



DEPARTAMENTO DE
**INGENIERÍA
MECÁNICA**



**Universidad de Santiago de Chile
Departamento de Ingeniería Mecánica
Ingeniería Civil Mecatrónica**

Laboratorio Fundamentos de Robótica: Robot Móvil

Tomas Araya
Fabián Andrés Miranda Aedo
Tomas Montaner
Profesor: Arturo Alvarez Cea



10 de diciembre de 2024

Índice

Índice	1
1. Introducción	2
2. Objetivo General	2
2.1. Objetivos Específicos	2
3. Marco Teórico	3
3.1. Robots Móviles Terrestres	3
3.2. Configuración diferencial de robot móvil con ruedas diferenciales	3
3.3. Control PID	4
3.4. Sensores Inerciales IMU	5
3.5. Sensores Infrarrojos de Proximidad	6
3.6. Conexión Bluetooth	7
3.7. Motor Paso a Paso 17HS4401S	8
3.8. Driver A4988	9
4. Presentación del Problema	10
5. Solución del Problema	11
5.1. Diagrama de Conexiones	12
6. Resultados	14
7. Conclusión	16
8. Referencias	17
Anexo I	18
Anexo II	18
Anexo III	18

1. Introducción

El presente informe documenta la implementación del Laboratorio 2 correspondiente al curso de Fundamentos de Robótica. En este proyecto se diseñó y desarrolló un robot móvil autoequilibrado de dos ruedas, cuya funcionalidad principal es mantener el equilibrio dinámico y seguir líneas para desplazarse en un espacio controlado.

El robot integra una combinación de sensores y actuadores conectados a un controlador central para garantizar un desempeño óptimo. Entre los sensores utilizados se encuentran módulos infrarrojos para detección de líneas y un sensor MPU6050 para medir la inclinación y las aceleraciones del sistema. Los actuadores principales son motores paso a paso controlados mediante drivers A4988, los cuales permiten un movimiento preciso y controlado.

El control del sistema se realiza mediante un controlador PID (Proporcional, Integral, Derivativo con el fin de generar un control autónomo preciso.

2. Objetivo General

Diseñar, construir e implementar un robot autónomo balanceador y seguidor de líneas, equipado con Bluetooth para el control remoto, que combine sistemas de control dinámico, sensores infrarrojos y que sea capaz de desplazarse mediante movimientos planificados y manuales, con el fin de realizar un recorrido competitivo por un circuito con irregularidades en el terreno.

2.1. Objetivos Específicos

- Diseñar y construir físicamente un chasis robótico, fabricado en base a impresión 3D, que permita:
 - Integración de un sensor Inercial (MPU6050).
 - Integración de sensor Infrarrojo (IR TCRT5000).
 - Instalación y conexión de actuadores motores (Paso a Paso).
 - Instalación y conexión Driver para control de motores paso a paso.
 - Instalación y conexión de hardware de control a Arduino.
 - Instalación y conexión de Modulo de comunicación Bluetooth.
 - Instalación y conexión de fuente de energía (batería o pilas), para energizar componentes electrónicos.
- Programación del sistema de control de estabilidad y movimiento, para Arduino.
- Desarrollo de Pruebas de movimiento estable en recorrido de circuito de línea negra.

3. Marco Teórico

3.1. Robots Móviles Terrestres

Los robots móviles terrestres son sistemas robóticos diseñados para desplazarse en el suelo utilizando diversas configuraciones de locomoción. Estos robots tienen aplicaciones en múltiples campos, como logística, exploración, asistencia y educación. La movilidad terrestre se logra mediante el uso de ruedas, orugas o patas, dependiendo del terreno y las necesidades específicas del sistema.

3.2. Configuración diferencial de robot móvil con ruedas diferenciales

La configuración diferencial es una de las más comunes en robots móviles. Este diseño utiliza dos ruedas motrices controladas independientemente para dirigir y propulsar el robot. La diferencia en la velocidad de rotación de las ruedas permite giros y movimientos precisos. La cinemática de esta configuración puede describirse mediante las siguientes ecuaciones:

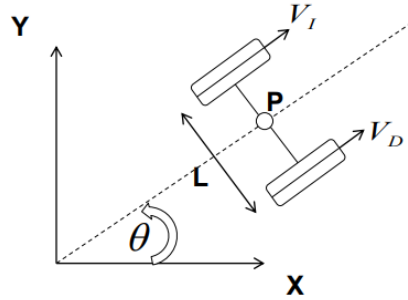


Figura 1: Modelo cinemático diferencial

Velocidades lineales de las ruedas

$$V_D = r_D \cdot \omega_D \quad V_I = r_I \cdot \omega_I \quad (1)$$

Velocidad instantánea del robot

$$v = \frac{V_D + V_I}{2} \quad (2)$$

Cambio de posición para tiempos pequeños

$$\dot{x} = x + V \cdot \cos(\theta) \cdot t \quad \dot{y} = y + V \cdot \sin(\theta) \cdot t \quad (3)$$

Velocidad angular del robot

$$\omega = \frac{V_D - V_I}{L} \quad (4)$$

Nueva orientación del robot

$$\dot{\theta} = \theta + \omega \cdot t \quad (5)$$

$$\begin{bmatrix} x(t + \Delta t) \\ y(t + \Delta t) \\ \theta(t + \Delta t) \end{bmatrix} \approx \begin{bmatrix} x(t) + v(t)\Delta t \cos(\theta(t)) \\ y(t) + v(t)\Delta t \sin(\theta(t)) \\ \theta(t) + \omega(t)\Delta t \end{bmatrix}$$

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} \cos(\theta(t)) & 0 \\ \sin(\theta(t)) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix}$$

3.3. Control PID

El controlador PID (Proporcional, Integral y Derivativo) es un tipo de control utilizado en sistemas de control automático por su simplicidad y efectividad. Este controlador ajusta la salida del sistema para que siga de manera precisa una referencia deseada, minimizando el error entre entrada y salida. La ecuación general del controlador es:

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt}$$

donde:

- $u(t)$ es la señal de control.
- $e(t)$ es el error.
- K_p, K_i, K_d son las constantes de ganancia proporcional, integral y derivativa, respectivamente.

1. Control proporcional (P):

Proporcional al error actual. Este término responde al valor instantáneo del error y busca corregirlo multiplicándolo por una constante proporcional K_p .

2. Control derivativo (D):

Responde a la tasa de cambio del error, anticipándose a cómo evolucionará. Este término suaviza la respuesta del sistema multiplicando la derivada del error por una constante derivativa K_d .

3. Control integrativo (I):

Acumula el error a lo largo del tiempo para corregir errores residuales. Este término elimina el error en estado estacionario al sumar las desviaciones acumuladas multiplicadas por una constante integral K_i .

3.4. Sensores Inerciales IMU

Los sensores inerciales IMU (Unidad de Medida Inercial) combinan acelerómetros y giroscopios para medir aceleraciones y velocidades angulares en tres ejes (X, Y y Z). Estos sensores son esenciales para la estabilización de robots, ya que permiten calcular la orientación del robot utilizando el filtro de Kalman o el filtro complementario.

- Acelerómetro: Mide la aceleración en los tres ejes (X, Y, Z) y proporciona datos sobre la aceleración total en cada eje.
- Giroscopio: Mide la velocidad angular en los tres ejes (X, Y, Z) y permite detectar la rotación del sensor.
- DMP: El procesador de movimiento digital (DMP) fusiona los datos de ambos sensores (acelerómetro y giroscopio) para proporcionar información sobre la orientación del sensor.
- Comunicación I2C: El sensor MPU6050 se comunica con un microcontrolador a través de I2C, enviando las lecturas de aceleración y rotación.

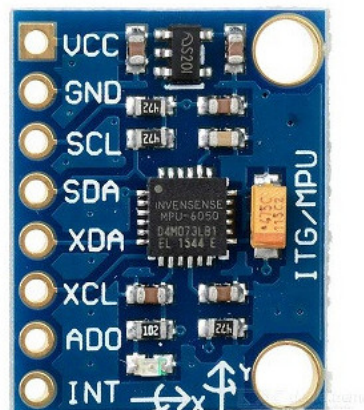


Figura 2: Sensor IMU MPU6050.

3.5. Sensores Infrarrojos de Proximidad

Estos sensores se utilizan para detectar líneas negras o la proximidad de objetos. Los sensores infrarrojos emiten luz y detectan su reflexión para identificar variaciones en la superficie. Esto es crucial para el seguimiento de líneas en el circuito.

En un sistema de seguimiento de línea, los sensores IR están colocados en la parte inferior del robot para detectar la línea que debe seguir. Los robots suelen usar varios sensores dispuestos en línea para poder diferenciar entre la posición de la línea y ajustarse a ella.

- **Detección de la línea:** Si el sensor está sobre la línea negra, refleja menos luz hacia el receptor, lo que indica la presencia de esta.
- **Ajuste del robot:** El robot ajusta su dirección (generalmente utilizando motores) para mantenerse alineado con la línea, girando en función de la información proporcionada por los sensores IR.

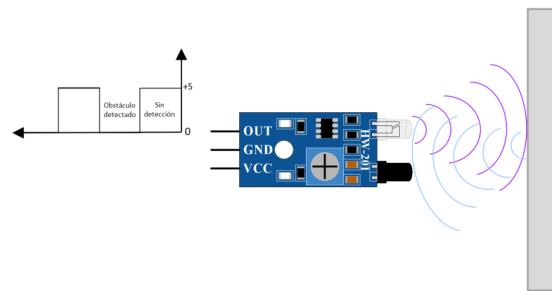


Figura 3: Funcionamiento de un sensor infrarrojo.

3.6. Conexión Bluetooth

El enlace Bluetooth permite la comunicación inalámbrica entre el robot y una aplicación móvil. Utilizando módulos como el HC-05, se puede enviar y recibir comandos desde un dispositivo Android. La configuración típica incluye el emparejamiento, la asignación de un baud rate y la comunicación mediante el protocolo UART.

Para este proyecto se utilizó el módulo bluetooth SPP-C3231, es un dispositivo que permite la comunicación inalámbrica a través de la tecnología Bluetooth. Es un módulo compacto diseñado para habilitar la transferencia de datos entre dispositivos como microcontroladores, computadoras y smartphones, usando el Perfil de Puerto Serie (SPP, Serial Port Profile). Este perfil permite que el módulo actúe como un puerto serie virtual, permitiendo la comunicación a través de un enlace Bluetooth como si fuera un puerto serial físico (RS232 o TTL).



Figura 4: Módulo Bluetooth HC-05 utilizado para control remoto.

3.7. Motor Paso a Paso 17HS4401S

El motor paso a paso modelo 17HS4401S es un dispositivo electromecánico bipolar de dos fases, diseñado para aplicaciones que requieren movimientos precisos y controlados, tales como impresoras 3D, máquinas CNC y proyectos de robótica. Sus principales características son:

- **Ángulo de paso:** 1.8° por paso, lo que equivale a 200 pasos por revolución, permitiendo un movimiento suave y exacto.
- **Corriente por fase:** 1.7 A, facilitando su integración con controladores de motores comunes y fuentes de alimentación estándar.
- **Par de retención:** 43 N·cm, proporcionando una fuerza adecuada para una variedad de aplicaciones.
- **Resistencia por fase:** 1.5 ± 10
- **Inductancia:** $2.8 \text{ mH} \pm 20$
- **Dimensiones:** Tamaño NEMA 17 (42 x 42 mm) con una longitud de cuerpo de 40 mm.
- **Eje:** Diámetro de 5 mm.
- **Peso:** Aproximadamente 280 g.

Este motor cuenta con cuatro cables, lo que facilita su conexión a controladores de motores bipolares. Su diseño robusto y especificaciones lo hacen ideal para proyectos que demandan alta precisión y confiabilidad en el control de movimientos.



Figura 5: motor paso a paso 17HS4401S.

3.8. Driver A4988

El driver A4988 es un controlador de motores paso a paso ampliamente utilizado para gestionar motores bipolares como el 17HS4401S. Este driver permite controlar la dirección y velocidad del motor con alta precisión, gracias a su capacidad de realizar micropasos. Sus principales características son:

- **Tensión de operación:** 8-35V.
- **Corriente máxima por fase:** 2A (con disipador térmico adecuado).
- **Resolución de micropasos:** Hasta 1/16 de paso completo.
- **Protecciones:** Contra sobrecorriente, sobretensión y desconexión por bajo voltaje.

Para integrar el A4988 con el motor 17HS4401S, es necesario conectar correctamente las fases del motor a las salidas del driver, ajustar el límite de corriente del driver usando un potenciómetro integrado y proporcionar las señales de dirección y paso desde un microcontrolador. Este setup permite aplicaciones precisas y confiables en diversos proyectos.

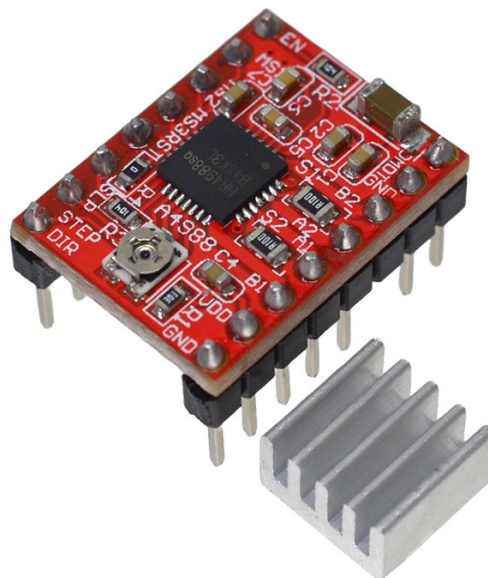


Figura 6: Imagen del driver A4988.

4. Presentación del Problema

En el marco de una competencia de robots móviles, se solicita desarrollar la integración y programación de un Balancing Robot. Particularmente, se requiere que el robot tenga una configuración autoestabilizada sobre dos ruedas y, además, se pueda monitorear remotamente desde una APK Android, con el fin de recorrer un circuito donde el robot sea capaz de desarrollar una trayectoria autónoma siguiendo una línea negra marcada en el suelo, y además pueda ser controlado de manera manual mediante Bluetooth.

Uno de los desafíos corresponde a corregir de manera autónoma la estabilidad del robot, por lo que será necesario implementar y calibrar un controlador PID, en base a los ángulos que entrega el sensor inercial (MPU6050). De este modo, el robot se debe mantener estable en un punto cuando no se está solicitando un cambio de posición a pesar de las perturbaciones. Además, el robot debe tener límites de inclinación y ser tolerante a las inclinaciones comandadas desde el control manual.

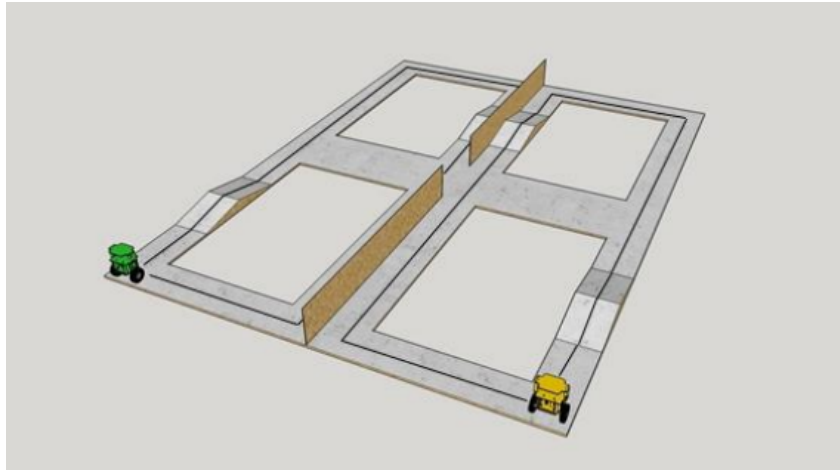


Figura 7: Ejemplo Trayectorias de Exploración y Retorno.

Una vez resuelta la estabilización y la aplicación móvil (APK) de control y calibración, el robot deberá competir en un circuito con otros robots de similares características. La competencia está enfocada en verificar la capacidad de estabilización del robot, memorización de rutas y cumplimiento en el menor tiempo posible de dichas rutas; los escenarios de la competencia serán similares a lo mostrado en la Figura 7.

5. Solución del Problema

1. **Diseño y construcción del chasis robótico:** El chasis fue fabricado mediante impresión 3D con PLA para garantizar una mayor estabilidad, considerando los requisitos de espacio para integrar sensores, actuadores (motores paso a paso) y el hardware de control necesarios, además el diseño está pensado para tener un centro de gravedad bajo para una mejor estabilidad. Este diseño incluye espacios para los siguientes elementos:
 - Sensor inercial MPU6050, ubicado en la parte superior del robot para obtener lecturas precisas de la inclinación del robot.
 - Sensores infrarrojos Tcrt5000, colocados en la parte inferior del robot para detectar líneas negras en el circuito.
 - Módulo de comunicación bluetooth SPP-C, instalado en la parte superior para una mejor señal.
 - Motores paso a paso (17HS4401S) y sus drivers, los paso a paso se encuentran en la parte inferior con sus soportes para atornillarlos al robot y los drivers (Pololu A4988) se instalaron en la parte superior del robot para una mejor ventilación.
 - Fuente de energía 3 Batería 18650 Zhv 3.7v, instaladas en el centro del robot con un soporte que conecta las pilas en serie para entregar un mayor voltaje.
2. **Programación del sistema:** La programación se realizó en el entorno de desarrollo de Arduino, implementando los siguientes módulos:
 - Control PID: Se desarrolla un algoritmo de control PID para que garantice la estabilidad del robot. Las lecturas del MPU6050 se procesan los cálculos de error de la inclinación para ajustar los motores en tiempo real.
 - Seguimiento de líneas: Los datos de los sensores infrarrojo se utilizan para implementar un algoritmo de control para que el robot se mantenga sobre una línea negra marcada en el circuito.
 - Control bluetooth: Se desarrolló una interfaz que permite enviar comandos de forma inalámbrica desde una app móvil, posibilitando un control manual para el robot.
3. **Calibración:**
 - PID: Se ajustan las ganancias del algoritmo PID, proporcional, integral y derivativa (K_p, K_i, K_d) para lograr una estabilidad óptima del robot ante las perturbaciones externas.
 - Sensores infrarrojos: Se verificó la sensibilidad de los sensores ante diferentes condiciones de iluminación.
 - Bluetooth: Se probó la conexión y respuesta del módulo para garantizar un tiempo de latencia mínimo.
4. **Implementación del control de estabilidad:**
 - Se utilizó el sensor inercial MPU6050 para medir el ángulo de inclinación del robot en tiempo real. Estos datos se utilizan para alimentar al controlador PID, que se encarga de ajustar las señales para que los motores puedan corregir la estabilidad del robot. El algoritmo permite que el robot mantenga su estado estable cuando se encuentra en reposo. Además el sistema se ajustó para tolerar inclinaciones comandadas manualmente mediante el control remoto.

5.1. Diagrama de Conexiones

Se detalla el esquema de conexiones entre los distintos implementos utilizados en el proyecto:

■ Drivers:

- **Driver 1:** El pin DIR se conectó al pin digital 2 del Arduino, el pin STEP al pin 3, y las entradas VCC, SLEEP, y RESET al pin de 5V del Arduino. La tierra (GND) y el pin ENABLE se conectaron a la tierra del Arduino.
- **Driver 2:** El pin DIR se conectó al pin digital 4 del Arduino y el pin STEP al pin 5. Las entradas VCC, SLEEP, y RESET también se conectaron al pin de 5V del Arduino. Adicionalmente, se conectaron una fuente externa de 12V y su correspondiente tierra externa para alimentar los motores.

- **Motores:** Los cables del motor se conectaron a los pines A1, A2, B1, y B2 de cada uno de los drivers siguiendo el esquema de color y conexión del motor paso a paso.

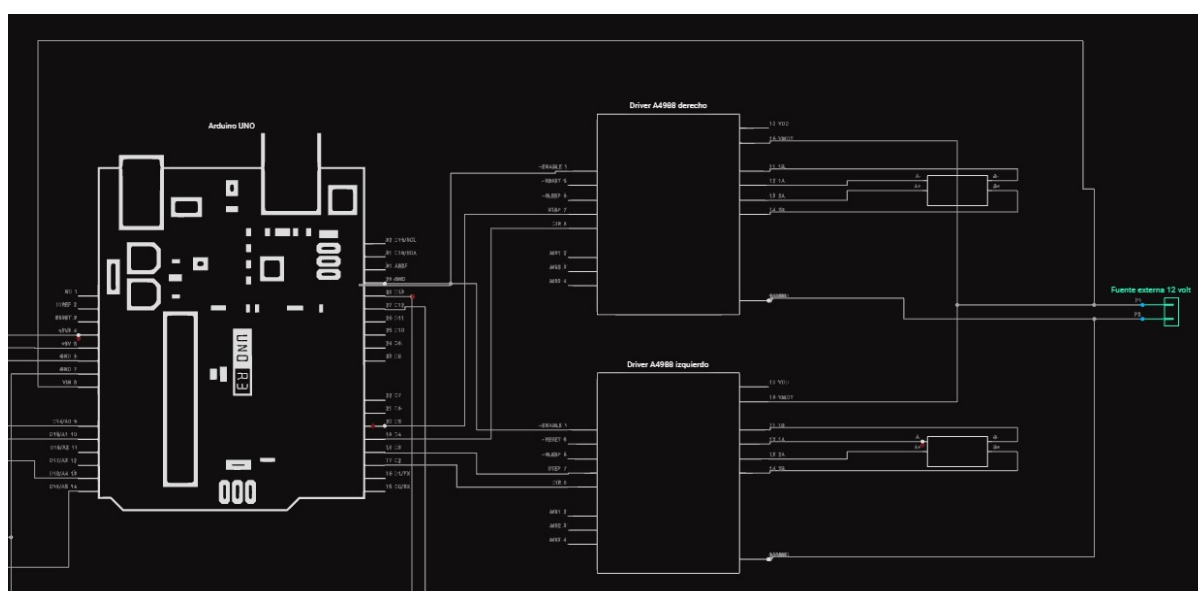


Figura 8: Diagrama de conexiones generales.

- **Sensor infrarrojo TCRT5000:** Cada sensor se conectó de la siguiente forma:
 - Tierra (GND) al pin de tierra del Arduino.
 - VCC al pin de 5V del Arduino.
 - Salida analógica al pin A4 y A5 para los sensores derecho e izquierdo, respectivamente.
 - Salida digital al pin D12 y D13 para los sensores derecho e izquierdo, respectivamente.
- **MPU6050:** El pin SCL se conectó al pin A0 del Arduino, el pin SDA al pin A1, y su alimentación al pin de 3.3V del Arduino. La tierra se conectó al pin GND del Arduino.
- **Batería de 12V:** La batería, compuesta por pilas de 3.7V conectadas en serie, se conectó al pin VIN del Arduino para energizarlo y a los drivers para alimentar los motores.

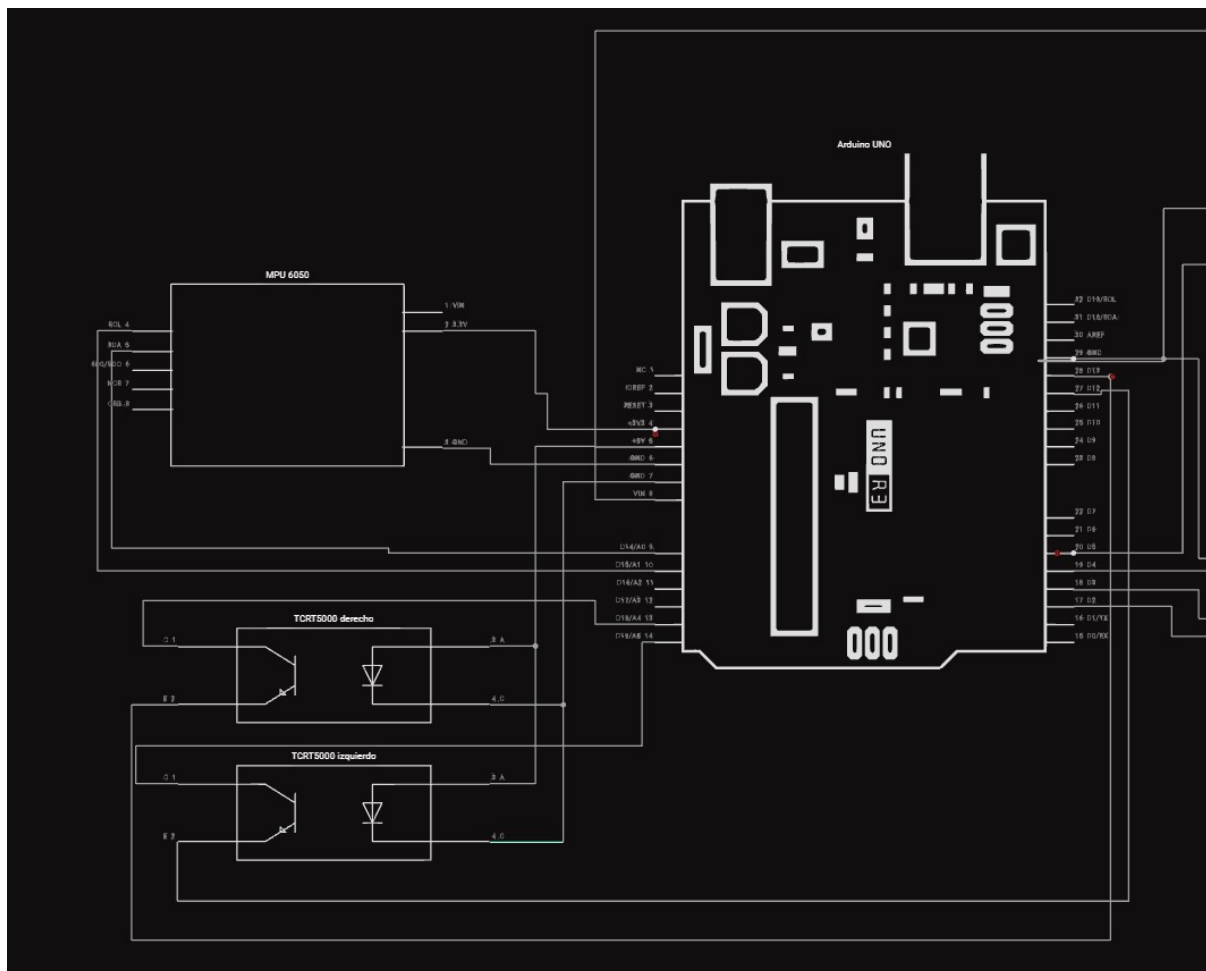


Figura 9: Diagrama detallado de conexiones de sensores y drivers.

6. Resultados

Las pruebas realizadas al robot balanceador revelaron diferentes problemas en el desempeño del sistema, relacionados principalmente con fallas de componentes electronicos, errores de diseño mecanico y desorden en el ensamblaje. A continuación, se presentan los resultados detallados.

1. Fallas en los componentes

a) Sensor inercial MPU6050:

Se quemaron dos sensores durante las pruebas iniciales, el primer sensor fue un error humano ya que se hizo contacto con el MPU y un cable con corriente, el segundo sensor se quemo debido a un corto circuito ocurrido en el robot por errores en el diseño del cableado. Esto comprometió la capacidad del robot para medir los angulos de inclinación por lo tanto no se podia realizar la mision de balancearse.

Sin datos confiables del MPU6050, el algoritmo utilizado PID no logro estabilizar el robot, ya que no se contaba con la información necesaria para ajustar los motores de manera precisa.

b) Drivers de Motores Pololu A4988:

Uno de los drivers para controlar los motores paso a paso se daño, especificamente se estropeo el regulador de corriente por lo que no se podia variar la velocidad de uno de los motores.

Esta disparidad en la velocidad causó que el robot girase constantemente en circulos en lugar de desplazarse en linea recta o mantenerse estatico.

c) Sistema electrico:

El cableado no se optimizó ni organizó. Los cables largos y desordenados causaron interferencias (ruido porque no eran cables aislantes), fallas de conexiones y ocupación innecesaria del espacio.

Esto dificultó las reparaciones, aumentó la posibilidad de cortocircuito y generó ineficiencia en la transmisión de las señales.

2. Problemas en el diseño

a) Modelo 3D del chasis:

El modelo 3D presentaba errores en su diseño que impidieron un correcto ensamble. Las piezas no encajaban con precisión, generando desalineaciones en los soportes de los motores. La falta de los soportes adecuados para los componentes electronicos como el MPU6050, los drivers de los motores, agujeros para atornillar el arduino y espacio suficiente para la bateria, provocaron que la estructura del robot fuera inestable, lo que agravó las fallas en el balance y comprometió la funcionalidad en general.

b) Centro de gravedad:

A pesar de que el diseño 3D si consideraba cierta simetria para el centro de gravedad, al introducir el sistema electrico el centro de gravedad se vio afectado haciendo

que se desbalance hacia un lado. Esto provoca que se necesite un mayor esfuerzo de parte de los motores para obtener estabilidad.

3. Retroalimentación teórica

a) MPU6050 (Unidad de Medición Inercial)

En sistemas de estabilización, el sensor MPU6050 mide inclinaciones con precisión hasta $\pm 250^\circ/\text{s}$. La inclinación ideal para un robot balanceador es mantener un ángulo dentro de $\pm 5^\circ$ de la vertical, con correcciones realizadas por el controlador PID en menos de 200 ms para mantener el equilibrio. Esto asegura una estabilidad constante incluso con pequeñas perturbaciones.

Existen algoritmos de filtro como el Kalman o sistemas de procesamiento digital de movimiento que mejoran la estabilidad al filtrar el ruido del MPU provocado por el resto de componentes.

b) Motores Paso a Paso y Drivers

Para realizar movimientos precisos, los motores deben operar a velocidades equivalentes en ambos lados, ajustándose rápidamente con cada corrección del PID. Las velocidades deseadas para un movimiento lineal son de 0.1-0.3 m/s y con respecto a los ángulos se busca 30-60°/s para drivers como los Pololu A4988.

La respuesta de los motores paso a paso 17HS4401S es de 1 a 2 ms por cada paso, esto depende de la frecuencia de entrada. El tiempo es suficiente para realizar correcciones rápidas si los sensores y el controlador trabajan con baja latencia.

c) Sensores Infrarrojos (IR TCRT5000)

Estos sensores cuentan con un tiempo de respuesta de 10-50 μs y cuentan con efectividad para velocidades de hasta 0.3 m/s en superficies con un contraste claro entre la línea y el fondo, esto es perfecto para el caso a estudiar ya que los motores paso a paso cuentan con esa velocidad recomendada.

Se deben utilizar varios sensores delante del robot para anticipar curvas y ajustar la dirección de los motores con cierto grado de precisión.

7. Conclusión

Si bien los resultados obtenidos no se acercaron a los planeados, esta experiencia aportó aprendizajes significativos sobre el funcionamiento de diversos componentes electrónicos, como los controladores de motores paso a paso, sensores inerciales, sensores de comunicación y sensores infrarrojos. Se logró comprobar el correcto uso de todos los componentes de manera independiente; sin embargo, durante la integración surgieron fallas que impidieron el funcionamiento óptimo del sistema en su conjunto como los errores en el diseño de la estructura y soporte del robot, así como en el circuito.

El objetivo de diseñar y construir físicamente un chasis robótico fabricado mediante impresión 3D fue cumplido con éxito. El diseño final cumplió con los requisitos establecidos, y el proceso de fabricación demostró ser efectivo, logrando un chasis funcional y adecuado para los fines del proyecto. Esto permitió sentar una base sólida para las etapas posteriores del desarrollo.

Sin embargo, el objetivo de programar el sistema de control de estabilidad y movimiento para Arduino no se logró. Durante el desarrollo se identificaron dificultades en la implementación de los algoritmos necesarios para el control eficiente del sistema. Esto afectó la integración entre hardware y software, lo que impidió alcanzar el desempeño deseado en las pruebas.

Finalmente, no se cumplió el objetivo de desarrollar pruebas de movimiento estable en un recorrido de circuito de línea negra. Las pruebas realizadas reflejaron una falta de precisión y estabilidad en el movimiento del robot, posiblemente debido a problemas relacionados con la programación del sistema de control o la calibración del chasis. Aunque se realizaron intentos para ajustar el desempeño, no se lograron resultados satisfactorios en esta etapa.

Para futuros trabajos, se recomienda una planificación más exhaustiva del proyecto, que incluya el diseño de un diagrama de conexiones detallado para verificar la correcta interacción entre los componentes, así como la construcción de una base estructural adecuada que facilite la estabilización de todos los elementos integrados. También, sería importante implementar el sistema de manera gradual, asegurando que cada etapa funcione correctamente antes de avanzar a la siguiente, y evaluar diferentes estrategias de control que permitan cumplir con los objetivos del proyecto de manera eficiente.

Asimismo, se considera valioso incorporar herramientas de simulación previas a la construcción física. Esto ayudaría a anticipar problemas de diseño e integración, optimizando los recursos disponibles y reduciendo la probabilidad de errores durante la implementación.

Aunque los resultados de la experiencia no fueron los esperados, esta representa un paso importante hacia la mejora de la planificación, ejecución y control en proyectos futuros. Los aprendizajes obtenidos servirán como base sólida para afrontar desafíos similares con mayor precisión y éxito.

8. Referencias

1. IEEE. *Enhancing Gimbal Stabilization Using DMP and Kalman Filter: A Low-Cost Approach with MPU6050 Sensor*. IEEE Conference Publication, 2024. Disponible en: <https://ieeexplore.ieee.org/document/10455683>.
2. IEEE. *On the Stability Bounds of Kalman Filters for Linear Deterministic Discrete-Time Systems*. IEEE Journals & Magazine, 2020. Disponible en: <https://ieeexplore.ieee.org/document/8957270>.
3. ALLDATASHEET. *MPU-6050 datasheet* [en línea]. [Sin fecha]. Disponible en: <https://www.alldatasheet.com/html-pdf/517744/ETC1/MPU-6050/818/8/MPU-6050.html>. [Consulta: 7 diciembre 2024].
4. ALLDATASHEET. *A4988 datasheet* [en línea]. [Sin fecha]. Disponible en: <https://www.alldatasheet.com/html-pdf/338780/ALLEGRO/A4988/292/1/A4988.html>. [Consulta: 7 diciembre 2024].
5. ALLDATASHEET. *17HS4401 datasheet* [en línea]. [Sin fecha]. Disponible en: <https://www.alldatasheet.com/html-pdf/1131976/MOTIONKING/17HS4401/700/1/17HS4401.html>. [Consulta: 7 diciembre 2024].

Apendice

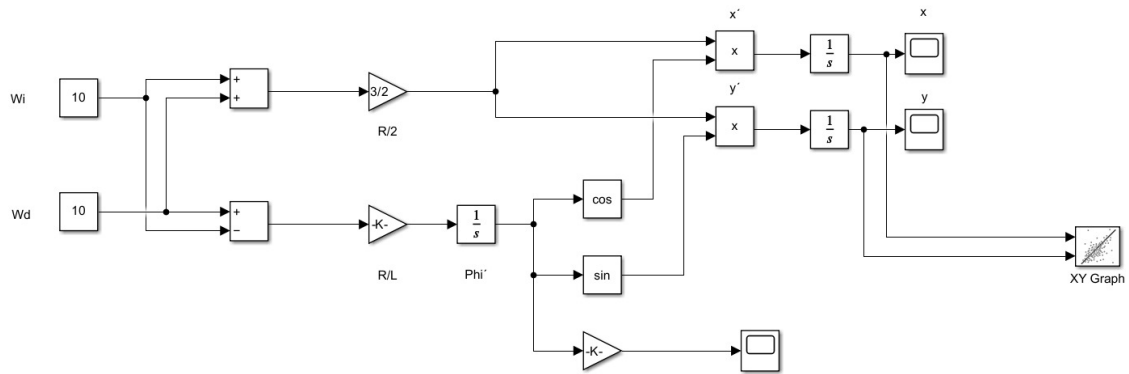


Figura 10: Modelación del sistema dinamico matlab simulink.