyY / 131.0; gyroX = GyX / 131.0;
53         gyroYfilt = alpha * gyroY + (1 - alpha) * gyroYfilt;
54         gyroZfilt = alpha * gyroZ + (1 - alpha) * gyroZfilt;
55         int pwm_X = constrain(K1 * angleX + K2 * gyroZfilt + K3 * motor_speed_X, -255, 255);
56         int pwm_Y = constrain(K1 * angleY + K2 * gyroYfilt + K3 * motor_speed_Y, -255, 255);
57         motor_speed_X += pwm_X;
58         motor_speed_Y += pwm_Y;

```

Figura 27: ESP32 Cube: Configuración inicial y motores. [Fuente: Autores]

```

1      if (balancing_point == 1) { XY_to_threeWay(-pwm_X, -pwm_Y); }
2      else if (balancing_point == 2) { Motor1_control(pwm_Y); }
3      else if (balancing_point == 3) { Motor2_control(-pwm_Y); }
4      else if (balancing_point == 4) { Motor3_control(pwm_X); }
5      } else {
6          XY_to_threeWay(0, 0);
7          digitalWrite(BRAKE, LOW);
8          motor_speed_X = 0;
9          motor_speed_Y = 0;
10     }
11     previousT_1 = currentT;
12 }
13 }
14 if (currentT - previousT_2 >= 2000) {
15     battVoltage((double)analogRead(VBAT) / 207);
16     if (!calibrated && !calibrating) { SerialBT.println("first_you_need_to_calibrate_the_
17         balancing_points..."); }
18     previousT_2 = currentT;
19 }

```

Figura 28: ESP32 Cube: Final del bucle principal y control de batería.[Fuente: Autores]

VI. RESULTADOS

VI-A. Diseño e Implementación de la Tarjeta PCB para el Control del Sistema

Para garantizar un sistema de control eficiente y con una integración electrónica compacta, se diseñó una tarjeta PCB que permitió la conexión de todos los componentes del sistema. La tarjeta incluyó el microcontrolador ESP32-WROOM-32, el sensor giroscópico MPU-6000, los circuitos de alimentación y los puertos de conexión para la interfaz del motor y el controlador VESC, como se evidencia en la figura 29.

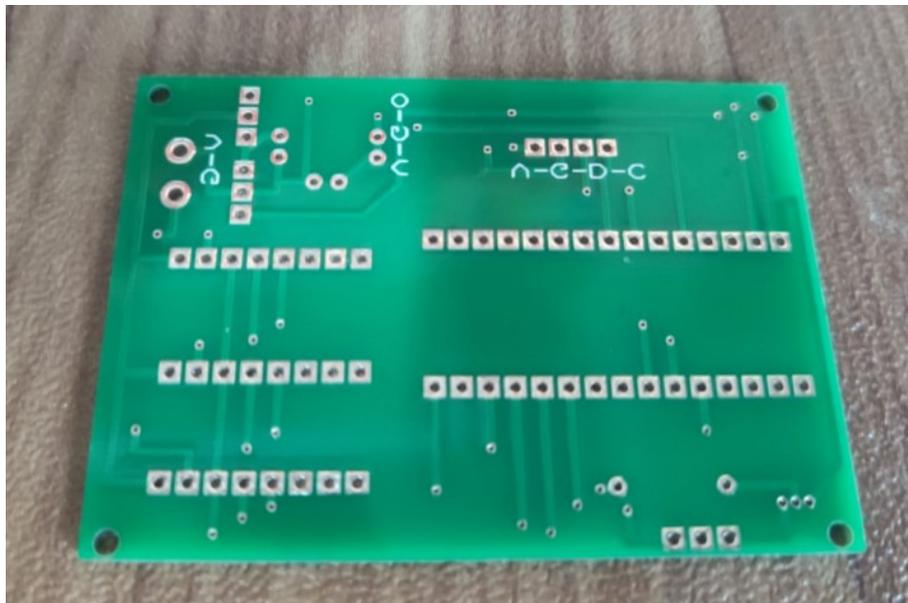


Figura 29: Placa electrónica PCB superior [Fuente: Autores].

El diseño de la PCB fue mejorado para reducir interferencias eléctricas y reducir el tamaño del sistema, asegurando una integración eficiente dentro de la estructura del cubo. La implementación de un diseño multicapa permitió la

VI-B. Implementación del Control de Lazo Cerrado en el Microcontrolador

El control del cubo autoequilibrante se basó en la implementación de un sistema de lazo cerrado, donde la señal del giroscopio fue procesada en tiempo real por el ESP32-WROOM-32 para generar la respuesta de control necesaria en el motor. Se utilizó un controlador PID para ajustar la señal de salida y compensar las desviaciones del sistema con base en la inclinación detectada por el MPU-6000 .

Durante la ejecución del algoritmo de control, se verificó que la adquisición de datos del sensor giroscópico se realizaba con una frecuencia de muestreo adecuada, permitiendo un procesamiento rápido y preciso de la información. La comunicación entre el MPU-6000 y el ESP32-WROOM-32 se estableció mediante la interfaz I²C, asegurando tiempos de respuesta mínimos y evitando pérdida de datos.

Se evaluaron diferentes configuraciones del controlador PID, ajustando los parámetros de ganancia proporcional (K_1), integral (K_2) y derivativa (K_3) para obtener la mejor respuesta dinámica del sistema. Los resultados indicaron que con una configuración adecuada, el sistema lograba corregir desviaciones con una respuesta estable, reduciendo oscilaciones y evitando sobrepasos no deseados.

VI-C. Validación del Funcionamiento y Evaluación del Tiempo de Respuesta

Para evaluar el desempeño del sistema, se realizaron pruebas experimentales en las que se midió la capacidad de estabilización del cubo en una arista y el tiempo de respuesta ante perturbaciones externas. Se aplicaron inclinaciones controladas al sistema y se registró el tiempo que tardaba en recuperar su posición de equilibrio.

Los resultados indicaron que el sistema lograba estabilizarse en un tiempo inferior a 2 segundos, lo que demuestra una respuesta rápida y eficiente del controlador PID. Además, se comprobó que el sistema mantenía el equilibrio en una arista sin desviaciones significativas durante periodos prolongados, lo que confirma la efectividad del control implementado.

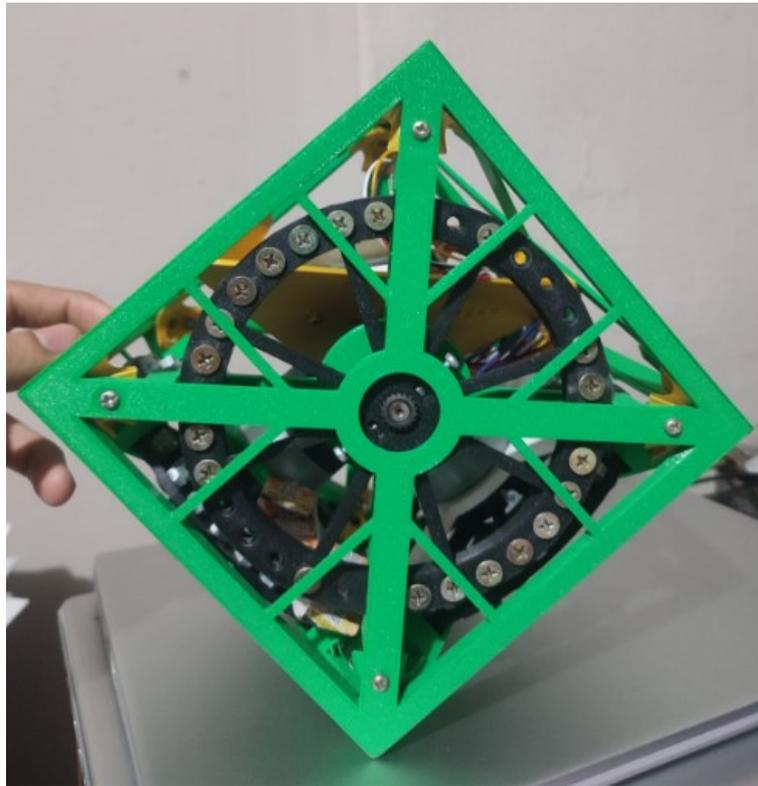


Figura 32: Microcontrolador ESP32 [23].

También se analizaron los efectos de variaciones en la carga del motor y en la velocidad de respuesta del sistema, demostrando que el controlador PID ajustado correctamente permitía una compensación eficiente ante perturbaciones externas. La capacidad de adaptación del sistema a diferentes condiciones operativas validó la eficacia del enfoque de control propuesto como se muestra en la figura 32.

VII. CRONOGRAMA

A continuación se muestra el cronograma de trabajo en la tabla I.

Tabla I: Cronograma

Actividad	Responsable	Mes										
		1	2	3	4	5	6	7	8	9	10	
Definición del alcance del proyecto	Carpio-Fernández	X										
Problema e investigación preliminar	Carpio-Fernández	X	X									
Objetivos generales y específicos	Carpio-Fernández		X									
Marco teórico	Carpio-Fernández		X	X								
Metodología	Carpio-Fernández			X								
Diseño estructural y modelado	Carpio-Fernández				X	X						
Desarrollo del software de control	Carpio-Fernández				X	X						
Simulación en software (MATLAB)	Carpio-Fernández					X	X					
Implementación del prototipo físico	Carpio-Fernández						X	X				
Pruebas funcionales del prototipo	Carpio-Fernández							X	X			
Validación del sistema y ajustes finales	Carpio-Fernández								X	X		
Redacción del informe final	Carpio-Fernández									X	X	

En la siguiente tabla se detallo los gatos para este proyecto.

VIII. PRESUPUESTO

Tabla II: Presupuesto

Nombre del elemento	Descripción	Cantidad	Valor total
Piezas Impresas en 3D	Estructuras, Soportes y Acoples.	1	248 \$
Motor brushless	Nidec 24H	3	45 \$
PCB Fabricada	PCBWay-32	1	15 \$
Acelerometro	MPU-6050 GY-521	1	15 \$
Filamento PETG	Sunlu 1 kg	3	60 \$
Tornillos	2mm	50	12.5 \$
Tuercas	2mm	100	15 \$
MicroControlador	ESP 32-WROOM-32	1	35 \$
Cables	#17	3m	5 \$
Horas trabajadas en ensamblaje	Armado de piezas 3d	30	60 \$
Horas trabajadas en conexion	Soldar y conectar	30	60 \$
Horas trabajadas en pruebas	Implementación final	40	80 \$
TOTAL			675.5\$

IX. CONCLUSIONES

Considerando en el desarrollo del proyecto, se logró implementar con éxito un sistema de autoequilibrio mediante el uso de un controlador PID en un microcontrolador. La correcta calibración de los parámetros de control permitió que el cubo mantuviera su estabilidad en una arista con tiempos de respuesta adecuados ante perturbaciones externas. La integración del sensor giroscópico MPU-6000 y el ESP32-WROOM-32 fue fundamental para obtener datos precisos en tiempo real, asegurando la estabilidad del sistema.

Otro logro significativo fue la implementación de la comunicación eficiente entre los componentes del sistema. La interacción entre el microcontrolador y los sensores mediante el protocolo I²C garantizó una adquisición de datos estable, con baja latencia en la transmisión de información. Además, la integración con el controlador de motor VESC permitió un control preciso del torque y la velocidad, mejorando el comportamiento dinámico del sistema.

Se logró la fabricación e integración de la tarjeta PCB diseñada específicamente para este proyecto. Este diseño permitió la distribución eficiente de los circuitos de alimentación, control y comunicación, reduciendo interferencias y mejorando la respuesta del sistema. La correcta disposición de los componentes en la PCB contribuyó a reducir el tamaño del sistema sin comprometer su funcionalidad, facilitando su ensamblaje y pruebas experimentales.

La elección estratégica de los materiales y componentes jugó un papel fundamental en el ajuste de ubicación del centro de masa y el desempeño del sistema. El uso del motor BLDC Nidec 24H proporcionó una respuesta rápida y eficiente, mientras que la estructura impresa en 3D con PLA permitió una reducción de peso sin comprometer la rigidez del cubo. Esta selección de materiales permitió obtener un sistema funcional con características mecánicas adecuadas para su aplicación.

El desarrollo e implementación del cubo autoequilibrante permitieron alcanzar los objetivos planteados. Se diseñó una tarjeta PCB funcional que permitió la integración del microcontrolador y los sensores dentro del sistema. La implementación del control en lazo cerrado aseguró la estabilidad del cubo en una arista, procesando la información del giroscopio en tiempo real. Finalmente, la evaluación del tiempo de respuesta y desempeño confirmó que el sistema lograba mantener su equilibrio de manera efectiva y sin oscilaciones excesivas.

Estos resultados demuestran la viabilidad del enfoque adoptado en el proyecto, validando el uso de un controlador PID en un sistema de autoequilibrio basado en microcontroladores y sensores inerciales.

Adicionalmente, el sistema demostró ser eficiente en términos energéticos. Se comprobó que el consumo promedio del ESP32-WROOM-32, junto con el motor y el sensor giroscópico, se mantuvo dentro de los límites operativos deseados, permitiendo una autonomía de varias horas con una batería de 2200 mAh. Esto valida la viabilidad del sistema en aplicaciones experimentales sin necesidad de una fuente de alimentación externa constante.

X. RECOMENDACIONES

Es fundamental realizar una calibración precisa de los sensores utilizados en el sistema, asegurando que los valores obtenidos sean fiables y representativos del comportamiento real del cubo autoequilibrante. En particular, el giroscopio MPU-6000 debe configurarse adecuadamente para reducir el ruido en la medición de la velocidad angular y garantizar una respuesta óptima del controlador PID. Asimismo, es recomendable establecer un filtro digital adecuado para eliminar perturbaciones y mejorar la precisión de los datos obtenidos.

Durante la implementación del sistema, se observó que la comunicación entre el microcontrolador ESP32-WROOM-32 y los módulos periféricos podría volverse más eficiente utilizando técnicas de sincronización en la transmisión de datos. Para evitar retrasos en la lectura de los sensores y mejorar la respuesta del sistema, se sugiere el uso de interrupciones en lugar de consultas periódicas, especialmente en la lectura del giroscopio y la ejecución de las rutinas de control.

Para lograr un ensamblaje eficiente del cubo, se recomienda verificar la distribución de los componentes electrónicos dentro de la estructura. La colocación del PCB debe realizarse de manera que reduzca la interferencia con otros módulos y garantice una correcta conexión con el motor y la batería. Además, es importante asegurar la fijación de los cables para evitar movimientos bruscos que puedan afectar el desempeño del sistema durante su operación.

Se sugiere mejorar la estabilidad mecánica del sistema mediante el ajuste de la rigidez estructural del cubo. La impresión 3D en PLA ha demostrado ser una solución efectiva para la construcción de la carcasa; sin embargo, es recomendable realizar pruebas con otros materiales o reforzar puntos críticos de la estructura con soportes adicionales. De este modo, se reducirá la posibilidad de deformaciones y se garantizará un mejor desempeño en condiciones de operación exigentes.

Para prevenir posibles fallos en la transmisión de señales de control, se recomienda evaluar la implementación de una capa de protección adicional en las conexiones eléctricas del sistema. El uso de recubrimientos aislantes y la disposición óptima de los conductores contribuirán a reducir interferencias electromagnéticas, mejorando la confiabilidad del sistema de control.

Adicionalmente, se recomienda realizar pruebas adicionales de validación del sistema, considerando diferentes escenarios de operación. Evaluar la respuesta del controlador PID ante variaciones en la inclinación inicial del cubo y diferentes condiciones de carga permitirá mejorar el ajuste de los parámetros de control y mejorar su rendimiento en aplicaciones futuras.

REFERENCIAS

- [1] M. González y J. Martínez, «Impacto de las nuevas tecnologías en la educación en ingeniería,» *Revista de Innovación Educativa*, vol. 15, n.º 2, págs. 45-58, 2020.
- [2] G. F. Franklin, J. D. Powell y A. Emami-Naeini, *Feedback Control of Dynamic Systems*. Pearson, 2015.
- [3] I. R. M. Association, *Virtual Reality in Education: Breakthroughs in Research and Practice*. Hershey, PA: IGI Global, 2019, ISBN: 9781522592819.
- [4] L. Ramírez y D. Pérez, «The Importance of Repeatability in Educational Engineering Systems,» *Journal of Mechanical Engineering Education*, vol. 22, n.º 4, págs. 299-312, 2017.
- [5] G. M. (Gajan), *Cubli: A Cube that Balances on its Edge or Corner*, YouTube video, Accessed: 2024-11-12, 2013. dirección: https://www.youtube.com/watch?v=n_6p-1J551Y&ab_channel=GajamohanMohanarajah%28Gajan%29.
- [6] D. G. Luenberger, *Introduction to Dynamic Systems: Theory, Models, and Applications*. New York, USA: John Wiley Sons, 1979, ISBN: 9780471025948.
- [7] K. J. Åström y R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*, 2nd. Princeton, NJ: Princeton University Press, 2012, ISBN: 9780691152004.
- [8] K. Ogata, *Ingeniería de Control Moderna*, 6th. Madrid, España: Pearson Educación, 2019, ISBN: 9788490357313.
- [9] T. Kailath, *Linear Systems*. Englewood Cliffs, NJ: Prentice Hall, 1980, ISBN: 9780135369616.
- [10] K. Ogata, *Ingeniería de control moderna*, 5.^a ed. Madrid, España: Pearson Educación, 2010, ISBN: 978-84-8322-660-5.
- [11] T. InvenSense, *MPU-6000 and MPU-6050 Product Specification*, Último acceso: 1 de febrero de 2025, 2015. dirección: <https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>.
- [12] T. InvenSense, *MEMS Motion Sensors - Applications and Technologies*, Último acceso: 1 de febrero de 2025, 2020. dirección: <https://www.invensense.tdk.com/mems-motion-sensors>.
- [13] S. J. Chapman, *Electric Machinery Fundamentals*, 4th. New York, USA: McGraw-Hill, 2004, ISBN: 9780072465235.
- [14] N. Corporation, *High-Performance DC Motors for Industrial Applications*, Último acceso: 1 de febrero de 2025, 2025. dirección: <https://www.nidec.com>.
- [15] K. Ogata, *Modern Control Engineering*, 5th. Upper Saddle River, NJ, USA: Prentice Hall, 2010, ISBN: 978-0136156734. dirección: <https://www.pearson.com/en-us/subject-catalog/p/modern-control-engineering/P200000003333/9780136156734>.
- [16] N. Corporation, *Nidec 24H DC Motors - Specifications and Performance*, Último acceso: 1 de febrero de 2025, 2025. dirección: <https://www.nidec.com/products/dc-motors-24H>.
- [17] E. Systems, *ESP32 Series Datasheet*, 2021. dirección: <https://www.espressif.com/en/products/socs/esp32>.
- [18] D. Lee y J. Park, «Performance Analysis of ESP32 in IoT Applications,» *Journal of Embedded Systems*, 2019.
- [19] K. Mohr y T. Schneider, «Low-Power Techniques in ESP32 for Embedded IoT Devices,» *IEEE Transactions on Power Electronics*, vol. 33, n.º 5, págs. 1205-1217, 2018.
- [20] A. Mohamed y R. El-Sayed, «IoT-based Smart Monitoring and Control Systems using ESP32,» *International Journal of Embedded Systems*, vol. 45, n.º 3, págs. 215-230, 2021.
- [21] R. Gupta y A. Sharma, «Wi-Fi Protocols for IoT: Analysis and Performance,» *IEEE IoT Journal*, 2020.
- [22] Y. Chen y L. Wang, «Real-Time Control Systems Using ESP32 in Robotics Applications,» *IEEE Transactions on Industrial Electronics*, vol. 67, n.º 9, págs. 7563-7572, 2020.
- [23] S. Pieters, *ESP32 Pinout Reference Guide*, Último acceso: 4 de febrero de 2025, 2024. dirección: <https://www.studiopieters.nl/esp32-pinout/>.
- [24] R. C. Dorf y R. H. Bishop, *Modern Control Systems*, 12th. Prentice Hall, 2011.
- [25] J. G. Ziegler y N. B. Nichols, «Optimum settings for automatic controllers,» *Transactions of the ASME*, vol. 64, n.º 8, págs. 759-768, 1942.
- [26] K. J. Åström y T. Hägglund, *PID controllers: theory, design, and tuning*. Instrument Society of America, 1995.

- [27] Y. Gao y S. Wang, «Evolutionary algorithms for PID controller tuning: A review,» *Applied Soft Computing*, vol. 8, n.º 4, págs. 1469-1481, 2008.
- [28] L. Schmidt y P. Conrad, «Advancements in Reaction Wheel Systems for Spacecraft Attitude Control,» *Acta Astronautica*, vol. 145, págs. 12-19, 2017.
- [29] R. C. Dorf y R. H. Bishop, *Modern Control Systems*, 12th. Upper Saddle River, NJ: Prentice Hall, 2011, ISBN: 9780136024583.
- [30] J.-J. E. Slotine y W. Li, *Applied Nonlinear Control*. Prentice Hall, 1991.
- [31] J. Furusho y M. Sakaguchi, «Control of an Inverted Pendulum Type Robot with a Reaction Wheel,» *IEEE Transactions on Robotics and Automation*, vol. 16, n.º 6, págs. 800-805, 2000.
- [32] M. W. Spong, S. Hutchinson y M. Vidyasagar, *Robot Modeling and Control*. Wiley, 2006.
- [33] Y.-J. Kim y S.-I. Oh, «Control of Balancing Robots: Theory and Applications,» *International Journal of Advanced Robotic Systems*, vol. 11, n.º 5, págs. 150-160, 2014. DOI: 10.5772/58975.
- [34] J. Wu, L. Zhang y M. Huang, «Self-Balancing Systems in Robotics: A Comprehensive Review,» *Robotics and Autonomous Systems*, vol. 75, págs. 101-115, 2016. DOI: 10.1016/j.robot.2016.01.003.
- [35] E. Systems, *ESP Development Kits*, Accessed: 2024-11-04, 2024. dirección: <https://www.espressif.com/en/products/devkits>.
- [36] D. Systèmes, *SolidWorks: 3D CAD Design and Engineering Solutions*, Último acceso: 1 de febrero de 2025, 2025. dirección: <https://www.solidworks.com/>.
- [37] MathWorks, *MATLAB and Simulink for Engineers*, Último acceso: 1 de febrero de 2025, 2025. dirección: <https://www.mathworks.com/products/matlab.html>.
- [38] E. Systems, *ESP32-WROOM-32 Datasheet*, Espressif Systems, 2023. dirección: <https://www.espressif.com>.
- [39] N. Corporation, *Nidec Corporation - Precision Motors and Manufacturing*, Último acceso: 1 de febrero de 2025, 2025. dirección: <https://www.nidec.com>.
- [40] InvenSense, *MPU-6000 and MPU-6050 Product Specification*, InvenSense, 2016. dirección: <https://www.invensense.com>.
- [41] A. R. Hambley, *Electrical Engineering: Principles and Applications*, 7th. Boston, MA: Pearson, 2018, ISBN: 9780134484143.
- [42] S. Ferrari, M. Herrero y R. Gomez, «The Role of Sensors in Control Systems for Industrial Applications,» *Journal of Control Engineering Practice*, vol. 27, n.º 4, págs. 412-426, 2019. DOI: 10.1016/j.conengprac.2019.03.004.
- [43] H. Becker, *Embedded Systems in Robotics: Practical Applications*. Berlin, Germany: Springer, 2018, ISBN: 9783319785643.
- [44] G. Huang, X. Li y P. Chen, «Embedded Systems for Motion Control: Design and Applications,» *Journal of Control Engineering Practice*, vol. 45, págs. 64-74, 2015. DOI: 10.1016/j.conengprac.2015.01.002.
- [45] J. Park y H. Kim, «Applications of Gyroscopes and Accelerometers in Control Systems,» *International Journal of Mechatronics and Automation*, vol. 6, n.º 2, págs. 85-92, 2017. DOI: 10.1504/IJMA.2017.100136.
- [46] R. Rem y contributors, *Self-Balancing Cube: An Open Source Project*, Último acceso: 2024-02-25, 2023. dirección: <https://github.com/remrc/Self-Balancing-Cube>.
- [47] G. F. Franklin, J. D. Powell y M. L. Workman, *Digital Control of Dynamic Systems*, 3rd. Addison-Wesley, 1998.
- [48] K. Ogata, *Modern Control Engineering*, 5th. Prentice Hall, 2010.
- [49] K. J. Åström y R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 2008.

ANEXO A
PROTOTIPO INICIAL



Figura 33: Prototipo inicial para próximas mejoras [Fuentes: Autores]

ANEXO B
PROTOTIPO FINAL

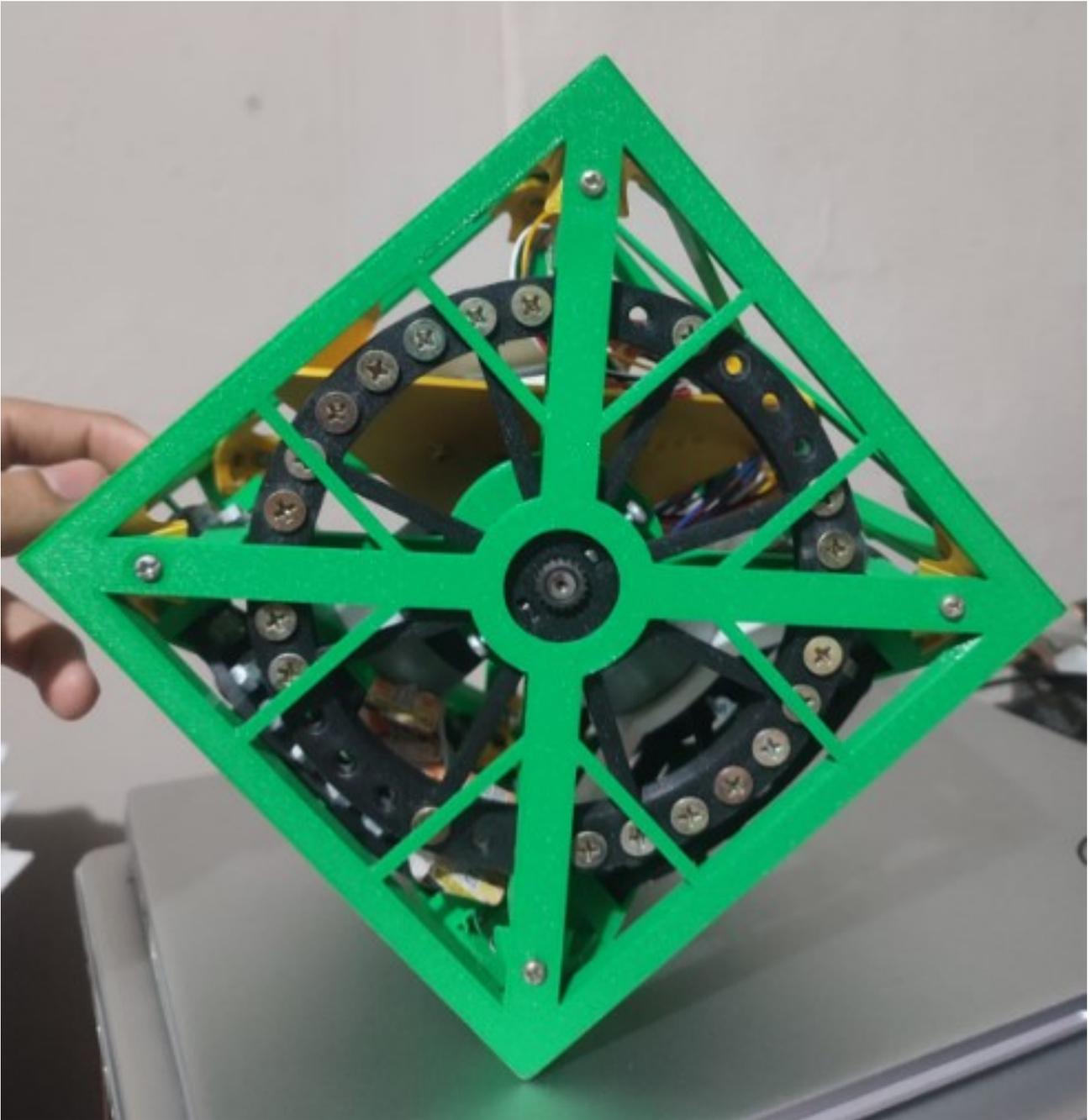


Figura 34: Prototipo final operativo Frontal [Fuentes: Autores]

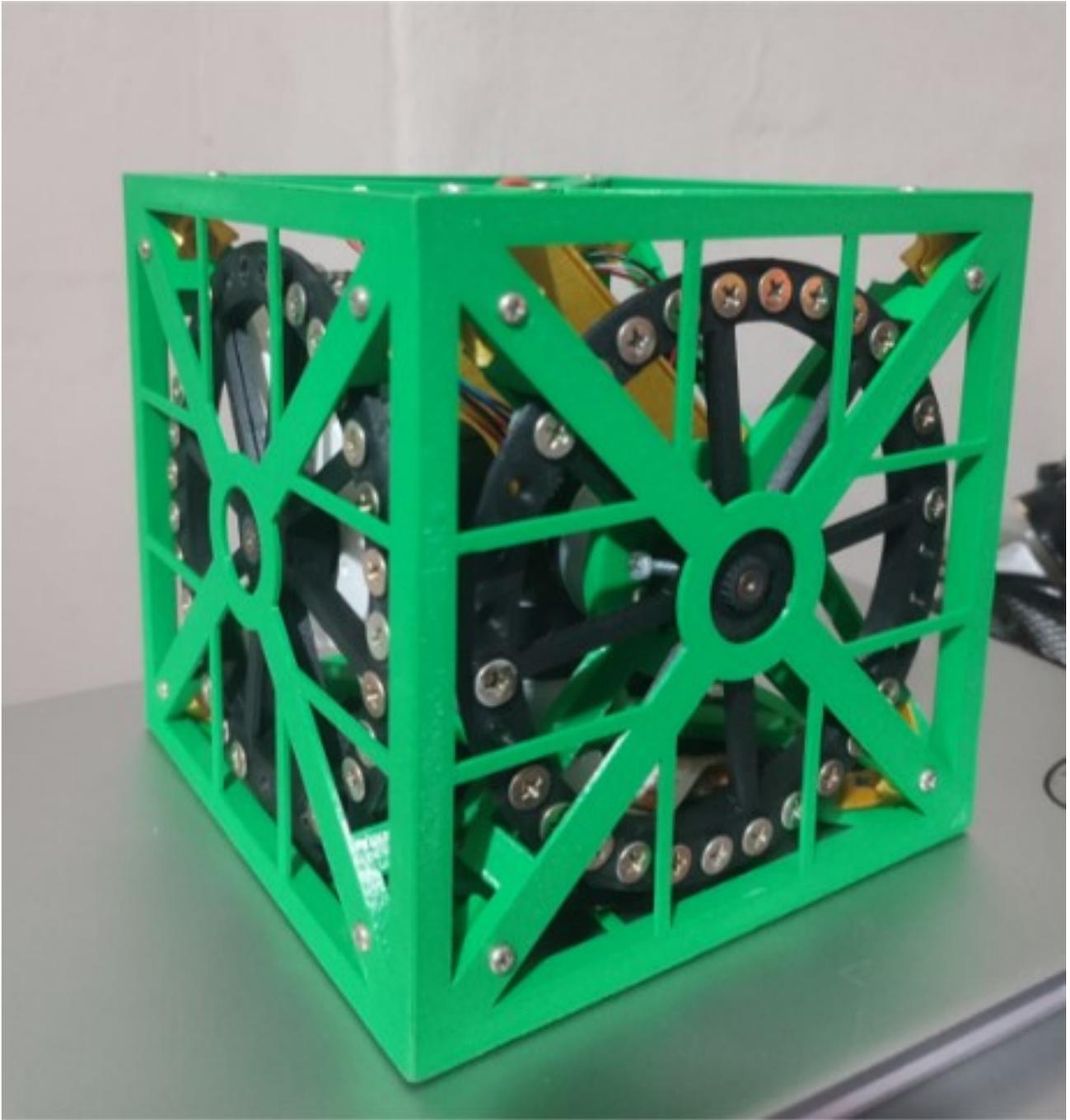


Figura 35: Prototipo final operativo lateral[Fuentes: Autores]

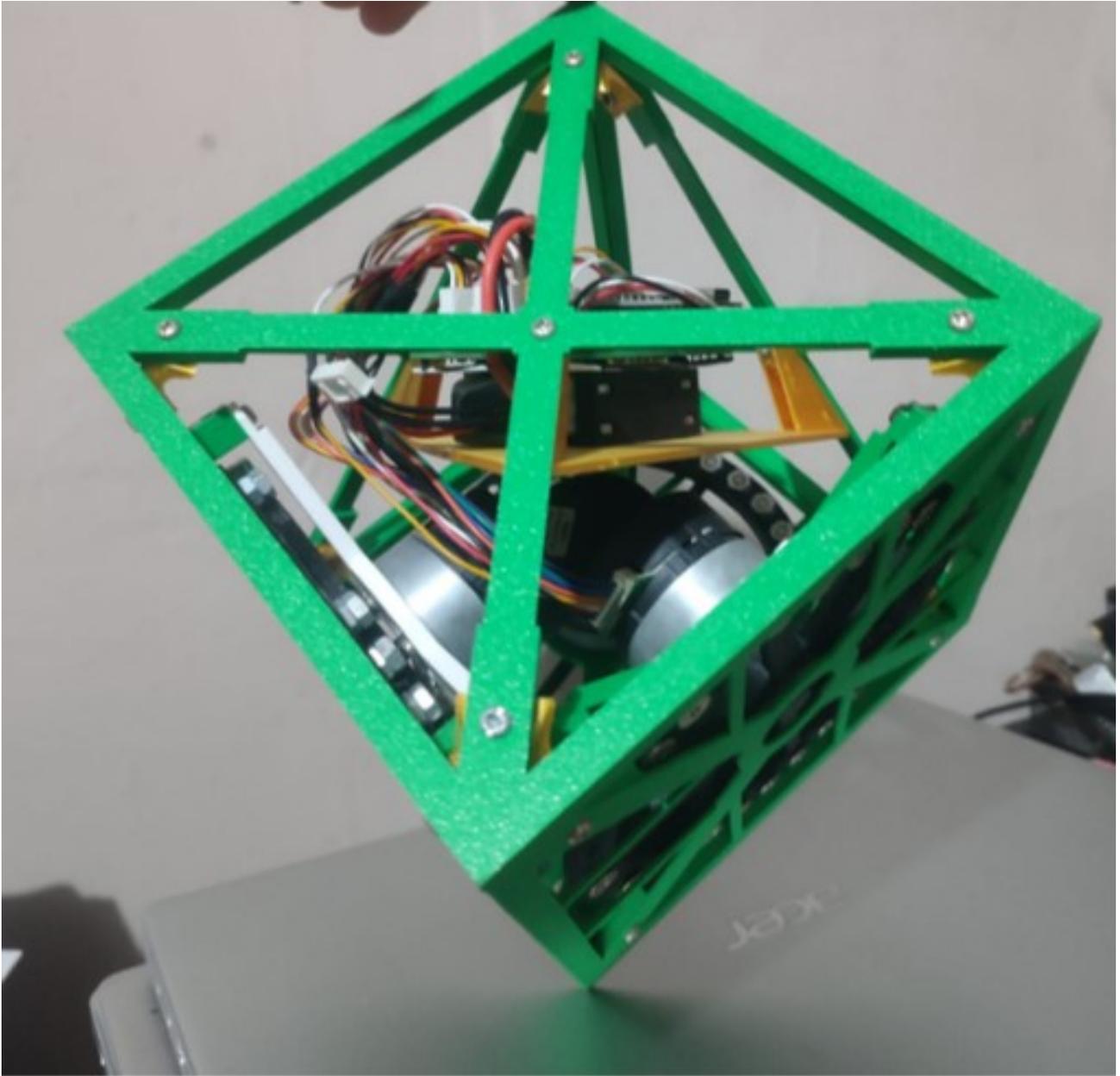


Figura 36: Prototipo final operativo parado[Fuentes: Autores]



Figura 37: Disco de inercia [Fuentes: Autores]



Figura 38: Pared de prototipo [Fuentes: Autores]

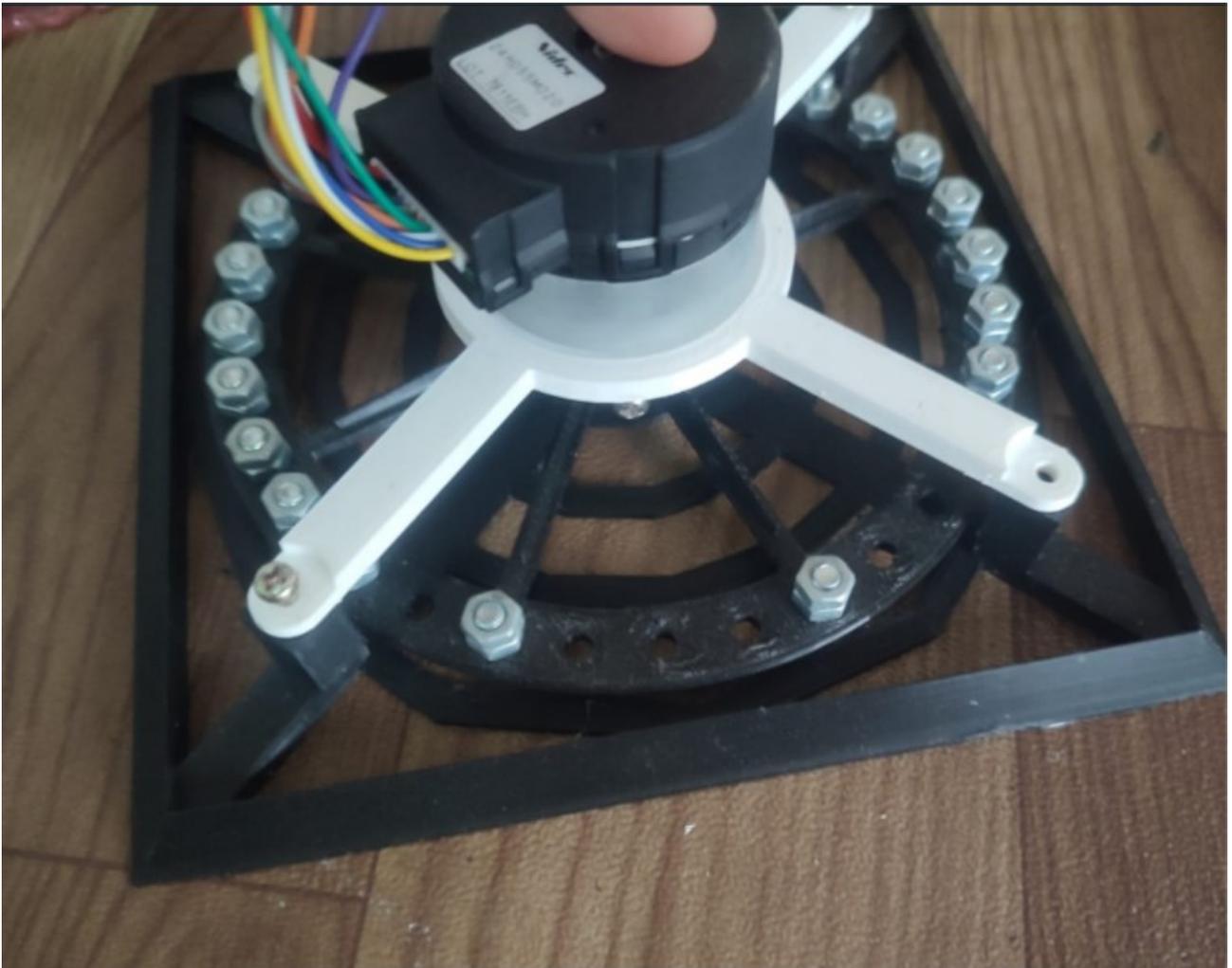


Figura 39: Pared con soporte de motor. [Fuentes: Autores]