



**UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
CARRERA DE ELECTRÓNICA Y TELECOMUNICACIONES**

TRABAJO DE TITULACION

Propuesta Tecnológica, previo a la obtención del Título de:

INGENIERA EN ELECTRÓNICA Y TELECOMUNICACIONES

Diseño de un sistema de navegación inercial para control autónomo de un robot móvil utilizando control PID y filtro de Kalman, aplicado en gestión de domótica y localización de personas.

MODALIDAD

Propuesta Tecnológica

AUTORA

Elizabeth Carolina Potes Lino

TUTOR

Ing. Sendey Vera González, MSc

La Libertad - Ecuador
2026

DEDICATORIA

Dedico este trabajo a Dios, por ser mi guía constante, brindarme fortaleza en los momentos difíciles y permitirme culminar esta etapa tan importante de mi vida.

A mi mamá, por su amor incondicional, su apoyo inquebrantable y por estar presente en cada paso de este camino, siendo mi mayor inspiración para seguir adelante.

En memoria a Violeta S., quien formó parte de este camino y, aunque ya no se encuentra presente, dejó un recuerdo significativo durante el proceso.

AGRADECIMIENTO

Expreso un sincero agradecimiento a mi mamá Mercy Lino, por su apoyo incondicional, su comprensión y su constante motivación a lo largo de toda mi formación académica.

De igual manera, agradezco a mi familia por su respaldo durante este proceso en especial a mis tíos: Domy Lino, Eloy Lino y Renato Rodríguez por su apoyo constante.

Gracias a Lady Lino, Ariel Ortega, Gabriel Lino, Geoconda Murillo por su disposición y valiosa colaboración en las pruebas experimentales, lo cual fue fundamental para el desarrollo de este trabajo.

A mi compañero de vida, por su apoyo, paciencia y compañía durante este proceso, siendo un respaldo importante en esta etapa.

A mi tutor de tesis, Ing. Sendey Vera, por su orientación, conocimientos compartidos y acompañamiento durante el desarrollo de esta investigación.

APROBACIÓN DEL TUTOR

En mi calidad de tutor del trabajo de titulación denominado: **“Diseño de un sistema de navegación inercial para control autónomo de un robot móvil utilizando control PID y filtro de Kalman, aplicado en gestión de domótica y localización de personas”**, elaborado por la estudiante **Elizabeth Carolina Potes Lino**, de la carrera de **Ingeniería en Electrónica y Telecomunicaciones** de la Universidad Estatal Península de Santa Elena, declaro que luego de haber orientado, estudiado y revisado, lo apruebo en todas sus partes y autorizo al estudiante que inicie los trámites legales correspondientes.

La Libertad, 23 de abril del 2026



Ing. Sendey Vera González, MSc
Tutor

TRIBUNAL DE GRADO



Validar únicamente en FirmaEC.
Firmado electrónicamente por:
**WASHINGTON DANIEL
TORRES GUIN**

Ing. Washington Torres Guin, Mgtr.
DECANO DE FACULTAD



Validar únicamente en FirmaEC.
Firmado electrónicamente por:
**JOSE MIGUEL SANCHEZ
AQUINO**

Ing. José Sánchez Aquino, Mgtr.
DIRECTOR DE CARRERA



Validar únicamente en FirmaEC.
Firmado electrónicamente por:
**OSCAR WLADIMIR
GOMEZ MORALES**

Ing. Óscar Gómez Morales, Ph.D
DOCENTE ESPECIALISTA



**Sendey Agustín
Vera González**



Ing. Sendey Vera González, Mgtr.
DOCENTE TUTOR



Validar únicamente en FirmaEC.
Firmado electrónicamente por:
**MARIA MARGARITA
RIVERA GONZALEZ**

Abg. María Rivera González, Mgtr.
SECRETARIA GENERAL

DECLARACIÓN DE AUTORÍA

Elizabeth Carolina Potes Lino, egresada de la Universidad Estatal Península de Santa Elena, Facultad de Sistemas y Telecomunicaciones de la Carrera de Electrónica y Telecomunicaciones, declaro que la responsabilidad del contenido recae exclusivamente en mí y el patrimonio intelectual de la misma le corresponde a la Universidad Estatal Península de Santa Elena.



Elizabeth Carolina Potes Lino
Autora

RESUMEN

El presente documento desarrolla un sistema de navegación inercial aplicado a un robot balancing, se utiliza una placa Arduino que se encarga de mantener el equilibrio del robot mediante un algoritmo basado en técnicas de control PID y Filtro de Kalman que contribuyen a reducir el ruido y optimizar la precisión de las señales. Se utiliza una tarjeta Raspberry Pi para la gestión de tres procesos, el primer proceso se trata del direccionamiento del robot utilizando un sensor inercial, el segundo proceso de detectar y medir una distancia con un sensor infrarrojo y el tercer proceso para la detección facial a través una cámara web. El robot logra su autonomía utilizando una brújula digital que ayuda en la toma de decisiones para determinar la dirección de los giros y desplazamientos con respecto al norte magnético sin depender de un GPS. Durante el recorrido del robot, el sensor de distancia puede detectar un obstáculo dentro del entorno de navegación, entonces el balancing detiene su desplazamiento y se activa el sistema de visión artificial para detectar un rostro, de esta manera se valida la presencia de una persona. Se diseña una aplicación Android guiada al ámbito de la domótica que sirve para visualizar de forma remota la información sobre la ubicación de una persona dentro de una habitación.

Palabras clave: Navegación inercial, Robot Balancing, Control PID, Filtro de Kalman, Domótica.

ABSTRACT

This paper describes an inertial navigation system applied to a balancing robot. An Arduino board is used to maintain the robot's balance through an algorithm based on PID control techniques and a Kalman filter, which help reduce noise and optimize signal accuracy. A Raspberry Pi board is used to manage three processes: the first process involves steering the robot using an inertial sensor; the second process involves detecting and measuring distance with an infrared sensor; and the third process involves facial detection via a webcam. The robot achieves autonomy by using a digital compass that aids in decision-making to determine the direction of turns and movements relative to magnetic north without relying on GPS. As the robot moves, the distance sensor can detect an obstacle within the navigation environment; the balancing mechanism then stops its movement, and the computer vision system is activated to detect a face, thereby confirming the presence of a person. An Android application designed for the home automation sector has been developed to remotely display information about a person's location within a room.

Keywords: Inertial navigation, Robot balancing, PID control, Kalman filter, Home automation.

Índice

Índice de figuras	10
Índice de cuadros	12
CAPÍTULO I	13
1 Generalidades del proyecto	13
1.1 Antecedentes	13
1.2 Descripción del Proyecto	16
1.3 Objetivos	18
1.3.1 Objetivo General	18
1.3.2 Objetivos Específicos	18
1.4 Justificación	18
1.5 Resultados Esperados	20
CAPÍTULO II	21
2 Marco Referencial	21
2.1 Marco Teórico	21
2.1.1 Sistema de Navegación Inercial	21
2.1.2 Componentes del proyecto	28
2.1.2.1 Componentes físicos	28
2.1.2.2 Componentes lógicos	34
2.2 Metodología del Proyecto	40
2.2.1 Investigación experimental	42
2.2.2 Alcance exploratorio	43
2.2.3 Técnicas de recolección de información	43
CAPÍTULO III	45
3 Desarrollo de la Propuesta	45
3.1 Diseño de la Propuesta	45
3.1.1 Diseño de navegación inercial	46
3.1.2 Diseño del control de dirección	55
3.1.3 Diseño de detección y localización	60
3.1.4 Diseño de aplicación móvil	63

CAPÍTULO IV	67
4 Pruebas y Resultados	67
4.1 Pruebas y validación experimental	67
4.2 Resultados	78
CONCLUSIONES	86
RECOMENDACIONES	88
BIBLIOGRAFIA	89
Referencias	89
ANEXOS	95

Índice de figuras

1	Self-balancing kit robot	22
2	Diagrama de bloque con control pid.[15]	23
3	Diagrama de bloque con filtro de kalman	26
4	Sensor mpu 6050 [15].	28
5	Módulo bluetooth hc-06[15].	29
6	Motores con sensor encoder.[15]	29
7	Arduino uno keyestudio.[15]	30
8	Motores dc.[15]	30
9	Bateria lipo.	31
10	Sensor mpu 9250.	31
11	Sensor vl53l0x.	32
12	Leds infrarojos.	32
13	Power bank	33
14	Camara web.	33
15	Raspberry pi 3.	34
16	Baquelita para cnc.	35
17	Sistema operativo raspbian	35
18	Virtual network computing (VNC)	36
19	Open source computer vision library (opencv)	37
20	Software proteus)	38
21	Herramienta flatcam)	39
22	Software de control numérico computarizado (cnc)	39
23	Android studio	40
24	Diagrama de metodología del proyecto	41
25	Arquitectura del sistema	45
26	Diagrama de interconexión	46
27	Diagrama de equilibrio	47
28	Diagrama representativo del péndulo invertido.[15]	47
29	Dinámica del péndulo invertido.[15]	48
30	Diagrama del control de dirección	56
31	Imágenes de los circuitos electrónicos diseñados	59

32	Fotografías de los circuitos impresos por la cnc	60
33	Diagrama de detección y localización	60
34	Fotografía del proceso de obtención de datos del sensor mpu6050	67
35	Gráfica comparativa mpu6050	69
36	Fotografía del proceso de obtención de datos del sensor mpu9250	70
37	Gráfica comparativa sensor mpu9250	71
38	Gráfica comparativa de detección	73
39	Obtención de datos sensor vl53l0x (21-40 cm)	74
40	Gráfica comparativa (1-20cm) sensor vl53l0x	76
41	Obtención de datos sensor vl53l0x (21-40 cm)	77
42	Gráfica comparativa (21-40cm) sensor vl53l0x	77
43	Propuesta final	78
44	Diagrama de flujo del sistema	80
45	Navegación sin detección	81
46	Interfaz sin detección	82
47	Navegación con detección	82
48	Interfaz con deteccion	83
49	Recorrido en el escenario	83
50	Menú de aplicación android	84
51	Opción de monitoreo	84
52	Opción balancing manual	85

Índice de cuadros

1	Datos de ángulo de inclinación para el control de equilibrio . .	68
2	Datos experimentales del sensor mpu9250	70
3	Datos de deteccion facial	72
4	Datos sensor vl53l0x (1-20cm)	75
5	Datos sensor vl53l0x (21-40cm)	76
6	Detalles técnicos del escaneo	96

CAPÍTULO I

1 . Generalidades del proyecto

1.1 Antecedentes

La necesidad de utilizar robots para facilitar las labores cotidianas de las personas , es mas comun ahora de lo que parece , los robot están diseñados e implantados con diferentes tipos de sensores y actuadores , para ubicarse y recorrer correctamente en un espacio bidimensional y tridimensional. En el trabajo investigativo de Rong, Ko, Wang y Shi, presenta un prototipo que emplea sensor inercial MEMs, IMU (unidad de medición inercial), implementando filtro de Kalman para obtener la medida de posición estable [1].

La navegación autónoma para un robot móvil se encuentra en constante investigación con la finalidad de tener un robot que cumpla con trayectorias predeterminadas evitando obstáculos, como se muestra en el estudio de Encalada y Álvarez en el que se utilizan componentes electrónicos y de control como Arduino mega, actuadores, modulo Xbee, controladores de motor, batería lipo, voltaje ajustable y regulador para el diseño del prototipo. Para la presentación de cálculos analíticos, funcionamiento y respectiva simulación se detalla el software ANSYS, en la aplicación de la navegación autónoma los autores utilizan control PID, cálculos de odometría, comunicación inalámbrica y una interfaz gráfica en Matlab con opción de manejo manual o automático de un prototipo omnidireccional [2].

En los temas de estudio para crear un prototipo con navegación autónoma se requiere de varios parámetros en la programación de la tarjeta controladora que coordina movimientos, ya sea para evitar tropezar con objetos, realizar un recorrido o llegar a un lugar específico. Esto se logra con el uso de elementos como sensores, controlador, procesador, actuadores, se tiene en cuenta que el modelo del robot depende de la necesidad del ser humano. La introducción de un autómeta logra generar oportunidades de empleo y aumentar el desarrollo de tecnologías emergentes, es decir que los robots no pueden reemplazar a los

humanos, su importancia tecnológica radica cuando el trabajo de una persona puede estar expuesta a contagio de enfermedades, en peligro de perder alguna de sus extremidades o inclusive perder la vida.

El prototipo móvil de modelo unicycle, se origina con el modelo Kazuo Yamafuj creado en el año 1986, diseño con un eje entre las ruedas en un móvil que contenía el dispositivo de estabilización y control. En el año 2000 fue creado el prototipo llamado “Joe” en los laboratorios de electrónica industrial [3]. En el 2001 el robot SEGWAY, un vehículo de transporte unipersonal eléctrico de dos ruedas, con sensores que detectan grados de inclinación, para regular la velocidad y mantener el equilibrio, el modelo de la empresa Hoverboard es un diseño mejorado que consta de dos ruedas unidas por dos plataformas que contiene más sensores y el modelo del robot con fines didácticos Balanbot de self Balancing con estructuras acrílicas, sensor IMU, que permite mantener la estabilidad vertical, estos sistemas requieren de un mecanismo adecuado y de diseño de circuitos eléctricos [4].

El diseño de una pista electrónica para interconexión de elementos en una placa de baquelita se construye frecuentemente en una maquina CNC conocida como una computadora de control numérico, esta tecnología es una herramienta que permite realizar circuitos eléctricos personalizados con movimientos controlados por medio de un computador que contiene el software que realiza la lectura del G-code (lenguaje de programación) contiene los comandos de acciones para fabricar la placa PCB. La CNC funciona de manera automática o manual para realizar los procesos de grabado, perforación y corte de acuerdo al diseño eléctrico [7].

La fabricación en máquinas CNC surge en el año 1940 por Johon Parsons con una máquina que se movilizaba como un sistema de coordenadas, en 1952 se crea el primer prototipo de mecanizado CNC como proyectos de laboratorio en las Fuerzas Aéreas de Estados Unidos, luego de un año demostraron todas sus aplicaciones. Tiempo después descubren herramientas de programación automática para comunicarse con la CNC y que la tecnología podía funcionar como fresadora, tornos y rectificadora [8]. Estos inicios desencadenan el uso de la máquina para el área de electrónica con circuitos eléctricos y en varios ámbitos industriales. Con la finalidad de crear sistemas

electrónicos inteligentes que evolucionan cada día.

El primer sistema de domótica aparece en el año de 1966 con el primer dispositivo que controlaba la televisión, la temperatura o programar el despertador entre otras tareas del hogar con un ordenador de gran tamaño. Luego en el año de 1991 surge la teleasistencia un sistema para personas mayores, con la aparición del internet en 1998 la domótica revoluciona por un lapso de 1999 a 2006 con electrodomésticos que se adaptan a la IOT (Internet de las cosas), cuando se tiene el primer teléfono inteligente en el año 2007, también se desarrollan las aplicaciones que permiten controlar los artefactos del hogar. Los avances de la tecnología aumentan con los equipos Smart y los sistemas Siri y Alexa que son IA (Inteligencia Artificial) estas herramientas permiten la automatización de las viviendas [9]. En tiempo actual existen más asistentes virtuales y sistemas de videovigilancia que son usados para la seguridad del hogar y aplicar detección facial es recomendable.

La aplicación de reconocimiento facial se ha usado desde 1883 por Alphonse Bertillon mediante medidas antropométricas descriptivas en rasgos faciales que distinguen a una persona. Los avances tecnológicos y estudios biométricos permiten que estos detecten rostros se realicen mediante algoritmos lógicos que analizan imágenes para reconocer a un ser humano [10]. Un sistema de detección de rostro de personas utiliza una cámara para capturar la imagen, procesarla e identificarla y su implementación incrementa en el tema de la seguridad, su finalidad es mejorar la calidad de vida, brindar comodidad para las personas del hogar.

El monitoreo de personas que tengan un algún tipo enfermedad o persona adulta mayores, en el estudio del seguimiento para pacientes en un entorno doméstico, en casos donde se requieren el uso de dispositivos en el cuerpo que resultan incómodos para el usuario. Proponen una solución sensorial en un suelo de madera donde caminen las personas. Como un caso de estudio para obtener de esto la ubicación e incluso lectura de peso para el reconocimiento en caso de haber varios usuarios [11]. Obtener datos de localización de

personas en el interior del hogar es inviable si se utiliza un GPS, porque la señal satelital es obstaculizada por el material de construcción, por este motivo no permite que la persona que utiliza el GPS sea encontrada.

De acuerdo a la revista Bit Steam en el 2018 que presenta una sección catalogada como monográfico de ciberseguridad, calidad de servicio y localización de persona u objetos en consecuencia a la tecnología iot en los edificios inteligentes. El autor expresa que en el interior de los edificios no es posible utilizar los servicios de localización satelital. Esto ha conducido a una actividad frenética de nuestra industria para buscar soluciones de localización que funcionen en zonas sin cobertura GPS (centros comerciales, almacenes, metros, aeropuertos, etc.) [12]. El uso de sensor inercial y de sensor para medir distancia resultan factibles ante lo expuesto, para generar ubicaciones por coordenadas de referencia mediante una interfaz gráfica.

La creación de programación en bloques para dispositivos Android comienza en el año 2010 con el desarrollo de una aplicación apk utilizando la herramienta MIT App Inventor, que es un software que permite crear una interfaz gráfica compatible con Android un poco más adelante, en el año 2013 aparece la herramienta Android Studio como plataforma oficial para crear aplicaciones de Android y actualmente se mantiene en continuas actualizaciones de versiones por compatibilidad de versión Android y Google. En el estudio comparativo de ambas herramientas demuestran que las diferencia no son significativas con respecto a rendimiento, pero si recomiendan Android Studio para aplicaciones de gran escala [13].

1.2 Descripción del Proyecto

El proyecto consiste en el desarrollo de un sistema inercial autónomo para aplicar en el control de un robot balancing, con la finalidad de localizar una persona que se encuentra en una habitación. Para demostrar el control sobre el robot y desarrollar la navegación inercial, se plantea la siguiente misión; en la que el prototipo comienza su recorrido desde un punto de origen y se desplaza en línea recta, luego cambia su dirección al norte magnético y en caso de detectar una persona termina su misión, entregando información a un usuario encargado del monitoreo en su dispositivo móvil, mediante un mensaje en el que indica que se ha detectado un rostro, la captura de imagen

de la persona y la distancia entre el robot y el objetivo, como prueba del uso de aplicación en gestión de domotica.

Para el sistema de navegación inercial se utiliza un sensor MPU 9250 que se encarga de brindar datos que sirve para determinar una posible dirección de la trayectoria del robot de esta manera se obtiene un angulo de orientación para determinar el rumbo en la que se desplaza el balancing sin necesidad de emplear un GPS.

El robot balancing utiliza un MPU 6050 y el encoder Hall, para el control del equilibrio y la dinámica de inclinación del robot, cuando el prototipo se encuentra en desequilibrio el trabajo de los motores se centra en proporcionar una fuerza opuesta, esta fuerza restauradora va a devolver al robot a una posición vertical, acelerando en dirección opuesta para generar una fuerza inercial contrarrestando la inclinación. Además se aplica el algoritmo de control y filtro de Kalman con los datos del acelerómetro, giroscopio y los pulsos generados por el encoder.

Los procesos de navegación inercial y equilibrio del prototipo se realizan por separado, por ello se utilizan dos placas electrónicas; una cumple con el control del equilibrio del robot y la otra con la función de procesar el entorno. Con ambas placas se pretende no colapsar las respuestas en el sistema, para que el robot balancing pueda realizar la misión de recorrido se necesita de las señales PWM administradas a los motores, estas señales son condicionadas por el sistema de equilibrio con un microcontrolador Arduino y por el sistema de navegación inercial aplicado en un microprocesador Raspberry.

Otros de los procesos que realiza el prototipo es obtener la distancia entre robot y la persona, mediante el uso del sensor VL53L0X, la luz emitida por este dispositivo se refleja ante cualquier objeto no transparente, esta medida de distancia es obtenida por el cálculo del tiempo entre ida y vuelta de la luz infrarroja. Cuando el sensor infrarrojo detecta un objeto o persona en el rango de distancia de 40 centímetros se activa la rutina de detección de imagen.

La detección facial se integra al sistema utilizando cámara web con resolución de 1920x1080 píxeles y algoritmo de visión artificial realizado en python para reconocer que el rostro de una persona que se encuentra en el entorno.

Incorporar el procesamiento de imágenes en tiempo real permite que el robot actúe como agente domótico con autonomía.

1.3 Objetivos

1.3.1 Objetivo General

- Diseñar sistema de navegación inercial en un prototipo de robot balancing, con Raspberry para gestión de domótica y localización de personas.

1.3.2 Objetivos Específicos

- Analizar la temática de navegación que emplean dispositivos MPU, para la determinación lógica de programación en la navegación inercial utilizada en el prototipo de robot balancing autónomo.
- Diseñar el sistema de interconexión entre dispositivos, para el monitoreo de sensores y cálculo de coordenadas referenciales del robot.
- Implementar la navegación inercial y autonomía del robot balancing utilizando control PID y filtro de Kalman para la localización de personas.
- Diseñar una interfaz gráfica de dispositivo Android para visualizar el entorno y los datos recolectados de la ubicación de coordenadas cartesianas de una persona, finalmente realizar las pruebas del sistema.

1.4 Justificación

La implementación de un algoritmo de navegación inercial aplicado en un robot autónomo permite que el prototipo pueda movilizarse y ser aplicado en tareas cotidianas como asistentes personales o laborales, para ello se requiere

realizar estudios con procesos experimentales que permitan obtener los conocimientos y resultados parecidos a prototipos industriales con avanzadas tecnologías muy poco conocidas. Los desafíos del proyecto es buscar mejoras y actualizar las bondades de un robot, de esta manera continuar con los avances de la tecnología, aportando conocimientos eficientes y sostenibles.

El prototipo de robot balancing aporta para investigaciones futuras que involucren avances tecnológicos, con nuevos dispositivos electrónicos, como por ejemplo: el efectuar una ruta de recorrido autónomo en lugares con espacios reducidos donde el recorrido que realice no resulte molesto para personas que se encuentren en el interior del hogar, oficina, edificio u otro establecimiento. Otro ejemplo es optimizar su capacidad de equilibrio, de responder en diferentes escenarios, en aplicaciones de un robot autónomo como acompañante de una persona o asistente doméstico e incluso en áreas de robótica médica,

Aplicar un sistema de detección facial resulta factible y eficiente, en situaciones cotidianas existen motivos relevantes para encontrar personas y detectarlas, en especial cuando están en estado de vulnerabilidad, accidente o desorientación o en motivos que resultan indispensables de vigilar por un supervisor o familiar encargado, por ejemplo: cuando el monitoreo de detección se utiliza en la búsqueda de personas en eventos catastróficos,

Como parte del sistema eléctrico del proyecto es decisivo obtener señales óptimas para la alimentación y la lectura de datos de los sensores, por ello surge la necesidad de realizar el diseño personalizado en una maquina CNC con una baquelita que se somete al proceso de tallado, perforado y corte según el diseño eléctrico. La placa PCB permite mejorar la posición y estabilidad de los elementos electrónicos de manera eficiente.

La seguridad ante situaciones de emergencia, ante incendios e incluso evacuaciones, justifica parte de la autonomía del robot que sirve para la detección de personas. La capacidad de ubicar una persona es útil en el caso de cuidado infantil por desorientación, en monitoreo de seguridad, en especial en los lugares que el sistema satélite GPS no tiene cobertura.

El diseño de una aplicación Android se requiere para brindar comodidad al usuario supervisor del sistema, de esta manera se justifica la creación de una

interfaz gráfica que tenga un entorno visual entendible, legible y compatible para celular o Tablet, que permita presentar los datos obtenidos del recorrido realizado por el robot balancing.

1.5 Resultados Esperados

En la propuesta del trabajo de titulación para implementar en el robot balancing, se espera garantizar la confiabilidad, la veracidad y un buen funcionamiento del prototipo. Una vez que se tenga armada la estructura mecánica, se debe realizar la conexión de los materiales electrónicos. La etapa inicial es calibrar los sensores para ver su funcionamiento y comportamientos, al analizar las lecturas obtenidas a través de las placas, en función del rol como procesador o controlador del sistema.

Los movimientos del robot dependen del algoritmo inercial propuesto; esta es la primera etapa de evaluación para que el prototipo realice un recorrido recto, con curvas y con inclinación ideal. Si se encuentra frente a una pared, un mueble, un mesón, una silla, un escritorio u otro objeto, proceda a un cambio de dirección. El motivo principal del proyecto es detectar a una persona que se encuentre en la habitación.

Se pretende obtener el aviso de detección al momento en el que se establezca que se trata de una persona por medio de algoritmos de programación, cuando se tenga una detección facial el robot debe realizar el envío de información de imagen y distancia entre persona-robot para expresar la localización. Entre los primeros resultados se debe tener una estimación de los objetivos establecidos mediante pruebas experimentales para corregir los errores que se observen, tanto para el movimiento del robot y el monitoreo de la habitación estos procesos se visualizan en el computador y la información se observan en el celular o tablet donde se instale la aplicación Android.

CAPÍTULO II

2 . Marco Referencial

2.1 Marco Teórico

2.1.1 Sistema de Navegación Inercial

Un sistema de navegación inercial contiene principios de acuerdo a las leyes de movimiento, la primera y la segunda ley de Newton están presentes. Los fundamentos de esta técnica consisten en colocar sensores inerciales a un vehículo, con la finalidad de obtener medidas de sensores como el acelerómetro, el giroscopio e incluso un sensor de orientación magnética

Navegación autónoma

La navegación es el proceso en el que un robot ejecuta desplazamientos de forma autónoma en su lugar de trabajo y realiza recorrido de trayectorias definidas, la orientación inercial, mediante el uso de dispositivos que permitan realizar una retroalimentación al sistema del robot con respecto a la posición actual del robot, se detalla el uso de sensores interno o externos a la estructura del robot.

Prototipo móvil con sensores fijos a la estructura

Este tipo de sistemas se los conoce como strapdown, según el tipo de sistema de navegación inercial aplicada a un vehículo, se recalca el colocar los sensores a una estructura de manera fija. Gracias a las mejoras tecnológicas, la complejidad mecánica se reduce y la factibilidad de implementar sensores incrementa. Se complementa con cálculos para disminuir los errores de medición sensorial de acuerdo a los movimientos del vehículo. [14]

Robot balancing Ks0193

Es un kit para uso educativo de robótica similar a un segway, que permite construir un mini vehículo de autoequilibrio, su placa controladora es



Figura 1: Self-balancing kit robot
[15]

un Arduino, con una placa shield que contiene conexión con los sensores y módulos que contribuyen al equilibrio del robot.[15]

Sensores propioceptivos

El uso de sensores se encuentra en la placa del robot, que contiene un MPU 6050 y se conecta cada encoder para los motores del móvil. Estos elementos pertenecen al diseño del robot balancing y cumplen con la función de medir parámetros y mediante programación obtener la lectura en cantidades numéricas para lograr el equilibrio del robot.

Sensores exteroceptivos

Entre los elementos externos se tiene el uso de un sensor MPU 9250 y un sensor VL53L0X para lograr la autonomía del robot, con la lectura de estos sensores se pretende orientar al robot y detener sus movimientos de acuerdo a las condiciones establecidas.

Controlador de equilibrio

Este control permite que el robot se mantenga en equilibrio con dos ruedas en posición vertical mediante la retroalimentación del sistema con sensores y algoritmos lógicos que ajustan los motores continuamente.

Controlador de Velocidad

En el control de velocidad intervienen factores de inclinación en caso de movimientos delanteros y traseros, para establecer un controlador de velocidad para los motores.

Controlador de dirección

Con el control de dirección se tiene en cuenta las diferencias de velocidad en estado de rotación en ambos motores del robot. Los comandos de dirección se establecen con comunicación bluetooth y la aplicación Android.

Algoritmos matemáticos

Estos algoritmos matemáticos se aplican para el control del robot, este funciona como un regulador para un sistema dinámico que debe mantener movimientos coordinados y un filtrado que mejora la precisión al estimar estados en las lecturas sensoriales.

Control PID

Implementar control PID en un robot balancing con diseño en cascada permite una retroalimentación en parámetros de inclinación y de velocidad del robot, este proceso regula las perturbaciones del sistema analógico.

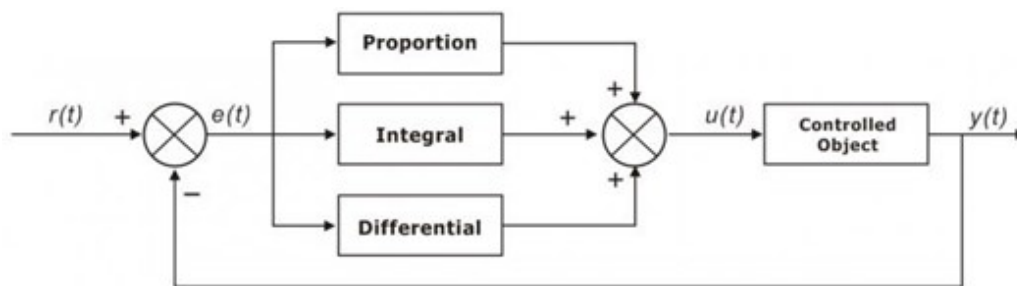


Figura 2: Diagrama de bloque con control pid.[15]

$$e(t) = r(t) - y(t) \quad (1)$$

- $e(t)$: Señal de error del ángulo.
- $r(t)$: Señal actual del sistema.
- $y(t)$: Señal de salida del sistema.

Control de Proporción

El control proporcional reacciona inmediatamente cuando el valor de K_p varía, es decir, cuando hay una desviación $e(t)$, el control ajusta la señal de los motores para disminuir la desviación. Según Key Studio, la fuerza de control depende netamente del coeficiente K_p ; si el valor es muy alto provoca oscilación e inestabilidad en el sistema; si es muy bajo, el control no funciona. Entonces, el valor de K_p debe ser ideal para el robot, acorde al tiempo de transición y error estacionario [15].

$$u_p(t) = K_p e(t) \quad (2)$$

- $u_p(t)$: Salida del controlador proporcional.
- K_p : Coeficiente proporcional.

Control Integral

El controlador integral acumula los valores de la señal de error al transcurrir el tiempo, permitiendo eliminar errores estacionarios en el sistema. Los efectos que ocasiona es eliminar el error estático, con valores altos de la constante integral se tiene una débil acumulación, esto disminuye el sobre impulso y brinda estabilidad del sistema, pero el proceso de corrección de error se hace lento, si el valor de T_i es bajo causaría oscilaciones y sobre impulsos [13].

$$u_i(t) = \frac{1}{T_i} \int e(t) dt \quad (3)$$

- $u_i(t)$: Salida del controlador integral.
- T_i : Tiempo integral o constante integral.

Control Diferencial

El control diferencial predice el comportamiento de desviación, calculando una tasa de cambio de la señal $e(t)$, de esta manera poder corregir la trayectoria del sistema anticipadamente. El control ajusta la respuesta dependiendo

de la señal de error, permitiendo reducir los sobre impulsos y brinda estabilidad al sistema. Si T_d : es mayor la reacción es rápida y con cambios bruscos de desviación, pero si la constante diferencial es menor el efecto de resistencia a desviación es débil.[15]

$$u_d(t) = T_d \frac{de(t)}{dt} \quad (4)$$

- $u_d(t)$: Salida del controlador diferencial.
- T_d : Tiempo diferencial o constante diferencial.

Filtro de Kalman

Este algoritmo establece soluciones cuando se tiene problemas para estimar en el sistema el estado con datos de entrada y salida de mediciones que contienen ruido e interferencias, el proceso de filtrado es mediante ecuaciones matemáticas en un sistema lineal, el estado se determina con diversos detalles que general el robot en relación al entorno de trabajo. En el estudio de Kalman se define como estados a los datos de lectura en pasado, presente y futuro de manera independiente, el sistema en análisis debe ser lineal en función a sus procesos de medición y transición.[14]

Ecuaciones generales

Predicción

$$x * k^- = A, x * k - 1 + B, u_k \quad (5)$$

$$P * k^- = A, P * k - 1 A^T + Q \quad (6)$$

Corrección

$$K_k = P_k^- H^T (H, P_k^- H^T + R)^{-1} \quad (7)$$

$$x_k = x_k^- + K_k(z_k - H, x_k^-) \quad (8)$$

$$P_k = (I - K_k H) P_k^- \quad (9)$$

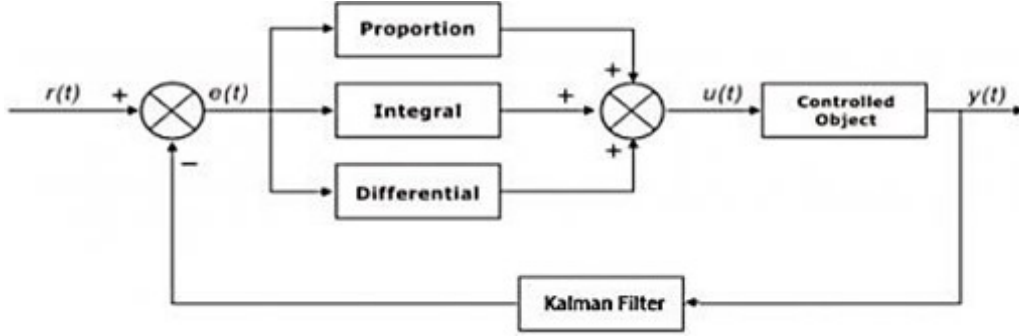


Figura 3: Diagrama de bloque con filtro de kalman

Domótica

La domótica es la tecnología que convierte a una vivienda normal en un hogar inteligente, este conjunto de tecnologías no solo se aplica en viviendas, sino también en edificios. El objetivo de este servicio es, mejorar la calidad de vida brindando seguridad, realizar un control mediante dispositivos móviles con acceso a internet permite que los usuarios controlen el sistema automático de la vivienda.[16]

Detección de Rostro

La detección de rostro es el proceso de análisis que consiste en capturar una imagen que contenga el rostro de una persona, en un ángulo de visión adecuado que le permita a la cámara obtener una imagen con buena resolución, para analizar las características e identificar donde se encuentra una cara o varias caras, la intensidad de luz, posición e inclinación, orientación, expresión facial son unas de las variables a controlar.[16]

Localización

La localización de personas mediante sensor de distancia, consiste en estimar distancia en unidades de longitud. método utilizado para entorno interno y domótica [17]

Aplicación Android

Una aplicación Android es diseñado como software para ejecutar en un sistema operativo Android, permite que los usuarios realicen distintas actividades en dispositivo móvil, estos incorporan datos de voz, mensajes, información de geolocalización e incluso video. Son eficientes por el tipo de comunicación inalámbrica que esta tecnología evolutiva integra para facilitar la interacción con el usuario.[18]

Comunicación Inalámbrica

Es el tipo de comunicación que permite la transmisión de datos sin cables, ni medios físicos para conectar a un emisor y un receptor. Utiliza dispositivos como antenas o drivers, las gestiones se realizan mediante software y protocolos de comunicación con señales de radiofrecuencia bajos y ancho de banda predeterminadas para las trasmisiones. El uso de esta tecnología de comunicación a distancias inferiores de transmisión para redes locales son el WIFI y el Bluetooth.[19]

Comunicación Bluetooth

Es la tecnología de comunicación inalámbrica que utiliza radiofrecuencia con banda de 2.4GHz para la transmisión de datos, este método de comunicación tiene la ventaja de un bajo consumo energético, una buena solución para comunicaciones de distancias pequeñas en interiores de una edificación.[19] El proceso que se realiza es de buscar dispositivos cercanos, seleccionar el dispositivo que se desea emparejar y conectar para definir la conexión entre dispositivos.

Cliente-Servidor

Es un paradigma conocido en arquitecturas de red en internet, consta de servidores que cuentan con servicios para el cliente que accede a la comunicación, la sincronización entre dispositivos, el objetivo de la comunicación es obtener los datos de información recopilados por la Raspberry y enviados al servidor para mantener la información actualizada.[19]

2.1.2 Componentes del proyecto

2. 1. 2. 1 Componentes físicos

MPU6050

El MPU6050 es un sensor inercial que combina un acelerómetro tri-axial y un giroscopio tri-axial, proporcionando 6 grados de libertad para medir la aceleración lineal y la velocidad angular del robot móvil. La medición precisa de estos parámetros es fundamental para estimar la orientación y posición del robot en el espacio tridimensional. La comunicación se realiza mediante el protocolo I2C, que permite una transferencia eficiente de datos hacia el microcontrolador [15].

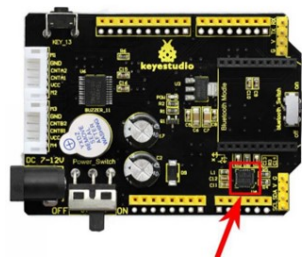


Figura 4: Sensor mpu 6050 [15].

Módulo Bluetooth HC-06

El módulo Bluetooth HC-06 facilita la comunicación inalámbrica entre el robot móvil y dispositivos externos, como teléfonos inteligentes o computadoras. Utiliza un protocolo Bluetooth Serial Port Profile (SPP), permitiendo

la transmisión y recepción de datos en tiempo real. Su antena integrada y compatibilidad con sistemas Android posibilitan el control remoto del robot, la recepción de comandos y la visualización de información de estado. Este módulo se integra al *shield* del microcontrolador y cuenta con un interruptor físico para encender o apagar la comunicación, garantizando un manejo sencillo del sistema inalámbrico [15].



Figura 5: Módulo bluetooth hc-06[15].

Encoders

Los encoders rotatorios se utilizan para medir la rotación y velocidad angular de las ruedas del robot. Esta información es esencial para calcular el desplazamiento y estimar la posición relativa del robot en su entorno. Los datos obtenidos se incorporan al filtro de Kalman, permitiendo una fusión multisensorial que mejora la precisión de la navegación y compensando errores inherentes a cada sensor individual [27].



Figura 6: Motores con sensor encoder.[15]

Arduino UNO

El Arduino UNO actúa como el controlador principal del sistema, encargado de ejecutar el algoritmo de control PID y el filtro de Kalman, así como de gestionar la adquisición de datos de todos los sensores integrados. Este

microcontrolador es responsable de enviar las señales de control a los motores DC para mantener la estabilidad y el equilibrio del robot en tiempo real [27].

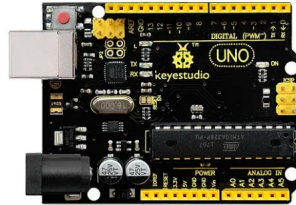


Figura 7: Arduino uno keystudio.[15]

Motores DC

Los motores DC proporcionan el torque necesario para el movimiento del robot. Mediante señales PWM generadas por el controlador PID, estos motores ajustan su velocidad para corregir la inclinación y mantener el equilibrio dinámico del robot. La caja reductora amplifica el torque y reduce la velocidad angular, mejorando la precisión del control y la capacidad de respuesta ante las perturbaciones externas [27].



Figura 8: Motores dc.[15]

Batería LiPo

La batería de polímero de litio (LiPo) es la fuente principal del sistema, caracterizada por su alta densidad energética y bajo peso. Estas propiedades la hacen ideal para aplicar en robots con autonomía y que requieran reducción

de masa. La batería alimenta tanto el microcontrolador como los actuadores y sensores del robot, asegurando un funcionamiento prolongado [3].



Figura 9: Bateria lipo.

MPU9250

El MPU9250 es una unidad de medición inercial avanzada que integra acelerómetro, giroscopio y magnetómetro, proporcionando un total de 9 grados de libertad. La inclusión del magnetómetro permite obtener datos de orientación absoluta mediante la detección del campo magnético terrestre, lo cual es vital para la navegación en entornos donde el GPS no está disponible o es poco fiable. Los datos de este sensor enriquecen el filtro de Kalman, mejorando la precisión del sistema de localización [14].



Figura 10: Sensor mpu 9250.

Sensor VL53L0X

Este sensor basado en tecnología Time-of-Flight (ToF) permite medir distancias con alta precisión mediante el tiempo que tarda un pulso de luz infrarroja en reflejarse en un objeto y regresar al sensor. En el robot móvil, se

utiliza para detectar obstáculos y objetos cercanos, posibilitando la activación de procesos automáticos como la detección facial o la reorientación en el trayecto [15].



Figura 11: Sensor vl53l0x.

LED Infrarrojo

El LED infrarrojo funciona como emisor de señales para sistemas de reconocimiento y detección de objetos cercanos. También puede emplearse como baliza para facilitar el seguimiento de rutas o la comunicación en redes de sensores, utilizando señales no visibles al ojo humano, lo cual reduce interferencias externas [15].



Figura 12: Leds infrarrojos.

Power Bank

La Power Bank funciona como fuente de alimentación secundaria o auxiliar para componentes que requieren estabilidad energética prolongada, como la Raspberry Pi 3 y la cámara web. A diferencia de baterías LiPo que alimentan los motores y controladores directamente, la Power Bank proporciona corriente constante a 5V con protección contra sobrecarga y cortes

súbitos, evitando reinicios inesperados durante la ejecución de algoritmos de navegación, reconocimiento facial o procesamiento de imágenes. Esta división energética mejora la eficiencia del sistema, especialmente en aplicaciones críticas de localización de personas y gestión domótica, donde la continuidad operativa es esencial para la respuesta en tiempo real [10].



Figura 13: Power bank

Cámara Web.

La cámara web se utiliza para la captura de imágenes y video en tiempo real, con aplicaciones en detección facial y reconocimiento de personas. El procesamiento de las imágenes se realiza en la Raspberry Pi, donde algoritmos de visión artificial analizan y extraen información relevante para el sistema de domótica y seguridad [10].



Figura 14: Camara web.

Raspberry Pi 3

La Raspberry Pi 3 actúa como una unidad de procesamiento embebido avanzada, encargada del análisis de imágenes, detección facial y gestión de datos del sistema. Se comunica con el Arduino UNO para sincronizar even-

tos y transmitir comandos, además de interactuar con la aplicación móvil Android para el control remoto y la visualización de información [10].

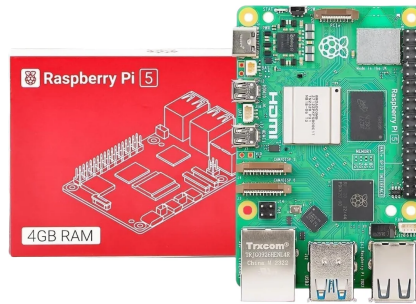


Figura 15: Raspberry pi 3.

CNC

Es una maquina utilizada como herramienta que es controlada por computadora, habilitada para trabajar con materiales como madera, baquelita, fomi, plástico, MDF, aluminio, y otros más, sus movimientos se desplazan en 3 ejes, con motores que sirven para grabar, perforar y cortar.[7]

Baquelita

La baquelita es el material utilizado para la fabricación de las placas de circuito impreso (PCB) personalizadas que soportan sensores y componentes electrónicos externos, como el MPU9250 y el VL53L0X. Sus propiedades mecánicas y térmicas la hacen adecuada para la fabricación mediante CNC, proporcionando un soporte estructural confiable para el sistema de navegación [7].

2. 1. 2. 2 Componentes lógicos

Arduino IDE

El Arduino Integrated Development Environment (IDE) es una plataforma de desarrollo abierta y multiplataforma diseñada para facilitar la progra-



Figura 16: Baquelita para cnc.

mación de placas Arduino. Ofrece una interfaz gráfica intuitiva que permite escribir, compilar y cargar código fuente en microcontroladores, además de gestionar bibliotecas y ejemplos. Su simplicidad y compatibilidad con múltiples sistemas operativos lo hacen ideal para prototipos rápidos en robótica y sistemas embebidos. El IDE soporta lenguajes basados en C/C++, brindando control eficiente sobre el hardware y facilitando la depuración mediante la monitorización serial en tiempo real [15].

Raspbian

Raspbian es un sistema operativo derivado de Debian, diseñado especialmente para aprovechar las características de un microprocesador Raspberry Pi, se reconoce por ser ligero y confiable, esto facilita ejecutar aplicaciones avanzadas relacionadas con procesamiento de imágenes, comunicación y control de sensores. Al estar basado en Linux, este permite integrar herramientas de desarrollo como Python, OpenCV son fundamentales para los algoritmos de visión artificial y sistemas de navegación autónoma. Además soporta interfaces gráficas para el uso mediante comandos para adaptarse a diferentes perfiles de usuario [10].

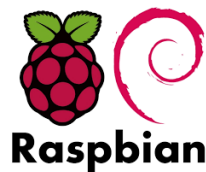


Figura 17: Sistema operativo raspbian

VNC

Virtual Network Computing (VNC) es una solución de acceso remoto que permite controlar la interfaz gráfica de la Raspberry Pi desde cualquier otro dispositivo conectado a la red. Este software transmite el escritorio completo del sistema operativo, posibilitando la supervisión, depuración y manipulación del robot sin necesidad de hardware periférico adicional, como teclado o monitor. VNC resulta fundamental en proyectos embebidos para realizar ajustes en tiempo real y facilitar el desarrollo remoto, incrementando la flexibilidad y eficiencia durante la etapa de implementación y pruebas [10].



Figura 18: Virtual network computing (VNC)

Python

Python es un lenguaje de programación de alto nivel, que se distingue por su versatilidad. permite trabajar bajo diferentes paradigmas de programación. En el entorno de la Raspberry Pi, es empleado para implementar algoritmos relacionados con navegación, detección facial. localización y comunicación entre dispositivos de forma remota. Por su extensa comunidad de desarrolladores, dispone de múltiples librerías especializadas, como NumPy y OpenCV, que agilizan la creación de aplicaciones en sistemas de inteligencia artificial. Así mismo permite la interacción con hardware mediante módulos a través de PySerial, facilitando la comunicación entre sensores y controladores [10].

OpenCV

OpenCV (Open Source Computer Vision Library) es una librería de código abierto diseñada para el procesamiento de imágenes y visión artificial. Esta biblioteca ofrece un amplio conjunto de algoritmos para detección, reconoci-

miento, segmentación y análisis de imágenes en tiempo real. En el proyecto, OpenCV se emplea para la detección y reconocimiento facial mediante el procesamiento de datos capturados por la cámara web, permitiendo al sistema identificar personas y responder ante eventos relevantes. Su eficiencia y soporte para múltiples lenguajes lo convierten en una herramienta esencial en robótica y sistemas de automatización [10].



Figura 19: Open source computer vision library (opencv)

Pyserial

Pyserial es un módulo de Python que facilita la comunicación serial entre la Raspberry Pi y otros dispositivos, como el Arduino UNO. Permite establecer conexiones a través de puertos serie, gestionar el envío y recepción de datos, y controlar el flujo de información en tiempo real. Esta biblioteca es vital para la sincronización de sensores y actuadores en sistemas embebidos, ya que simplifica la transferencia de comandos y la recopilación de mediciones, asegurando la correcta interacción entre los diferentes componentes del sistema [10].

Arduino Core

Arduino Core comprende el conjunto de bibliotecas y herramientas de software que permiten programar microcontroladores Arduino utilizando lenguajes como C y C++. Estas bibliotecas ofrecen funciones para la gestión de hardware básico, tales como entradas y salidas digitales, comunicación serial, temporizadores y manejo de interrupciones. Su arquitectura modular facilita la expansión y personalización del firmware, permitiendo al desarrollador optimizar el rendimiento del robot y garantizar la respuesta adecuada a las señales de los sensores y actuadores [15].

Thonny

Thonny es un entorno de desarrollo integrado (IDE) diseñado para facilitar la programación en Python, se destaca por su simplicidad y facilidad de uso. lo que es adecuado para entornos educativos y usuarios con menor experiencia. en la Raspberry, permite desarrollar, ejecutar y probar scripts de manera eficiente, contribuye a agilizar la implementación del sistema. entre sus funciones se encuentra la ejecución paso a paso y la visualización del estado de las variables, esto favorece al análisis de comportamiento el programa y la identificación de errores en etapas tempranas en especial para aplicaciones en proyectos de robótica y control [10].

Proteus

Proteus es un software profesional de simulación electrónica que permite diseñar, simular y probar circuitos electrónicos y sistemas embebidos antes de su implementación física. En el contexto de este proyecto, Proteus se utiliza para validar esquemas de controladores y sensores, verificando la lógica de funcionamiento y detectando posibles fallos en el diseño. Su capacidad para simular microcontroladores y periféricos digitales y analógicos reduce costos y tiempos de desarrollo, asegurando un prototipo funcional y optimizado [7].

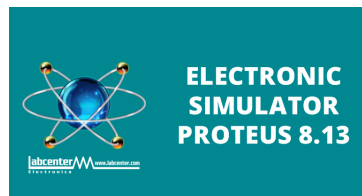


Figura 20: Software proteus)

FlatCAM

FatCAM es una herramienta de código abierto orientada a la conversión de archivos Gerber, ampliamente utilizados en el diseño de circuitos impresos (PCB), en trayectorias de mecanizado compatibles con fresadoras CNC. En el presente proyecto, esta aplicación se emplea para el diseño y preparación de placas de baquelita personalizadas, donde se integran los sensores

y demás componentes electrónicos. Su uso permite obtener un proceso de fabricación más preciso y eficiente, garantizando una adecuada distribución de los elementos en la placa [7].



Figura 21: Herramienta flatcam)

Mach3Mill

Mach3Mill es un software de control numérico computarizado (CNC) utilizado para la operación de máquinas fresadoras, permitiendo ejecutar procesos de fabricación mediante el control preciso del movimiento del cabezal de mecanizado. En este proyecto, se emplea en conjunto con las trayectorias generadas por FlatCAM, lo que facilita la elaboración de placas de circuito impreso y estructuras mecánicas con un alto nivel de exactitud. Esta integración resulta fundamental para asegurar un adecuado ensamblaje y la estabilidad del sistema de navegación implementado [7].



Figura 22: Software de control numérico computarizado (cnc)

Android Studio

Android Studio es el entorno oficial para el desarrollo de aplicaciones móviles para Android, que integra herramientas para diseñar interfaces gráficas, escribir código en Java, Kotlin o C++, y realizar pruebas y depuraciones. En este proyecto, Android Studio se utiliza para crear la aplicación que controla

el robot de manera remota mediante Bluetooth, facilitando la interacción del usuario con el sistema y la visualización de datos en tiempo real [13].



Figura 23: Android studio

Kotlin

Kotlin es un lenguaje de programación moderno y expresivo, diseñado para interoperar plenamente con Java, y es el lenguaje oficial para el desarrollo de aplicaciones Android. Destaca por su sintaxis concisa, seguridad frente a errores comunes y soporte para programación funcional. Su uso en la aplicación móvil del proyecto permite desarrollar un software robusto y eficiente para el control y monitoreo del robot, mejorando la experiencia del usuario [26].

Firebase

Firebase es una plataforma de servicios en la nube que integra diversas funcionalidades, como bases de datos en tiempo real, autenticación de usuarios, almacenamiento y mensajería. En el desarrollo de este proyecto, se emplea para gestionar la comunicación entre la aplicación móvil y el robot, permitiendo mantener actualizada la información relacionada con el estado y control del sistema. Asimismo, su utilización simplifica la implementación del backend, evitando la necesidad de configurar y mantener servidores propios [18].

2.2 Metodología del Proyecto

En el presente trabajo se fundamenta la investigación del tema y se establece una metodología para el avance del proyecto de acuerdo a la figura 24 se determina que el tipo de trabajo contiene un enfoque experimental, con

un alcance exploratorio esto se debe a la existencia de textos con información de estudios o implementaciones parecida y complementaria, definiendo que la creación del prototipo tiene fin evaluativo mas no comercial.

En el desarrollo del proyecto se contempla la aplicación de aprendizaje en el área de electrónica y telecomunicaciones como base principal para la implementación y funcionamiento del prototipo, se define que el trabajo de titulación permite aplicar de manera práctica los conocimientos adquiridos en el transcurso de la carrera y la investigación como auto-aprendizaje para temas de navegacion inercial, visión artificial y aplicaciones para dispositivos Android.

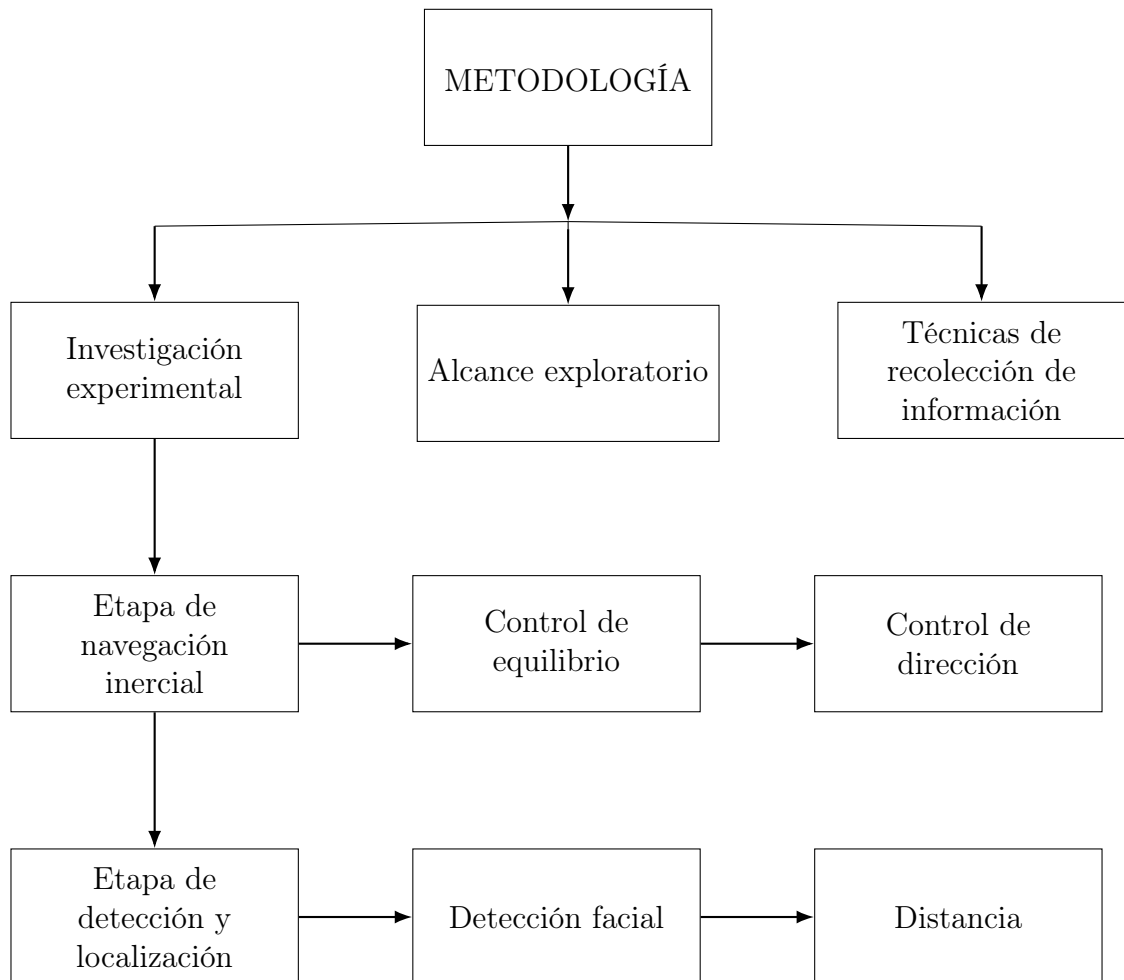


Figura 24: Diagrama de metodología del proyecto

2.2.1 Investigación experimental

El proyecto de investigación se define como experimental, a causa de los procesos en los que se pretende observar efectos o reacciones de acuerdo a la manipulación del autor y se tiene como propósito el análisis de datos experimentales de acuerdo a las variables que intervienen, se menciona un alcance exploratorio de acuerdo a los métodos que se realizan en el desarrollo del proyecto.[20] Cuando se tenga la implementación del prototipo se podrá cumplir con las pruebas experimentales en varias etapas para obtener la movilidad del robot, luego se plantean pruebas específicas para la navegación inercial, la detección facial de personas, las lecturas de distancia y también la comunicación con dispositivo móvil.

Etapas Navegación Inercial

Control de equilibrio

En las pruebas experimentales inerciales se busca el correcto funcionamiento del sensor MPU 6050 para lectura de aceleración y de giroscopio, que sirven para el equilibrio vertical del robot balancing. Además del uso del encoder para lecturas de pulsos en tiempo transcurrido, se utiliza la velocidad en rpm con la dirección de movimiento del robot de adelante y atrás, incluso para giro de izquierda o derecha. El experimento para demostrar el equilibrio se basa en comparar lectura de sensores en ángulo real, la lectura natural, aplicando control PID con el filtro de Kalman, este método me ayuda a analizar el ángulo necesario para mantener el equilibrio del robot con o sin perturbaciones.

Control de dirección

En la segunda etapa se detalla los procesos experimentales para cumplir una navegación inercial estable, para esto se observa la lectura obtenida por el sensor MPU 9250, esto es el dato del ángulo yaw, la comparación entre grados, la lectura natural y la lectura con filtro complementario, estos valores me indican que tan cierto es el rumbo que debe tomar el robot para dirigirse

al norte magnético.

Etapas de detección y localización

Detección facial

Otra etapa de los procesos experimentales está relacionado a la detección facial en esta se incluye técnicas de visión artificial en algoritmos que se implementa con lenguaje de programación, se tiene el uso de una cámara y una tarjeta Raspberry Pi 3 que son acoplados al robot balancing y para estas pruebas se comparan el nivel de confianza con cuatro rostros diferentes, en un rango de distancia entre 20 y 60 centímetros, para validar este experimento se espera tener un mínimo a nivel de confianza de 0.95.

Distancia

En las pruebas experimentales para obtener la ubicación de una persona se utiliza el sensor VL53L0X que proporciona la medida de longitud entre persona y robot, este proceso permite comparar la distancia real con la lectura natural y la lectura con filtro EMA, las señales muestran un alcance estable entre 20 y 40 centímetros.

2.2.2 Alcance exploratorio

Una investigación con alcance exploratorio es la que se aplica métodos cuantitativos o cualitativos. En los procesos para analizar los datos el método cuantitativo es eficiente porque permite identificar cambios o características. Al aplicar un método cualitativo se obtiene la identificación subjetiva en interacciones de las variables que intervienen. El propósito de aplicar estos métodos es obtener soluciones factibles de acuerdo a los objetivos establecidos en el proyecto.[21]

2.2.3 Técnicas de recolección de información

El aplicar técnicas de recolección de información para este proceso es de alta relevancia, en este proyecto se establece la técnica experimental y además

análisis de documentos. Al realizar los experimentos se aplica el método de evaluar el comportamiento de variables y al aplicar el método de análisis de documentos y registros, se considera la revisión de textos como informes, tesis de titulación, entre otros.[22]

La búsqueda de referencias en artículos e investigaciones realizadas es primordial, por ello el tema principal es la navegación inercial aplicados en un robot móvil, se encontró información de controles manuales o control autónomo con recorrido establecido por el usuario, en otro tema citado está el estudio del algoritmo de navegación aplicando filtro complementario como método para corregir y obtener precisión en la navegación.

Ademas se tiene la búsqueda de investigaciones con temas relacionados a la domótica y seguridad para personas que por algún motivo deben estar en supervisión por la persona encargada de su vigilancia. Se recolecta también información de plataforma para crear una aplicación compatible en dispositivos Android y se finaliza con enumerar los materiales y elementos a utilizar.

Con este trabajo se obtiene un robot balancing con un buen nivel de estabilidad, junto a la implementación de un sistema de navegación inercial basado en un sensor MPU integrado al prototipo. Asimismo, se desarrolla un proceso de detección facial que alcanza un nivel de confianza satisfactorio, aunque con limitaciones en el tiempo de procesamiento.

CAPÍTULO III

3 . Desarrollo de la Propuesta

En la propuesta para que el robot balancing cumpla con la misión establecida, se debe empezar con el control de equilibrio. Para tomar un rumbo específico se aplica el control de dirección de las ruedas del prototipo al mismo tiempo que debe estar equilibrado, este inicia el recorrido en línea recta hasta cambiar el sentido de dirección hacia al polo magnético. Para el proceso de detección de personas depende del algoritmo de visión artificial y de la lectura de un sensor de distancia para esto se utiliza el detector de obstáculo que activa el proceso de detección de rostro con esto se indica si una persona está cerca del robot mostrando valores y mensaje informado al usuario.

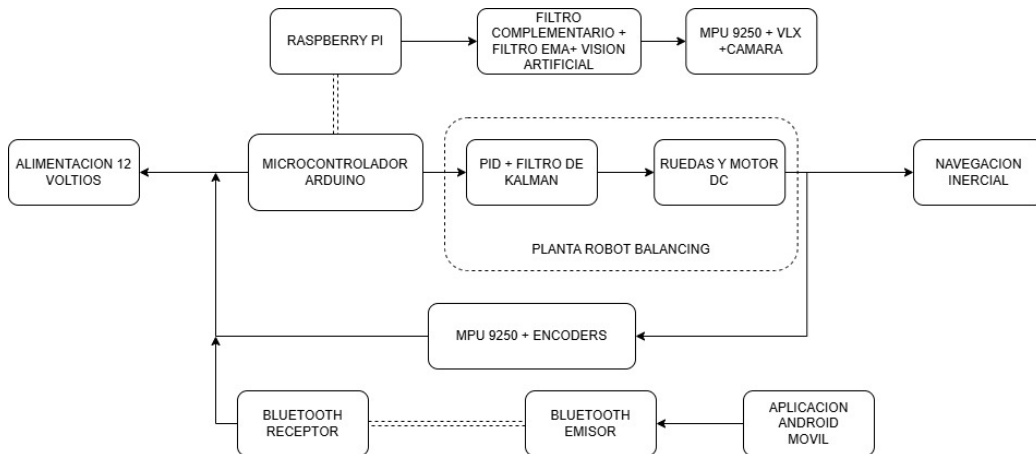


Figura 25: Arquitectura del sistema

3.1 Diseño de la Propuesta

El proyecto tiene como diseño principal a un robot balancing que está interconectado a bloques principales, en la Figura 26, se especifica el uso de un power bank para la alimentación del sistema que permite el funcionamiento de los módulos de control de equilibrio, el control de dirección, la detección y una aplicación Android.

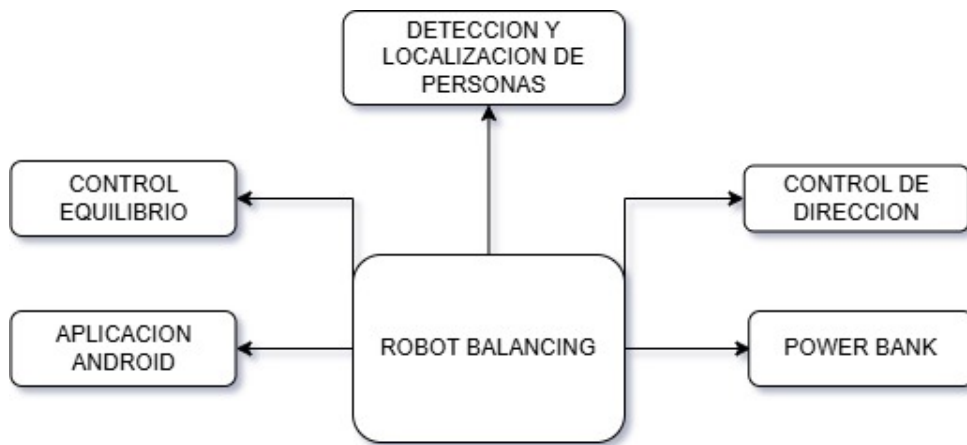


Figura 26: Diagrama de interconexión

3.1.1 Diseño de navegación inercial

En el diseño para aplicar navegación inercial en el robot balancing se procede a validar el algoritmo de equilibrio del robot y analizar el comportamiento del prototipo de manera manual y para el control de dirección se aplican técnicas de filtrado para que las lecturas angulares no tengan ruido, de esta manera establecer un rumbo sin distorsión hacia al norte magnético.

Diseño del control de equilibrio en cascada

El control de equilibrio se expresa en el diagrama de la Figura 27, donde tiene como principal elemento un arduino que cumple la gestión de manejar los motores DC, con medidas proporcionales a los datos de los sensores MPU 6050 y los encoders, el microcontrolador también gestiona la comunicación inalámbrica por medio del módulo bluetooth. Se emplean los algoritmos de PID y Filtro de Kalman como una dinámica de corrección para el equilibrio y la velocidad aplicada en los motores.

Teoría de péndulo invertido

El funcionamiento del péndulo invertido es un sistema inestable, el objetivo es controlar el equilibrio y permanecer en posición vertical. El diseño del prototipo es un péndulo con su masa sobre dos ruedas, el robot a medida

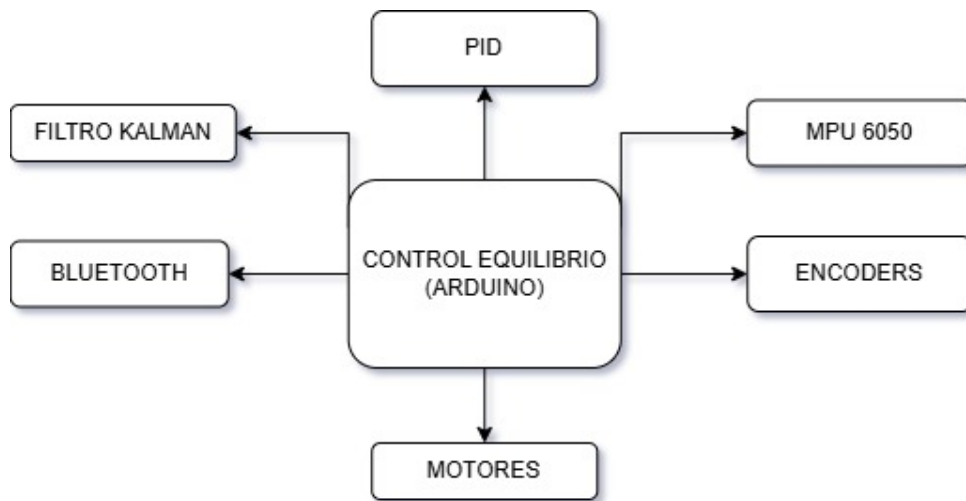


Figura 27: Diagrama de equilibrio

que se desplaza debe compensar con una fuerza contraria al movimiento de caída para mantener su centro de gravedad y equilibrio.[3]

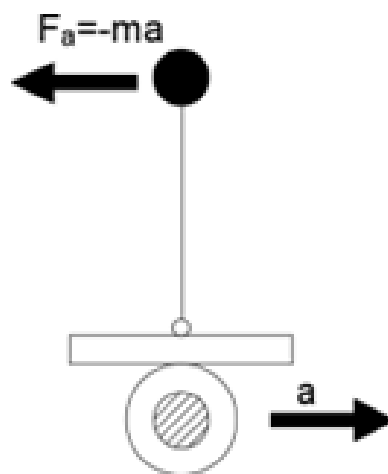


Figura 28: Diagrama representativo del péndulo invertido.[15]

De acuerdo al análisis de equilibrio de fuerzas que actúan sobre el péndulo invertido, la siguiente ecuación describe las fuerzas involucradas en el sistema:

$$mg \cdot \sin \theta = ma \cdot \cos \theta \quad (10)$$

donde:

- mg representa la fuerza gravitatoria, dirigida verticalmente hacia abajo desde el centro de masa del péndulo,
- ma es la fuerza de aceleración generada por el desplazamiento de las ruedas,
- m es la masa total del robot,
- g es la aceleración debida a la gravedad,
- a es la aceleración lineal del sistema,
- θ es el ángulo de inclinación del péndulo respecto a la vertical.

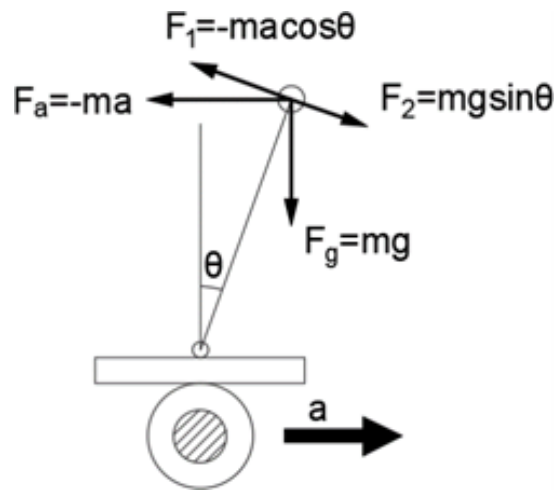


Figura 29: Dinámica del péndulo invertido.[15]

Con el equilibrio del robot se busca relacionar el ángulo de inclinación y la acción correctiva aplicada para mantener una posición vertical. Estos valores se tienen a partir de la medición proporcionada por el sensor MPU6050, se establece un modelo proporcional:

$$\alpha = k_1 \theta \quad (11)$$

Para este modelo, la aceleración angular α es directamente proporcional al ángulo de inclinación, cuando k_1 que es el parámetro que regula la respuesta

del sistema. Este enfoque no garantiza la estabilidad completamente, ya que la ausencia de un término derivativo provoca oscilaciones en el movimiento del robot. Para mitigar este efecto, se implementa un control proporcional derivativo (PD), que incorpora la velocidad angular ω para amortiguar las oscilaciones. La ecuación de control se modifica a:

$$\alpha = k_1\theta + k_2\omega \quad (12)$$

Donde k_2 es el coeficiente de ganancia derivativa, y ω es la velocidad angular obtenida del giroscopio del sensor MPU6050. Esta formulación permite ajustar la aceleración del sistema en función tanto del ángulo como de su velocidad de cambio, logrando una respuesta más estable y eficiente para el equilibrio del robot balancing.

Control PD para el bucle de ángulo de inclinación

Para mantener el equilibrio vertical del robot balancing, es necesario aplicar una fuerza correctiva que contrarreste cualquier inclinación detectada. Esto permite evitar caídas ante perturbaciones externas. El sistema utiliza un controlador Proporcional-Derivativo (PD), el cual toma en cuenta dos variables principales: el ángulo de inclinación y la velocidad angular del robot, ambas medidas obtenidas con el sensor inercial MPU6050.[29] La ecuación que representa la aceleración angular es:

$$\alpha = k_1\theta + k_2\omega \quad (13)$$

Donde:

- θ : es el ángulo de inclinación,
- ω : es la velocidad angular,
- k_1 y k_2 : son constantes de ganancia que ajustan la contribución proporcional y derivativa.

La expresión general del controlador PD para el equilibrio dinámico se modela

como:

$$PD = K_p \cdot \theta + K_d \cdot \omega' \quad (14)$$

donde:

- K_p es la ganancia proporcional que actúa sobre el ángulo de inclinación,
- K_d es la ganancia derivativa aplicada sobre la velocidad angular filtrada,
- ω' representa la velocidad angular corregida a través del filtro de Kalman.

Además, se considera un ángulo de referencia θ_0 , que idealmente corresponde a 0° , y se define el ángulo total como:

$$\theta = \theta_{post} + \theta_0 \quad (15)$$

Así también, se ajusta la velocidad angular usando la estimación filtrada del Kalman, restando el término de incertidumbre Q :

$$\omega' = \omega - Q \quad (16)$$

Por tanto, la ecuación final del controlador PD, con ambas correcciones aplicadas, se expresa como:

$$PD = K_p \cdot (\theta_{post} + \theta_0) + K_d \cdot (\omega - Q) \quad (17)$$

La constante K_p regula la magnitud de la fuerza proporcional al ángulo de inclinación θ , mientras que K_d ajusta la fuerza derivativa en función del cambio en la velocidad angular ω , lo cual contribuye a amortiguar las oscilaciones del sistema y mejorar la estabilidad dinámica del robot.[29]

Control PI para bucle de velocidad

Para el ajuste dinámico de la velocidad del robot, se utiliza una estrategia de retroalimentación basada en los encoders montados en cada motor. Esta información es esencial para mantener el equilibrio incluso ante perturbaciones externas o desplazamientos en superficies inclinadas. El sistema de

control ajusta la velocidad para conservar la estabilidad del robot, permitiendo también la navegación precisa en condiciones variables.

Con el fin de reducir el ruido presente en la señal obtenida de los encoders y mejorar la estabilidad del control de velocidad, se aplica un filtro complementario de primer orden. Este filtro suaviza las fluctuaciones al ponderar el valor anterior con el valor actual. La ecuación utilizada para aplicar dicho filtro es:

$$v_{FilterOld} = v_{FilterOld} \cdot 0,7 \quad (18)$$

Esta fórmula conserva el 70 % de la velocidad previamente filtrada. A continuación, se calcula la velocidad actual del robot a partir de la lectura de pulsos de los encoders del motor izquierdo ($pulso_i$) y derecho ($pulso_d$):

$$v = (pulso_i + pulso_d) \cdot 1,0 \quad (19)$$

Esta expresión convierte la suma de los pulsos en una magnitud proporcional a la velocidad. Posteriormente, se actualiza la señal filtrada de velocidad incorporando un 30 % del valor recién medido:

$$v_{filter} = v_{FilterOld} + (v \cdot 0,3) \quad (20)$$

Esta combinación ponderada de valores anteriores y actuales (70/30) permite reducir el ruido sin perder la respuesta dinámica del sistema ante cambios reales de velocidad. El control de velocidad se implementa mediante un controlador Proporcional-Integral (PI), cuya ecuación general es:

$$PI = k_p \cdot v_{error} + k_i \cdot v_{integral} \quad (21)$$

donde:

- v_{error} representa el error de velocidad instantáneo entre la velocidad deseada y la medida,
- $v_{integral}$ es la acumulación del error en el tiempo para corregir desviaciones sostenidas,

- k_p y k_i son las ganancias proporcional e integral, respectivamente.

La acción proporcional que actúa sobre el error en la velocidad, permite una respuesta mejorada ante desviaciones. Además la componente integral compensa errores temporales acumulados, disminuyendo el error en estado estacionario. La capacidad del sistema es mantener la velocidad deseada, incrementando las correcciones frente a perturbaciones y variaciones en la dinámica del robot móvil.

Control PD para bucle de dirección

El sistema de navegación del robot también contempla el ajuste dinámico de la dirección, especialmente útil cuando se presentan cambios de masa, como la incorporación de objetos adicionales o carga variable. Estas variaciones pueden generar desequilibrio o desviaciones en la trayectoria, por lo que se implementa un controlador Proporcional-Derivativo (PD) que ajusta el PWM de cada motor de manera diferencial.

Los giros del robot para que se dirija hacia atrás o adelante o de izquierda a derecha empiezan con el control de dirección este se basa en el cálculo de la velocidad diferencial entre ambos motores, así como en la velocidad angular medida en el eje Z por el giroscopio del sensor MPU6050. De esta forma, el sistema puede corregir desviaciones de rumbo de forma autónoma. La ecuación de control PD utilizada para el ajuste de dirección se expresa como:

$$PD = K_p \cdot v_{dif} + K_d \cdot \omega' \quad (22)$$

donde:

- K_p es la ganancia proporcional que ajusta el PWM según la diferencia de velocidad entre las ruedas,
- K_d es la ganancia derivativa que actúa sobre la velocidad angular en el eje z ,
- v_{dif} es la velocidad diferencial calculada entre ambos motores,
- ω' es la velocidad angular filtrada del eje z , proporcionada por el giroscopio.

El equilibrio necesita un control que permita mantener un recorrido autónomo aunque este expuesto a perturbaciones externas y cambios en la dinámica del sistema. La retroalimentación de la velocidad diferencial y la velocidad angular, se logra con correcciones para generar lecturas precisas para la orientación del robot, lo que optimiza su capacidad de maniobra y mejora el desempeño global del sistema de navegación.

Filtro de Kalman aplicado al robot

El proceso para la estimación de estado se desarrolla por etapas estas se conocen como etapa de predicción y etapa de corrección, este filtro de Kalman se utiliza para suavizar la señal del ángulo calculado, este interviene en el equilibrio del robot, con el objetivo de disminuir el efecto de ruido, errores o interferencias que se pueda presentar en las lecturas del sensor MPU6050.[32]

Etapas de predicción

Durante esta etapa, se estima el estado futuro del sistema (ángulo) en función del estado anterior y del modelo dinámico. Esta predicción se realiza utilizando la velocidad angular proporcionada por el giroscopio, ajustada mediante el intervalo de muestreo Δt . La ecuación utilizada para predecir el ángulo es:

$$\theta_{pred} = \theta_{prev} + (\omega' - q) \cdot \Delta t \quad (23)$$

Donde:

- θ_{pred} : ángulo de predicción,
- θ_{prev} : ángulo estimado en el instante anterior,
- ω' : velocidad angular medida por el giroscopio,
- q : corrección estimada del filtro de Kalman,
- Δt : intervalo de tiempo entre mediciones.

La predicción también se extiende a la incertidumbre del sistema, modelada por la matriz de covarianza del error. Esta se actualiza con la siguiente

expresión:

$$P_{pred} = P_{prev} + Q \cdot \Delta t \quad (24)$$

Donde:

- P_{pred} : es la covarianza de predicción,
- P_{prev} : es la covarianza anterior,
- Q : es la covarianza del ruido del sistema.

Etapas de corrección

En esta etapa de corrección las medidas deben estar actualizadas constantemente para la estimación previa del estado, con el fin de mejorar la precisión del sistema. El ángulo estimado es ajustado a partir de la lectura del acelerómetro y compensado en desviaciones acumuladas de la predicción. Se calcula el error de medición, definido como la diferencia entre el ángulo obtenido por el acelerómetro y la estimación previa:

$$Error = \theta_{ac} - \theta_{prev} \quad (25)$$

Donde:

- θ_{ac} : ángulo medido por el acelerómetro,
- θ_{prev} : ángulo previamente obtenido.

La ganancia de Kalman K se calcula con la siguiente ecuación:

$$K = \frac{P \cdot C}{R + P \cdot C} \quad (26)$$

Donde:

- K : ganancia de Kalman,
- P : matriz de covarianza del error,
- C : matriz de observación (generalmente 1),

- R : covarianza del ruido de la medición.

Al obtener esta ganancia, se actualiza el ángulo estimado (estado posterior):

$$\theta_{post} = \theta_{post} + K \cdot Error \quad (27)$$

Y cuando, se actualiza la covarianza asociada a la estimación se tiene la siguiente ecuación:

$$P_{act} = (I - K \cdot C) \cdot P_{pred} \quad (28)$$

Donde:

- P_{act} : es la covarianza actualizada,
- I : es la matriz identidad.
- K : es la ganancia de Kalman,
- C : es la matriz de observación (generalmente 1),
- P_{pred} : es la covarianza de predicción.

Este proceso iterativo de predicción y corrección permite obtener un ángulo confiable, fusionando la información de dos sensores: acelerómetro y giroscopio, lo cual es esencial para el control preciso del equilibrio y navegación del robot móvil.

3.1.2 Diseño del control de dirección

El diagrama representativo de la Figura 30 muestra la interconexión de los componentes para el control de dirección, el principal es la placa Raspberry pi 3, el sensor MPU 9250 que esta conectado directamente a los puertos I2C, su función es brindar información de ángulos con respecto al norte magnéticos como un magnetómetro digital. Se aplica un filtro complementario en las lecturas del sensor inercial para suavizar la señal, el algoritmo utiliza lenguaje Python en un sistema Raspbian.

La aplicación android MiaMav para un dispositivo móvil sirve para realizar rutinas de giro, asignando dirección de adelante o atrás y de izquierda a derecha en modo manual, antes de aplicar el sistema de navegación inercial.

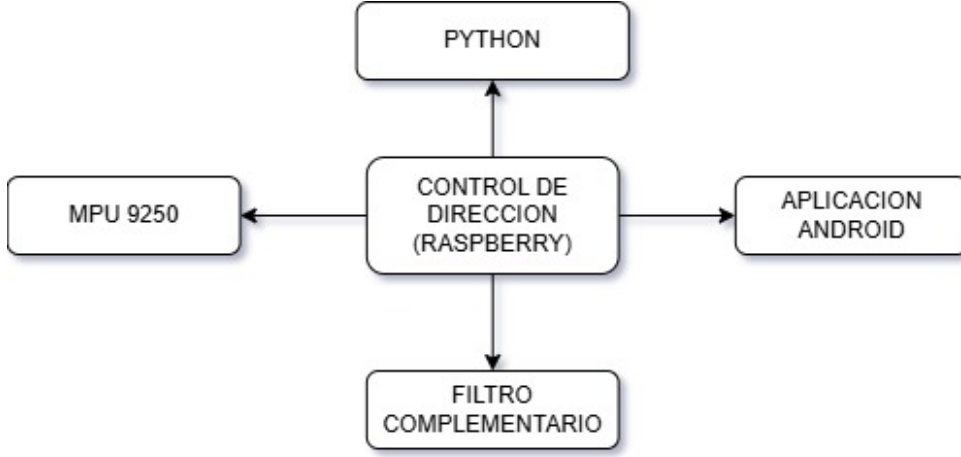


Figura 30: Diagrama del control de dirección

Filtro complementario

Las lecturas de los ejes magnéticos M_x y M_y obtenidas por la comunicación I2C que se conecta con el MPU9250, contienen errores debido a los materiales ferrosos, este efecto es hard-iron y soft-iron [35]. Para corregir estas alteraciones se aplican correcciones de offset y factor de escala para calibrar el sensor, de esta manera se puede disminuir los desplazamientos para campos magnéticos del robot causados por factores ambientales. Las ecuaciones de compensación Hard-Iron para cada eje son:

$$M_{corr_x} = (M_x - offset_x) \cdot scale_x \quad (29)$$

$$M_{corr_y} = (M_y - offset_y) \cdot scale_y \quad (30)$$

donde:

- $M_{x,y}$: son las lecturas naturales.
- $offset_{x,y}$: son desplazamientos que causa el efecto Hard-Iron.

- $M_{corr_{x,y}}$: son las mediciones con correcciones.
- $scale_{x,y}$: son los factores de escala o ganancia para cada eje.

Los factores de escala se aplican para compensar deformaciones elípticas del campo magnético por la calibración, luego se calcula en rango de $0^\circ - 360^\circ$ el ángulo $\text{Yaw}(\theta)$, aplicando tangente inversa a las mediciones corregidas, la ecuación es:

$$\theta = \tan^{-1} \left(\frac{M_y^{corr}}{M_x^{corr}} \right) \quad (31)$$

$$\theta = \begin{cases} \theta + 360^\circ & \text{si } \theta < 0 \\ \theta & \text{si } \theta \geq 0 \end{cases} \quad (32)$$

El filtro complementario circular para el magnetómetro cuando se expone ante un ruido o perturbación debido a las vibraciones mecánicas, se expresa con la siguiente ecuación:

$$\theta_{filtrado} = \alpha \theta_{previo} + (1 - \alpha) \theta_{nuevo} \quad (33)$$

$$0 < \alpha < 1 \quad (34)$$

donde θ_{nuevo} es la medición del magnetómetro y θ_{previo} es la estimación anterior. La contribución relativa de la estimación anterior y la nueva medición es controlada por la ganancia α , entonces los valores altos de α ayudan a reducir el ruido debido al retardo.

Instalación y preparación del sistema para la navegación inercial

Instalación de Paquetes imusensor en Rasbian

Para la configuración del sensor inercial MPU 9250 (acelerómetro, giroscopio y magnetómetro), se instalaron los siguientes paquetes esenciales:

```

1 pip install imusensor
2 sudo apt-get install i2c-tools
3 sudo pip install smbus

```

Configuración del Bus I2C

Se verifica la conexión I2C del sensor MPU 9250, encontrando inicialmente problemas de configuración que fueron resueltos con:

```
1 sudo i2cdetect -y 1
```

Se instalan paquetes para la comunicación serial entre Arduino y Raspberry, para continuar con el proceso de asignar direcciones al robot balancing, con referencia a los polos magnéticos obtenidos por el sensor inercial.

Configuración de Puertos Seriales

Para establecer la comunicación con el Arduino, se configuraron los puertos seriales y se instaló el paquete pyserial:

```
1 dmesg | grep "tty"
2 pip install pyserial
```

Instalación de paquete Arduino IDE

Para permitir la programación y modificación del código del Arduino directamente desde la Raspberry Pi, se instaló el entorno de desarrollo Arduino IDE mediante el siguiente comando:

```
1 sudo apt install arduino arduino-core
```

Configuración de Dispositivos USB

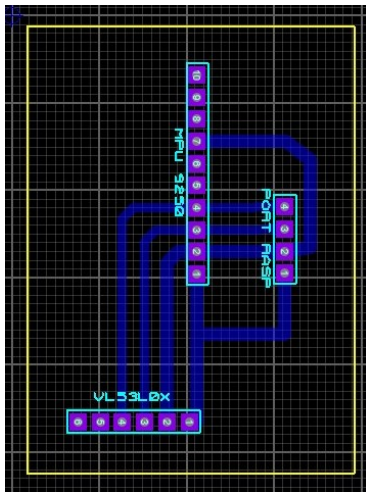
Comando para verificar dispositivos conectados en puerto usb.

```
1 lsusb
```

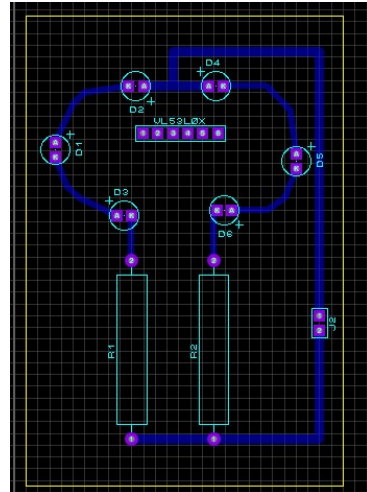
Circuito eléctrico para sensores

Los circuitos eléctricos del proyecto se elaboran para optimizar las líneas de conexiones de los sensores que se utilizan en los distintos procesos, el diseño de las pistas electronicas en una placa PCB, se generan en el software proteus como un archivo Gerber que es una información en formato vectorial ASCII, luego se crea el archivo g-code con el software FlatCAM, el archivo que se obtiene en FlatCAM sirve para el software Mach3 que permite el uso de la maquina CNC. que se encarga de construir las pistas en la baquelita.

En Proteus se define el diseño del circuito de pistas electronica, para ello se selecciona cada dispositivo que debe ir en la placa PCB, en el área de trabajo se ubica cada elemento optimizando espacios. Luego se conecta los elementos electrónicos como se muestra en la figura 31 y se crea el modelo para el PCB Layout para obtener las pistas del circuito.



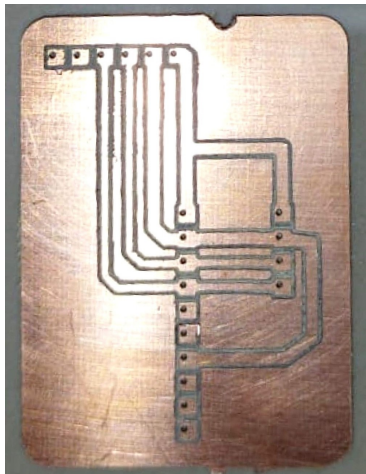
(a) Diseño para mpu9250



(b) Diseño para vl5310x

Figura 31: Imágenes de los circuitos electrónicos diseñados

Los archivos gerber que se generan en Proteus deben tener las características de la pista de todo el circuito, tamaño de perforaciones, dimensión, espacios entre elementos, etc. Este proceso es necesario para crear los archivos G-code en el software FlatCAM, en esta plataforma se parametriza las dimensiones y configuraciones adecuadas para la impresión de la placa, se obtiene 3 archivos G-code: uno con información de la pista, otro con el perforado y otro con el corte de la placa. En el software Mach3Mill se importa cada archivo G-code de la anterior etapa, que permitirá controlar la máquina CNC mediante el lenguaje de programación de cada proceso que son el tallado de la pista del circuito, las perforaciones y el corte de los bordes de la placa, cada etapa del tallado se realiza en tiempos no simultáneos, el diseño se muestra en las fotografías en la figura 32.



(a) Circuito para mpu9250



(b) Circuito para vl53l0x

Figura 32: Fotografías de los circuitos impresos por la cnc

3.1.3 Diseño de detección y localización

El diseño se expresa en el diagrama de la figura 33 mediante elementos físicos, codificación y plataformas que se comunican con la tarjeta procesadora.

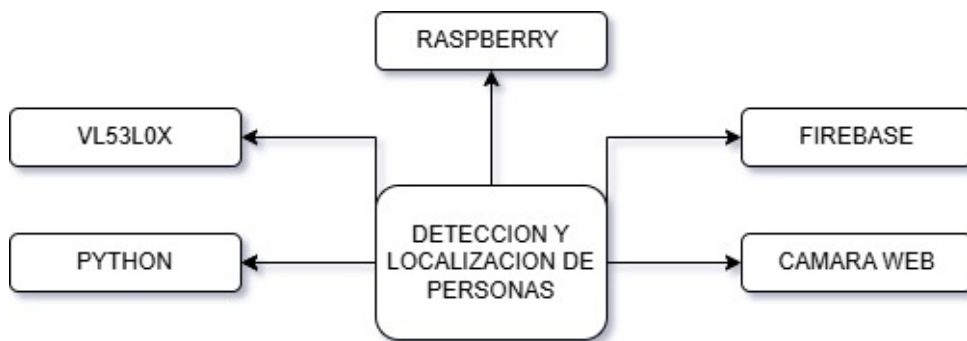


Figura 33: Diagrama de detección y localización

Detección facial

El sistema de detección facial implementado en el robot se basa en el procesamiento de imágenes, mediante una cámara web conectada a la Raspberry Pi 3. Para ello, se emplea el lenguaje de programación Python junto con la

librería OpenCV, la cual permite implementar algoritmos de visión artificial como el detector facial, tales como los clasificadores basados en Haar Cascades o metodos fundamentados en HOG (*Histogram of Oriented Gradients*) y SVM (*Support Vector Machine*) [10, 14].

Instalación y preparación del sistema para la detección facial

Instalación de paquete fswebcam en Rasbian

Para habilitar la herramienta que permite capturar imágenes desde la cámara conectada a la Raspberry Pi mediante terminal, se ejecutan el siguiente comando:

```
1 sudo apt-get install fswebcam
```

Instalación de paquetes OpenCV

El uso de OpenCV permite capturar y procesar imágenes en tiempo real desde una cámara web, siendo una opción ampliamente utilizada por su compatibilidad con diversos modelos de cámaras USB y su integración eficiente con sistemas embebidos como la Raspberry Pi [37].

```
1 sudo apt-get update
2 sudo apt-get install python3-opencv
3 pip3 install opencv-python
```

Instalación de paquete Pillow

Como complemento esencial para la interfaz gráfica, se instaló la biblioteca Pillow para el procesamiento y manejo de imágenes:

```
1 pip install Pillow
```

Instalación de paquete imutils

La librería imutils se instala para simplificar las operaciones de procesamiento de imágenes con OpenCV:

```
1 pip install imutils
```

Instalación de paquete mencoder

Para el procesamiento de video, necesario para la detección facial en tiempo real, se ejecuta la siguiente línea de comandos:

```
1 sudo apt-get install mencoder
```

Localización

El método para ubicar a una persona con referencia al robot y determinar la posición relativa se lo realiza integrando los sensores que ayudan a obtener distancia y ángulo de orientación [14].

El sensor VL53L0X mide con alta precisión la distancia frontal a un objeto, y permite detectar la proximidad de una persona. Cuando se registra la detección, se asocia con la posición actual del robot estimada por el tiempo de navegación. Este enfoque brinda ubicaciones relativas dentro de un entorno cerrado sin necesidad del uso de GPS, ideal para aplicaciones de monitoreo en interiores. [11, 10].

Filtro EMA aplicado a las lecturas de distancia

Aplicación de Filtro EMA (Media Móvil Exponencial) para suavizar la señal expuesta al ruido, a las variaciones por iluminación y reflectividad e incluso por lecturas inestables en tiempo real, el filtro digital se detalla mediante la ecuación:

$$d_f(k) = \alpha d(k) + (1 - \alpha) d_f(k - 1) \quad (35)$$

Donde $d(k)$ es la distancia medida en centímetros en el instante k , $d_f(k)$ es la distancia instantánea filtrada k , $d_f(k - 1)$ es la distancia anterior filtrada, α es el factor de suavizado del filtro, con $0 < \alpha < 1$, el valor de alpha recomendado en un montaje utilizando un robot balancing es de 0.3 para la veracidad de lectura y el evitar los retardos en el cambio de distancias [31, 30].

Instalación y preparación del sistema para la localización

Instalación de paquetes para operaciones numéricas

Para el procesamiento de datos del sensor, se instalan las siguientes bibliotecas en el sistema:

```
1 sudo pip install scipy matplotlib numpy
```

Optimización de Rendimiento Numérico

Se instalaron librerías de optimización para el procesamiento eficiente de los datos del sensor:

```
1 sudo apt-get install libatlas-base-dev
```

Ajuste de Versiones de Numpy

Se realizaron ajustes en la versión de la librería para garantizar compatibilidad:

```
1 sudo pip3 install numpy
2 pip install numpy==1.26.4
```

Instalación de paquete vls31ox

Para implementar el funcionamiento del sensor VLS31OX, se instalaron los siguientes paquetes específicos:

```
1 sudo pip3 install adafruit-circuitpython-vls31ox
```

3.1.4 Diseño de aplicación móvil

Como parte del tema propuesto, está el desarrollo de una aplicación móvil en Android Studio empleando lenguaje Kotlin, este debe actuar como una interfaz para usuario, que permite visualizar datos e manipular manualmente el robot. La aplicación incluye eventos como la detección de personas, presentar la medida de distancia del sensor y la imagen capturada durante el recorrido.[36]

La comunicación entre el robot y la aplicación se realiza a través del módulo Bluetooth HC-06 o por medio de la nube mediante Firebase. La comunicación inalámbrica permite sincronizar los eventos registrados por el microprocesador Raspberry con el dispositivo Android [18, 26].

La interfaz para usuario está diseñada para ser espontanea, visualizando mensajes de estado, los datos de posición y la activación de funciones para que el robot pueda avanzar o girar. Esta integración de hardware y software ofrece una solución para sistemas de asistencia, vigilancia y gestión domótica centrados en la detección y localización de personas.[36]

■ Configuración de Firebase en Android Studio Koala

Para implementar la integración con Firebase en el proyecto Android desarrollado en Android Studio Koala, se realizó un proceso detallado de configuración que incluyó el registro de la aplicación en la consola de Firebase, la descarga del archivo de configuración específico para el proyecto, y la modificación de los archivos de construcción Gradle para incluir las dependencias necesarias. Esta configuración es necesaria para habilitar servicios como Analytics, Authentication, Realtime Database y Storage que serán utilizados en el desarrollo de la aplicación.

■ Registro de la aplicación en Firebase Console

El primer paso consistió en registrar la aplicación en la consola de Firebase, donde se especificó el nombre del paquete Android (com.miamav) y el sobrenombre de la aplicación (MiaMav). Este proceso generó el archivo de configuración google-services.json que contiene todas las credenciales necesarias para la conexión.

■ Configuración del archivo build.gradle.kts de nivel de proyecto

Se agregó el plugin de Google Services al archivo build.gradle.kts de nivel de proyecto, lo que permite el procesamiento del archivo de configuración google-services.json. Este paso establece las bases para todas las integraciones posteriores con los servicios de Firebase:

```
1      plugins
2      {
3          id("com.google.gms.google-services") version
              "4.4.2" apply false
4      }
```

- **Configuración del archivo build.gradle.kts, nivel de aplicación**

Posteriormente, se configuró el archivo build.gradle.kts del módulo de aplicación, donde se agregó el plugin de Google Services y se incluye la Bill of Materials (BoM) de Firebase que simplifica la gestión de versiones de las bibliotecas:

```
1      plugins
2      {
3          id("com.android.application")
4          id("com.google.gms.google-services")
5      }
6
7      dependencies
8      {
9          implementation(platform("com.google.firebase:
10             firebase-bom:33.1.2"))
11             implementation("com.google.firebase:firebase-
12                 analytics")
13     }
```

Se verifica que todos los pasos de configuración fueron registrados correctamente en la consola de Firebase. Para continuar con el siguiente proceso de interconexión entre plataformas se realiza lo siguiente:

- **Configuración para establecer conexión en Android Studio Koala a Firebase**

Para implementar la conexión entre Android Studio Koala y Firebase, se siguieron los pasos de configuración necesarios para integrar los diversos servicios que ofrece Firebase, comenzando por la configuración básica del proyecto hasta la implementación de Analytics y otros servicios complementarios.

- **Selección e Habilitación de Servicios Firebase**

Luego de establecer la conexión con la plataforma, se procede a configurar y habilitar los servicios de Firebase necesarios en el proyecto,

incluyendo Analytics, Storage y Realtime Database como servicio principal y otros servicios complementarios.

■ **Configuración Detallada de Firebase Analytics**

Con los servicios seleccionados, se realizó la configuración específica de Firebase Analytics, habilitando todas las funcionalidades de seguimiento y monitoreo de la aplicación.

Instalación y preparación para integrar funciones de Firebase en Rasbian

■ **Instalación de dependencias para Python**

Para garantizar el desarrollo de las dependencias del sistema, para la compilación y correcta ejecución de bibliotecas que requieren extensiones nativas, se instalan los paquetes de desarrollo de Python en la Raspberry

```
1      sudo apt-get install python-dev
```

- **Instalación de biblioteca Pyrebase** Se instala para la configuración del entorno, la biblioteca Pyrebase que permite integrar los servicios de Firebase en Python. Esta herramienta gestiona operaciones para el envío y recepción de datos en la nube.

```
1      sudo pip install pyrebase
```

■ **Instalación de Google Cloud Firestore**

Como parte de la configuración del backend, se instaló el cliente de Firestore para Python, permitiendo la integración con la base de datos en la nube de Firebase.

```
1      pip install google-cloud-firestore
```

Entonces con estos paquetes instalados en las diferentes sistemas se logra interactuar de forma inalámbrica entre el teléfono celular , la tarjeta Raspberry utilizada en el robot y la nube.

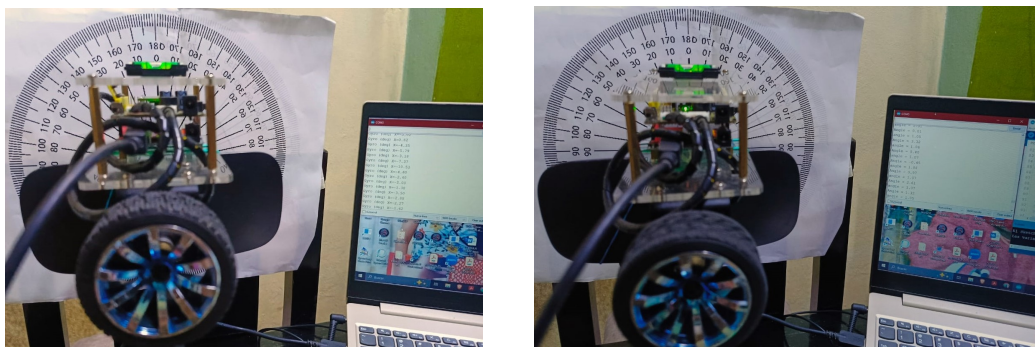
CAPÍTULO IV

4 . Pruebas y Resultados

4.1 Pruebas y validación experimental

Experimentos control de equilibrio

El experimento consiste en comparar tres medidas provenientes del sensor MPU 6050 estas son la lectura del angulo de inclinación, la lectura del angulo aplicando Filtro de Kalman y la lectura del angulo filtrado con Kalman aplicado al control PID. Estas pruebas se realizan para verificar que la lecturas angulares obtenidas son lo mas cercano al angulo real. Este angulo es esencial para establecer el equilibrio del robot balancing.



(a) Lectura natural

(b) Lectura con filtrado y control PID

Figura 34: Fotografía del proceso de obtención de datos del sensor mpu6050

En el proceso para la extraer los datos se utilizan instrumentos mecánicos como graduador angular fijado a una pared y tambien se coloca un nivel de burbuja sobre la estructura del robot en la parte superior del robot para garantizar una posición física estable, como se muestra en la figura 34. Los datos de muestreo registran la inclinación del robot hacia delante como positivo y al contrario como negativo.

Cuadro 1: Datos de ángulo de inclinación para el control de equilibrio

Nº	Ángulo	Ángulo/inclinación(θ)	θ con Kalman	θ con Control PID
1	25	25.51	25.27	25.14
2	20	20.11	20.08	20.03
3	15	15.15	15.08	15.06
4	10	10.51	10.16	10.08
5	5	5.10	5.08	5.05
6	0	0.82	0.08	0.01
7	-15	-5.47	-5.22	-5.04
8	-10	-10.16	-10.09	-10.03
9	-15	-15.61	-15.45	-15.10
10	-20	-20.66	-20.30	-20.12
11	-25	-25.98	-25.20	-25.10

En la Tabla 1 se presenta ángulos de valores referenciales desde $+25^\circ$ hasta -25° , correspondientes a las inclinaciones del robot. Para obtener los datos del MPU 6050 se compila el algoritmo para calcular el angulo del sensor, los valores obtenidos contienen ruido debido a las vibraciones del sistema. mostrando para valores entre $0^\circ \leq \theta \leq 25^\circ$, un intervalo de error de $0,10^\circ \leq e \leq 0,68^\circ$ y para ángulos $-25^\circ \leq \theta \leq 0^\circ$ el intervalo de error de $-0,98^\circ \leq e \leq -0,16^\circ$.

Al introducir el filtro de Kalman en el algoritmo anterior fusiona las características del acelerómetro y giroscopio, con este metodo los valores del intervalo de error se reducen y la señal del sensor se suaviza obteniendo datos cercanos a la realidad. La muestra de valores son para angulos de $0^\circ \leq \theta \leq 25^\circ$, con un intervalo de error de $0,08^\circ \leq e \leq 0,27^\circ$ y para ángulos $-25^\circ \leq \theta \leq 0^\circ$ el intervalo de error de $-0,45^\circ \leq e \leq -0,09^\circ$.

El valor filtrado ingresa a un lazo de control PID, de esta manera corrige y estabiliza con mayor precisión al sistema de equilibrio. los valores para los ángulos positivos son de $0^\circ \leq \theta \leq 25^\circ$, con intervalo de error de $0,01^\circ \leq e \leq 0,14^\circ$ y para ángulos negativos de $-25^\circ \leq \theta \leq 0^\circ$ el intervalo de error es $-0,12^\circ \leq e \leq -0,03^\circ$.

En la Figura 35 se tiene la gráfica de las señales de datos de inclinación, datos con filtro de Kalman y datos con PID mostrando en el eje horizontal los valores reales y en el eje vertical ángulos obtenidos. se muestra la comparativa

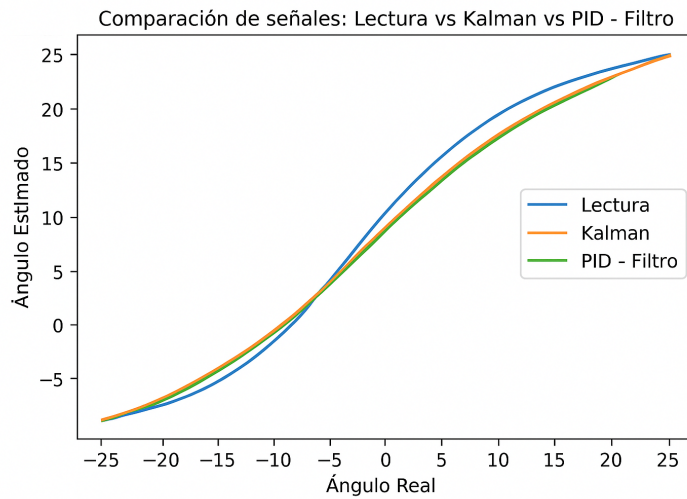


Figura 35: Gráfica comparativa mpu6050

demostrando que el sistema mejora significativamente porque las lecturas se acercan a los ángulos reales y esto sirve para la estabilidad del sistema de robot balancing.

Experimentos control de dirección

En el experimento se pretende comparar las lecturas obtenidas del sensor MPU 9250, estas son el ángulo yaw y la lectura del ángulo con el filtro complementario, se realiza la recolección de datos colocando el instrumento mecánico para este caso; una brújula sobre la estructura del robot en la parte superior central, el nivel de burbuja se mantiene para verificar la dirección estable del mpu 9250 ubicado a un lado, el sensor inercial está conectado a la tarjeta procesadora Raspberry Pi3 que está alimentada con un power bank, la placa que contiene el sensor magnetómetro ha sido montado al robot balancing tal como se visualiza en la figura 36.

El propósito de este método es observar que el algoritmo es el adecuado para el direccionamiento autónomo del robot, teniendo en cuenta aspectos físicos del balancing y a las vibraciones a las que se exponen los sensores es necesario introducir un filtrado. En la tabla 2 se registran los datos de las medidas de los movimientos rotatorios esto es el ángulo yaw y los valores del



(a) Lectura natural



(b) Lectura con filtro complementario

Figura 36: Fotografía del proceso de obtención de datos del sensor mpu9250

mismo aplicando el filtro complementario .

Cuadro 2: Datos experimentales del sensor mpu9250

Nº	Grados	Ángulo Yaw	Filtro Complementario
1	360	360,18	360,02
2	340	341,10	340,30
3	320	319,96	320,02
4	300	300,91	300,35
5	280	280,04	280,01
6	260	260,59	260,17
7	240	240,28	240,05
8	220	220,98	220,10
9	200	200,58	200,08
10	180	180,87	180,34
11	160	160,27	160,15
12	140	140,42	140,23
13	120	120,92	120,50
14	100	100,97	100,34
15	80	80,83	80,68
16	60	60,90	60,76
17	40	40,95	40,28
18	20	20,81	20,32
19	0	0,18	0,02

Los parámetros registrados del angulo yaw para medidas de $0^\circ \leq \theta \leq 360^\circ$ tienen intervalos de error entre $0,10^\circ \leq e \leq 0,98^\circ$ en comparación a las medidas de la brújula. Los datos que se obtienen aplicando el filtro

complementarios para sistemas angulares de 360° , se obtiene un intervalo de error de $0,01^\circ \leq e \leq 0,76^\circ$.

Comparación de señal: Real vs Lectura vs Filtro complementario

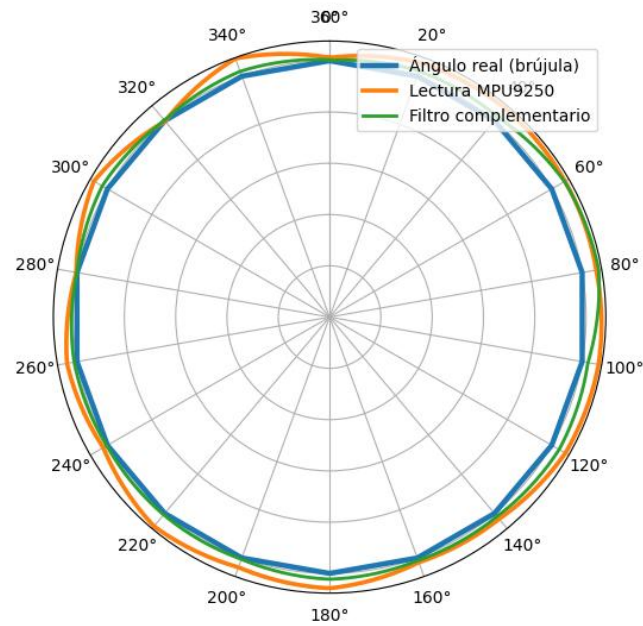


Figura 37: Gráfica comparativa sensor mpu9250

En la figura 37 se tiene la gráfica de los parámetros analizados en la tabla anterior, donde se visualiza cada señal comparadas ante los ángulos de una brújula física, se observa que la señal que se acerca a estos valores es la señal con filtrado representada con color verde.

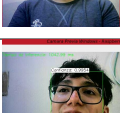
Prueba de detección facial y localización de personas

Experimentos de detección

Se realizan los experimentos del algoritmo de detección facial para analizar el nivel de confianza al detectar rostros expuestos a diferentes distancias. Se tiene la siguiente tabla 3 con una captura de una persona a diferentes distancias hasta completar de cuatro rostros distintos. Los resultados obtenidos del valor de confianza proporcionada muestran la factibilidad y confiabilidad

del algoritmo de detección de rostros.

Cuadro 3: Datos de detección facial

Nº	Persona	Distancia (cm)	Confianza
1		20	1.0000
2		40	0.9881
3		60	0.9482
4		20	0.9999
5		40	1.0000
6		60	0.9731
7		20	0.9998
8		40	1.0000
9		60	0.9678
10		20	0.9954
11		40	1.0000
12		60	0.7822

En los resultados mostrados, se observa un alto nivel de precisión a cortas distancias, los valores de confianza se acercan a 1.0000 cuando la persona se encuentra a veinte centímetros, esto indica la fiabilidad del algoritmo. Cuando la distancia aumenta a cuarenta centímetros el rendimiento del sistema es estable, se obtiene ligeras variaciones debido a las condiciones de iluminación y ángulo de rostro e incluso depende de los accesorios que tenga la persona en el rostro. Al analizar los resultados de exposición a sesenta centímetros, su nivel disminuye en ciertos casos, la causa podría ser por reducción de la resolución en las características faciales.

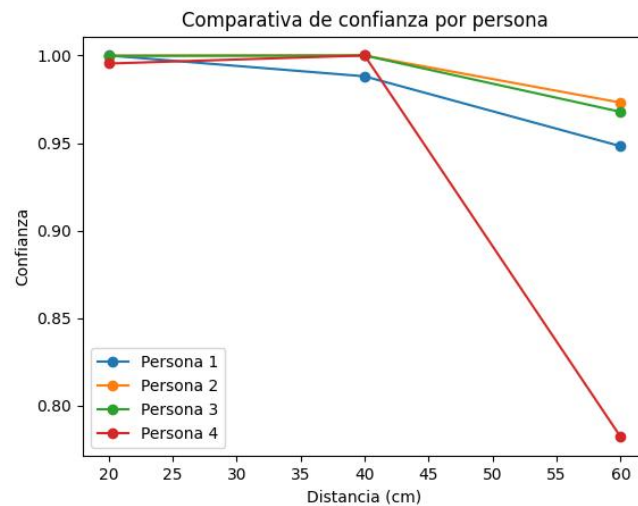


Figura 38: Gráfica comparativa de detección

En la figura 38 se presenta la comparativa de confianza vs distancia de cada persona es la relación que se muestra con valores en centímetros entre una persona y la cámara, el nivel de rendimiento del sistema opera establemente para distancias menores a cuarenta centímetros, Este proceso permite obtener valores de confianza para condicionar el sistema a un rango mayor a 0.9500 para validar un rostros detectados.

Experimentos de localización

Los experimentos para determinar la ubicación referencial entre robot y una persona, se realiza con un sensor VL53L0X que mide la distancia cuando detecta obstrucción, con este método se pretende evaluar la precisión en las lecturas. Se procede a colocar una cinta métrica como se muestra en la figura 39, para establecer valores de distancia física como referencia para comparar las lecturas de distancia del sensor - objeto y la lectura del mismo aplicando el filtro EMA.



Figura 39: Obtención de datos sensor vl53l0x (21-40 cm)

Los datos de distancia se registran en la tabla 4, el procedimiento consiste en mover el obstáculo cada un centímetro y luego se recolecta las lecturas de distancia en un rango de $1 \leq dist \leq 20$ donde se tienen valores de error de $0,06 \leq e \leq 0,93$, por condiciones de estructura y propiedades del sensor, la señal de distancia medida contiene ruido y cuando se aplica el filtro EMA el intervalo de error disminuye a $0,01 \leq e \leq 0,18$.

Cuadro 4: Datos sensor vl53l0x (1-20cm)

DISTANCIA	Lectura de Distancia	FILTRO EMA
1	1,29	1,09
2	2,74	2,05
3	3,11	3,04
4	4,20	4,02
5	5,42	5,09
6	6,64	6,04
7	7.06	7.02
8	8.85	8.14
9	9.32	9.01
10	10.54	10.16
11	11.63	11.08
12	12.26	12.06
13	13.58	13.03
14	14.76	14.18
15	15.93	15.02
16	16.69	16.13
17	17.79	17.12
18	18.65	18.09
19	19.90	19.14
20	20.50	20.15

A continuación se muestra la figura 40 con las señales de la atenuación en las rectas en el rango de 1 a 20 centímetros, la señal con filtro EMA se aproxima a la natural con menores ruidos.

Como segunda etapa se utiliza led infrarrojos adicionales colocados alrededor del sensor, se pretende aumentar el alcance de la señal de distancia para detectar obstrucciones. Los datos recolectados se ven aumentados hasta en cuarenta centímetros, mostrando el mismo comportamiento pero con mayor rango de distancia como se muestra en la figura 41.

De la misma manera se tiene la tabla 5 con los datos del sensor, con distancias físicas entre $21 \leq dist \leq 40$ en centímetros, para estas señales el intervalo de error en las lecturas del sensor son de $0,08 \leq e \leq 0,96$ y los resultados con el filtro tiene un rango de $0 \leq e \leq 0,19$.

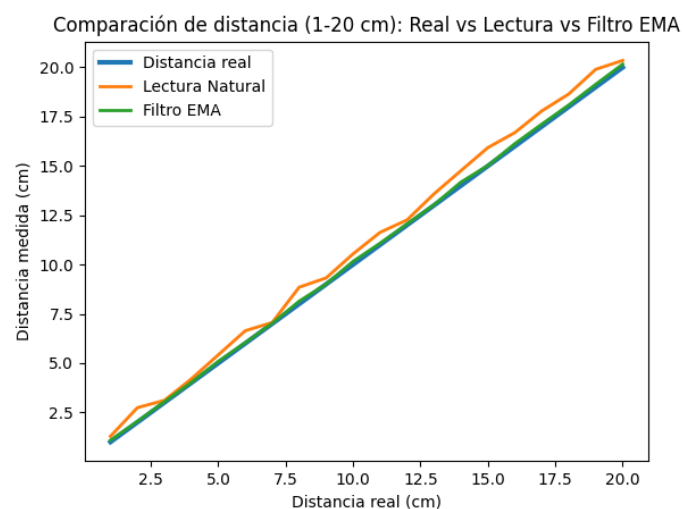


Figura 40: Gráfica comparativa (1-20cm) sensor vl53l0x

Cuadro 5: Datos sensor vl53l0x (21-40cm)

DISTANCIA	LECTURA NATURAL	FILTRO EMA
21	21,23	21,07
22	22,73	22,18
23	23,96	23,12
24	24,38	24,18
25	25,83	25,12
26	26,30	26,03
27	27,72	27,16
28	28,08	28,01
29	29,54	29,08
30	30,83	30,13
31	31,90	31,19
32	32,65	32,03
33	33,73	33,18
34	34,46	34,07
35	35,56	35,10
36	36,87	36,17
37	37,83	37,05
38	38,54	38,19
39	39,35	39,00
40	40,34	40,07

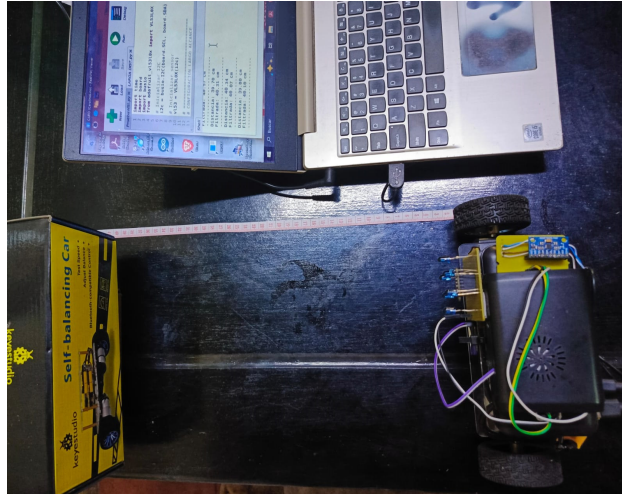


Figura 41: Obtención de datos sensor vl53l0x (21-40 cm)

La gráfica 42 muestra las señales de lectura natural y con filtro EMA en comparación a una distancia real, se requiere encontrar la precisión del sensor y se observa la señal filtrada utilizando adicionalmente una placa con leds infrarojo.

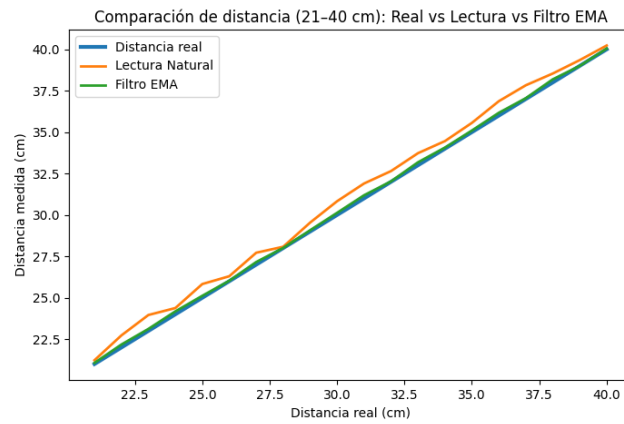


Figura 42: Gráfica comparativa (21-40cm) sensor vl53l0x

4.2 Resultados

Como resultado se tiene un sistema de navegación inercial aplicado a un robot balancing, permitiendo integrar placas de control y de procesamiento. A partir de los experimentos realizados se logran obtener respuestas satisfactorias en cuanto a la estabilidad del robot, ante los desplazamientos, los giros del robot para su orientación. El algoritmo aplicado al robot cumple con detenerse al detectar obstáculo y detectar rostros, además brindar información del lugar donde se encuentra la persona. En la Figura 43 se muestra el prototipo desarrollado que contiene los componentes mencionados anteriormente, esta arquitectura permite distribuir cada tarea para los actuadores del robot y de procesamiento de datos de los sensores, cada detalle influye al desempeño general del prototipo.



Figura 43: Propuesta final

Diagrama de flujo del sistema

La secuencia logica se presenta mediante diagrama de flujo en la Figura 44, el proceso inicia con la ejecución del sistema de equilibrio del robot balancing, codificado en un microcontrolador Arduino que contiene algoritmos de control PID y el Filtro de Kalman. Adicionalmente se ejecuta el control de dirección, de este proceso se encarga el micropocesador Raspberry Pi, que gestiona las decisiones de movimiento para que el robot actúe de acuerdo a su entorno, para la lectura del sensor inercial y del sensor de distancia, se emplean Filtro Complementario y Filtro EMA para suavizar las señales.

A continuación el siguiente bloque correspondiente a la detección facial utiliza de los datos del sensor de distancia para detectar cualquier presencia frente al robot, dando paso a la activación de la cámara para poder analizar si se trata de un rostro y en caso de serlo guardar la captura de la detección y datos de distancia entre robot y la persona.

La condición de persona detectada haria que el robot se encuentre determinado por una detección facial, presntandouna interfaz gráfica para el usuario (GUI), en la cual se detalla la orientación del robot en grados y la distancia entre el robot y la persona, con la imagen capturada de la detección, ademas del mensaje de estado. Caso contrario retorna a su actividad inicial.

La condición de aplicación android corresponde a las opciones de botón para seleccionar monitoreo y el control manual del robot. Si se ha seleccionado el modo monitoreo se visualiza los resultados, la información envianda a la plataforma Firebase permite visualizar en el móvil de manera remota la imagen la distancia y el estado. Si se ha seleccionado el modo control manual se activa el bluetooth para conectar el dispositivo android con el robot balancing, con el fin de permitir que el usuario mediante botones o con comandos de voz conduzca los movimientos del prototipo.

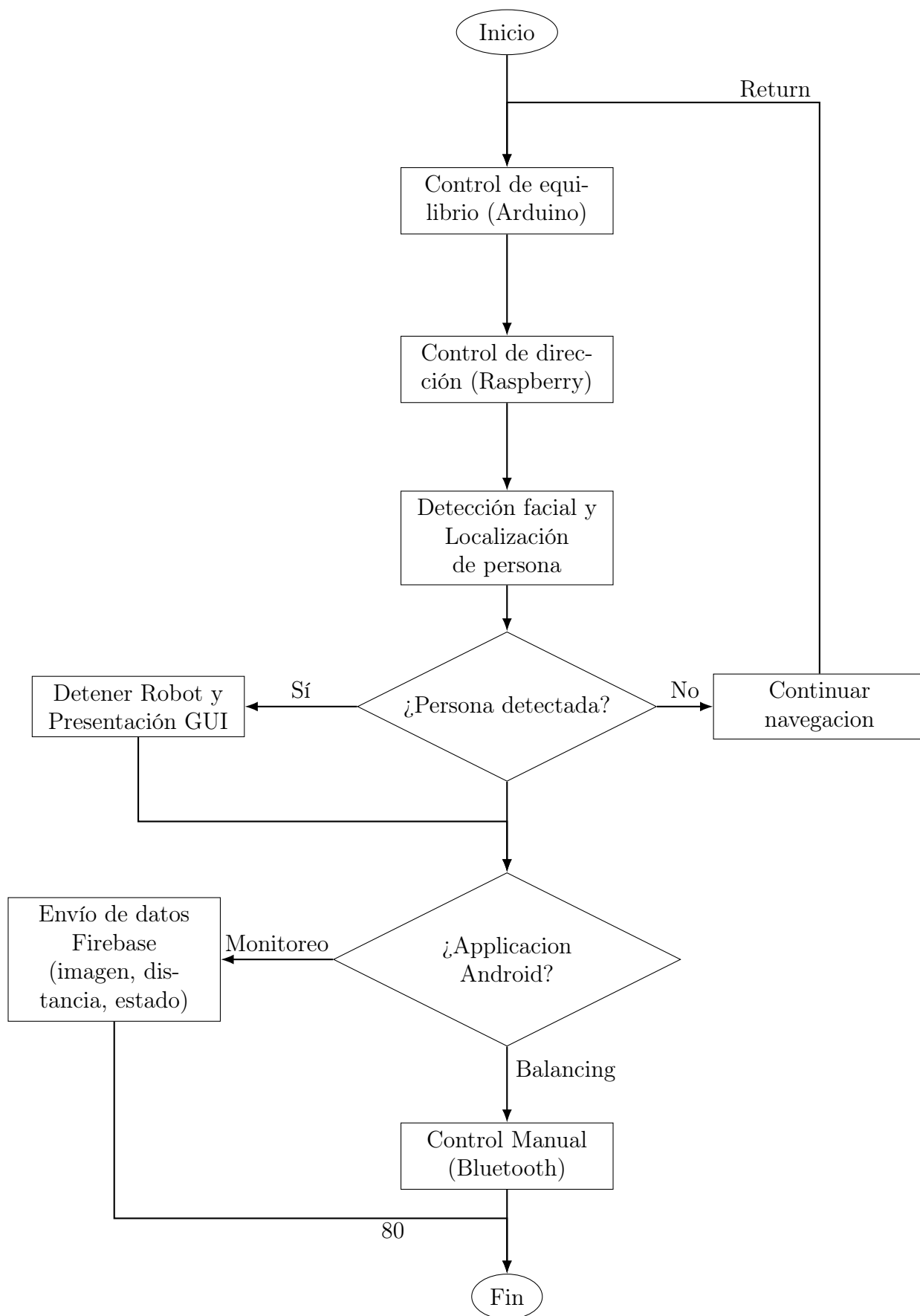


Figura 44: Diagrama de flujo del sistema

Funcionamiento

En este apartado se pretende confirmar el comportamiento del robot Balancing con el sistema de navegación inercial en presencia de un objeto, en la Figura 45, se muestra un escenario creado para el recorrido, al encender el robot se desplaza hacia delante, se detiene a 28.40 centímetros de acuerdo a la medida de la cinta métrica y el algoritmo activa la cámara sin detectar rostro y continua con la rutina de navegación.



Figura 45: Navegación sin detección

Los resultados se observan en la interfaz de python , en la figura 46 se muestra los datos de ángulo yaw igual a 89.55° , valor que hace referencia al polo magnético Este. La distancia que se presenta es de 34.85 cm, muy cercano a los 35 centímetros que se aproxima a la medida física entre robot - obstáculo, la diferencia de longitud mínimas que se tiene es debido al balanceo. Además presenta el mensaje de "Persona no detectada. Continuar navegación".

De acuerdo a la figura 47 se observa el escenario de la detección de rostro, al continuar la navegación, el prototipo procede a girar hacia la izquierda, buscando el polo magnético codificado en este caso hacia el norte, avanza hacia delante y luego se detiene a una distancia de 61.6 centímetros medida con la cinta métrica, para analizar la presencia detectada. En este caso se detecta un rostro y el robot se mantiene estacionado procesando la información para guardarse respectivamente.

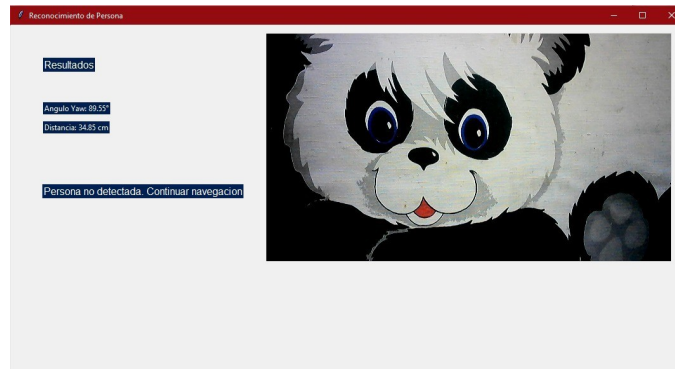


Figura 46: Interfaz sin detección



Figura 47: Navegación con detección

Los resultados que se muestran en la figura 48 indican que se detecta el rostro de una persona con un valor de confianza de 0.99, con la distancia entre robot - persona de 39.08 centímetros, la lectura de angulo yaw es de 1.08° cercano al 0° en referencia hacia el norte y la presentación del respectivo mensaje "Persona detectada. Deteniendo robot".

En la figura 49 se tiene el escenario completo que inicia desde el Este, con el primer obstáculo que se encuentra a una distancia de 65 centímetros, la segunda obstrucción esta a 102 centímetros en dirección al norte. El robot

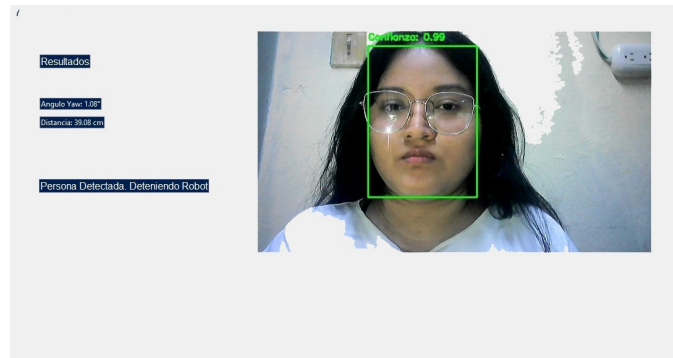


Figura 48: Interfaz con deteccion

se encuentra detenido, a causa del rostro detectado luego de retroceder su posición por condiciones del algoritmo, esperando que el usuario supervisor revise la información guardada y tome acciones correspondientes.

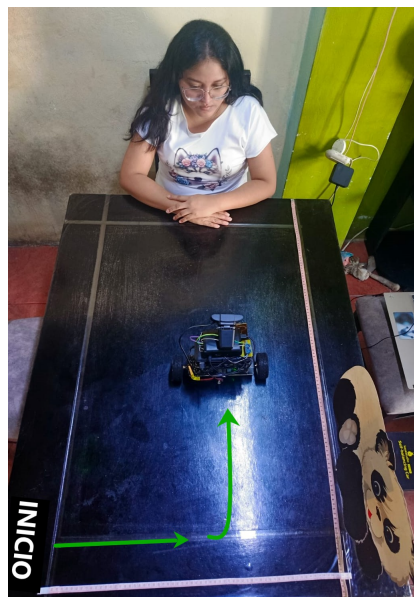


Figura 49: Recorrido en el escenario

El usuario supervisor del sistema cuenta con el APK para instalar en su dispositivo android, luego de conceder todos los permisos para el uso del Bluetooth y acceso de ubicación que es un requisito de acuerdo a las políticas de google. En la sección de menú del aplicativo se dispone de botones para interactuar, en la interfaz que se muestra en la figura 50 se tiene el diseño de

la aplicación MiaMav.

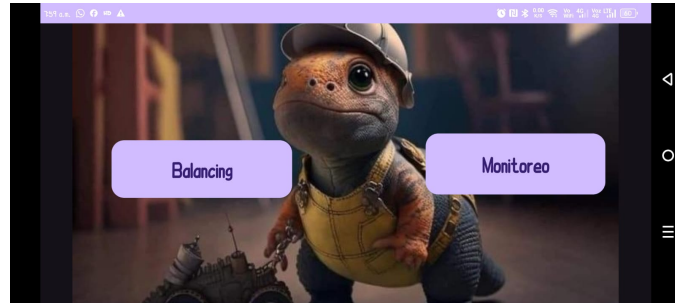


Figura 50: Menú de aplicación android

Al seleccionar la opción de monitoreo se activa un evento para visualizar la información registrada en la nube. Los datos que se generan son la distancia entre robot y la persona detectada, con un mensaje de estado y la respectiva captura de detección tal como se observa en la figura 51, esta información es enviada y se almacenada en Firebase, demostrando que al integrar domotica al prototipo permite que el usuario este pendiente de manera local o remota al entorno de supervisión.

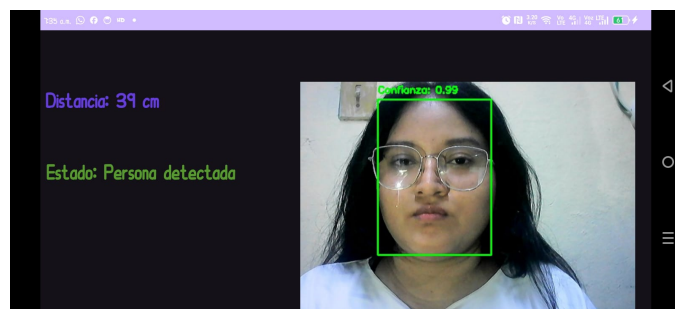


Figura 51: Opción de monitoreo

La opción de Balancing contiene varios botoneras para conducir de manera manual el robot, esta herramienta mejora la interacción y complementa el funcionamiento del sistema. En la figura 52 se tiene la interfaz completa, al pulsar el botón de "Buscar Bluetooth" se despliega una lista de los dispositivos, para seleccionar se tiene el botón de conectar que empareja con el Bluetooth del robot, los botones con flechas son para asignar movimientos

y el botón stop para detener, el icono de micrófono permite que el usuario emita comandos de voz para mover el prototipo y el botón desconectar para finalizar el emparejamiento.

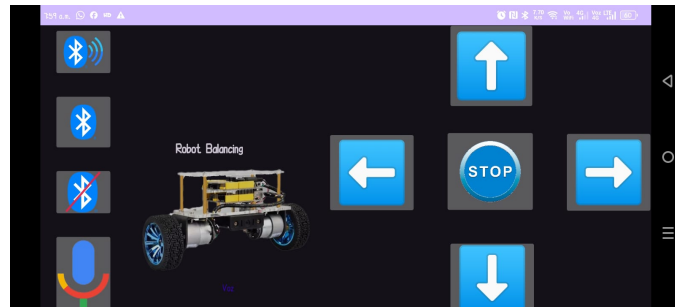


Figura 52: Opción balancing manual

CONCLUSIONES

En el trabajo de titulación se diseñó e implementó un sistema de navegación inercial aplicado a un robot Balancing, las etapas de pruebas de cada plataforma de control, procesamiento y sensores integrados, a demás del uso de tecnología de visión artificial y domotica que ayudaron a cumplir los objetivos planteados.

Los análisis experimentales mostraron como resultados datos con un mínimo de errores, en cada sensor para cada etapa definida se involucró el respectivo componente para su completo funcionamiento. el control de equilibrio tiene como principal componente el sensor MPU 6050, ya integrado en la placa del fabricante, cabe recalcar que se utilizó un robot balancing diseñado por KeyStudio, de igual manera se realizaron pruebas al prototipo para mejorar sus condiciones de funcionamiento.

Se comprobó que el sistema de equilibrio basado en algoritmos de control PID y filtro de Kalman para la placa Arduino, mantiene el equilibrio garantizando la estabilidad del robot, las pruebas del angulo de inclinación con el que trabaja registraba errores entre $-0,98^\circ$ y $0,82^\circ$. al filtrar la señal el error disminuye a valores entre $-0,45^\circ$ y $0,27^\circ$ y cuando se aplica el control PID los errores se minimizan hasta $-0,12^\circ$ y $0,14^\circ$ evidenciando alta precision para la estabilidad. Este proceso aseguró que sea posible la incorporación de elementos adicionales como la batería, la Raspberry, la cámara e incluso las placas impresas para los sensores, que incrementan el peso y la meta era mantener el equilibrio.

Se verificó con el mpu 9250 las condiciones del control de direcciones, el objetivo fue alinear al robot con respecto a los polos magnéticos, para analizar los cambios de orientación y aplicar correcciones con filtro complementario especialmente para sistemas que trabajan a 360° , se confirmó la efectividad en reducción de ruido y estabilidad hasta un rango de $0,01^\circ$ y $0,76^\circ$ en las lecturas, pese a las condiciones físicas y ambientales de la estructura y movimientos que en un principio registraban errores entre $0,10^\circ$ y $0,98^\circ$.

Se continuó con las pruebas de detección facial para validar que el robot

se encuentra frente a un obstáculo y que se trata de una persona, se realizó captura del comportamiento del sistema que mostró respuestas hasta 1.0000 como mayor valor significativo, demostrando el nivel de confianza al detectar un rostro humano, además se establecieron distancias de exposición de la cámara hasta el humano teniendo en cuenta el alcance del sensor de distancia. Se determinó una efectividad del sistema de detección con una distancia adecuada de enfoque de 40 centímetros y una confianza mínima hasta 0.9500 para validar el rostro. Se observaron las variantes que influyen en el rostro de una persona estos pueden ser lentes, cabello, incluso cicatrices que podría disminuir el nivel de confianza de manera mínima.

En los experimentos que se realizó con el sensor vl53l0x al medir las distancias entre robot y obstáculo, se obtuvieron valores de error entre 0,06 y 0,93 y para suavizar la señal se implementó el filtro EMA a las lecturas, registrando errores menores entre 0,01 y 0,18. para distancias de 1 hasta 20 centímetros. Además se agregó al rededor del sensor varios leds infrarrojos para incrementar el alcance de distancia en la que trabaja el sensor, se tuvo mediciones de la señal hasta 40 centímetros estables con valor de error entre 0,08 y 0,96, al aplicar el filtro el error se encuentra entre 0 y 0,19, se demostró mejoras en la señal afectada por el balanceo y condiciones de iluminación en el ambiente.

Con el desarrollo de la aplicación móvil, el usuario pudo supervisar, monitorear e incluso controlar el prototipo, el app es complementaria para visualizar datos resumidos de manera remota o local. Los datos se mostraron también en la interfaz gráfica para usuarios de la Raspberry, se visualizó la posición del robot con respecto a la trayectoria recorrida mediante navegación inercial, lectura de ángulo yaw que hace referencia al polo magnético, la distancia entre el robot-humano, mensaje de estado y la visualización captada por la cámara en tiempo real.

La integración de las tecnologías demuestran la posibilidad de implementación con un funcionamiento eficiente el desarrollo del sistema, utilizando controles para la navegación, detección y monitoreo, ideales para soluciones que favorezcan al humano en cualquier tipo de entornos con domótica.

RECOMENDACIONES

En futuros trabajos, se recomienda mejorar la precisión del sistema de navegación agregando sensores o cambiando a componentes mas avanzados e incluso agregar técnicas de fusión sensorial que reduzcan los errores acumulativos de estimación. guiar el proyecto a una aplicación específica y adecuar el prototipo de manera correspondiente.

Se sugiere implementar de forma personalizada al robot balancing para trabajar con el mismo sensor mpu, de esta manera reducir componentes y espacios en la estructura, esto también disminuye trabajo en los procesos de la Raspberry, así mismo el consumo de energía. En este trabajo se utilizó un prototipo fabricado para tener un sistema de planta validado y optimo, centrando el trabajo al diseño y la implementación de la navegación inercial, proceso de datos, detección, localización y domotica.

La aplicación de visión artificial en este proyecto es detección de rostro, que resulta satisfactoria para distancias cortas, esto podría mejorar con reconocimiento facial o para distancias largas con detección de silueta de humano, e incluso agregar sensor de temperatura corporal para validar que es una persona si el objetivo es la video vigilancia a humanos.

Con respecto a la distancia, el sensor capta señales cortas menores a un metro; por lo tanto, se sugiere utilizar sensores mas sofisticados que puedan dar un alcance mayor y permitir que el trabajo se realice en entornos amplios y con fines diferentes. Además, obtener distancias de posición aceptables por ende varia el algoritmo de filtrado de señal con técnicas de calibración robusta.

Como posible mejora en la aplicación Android se propone incorporar función de notificaciones, registro historial de eventos y la presentación visual de la trayectoria del robot. Estas características contribuyen a la optimización en la experiencia para el usuario y fortalece la aplicación de domotica destacando su importancia. Esto implica complejidad pero genera un gran impacto en la sociedad moderna al integrar nuevas tecnologías.

BIBLIOGRAFIA

Referencias

- [1] Rong, Y., Ko, Y., Wang, X., & Shi, G. (2019). Algorithm research of magnetometer assisted inertial navigation system for mobile robots. En Conference on Micro/Nano Sensors for AI, Healthcare and Robotics (pp. 92–96).
- [2] Álvarez Tobar, S. J. (2019). Navegación autónoma de un robot móvil omnidireccional en trayectorias predeterminadas con evitación de obstáculos [Tesis de maestría, Universidad Técnica de Ambato]. <http://repositorio.uta.edu.ec/handle/123456789/29173>
- [3] Cabrera, F. R., & Armijo, D. G. (2018). Robot equilibrista sobre dos ruedas en configuración diferencial para concurso de robótica RioTronic 2018. Revista Caribeña de Ciencias Sociales, (11), 23..
- [4] Sánchez, A., & Casanova, V. (2024). Diseño, implementación y control de un vehículo [Tesis de grado, Universidad Politécnica de Valencia].
- [5] Sendey, G., Chuquimarca Jiménez, L., y Plaza, D. (2024). *Kalman Filter Applied To A Differential Robot*. <https://doi.org/10.48550/arXiv.2410.04011>
- [6] Prasad, V., y Ajay, C. (2026). Self-Balancing Robot. *International Journal of Advanced Research in Science Communication and Technology*, 274. <https://doi.org/10.48175/IJARSCT-32741>
- [7] Rosales Tomalá, H. E. (2020). Prácticas para la fabricación de placas de circuitos impresos (PCB) utilizando máquina fresadora de control numérico por computadora (CNC) [Tesis de grado, Universidad Estatal Península de Santa Elena]..

- [8] Mecánica Curiel S. L. (2018). Origen del mecanizado CNC. Onlinevalles.com. <https://mecanicacuriel.com/2018/06/05/origen-del-mecanizado-cnc>
- [9] Túquerres, O. A. (2022). Aportes de la seguridad domótica a la vigilancia de las viviendas. Recovered from: <https://hdl.handle.net/10654/44665>.
- [10] Borrero, L., & Gómez B., (2021). Implementación de un sistema de seguridad con detección facial para la empresa Publimoda del Caribe [Tesis de grado, Universidad Autónoma del Caribe].
- [11] Liao, W.-H., Wu, C.-L., & Fu, L. C. (2008). Inhabitants tracking system in a cluttered home environment via floor load sensors. *IEEE Transactions on Automation Science and Engineering*.
- [12] Cámara Rodríguez, J. M. (2018). Ciberseguridad, calidad de servicio y localización. *Revista Bit Steam*..
- [13] Pekyürek, M., Sağlam, Z., & Ustun, A. (2020). MIT App Inventor ve Android Studio kullanılarak tasarlanmış mobil uygulamanın performans karşılaştırması. *Bilgi ve İletişim Teknolojileri Dergisi*, 2(2), 161–181..
- [14] Couoh Ku, R. I. (2020). Estimación de pose de robots móviles mediante mediciones de sensores inerciales y una cámara externa [Tesis de grado, Universidad Autónoma de Yucatán]..
- [15] Keyestudio. (2024). KS0193 self-balancing car kit documentation. <https://docs.keyestudio.com/projects/KS0193/en/latest/KS0193.html>
- [16] Mollocuaquira Puña, V. (2020). Desarrollo de un sistema domótico con reconocimiento facial [Tesis de grado, Universidad Pública de El Alto].
- [17] Arora, S., Kumar, S., & Sahin, F. (2019). Human position detection and tracking with on-robot time-of-flight laser ranging sensors. *arXiv*. <https://arxiv.org/abs/1909.09750>
- [18] Chimarro Amaguaña, J. D. (2020). Sistema integrado para la operación de un brazo robótico teleoperado en tiempo real mediante la plataforma

- Firestore con el uso de dispositivos móviles [Tesis de grado, Universidad Tecnológica Israel].
- [19] Postigo Gómez, V. M. (2018). Desarrollo de aplicación Android para localización de dispositivos Bluetooth en interiores [Tesis de grado, Universidad de Valladolid].
 - [20] Guevara, G. P., Verdesoto, A. E., & Castro, N. E. (2020). Metodologías de investigación educativa (descriptivas, experimentales, participativas y de investigación-acción). *RECIMUNDO*, 4(3), 163–173.
 - [21] Ramos Galarza, C. A. (2020). Los alcances de una investigación. *Revista CienciAmérica*, 9(3).
 - [22] Mohamed, M. M. H., Martel Carranza, C. P., Huayta Meza, F. T., Rojas León, C. R., & Arias Gonzáles, J. (2023). Metodología de la investigación: Guía para el proyecto de tesis. Instituto Universitario de Innovación Ciencia y Tecnología Inudi Perú. <https://doi.org/10.35622/inudi.b.073>
 - [23] Raspberry Pi Foundation. (2023). Raspberry Pi Imager documentation. <https://www.raspberrypi.com/software/>
 - [24] Famatech. (2023). Advanced IP Scanner. <https://www.advanced-ip-scanner.com/>
 - [25] RealVNC. (2023). VNC Connect documentation. <https://www.realvnc.com/en/connect/docs/>
 - [26] Guimerá Orozco, A. (2021, abril). ¿Qué es Kotlin? OpenWebinars. <https://openwebinars.net/blog/que-es-kotlin/>
 - [27] Gracia Moisés, A., & Tainta Ausejo, S. (2017). Diseño y construcción de un robot auto-balanceado mediante Arduino [Tesis de grado, Universidad Pública de Navarra].
 - [28] Couoh Ku, R. I. (2020). Estimación de pose de robots móviles mediante mediciones de sensores inerciales y una cámara externa [Tesis de grado, Universidad Autónoma de Yucatán].

- [29] Bouabdallah, S., Murrieri, P., & Siegwart, R. (2004). PID vs LQ control techniques applied to an indoor micro quadrotor. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2451–2456.
- [30] Brown, R. G. (2019). Smoothing, forecasting and prediction of discrete time series. Dover Publications.
- [31] Smith, J., & Lee, A. (2020). Sensor data filtering techniques for IoT and embedded systems. *IEEE Internet of Things Journal*, 7(5), 3984–3996.
- [32] Welch, G., & Bishop, G. (1995). An introduction to the Kalman filter. University of North Carolina at Chapel Hill, Department of Computer Science.
- [33] Hähnel, D., Burgard, W., Fox, D., & Thrun, S. (2003). Learning compact 3D models of indoor and outdoor environments with a mobile robot. *Robotics and Autonomous Systems*, 44(1), 15–27.
- [34] Bradski, G. (2000). The OpenCV library. *Dr. Dobb’s Journal of Software Tools*.
- [35] Ahmad, M. A., & Muad, A. M. (2025). Fusion of image and inertial measurement unit (IMU) data to remedy inverse perspective mapping imagery. *Jurnal Kejuruteraan (Journal of Engineering)*, 37(4), 1637–1647. [https://doi.org/10.17576/jkukm-2025-37\(4\)-03](https://doi.org/10.17576/jkukm-2025-37(4)-03)
- [36] Hashem, A. S., & Kadhim, A. A. (2021). Design and implementation of a mobile application for smart home control on Android platform using Firebase real-time database. *Materials Today: Proceedings*, 42, 2556–2561.
- [37] OpenCV. (2023). Open source computer vision library. OpenCV Documentation. <https://docs.opencv.org/>
- [38] OpenCV. (2024). Deep neural network (DNN module). https://docs.opencv.org/4.x/d6/d0f/group__dnn.html

- [39] OpenCV. (2025). DNN-based face detection.
https://docs.opencv.org/4.x/d0/dd4/tutorial_dnn_face.html
- [40] Jia, Y., et al. (2024). Caffe: Deep learning framework. BVLC.
<https://caffe.berkeleyvision.org/>
- [41] Android Developers. (2024). Bluetooth overview.
<https://developer.android.com/guide/topics/connectivity/bluetooth>
- [42] Think Robotics. (2025). VL53L0X time-of-flight sensor tutorial: Complete setup and programming guide.
<https://thinkrobotics.com/blogs/tutorials/vl53l0x-time-of-flight-sensor-tutorial-complete-setup-and-programming-guide>
- [43] Maker Portal. (2024). MPU9250 blog tag resources.
<https://makersportal.com/blog/tag/MPU9250>
- [44] Google Firebase. (2025). Firebase documentation.
<https://firebase.google.com/docs>
- [45] Android Developers. (2024). Activity overview.
<https://developer.android.com/guide/components/activities/intro-activities>
- [46] Android Developers. (2024). Intent and intent filters.
<https://developer.android.com/guide/components/intents-filters>
- [47] Android Developers. (2024). User interface components.
<https://developer.android.com/develop/ui/views/components/button>
- [48] Android Developers. (2024). Bluetooth permissions.
<https://developer.android.com/guide/topics/connectivity/bluetooth/permissions>
- [49] Android Developers. (2024). Speech recognition.
<https://developer.android.com/reference/android/speech/RecognizerIntent>
- [50] Google Firebase. (2025). Firebase Realtime Database documentation.
<https://firebase.google.com/docs/database>

- [51] Google Firebase. (2025). Cloud Storage for Firebase.
<https://firebase.google.com/docs/storage>
- [52] Square. (2024). Picasso image loading library.
<https://square.github.io/picasso/>
- [53] Google Firebase. (2025). Agrega datos a Cloud Firestore.
<https://firebase.google.com/docs/firestore/manage-data/add-data>

ANEXOS

Los archivos ejecutables del proyecto se encuentran disponibles en el repositorio GitHub. Los enlaces se presentan a continuación:

- <https://github.com/Tecnotronick1/Tesis-Balancing-v1>
- https://github.com/elizabethpotes0325-cpu/Tesis_Potes_Elizabeth

Configuraciones previas para el sistema Rasbian Verificar el Sistema Operativo Para la configuración, se utilizó el Raspberry Pi Imager (v1.7.5) para instalar el sistema operativo Rasbian. Se establece la versión de 64 bits por su compatibilidad y mejor rendimiento en procesamiento de sensores [23].

Preparación de la Tarjeta SD Se utilizó una tarjeta SD de 125.1 GB, esta se insertó en el computador para flashear el sistema operativo Rasbian. Esta capacidad permite almacenar todos los datos de navegación y detección facial sin problemas de espacio, además de garantizar velocidad adecuada para operaciones de lectura y escritura [23].

Confirmación de Escritura El software Raspberry Pi Imager muestra una advertencia de seguridad sobre el borrado irreversible de todos los datos en la tarjeta SD. Esta verificación previene la pérdida accidental de información[23].

Proceso de Instalación La imagen del sistema operativo se escribe en la tarjeta SD con verificación automática. Este proceso tarda aproximadamente 10 minutos, dependiendo de la capacidad de la tarjeta y velocidad del computador utilizado. Durante esta fase, el sistema valida la integridad de los datos escritos para garantizar una instalación sin errores [23].

Instalación Exitosa Al finalizar, el software muestra un mensaje de confirmación. La tarjeta ya está lista para ser insertada en la Raspberry Pi y proceder con la configuración inicial [23].

Configuración de Video HDMI Para garantizar una visualización correcta, se modificó el archivo config.txt para forzar la salida HDMI y ajustar los parámetros de visualización [23].

Escaneo de Red Inicial Antes de proceder con la configuración del sistema, se realizó un escaneo de la red local utilizando **Advanced IP Scanner** para identificar la dirección IP asignada a la Raspberry Pi y verificar los dispositivos conectados [24]. Este paso es crucial para:

- Confirmar que la Raspberry Pi está correctamente conectada a la red.
- Identificar posibles conflictos de IP.
- Preparar la conexión remota (SSH/VNC).

Parámetro	Valor
Rango escaneado	10.177.49.1-254
Fabricante detectado	Raspberry Pi Foundation
Nombre de host	raspberrypi

Cuadro 6: Detalles técnicos del escaneo

Configuración de Acceso Remoto con RealVNC Para habilitar el control remoto de la Raspberry Pi, se configuró RealVNC mediante una cuenta personal asociada al correo institucional. Esta solución permite una conexión remota desde cualquier dispositivo, transferencia segura de archivos y configuración inicial sin necesidad de monitor físico [24].

Creación de Nueva Conexión en RealVNC Para gestionar múltiples accesos remotos, se configuró una nueva conexión en RealVNC Viewer, [25] con los siguientes parámetros:

Verificación de Seguridad en Conexión VNC Durante el establecimiento de la conexión remota, el sistema detectó un cambio en la identidad del servidor VNC, generando una advertencia de seguridad estándar. Este mecanismo protege contra posibles ataques de tipo "Man-in-the-middle".

Autenticación en el Servidor VNC Para establecer la conexión remota con la Raspberry Pi, se requirió la autenticación directa en el servidor VNC. Este paso es independiente de las credenciales de la cuenta RealVNC y utiliza las credenciales locales del sistema.

Autenticación Personalizada en VNC Server Tras la configuración inicial, se establecieron credenciales personalizadas para el servidor VNC,

reemplazando los valores por defecto. Este proceso fortalece la seguridad del acceso remoto.

Conexión Remota Exitosa Tras completar la autenticación, se estableció una conexión estable con el escritorio remoto de la Raspberry Pi mediante VNC. Esta interfaz permite el control completo del sistema para la configuración del proyecto [25] .

Configuración de Usuario Seguro Para mejorar la seguridad del sistema, se creó un usuario administrativo dedicado con privilegios controlados, reemplazando la cuenta predeterminada 'pi' [25].

Actualización Inicial del Sistema Tras el primer arranque, se ejecutaron los comandos básicos de actualización para asegurar que todos los paquetes estén en su última versión estable [31, 30]. Primero, se actualizó el sistema operativo Raspbian ejecutando los siguientes comandos en la terminal:

```
sudo apt-get update
sudo apt-get upgrade
```

Instalación de paquetes PIP

Para la correcta gestión de dependencias en Python [31, 30], se configuró el entorno mediante la instalación de PIP siguiendo estos pasos:

```
sudo apt remove python-pip
wget https://bootstrap.pypa.io/get-pip.py
sudo python get-pip.py
```

Configuración de GPIO Zero Para el control de los pines GPIO en Raspberry Pi, se procedió a instalar el paquete Python GPIO Zero [31, 30].

```
sudo apt-get install python-gpiozero python3-gpiozero
```

Instalación de Tkinter para Interfaz Gráfica

Para implementar una interfaz gráfica en la Raspberry Pi [31, 30], se instaló el paquete Tkinter mediante el siguiente comando:

```
sudo apt-get install python-tk
```

La Libertad, 23 de abril del 2026.

Señor

Ing. José Sánchez Aquino Msc.

DIRECTOR DE CARRERA DE ELECTRÓNICA Y TELECOMUNICACIONES

En su despacho.

De mi consideración:

Cumplo en informar el resultado obtenido en la revisión de plagio realizado por el software COMPILATO, del trabajo de titulación, "Diseño de un sistema de navegación inercial para control autónomo de un robot móvil utilizando control PID y filtro de Kalman, aplicado en gestión de domótica y localización de personas."



elaborado por la estudiante de la Carrera de Electrónica y Telecomunicaciones, Srta.

ELIZABETH CAROLINA POTES LINO, el cual dio como resultado el 7% de similitudes.

Atentamente,



Ing. Sendey Vera Gonzalez

TUTOR DE TRABAJO DE TITULACIÓN