



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

“Estimación de Estados Empleando Técnicas Basadas en la Teoría
del Filtro de Kalman para la Localización de un Robot

Móvil”

TRABAJO DE TITULACIÓN

Previo a la obtención del Título de:

MAGISTER EN AUTOMATIZACIÓN Y CONTROL INDUSTRIAL

ING. LEONELA DE LA A SALINAS

ING. MIGUEL ANGEL VELASCO

GUAYAQUIL – ECUADOR

2023

AGRADECIMIENTO

A ti, amado Jesucristo, Tú, con tu amor incondicional y tu gracia infinita, has sido mi roca en momentos de duda, por ser mi guía y acompañarme en el transcurso de mi vida.

Mi gratitud también se extiende a mi amada familia, a mis padres, cuyo amor inquebrantable y apoyo incondicional me han sostenido en cada paso del camino. A mis hermanos, cuyas risas y ánimos me han recordado la importancia de encontrar alegría en medio de los desafíos. Su presencia ha sido un regalo invaluable que ha enriquecido mi vida.

A mi querido novio, palabras no pueden expresar cuánto agradezco tu amor incondicional, paciencia y apoyo constante. Tu amor me ha inspirado a seguir adelante incluso en los momentos más difíciles.

A nuestro Coordinador de Programa, el PhD. Douglas Plaza G., por su gran apoyo y orientación en todo el proceso de formación y titulación.

Las palabras de gratitud y admiración a ti, mi Salvador, Tu amor ha sido mi fuerza motriz. Que mi vida y mi carrera sean un testimonio de tu gracia y poder transformador.

Con humildad y gratitud eterna,

Leonela De La A Salinas

AGRADECIMIENTO

Agradezco a Dios, primeramente, con su paciencia y misericordia me ha llevado por sus senderos, y me ha logrado sostener en tiempos difíciles, le agradezco a él por amor fraterno e incondicional.

Agradezco mucho a mis padres, por ser mi guía, por su esfuerzo y sacrificio en cada momento, en cada detalle, por su motivación en ayudarme a no rendirme, les debo todo.

A mis hermanos les agradezco por su respeto y paciencia, sus consejos me han ayudado a seguir cada paso con determinación, a no rendirme y soñar con un nuevo mañana.

A nuestro Coordinador de la Maestría, el PhD. Douglas Plaza G, por su apoyo incondicional hemos logrado culminar este trabajo con éxito.

Agradezco a todos ustedes,

Miguel Angel Velasco .

TRIBUNAL DE SUSTENTACIÓN



firmado electrónicamente por:
LIVINGTON ALFREDO
MIRANDA DELGADO

Ph.D. VANESSA CEDEÑO
SUB-DECANO FIEC

M.Sc. LIVINGTON MIRANDA D.
DIRECTOR

A handwritten signature in blue ink, appearing to read 'Wilton Agila Gálvez', written over a horizontal line.

PhD. Wilton Agila Gálvez
VOCAL PRINCIPAL

DECLARACIÓN EXPRESA

“La responsabilidad y la autoría del contenido de este Trabajo de Titulación, nos corresponde exclusivamente; y damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promoverla consulta, difusión y uso público de la producción intelectual”.



DE LA A SALINAS LEONELA DEL ROCIO



VELASCO PARRA MIGUEL ANGEL

Resumen

Durante la investigación se han utilizado los fundamentos matemáticos y algorítmicos del filtro de Kalman para la estimación de estados, y aplicarlos en la navegación autónoma de robots diferenciales móviles. El robot diferencial es un tipo de robot móvil que se desplaza utilizando dos ruedas independientes y unidas a un cuerpo.

Como solución se desarrolló un filtro de Kalman para estimar la posición y trayectoria de un robot móvil diferencial, adaptado y optimizado para la navegación autónoma de robots diferenciales móviles. Los datos recogidos de los sensores inerciales, la calibración y ajuste de los parámetros, logran estimar la trayectoria real del robot. A medida que el filtro de Kalman combina las mediciones y las predicciones, se ajusta los datos en el filtro.

Para planificar una trayectoria el robot recoge datos de velocidad, dirección y se controlan variando las velocidades de las ruedas izquierda y derecha. Los sensores utilizados para medir la posición y el entorno del robot fueron los encorder y, están sujetos a diversas fuentes de ruido, la presencia de ruido, incertidumbre y las interacciones dinámicas del entorno. Existen diferentes fuentes de ruido en un robot diferencial, algunas de las cuales son las siguientes: Ruido en los sensores, en un robot diferencial es una perturbación no deseada que afecta la precisión y el rendimiento del robot.

Se construyó un robot diferencial el cual corrige su trayectoria a partir de una trayectoria referencia mediante la aplicación del algoritmo de precisión, los resultados se validan a través de la simulación en Matlab-Simulink así también como del prototipo implementado.

El Filtro de Kalman analiza los datos y ajusta a la trayectoria establecida por el robot. El filtro de Kalman en la navegación autónoma de robots diferenciales móviles no solo representa un logro científico, sino también un salto hacia adelante en el campo de la robótica.

Palabras Claves.

Filtro de Kalman, Robot diferencial, Robot Móvil, Trayectoria, Matlab-Simulink, Sensores, Enconder, Estimar, Ruido.

ÍNDICE GENERAL

Contenido

AGRADECIMIENTO	3
AGRADECIMIENTO	2
TRIBUNAL DE SUSTENTACIÓN	3
DECLARACIÓN EXPRESA	4
Resumen	5
ÍNDICE GENERAL.....	6
ÍNDICE DE FIGURA.....	9
ÍNDICE DE TABLAS	¡Error! Marcador no definido.
INTRODUCCION.....	11
CAPÍTULO I	13
PLANTEAMIENTO DE LA PROBLEMÁTICA.....	13
1.1. Descripción del problema.	13
1.1.2. Problema de localización.....	14
1.2. Justificación.	15
1.4. Objetivos	17
1.4.1. Objetivos generales.....	17
1.4.2. Objetivos específicos.....	17
1.5. Metodología.	17
CAPÍTULO 2.	21
ESTADO DEL ARTE.	21
2.2. Sistemas lineales y no lineales.	21
2.3. Sistema Lineal:.....	21
2.4. Función Transferencia.	22
2.5. Sistema de control en espacios de estados	23
2.6. Análisis y diseño de variables de estado.	23
2.8. Análisis de controlabilidad y observabilidad.....	24
2.9. Que es un estimador de Kalman	27
2.15. Algoritmos empleados en la estimación de estados.....	30
2.15.1. Información a priori: Al determinar su ubicación, un robot tiene acceso a dos tipos de información.	30
CAPÍTULO 3.	33

MODELADO DEL ROBOT DIFERENCIAL DE NAVEGACION.	33
3.1. Construcción del robot diferencial.....	34
3.2. Componentes del robot diferencial	34
3.3. Diseño del robot.....	35
3.4. Diagrama de conexión del robot diferencial de navegación.....	37
3.5. Componentes electrónicos, sensores y actuadores del robot diferencial de navegación autónoma.	38
3.6. Modelado matemático del Robots Móviles de navegación.....	39
3.7. Ecuaciones para el observable o estimador de Kalman.....	43
3.8. Etapas del algoritmo del filtro de Kalman.....	46
CAPÍTULO 4	49
SIMULACION Y COMPARACION DEL FILTRO	49
4.1. Diseño y Simulación del Filtro de Kalman	49
4.2. Simulación del filtro de Kalman en MATLAB Simulink	49
4.3. Descripción de cada bloque del código.....	50
CAPÍTULO 5.	55
ANALISIS DE RESULTADOS.....	55
5. Simulación y resultados obtenidos	55
5.1. Navegación autónoma	56
5.2. Resultados de las estimaciones predichas por el filtro de Kalman	58
CONCLUSIONES	62
RECOMENDACIONES Y TRABAJOS FUTUROS	63
BIBLIOGRAFÍA.....	64
ANEXOS	67
1.- CODIGO DE PROGRAMACION DEL ROBOT NAVEGACION EN EL IDE ARDUNIO.....	67
2. CODIGO DE PROGRAMACION EM MATLAB Y RAFICAS DEL FILTRO DE KALMAN	71
3. FUNCIÓN PARA GRAFICAR AL ROBOT EN ANIMACIÓN MÓVIL.....	72
3. ETAPA DE CONSTRUCCION DE ROBOT DIFERENCIAL DE NAVEGACION AUTOMANA.....	77
4. MEDICION Y LECTURA DE LAS SEÑALES ENTREGADAS POR LOS SENSORES, ENCODERS, IMU.	79

ABREVIATURAS Y SIMBOLOGÍA

V	VELOCIDAD LINEAL DEL ROBOT
W	VELOCIDAD ANGULAR DEL ROBOT
FK	FILTRO DE KALMAN
EKF	FILTRO EXTENDIDO DE KALMAN
A PRIORI	VALOR ANTERIOR DEL VECTOR DE ESTADO
A POSTERIORI	VALOR CORREGIDO DEL VECTOR DE ESTADO
DC	DIRECT CURRENT
AC	ALTERNATE CURRENT
IMU	UNIDAD DE MEDICION INERCIAL
PID	CONTROL PROPORCIONAL INTEGRAL Y DERIVATIVO
IoT	INTERNET DE LAS COSAS
m	MASA
θ	ORIENTACION
I	INERCIA
a	ACELERACION

ÍNDICE DE FIGURA

Figura: 1 Sistemas lineales invariantes en el tiempo	21
Figura: 2 Sistemas no lineales.	22
Figura: 3 Rango y dominio entre el tiempo y de la frecuencia	23
Figura: 4 Sistema de control en variables de estados.....	24
Figura: 5 Matriz de variables	24
Figura: 6 Controlabilidad en espacios de estados.....	25
Figura: 7 Observabilidad de los estados.....	26
Figura: 8 Diagrama de flujo de la ejecución del algoritmo	30
Figura: 9 Diseño en 3D del robot diferencial	35
Figura: 10 Vista frontal del robot.	36
Figura: 11 Vista de atrás del robot diferencial.....	36
Figura: 12 Vista lateral del robot.....	37
Figura: 13 Esquema electrónico.....	38
Figura: 14 Disco del encoder	39
Figura: 15 Líneas de datos del Encoder.....	39
Figura: 16 Variables de la característica del vehículo	40
Figura: 17 Esquema del robot de navegación.	41
Figura: 18 Movimiento Acelerado Uniforme	43
Figura: 19 Diagrama de bloques de conexión del modelo dinámico del robot de navegación y del estimador de Kalman.....	45
Figura: 20 Predicción y Corrección.....	46
Figura: 21 Secuencia de ejecución de algoritmo.....	48
Figura: 22: Diagrama de bloque general del Simulink del filtro de Kalman.....	50
Figura: 23 Bloques de variables de estado del sistema	51
Figura: 24: Actualización y medición de las variables de estado del sistema	51
Figura: 25 Bloque del modelo dinámico del robot.....	52
Figura: 26 Ecuación dinámica del robot de navegación autónoma	52
Figura: 27 Bloque del modelo del espacio de estado KALMAN	53
Figura: 28 Modelo de estado del estimador de Kalman	53
Figura: 29 Bloque de programación de filtro de Kalman	54
Figura: 30 Bloque de covarianza de ruidos del proceso en el tiempo	54
Figura: 31 Matriz de covarianza de ruidos de las mediciones y de los estados.....	55
Figura: 32 Trayectoria real del robot diferencial de navegación autónoma.....	56
Figura: 33 Curvas de las velocidades lineal y angular del robot diferencial de navegación autónoma.....	57
Figura: 34 Curvas de error en la medición de las señales del vector posición $\{x, y\}$ en cada instante de tiempo	58
Figura: 35 Estimación de la trayectoria del robot mediante el filtro de Kalman	59
Figura: 36 Trayectoria del robot.....	60
Figura: 37 Error en la medición señales de las posiciones en $\{X,Y\}$	60

Figura: 38 Error en la medición de los sensores de velocidad del robot. Velocidad real del robot (Superior), Titulo (Inferior)	61
Figura: 39 Diseño del robot 3D	77
Figura: 40 Diseño lateral del robot 3D	78
Figura: 41 Estructura para sensores impreso 3D	78
Figura: 42 Robot con sus componentes.	79
Figura: 43 Prueba de datos Obtenidos del robot vía Bluetooth	79
Figura: 44 Visualización de datos obtenidos del robot vía Bluetooth	80
Figura: 45 Datos obtenidos Serial	81
Figura: 46 Datos obtenidos de velocidad Lineal y Angular.	82

INTRODUCCION

La robótica ha ayudado en la automatización de muchas actividades. Los robots han revolucionado múltiples procesos industriales para producir productos en masa hace varias décadas, pero estos robots no son más que máquinas versátiles que ejecutan un programa complejo pero fijo. En general, los robots de fabricación no exhiben inteligencia autónoma. En su mayoría no son conscientes de su entorno y se utiliza información sensorial limitada durante la operación. Por lo tanto, se tiene mucho cuidado para que el entorno del robot sea lo más predecible posible y logre autonomía. (Angeles, 2003)

La autonomía de los robots depende de hasta qué punto un robot se basa en conocimientos previos o información del entorno para realizar sus tareas. Esta falta de autonomía es una de las principales limitaciones que deben superarse para permitir que los robots se vuelvan móviles en un entorno previamente conocido. (SICMA21, 2021). En muchos casos, los robots se controlan manualmente para moverse desde el origen hasta su destino. Sin embargo, se han llevado a cabo varios estudios sobre robots autónomos que han dado lugar a toda una serie de aplicaciones potenciales. A medida que la tecnología continúa avanzando, es probable que surjan aún más posibilidades en diversas áreas, mejorando la eficiencia, seguridad y comodidad en numerosos aspectos de nuestras vidas. (Hilario López García, 1996)

La navegación es una tarea compleja que se basa en el desarrollo de una representación interna del espacio, basada en puntos de referencia reconocibles y un procesamiento visual, que puede soportar simultáneamente la auto localización continua y una representación del destino. (Díaz, 2013)

El análisis de datos se realizó utilizando técnicas estadísticas y herramientas de procesamiento de datos, permitiendo examinar las estimaciones de estado proporcionadas por el filtro de Kalman y compararlas con los valores de referencia. El robot primero deberá crear un modelo, este modelo generalmente debe comprender una trayectoria deseada, con la ayuda de los sensores y así obtener resultados.

Los resultados obtenidos fueron discutidos en relación con los datos obtenidos de los sensores. Se analizaron los resultados, y se realizaron pruebas sobre la efectividad de las técnicas basadas en el filtro de Kalman utilizadas en el estudio. Finalmente, se obtuvo conclusiones fundamentadas a partir de los resultados obtenidos.

En este estudio se detallan los siguientes capítulos:

En el Capítulo I, se presenta descripción del problema y la solución del problema del desarrollo de un algoritmo de estimación empleando las técnicas basadas en el filtro de Kalman para la localización de un robot móvil, también se da a conocer metodología y el alcance del tema propuesto.

En el capítulo II, se expone el estado del arte, se describe toda la teoría de control desde los sistemas dinámicos hasta los algoritmos de predicción en espacio de estados.

El capítulo III. Explica la construcción del robot móvil diferencial, el modelado matemático y la estimación del filtro de Kalman. Posteriormente, se hace una descripción general de cada uno de los componentes electrónicos y mecánicos.

El capítulo IV se describe la simulación y la comparación del filtro con su respectivo análisis de resultados.

En el capítulo V se presenta el análisis de los resultados del robot implementado.

El capítulo culmina con las conclusiones y recomendaciones anexos y bibliografía.

Se resumieron los resultados más relevantes y se proporcionaron recomendaciones prácticas para mejorar el sistema de localización.

CAPÍTULO I

PLANTEAMIENTO DE LA PROBLEMATICA.

1.1. Descripción del problema.

Hoy en día contamos con máquinas que facilitan las tareas para el ser humano, como la limpieza en el hogar, el envío de mercaderías en zonas menos transitadas, la inteligencia artificial para la toma de decisiones, la automatización de procesos en la industria, la cuarta revolución industrial, inclusive en el direccionamiento de misiles y navegación en el espacio. (Bonilla, 2018)

Tenemos como ejemplo el robot SWEEP que se comercializan en los supermercados para realizar la limpieza en el piso de los hogares, también tenemos al dron mensajero de Amazon el Prime Air, que está direccionado a enviar paquetes de hasta 5 libras en un rango de 24km en menos de 30 minutos, (Rubin, 2019) también tenemos a los robots móviles industriales cuya función es cargar con objetos pesados y enviarlos de un lugar a otro, y tenemos otros ejemplos como el auto autónomo de Google y el robot espacial Curiosity. (MARTÍ, 2018)

La mayoría de las funciones anunciadas anteriormente se los puede ejecutar por medio de robots móviles, que poseen la tecnología de navegación autónoma, cuyo fin es que el robot móvil se traslade desde un punto de origen a un punto final, realizando una acción específica de una manera segura y eficiente, determinando así su recorrido y localización en un entorno desconocido.

En la actualidad en el país se utilizan sistemas móviles para obtener información específica adquirida de los sensores, de distinta naturaleza con diversos fines por ejemplo ubicación de postes de alumbrado eléctrico, actualización de mapas digitales, etc. El poder contar con un sistema de estimación de precisión contribuirá a mejorar la calidad de los datos obtenidos de los sensores. (Bonilla,2018).

En el Ecuador la navegación autónoma está en vías de desarrollo con respecto al avance tecnológico, una de las razones importantes considerada como una problemática es el control de la precisión, y seguimiento de trayectorias, puesto que todos los robots deben poseer sensores, tales como: sensores de localización, sensores de velocidad, sensores de sonido, sensores de visión, sistemas de unidades inerciales, odometría, GPS, etc. (Villa Lopez, 2018).

Estos sensores por lo general presentan errores de magnitud en sus variables físicas. Se requiere entonces disminuir estos errores de medición a través de algoritmos de estimación de estados que eviten posibles fallas de lectura en estos dispositivos que son sensibles al ruido. (Santos, 2019).

1.1.2. Problema de localización.

El problema de la localización es un problema importante. Es un componente clave en muchos sistemas de robots autónomos exitosos. Si un robot no sabe dónde está en relación con el entorno, es difícil decidir qué hacer. Lo más probable es que el robot necesite tener al menos una idea de dónde está para poder operar y actuar con éxito. Algunos autores han declarado que el problema de la localización de robots es el "problema más fundamental para proporcionar capacidades verdaderamente autónomas a los robots". (Gustavo Velasco Hernandez, 2019)

El problema general de localización tiene una serie de instancias de problemas cada vez más difíciles. En el problema de seguimiento de posición, el robot conoce su ubicación inicial. El objetivo de la localización es realizar un seguimiento de la posición mientras el robot navega por el entorno. Las técnicas que resuelven este problema se denominan técnicas de rastreo o técnicas locales, ya que el robot no conoce su posición inicial. Tiene que localizarse desde cero. Por la presente, posiblemente necesite poder manejar múltiples ideas sobre su ubicación. (Gustavo Velasco Hernandez, 2019).

Las técnicas que resuelven este problema de localización también se pueden utilizar para resolver el problema del robot. Un factor que complica cada uno de estos problemas es la dinámica del entorno en el que se desplaza el robot. La mayoría de las investigaciones sobre localización se han centrado en realizar la localización en entornos estáticos.

Esto significa que el robot es el único objeto en movimiento en el entorno. Obviamente este no es el caso en el mundo real. Los entornos dinámicos contienen otros. Objetos en movimiento y en estos entornos la localización es significativamente más difícil, ya que estos otros objetos pueden confundir al robot acerca de su ubicación al corromper la información utilizada para la localización (Negenborn, 2003).

Se busca abordar un sin número de problemas presentes en los robots como son los siguientes:

- **Mapeo de trayectoria:** el problema del mapeo de una trayectoria mediante la construcción e implementación de un robot móvil diferencial es requerido por que hoy en día lograr la autonomía de un robot está en proceso de desarrollo.
- **Localización:** La localización en la navegación es un problema a partir de una posición desconocida. En donde es difícil estimar la posición de un robot.
- **Error de estimación:** El error aparece por el ruido presente en los sistemas de lectura y sensores de medición, se debe estimar el estado instantáneo del sistema dinámico lineal perturbado.
- **Adquisición de datos. Sensores:** Los valores de los sensores, muchas veces dan falsas lecturas, o datos erróneos, o también hay ruidos presentes y es difícil predecir el futuro valor que debería dar el sensor.

1.2. Justificación.

Las aplicaciones novedosas de los sistemas autónomos como por ejemplo la entrega de paquetes con drones o vehículos sin conductor dependen fuertemente de la precisión de la ubicación hacia donde deben dirigirse, ya que al ser autónomos no cuentan con un conductor que pueda corregir posibles errores. En este sentido es importante combinar la información de todos los sensores presentes en el vehículo móvil para mejorar la precisión de la ubicación actual del robot con respecto a la ubicación hacia donde se dirige. La combinación de localización con seguimiento de trayectoria es la razón que permite al robot lograr la autonomía (Campoverde, 2018).

La ubicación de personas y objetos es una información relevante que se la puede utilizar en distintas aplicaciones. Hoy en día contamos con tecnología que nos permite acceder a dicha

información, sin embargo; la exactitud de la información dependerá de la precisión de los sensores. (Armada, 2015)

En esta investigación, se desarrolla un algoritmo de estimación de Kalman para disminuir los errores de ruido y los errores generados por el sensor, cuando se mueve una trayectoria predefinida o en el entorno de una plataforma robótica autónoma. Adaptar esto continuamente podría volverse tedioso y costoso, por lo que proponemos un algoritmo, mediante la odometría adquirido en el vehículo autónomo con el que podemos corregir los datos en la salida del sensor para garantizar que el robot móvil mejore su trayectoria.

1.3. Solución Propuesta.

1. **Robot móvil:** El estudio de la localización del robot móvil está basado en técnica de la teoría del filtro de Kalman, con el propósito de la precisión y la confiabilidad de la estimación del estado del robot, otorgándole la capacidad de navegar y orientarse de manera autónoma.
2. **Componentes:** En la solución propuesta el robot cuenta con componentes y algoritmos. Entre los sensores comunes utilizados en la localización de robots móviles se encuentran los sensores de adquisición de datos de información los encoders y los IMU, que define la orientación y el entorno del robot móvil.
3. **Datos:** Los datos obtenidos de los sensores se someten a un filtrado de señales, donde se corrigen los errores sistemáticos presentados en la lectura de los encoders.
4. **Modelo:** La creación del modelo describe el estado del robot móvil en el transcurso del tiempo, encierran la dinámica del movimiento del robot, las mediciones de los sensores y las inciertas perturbaciones externas, este modelo permitirá predecir el estado del robot.
5. **Filtro de Kalman:** El filtro de Kalman recoge los datos, de los sensores y las predicciones del modelo, esta estadística matemática del filtro de Kalman hace que funcione con exactitud, revelando estimaciones del estado del robot móvil que muestran precisión y confiabilidad. Así, el filtro de Kalman guía al robot, mejora las estimaciones a medida que recoge los datos en tiempo real.
6. **Estimación:** Se actualiza y corrige las estimaciones del estado del robot utilizando el filtro de Kalman, lo que permite corregir y adaptar el modelo del sistema, mejorando la precisión de la localización, permitiendo que el robot se mueva en su entorno.

La solución propuesta también permite realizar pruebas y experimentos para validar su eficacia y evaluar su desempeño en diversos escenarios. Además, los parámetros del filtro de Kalman y los modelos empleados buscan precisión.

El problema de localización más sencillo es la navegación a partir de una posición inicial conocida. En este caso, la solución consiste en estimar la posición final del robot compensando los errores incrementales de odometría acumulados por sus encoders. En resumen, esta solución propuesta abre las puertas a un futuro de autonomía y precisión, donde los robots móviles se despliegan en entornos complejos.

1.4. Objetivos

1.4.1. Objetivos generales

Desarrollar un algoritmo de estimación para la localización y seguimiento de trayectorias de un robot móvil diferencial empleando las técnicas basadas en el filtro de Kalman

1.4.2. Objetivos específicos

- Formular un modelo cinemático que describa el comportamiento del robot diferencial.
- Desarrollar un estimador de Kalman para la estimación de la trayectoria realizada por el robot diferencial
- Simular el modelo matemático y los estimadores con la herramienta Matlab.
- Identificar la instrumentación necesaria para la localización del robot para el sistema de medición inercial (IMU).
- Validar la respuesta de los estimadores en MATLAB-SIMULINK mediante comparaciones entre los estimadores.

1.5. Metodología.

En este estudio sobre la estimación de estados empleando técnicas basadas en desarrollar un algoritmo de estimación para la localización y seguimiento de trayectorias de un robot móvil diferencial empleando las técnicas basadas en el filtro de Kalman, se ha seguido una metodología de investigación rigurosa y cuidadosamente diseñada.

- 1. Diseño metodológico:** La metodología se ha desarrollado con el objetivo de obtener resultados confiables y significativos, contribuyendo al avance en el campo de la robótica y la navegación autónoma.
- 2. Método Hipotético deductivo:** La metodología de investigación abarcó varias etapas esenciales para lograr los objetivos planteados. Se incluyeron aspectos como el diseño de investigación, la definición de variables, la recopilación de datos, la implementación del sistema de localización, las pruebas y validación, el análisis de datos, la discusión de resultados, y las conclusiones y recomendaciones.
- 3. Método Experimental.** El diseño de investigación seleccionado se adaptó a las necesidades específicas del estudio, considerando si se requería un enfoque experimental o un diseño de estudio de caso, entre otros. Esto permitió establecer una estructura sólida para llevar a cabo la investigación de manera efectiva.

Se definió un conjunto de variables relevantes para el estudio, incluyendo variables dependientes e independientes, que permitieron analizar y medir los aspectos clave de la estimación de estados en la localización del robot móvil.

Medición: Esto incluyó la selección y configuración de sensores, así como la realización de simulaciones, experimentos en laboratorio o pruebas en entornos reales, según fuera necesario.

Localización: La implementación del sistema de localización implicó el desarrollo y la integración de técnicas y algoritmos basados en el filtro de Kalman. Esto incluyó la programación de software, la configuración de sensores y la integración de componentes hardware, según los requisitos específicos del estudio.

Validación: Se llevaron a cabo pruebas exhaustivas para validar el sistema de localización y evaluar su desempeño en diferentes escenarios. Esto permitió establecer métricas de evaluación apropiadas y obtener resultados representativos.

1.5.2. Etapa 1: Diseño del robot

En la primera etapa se va a implementar un robot móvil autónomo diferencial que pueda seguir una trayectoria específica, para su funcionamiento este incluirá sistemas embebidos, sensores (enconder.) y actuadores (motores).

De los sensores se toman los datos para obtener información del entorno ya sea posición, velocidad, aceleración o proximidad, en función de la lectura de los transductores, que son señales eléctricas que ingresan al controlador, para que realicen una función específica programada por el usuario.

De los actuadores tenemos a los motores que permiten realizar una acción es decir una ejecución de los movimientos de su trayectoria, esto permite al robot moverse en distintas direcciones al entrar en acción.

1.5.3. Etapa 2: Creación del modelo matemático para estimar los estados del robot

En la segunda etapa se desarrollará un modelo matemático que describa el robot el cual se lo modelará en espacio de estados, nos servirá para desarrollar los estimadores de Kalman bayesiano, para predecir las posiciones y las velocidades reales del robot

Se desarrolla el modelo matemático de un robot móvil tipo diferencial, se consideran las características cinemáticas, dinámicas y eléctricas del motor. Se realizan algunas simulaciones empleando Simulink para comprobar el correcto funcionamiento del modelo.

1.5.4. Etapa 3: Validación de los datos

Analizar los resultados de los filtros implementados como es el filtro de Kalman y el filtro bayesiano (Partículas) para determinar los estados de posición y orientación del robot móvil autónomo en cada instante de tiempo, usando el sistema de odómetro basado en los enconder y así minimizar los errores entregado por los sensores, cuanto menor es el error de los sensores los datos serán más exactos.

1.6. Alcance.

En Ecuador se utilizan sistemas móviles para obtener información específica adquiridas de los sensores, de distinta naturaleza con diversos fines. El poder contar con un sistema de estimación de precisión contribuirá a mejorar los datos obtenidos de los sensores. Los

robots poseen sensores, tales como: sensores de localización, sensores de velocidad, sensores de sonido, sensores de visión, sistemas de unidades inerciales, odometría, GPS, etc.

El desarrollo de un algoritmo de estimación para la localización y seguimiento de trayectorias de un robot móvil diferencial empleando las técnicas basadas en el filtro de Kalman, que pueda tomar decisiones previamente controladas mediante la detección de señales y errores en la lectura definidas por la odometría, basándose en los principios de la cinemática y dinámica de un cuerpo móvil.

Para el control y seguimiento de las trayectorias, se pueden trabajar en otros tipos de estimadores que permitan de manera eficiente encontrar los puntos comunes y críticos en la eliminación de todo el ruido presente que causan las malas lecturas de los sensores.

Se pretende identificar los resultados obtenidos de la implementación a través de un software de simulación Simulink, y comprobar los resultados que puedan predecir la corrección en la implementación del robot móvil.

La aplicación de la teoría del filtro de Kalman en la estimación del estado del robot desafía los límites de lo posible, en la localización robótica, la navegación autónoma.

CAPÍTULO 2.

ESTADO DEL ARTE.

2.1. Teoría de control.

Es la acción o el efecto de poder decidir sobre el desarrollo de un proceso o sistema. También se puede entender como la forma de manipular ciertas variables para conseguir que ellas u otras variables actúen en la forma deseada. (Hernandez Gaviño, 2010, pág. 2)

2.2. Sistemas lineales y no lineales.

Un sistema lineal es aquel que obedece a las propiedades de escala (homogéneo) y superposición (aditivo), mientras que un sistema no lineal es todo sistema que no obedece al menos a una de estas características. La mayoría de los sistemas de la vida real tienen características no lineales. (Hernandez Gaviño, 2010, pág. 118)

2.3. Sistema Lineal:

Un sistema lineal la salida deberá reflejar los mismos cambios generados en la entrada. Un sistema invariante en el tiempo es aquel que no depende de cuando ocurre: la forma de la salida no cambia con el retraso de la entrada.



Figura: 1 Sistemas lineales invariantes en el tiempo

Los sistemas lineales son aquellos sistemas que están representado por ecuaciones diferenciales lineales:

Ecuaciones diferenciales lineales: Son aquellas ecuaciones en donde la

variable dependiente y todas sus derivadas son de primer grado, es decir la potencia de todo término función de la variable dependiente es uno y además los coeficientes de todos los términos son constantes o si son variables, solo dependen del tiempo (t), que es la variable independiente. (Hernandez Gaviño, 2010, pág. 75)

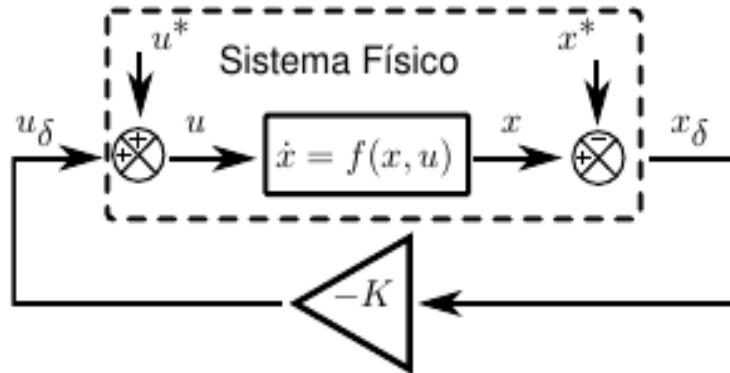


Figura: 2 Sistemas no lineales.

2.4. Función Transferencia.

Es una expresión matemática que caracteriza las relaciones de “Entrada – Salida” de sistemas lineales invariantes en el tiempo, se define como la relación de la transformada de Laplace de la salida (Dorf & H.Bishop, 2022, pág. 95)

Sea el sistema lineal invariante en el tiempo definido por la siguiente ecuación: $a_0 y^n + a_1 y^{n-1} + \dots + a_n y = b_0 x^n + b_1 x^{n-1} + \dots + b_m x$ [1]

Aplicando la transformada de Laplace, nos queda

$$(a_0 S^n + a_1 S^{n-1} + \dots + a_n) Y(S) = (b_0 S^n + b_1 S^{n-1} + \dots + b_m) X(S) \quad [2]$$

$$G(S) = \frac{b_0 S^n + b_1 S^{n-1} + \dots + b_m}{a_0 S^n + a_1 S^{n-1} + \dots + a_n} \quad [3]$$

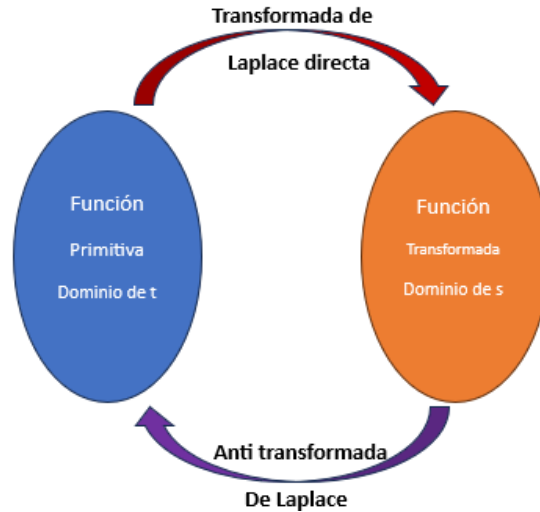


Figura: 3 Rango y dominio entre el tiempo y de la frecuencia

2.5. Sistema de control en espacios de estados

El análisis de espacio de estados es un excelente método para el diseño y análisis de sistemas de control. El método convencional y antiguo para el diseño y análisis de sistemas de control es el método de la función de transferencia. (Dominguez & Campoy, 2006, pág. 17)

2.6. Análisis y diseño de variables de estado.

Estado: El estado de un sistema dinámico es el conjunto más pequeño de variables llamadas variables de estado tal que el conocimiento de estas variables en el tiempo $t=t_0$ (Condición inicial), junto con el conocimiento de la entrada para $t \geq t_0$ determina completamente el comportamiento del sistema para cualquier tiempo.

2.7. Modelo de estado

Consideremos que un sistema de múltiples entradas y múltiples salidas tiene ingresos $u_1, u_2, t, \dots, u_r(t)$ y de las salidas $y_1, y_2, t, \dots, y_m(t)$ de las variables de estado $x_1, x_2, t, \dots, x_n(t)$ Entonces el modelo de estado viene dado por la ecuación estado y salida. (Dominguez & Campoy, 2006, pág. 12)

$$\frac{dX}{dt} = A x + B u \quad [4]$$

$$Y = C x + D u \quad [5]$$

- A, B es una matriz de estado de tamaño ($n \times n$)
- C es la matriz de salida de tamaño ($m \times n$)
- D es la matriz de transmisión directa de tamaño ($m \times r$)
- X(t) es el vector de estado de tamaño ($n \times 1$)
- Y(t) es el vector de salida de tamaño ($m \times 1$)
- U(t) es el vector de entrada de tamaño ($r \times 1$)

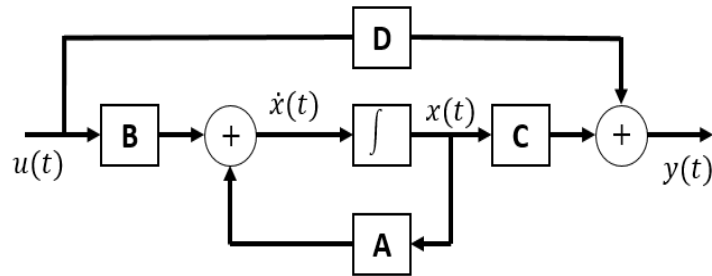


Figura: 4 Sistema de control en variables de estados

Las ecuaciones de estado son un conjunto de n expresiones operativas simultáneas. Los métodos comunes para resolver ecuaciones algebraicas lineales, por ejemplo, la eliminación de Gauss, la regla de Cramer, la matriz inversa, la eliminación y la sustitución, pueden aplicarse directamente a ecuaciones operativas lineales. (Dominguez & Campoy, 2006, pág. 10)

$$[sI - A] = \begin{pmatrix} (s-a_{11}) & -a_{12} & \dots & -a_{1n} \\ -a_{21} & (s-a_{22}) & \dots & -a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -a_{n1} & -a_{n2} & \dots & (s-a_{nn}) \end{pmatrix}$$

Figura: 5 Matriz de variables

2.8. Análisis de controlabilidad y observabilidad

Controlabilidad: Un sistema de control es de estado completamente controlable, si es posible transferir el sistema desde un estado inicial arbitrario a cualquier otro estado deseado en un intervalo de tiempo finito. (Dominguez & Campoy, 2006, pág. 132)

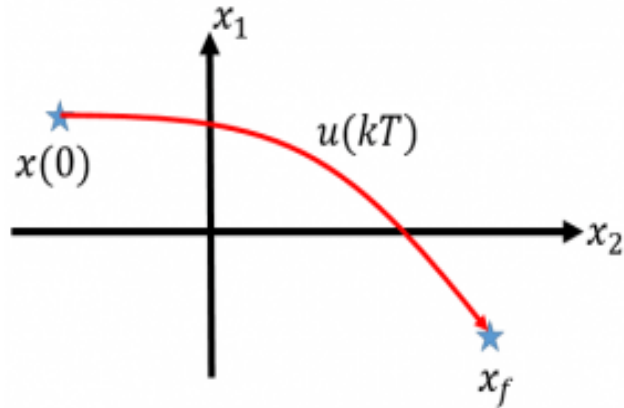


Figura: 6 Controlabilidad en espacios de estados.

Si se tiene un sistema dado por:

$$\begin{aligned}x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k)\end{aligned}$$

La variación del estado solo depende de las matrices A y B, así que, expandiendo el estado hasta n, tenemos que:

$$x(1) = Ax(0) + Bu(0)$$

$$x(2) = Ax(1) + Bu(1) = A^2x(0) + ABu(0) + Bu(1)$$

⋮

$$x(n) = A^n x(0) + A^{n-1}Bu(0) + A^{n-2}Bu(1) + \dots + ABu(n-2) + Bu(n-1)$$

La expresión anterior se puede escribir en la forma:

$$x(n) - A^n x(0) = [B \quad AB \quad A^2B \quad \dots \quad A^{n-1}B] \begin{bmatrix} u(n-1) \\ u(n-2) \\ \vdots \\ u(1) \\ u(0) \end{bmatrix}$$

El vector de estados $x(k)$ tiene una dimensión de n, de esta forma implica que se deben generar n ecuaciones simultaneas, y para ello la matriz debe ser igual a n, o sea debe tener rango completo para afirmar que el sistema es controlable.

La observabilidad es una propiedad importante de un sistema de control, y la propiedad de observabilidad juega un papel crucial en muchos problemas de control, como la estabilización de sistemas inestables por retroalimentación o el control óptimo. (Dominguez & Campoy, 2006, pág. 198)

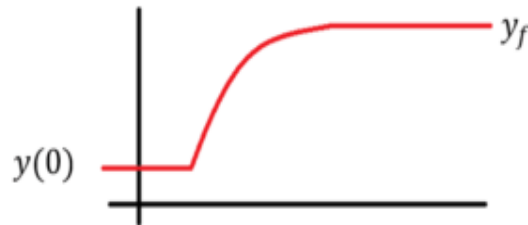


Figura: 7 Observabilidad de los estados

Considere un sistema lineal, invariante en el tiempo y de tiempo discreto en la forma de espacio de estado $X(k+1) = A X(k)$ $X(0) = x_0$ con medición de salida en la variable de salida $Y(k) = C X(k)$, donde $X(k) \in \mathbb{R}^n$ y $Y(k) \in \mathbb{R}^m$ A y C son matrices constantes de dimensiones. (G.C. Goodwin, 2000-)

$X(0) = x_0$, desde el n -vector dimensional $X(0)$ tiene n componentes desconocidos, se espera que n mediciones son suficientes para determinar X_0 toma $k = 0, 1, \dots, n-1$, y generar la siguiente secuencia (G.C. Goodwin, 2000-)

$$\begin{aligned} y(0) &= C_d x(0) \\ y(1) &= C_d x(1) = C_d A_d x(0) \\ y(2) &= C_d x(2) = C_d A_d x(1) = C_d A_d^2 x(0) \\ &\vdots \\ y(n-1) &= C_d x(n-1) = C_d A_d^{n-1} x(0) \end{aligned}$$

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ \vdots \\ y(n-1) \end{bmatrix}^{(np) \times 1} = \begin{bmatrix} C_d \\ C_d A_d \\ C_d A_d^2 \\ \vdots \\ C_d A_d^{n-1} \end{bmatrix}^{(np) \times n} \times x(0)$$

Sabemos por álgebra lineal que el sistema de ecuaciones algebraicas lineales con n incógnitas tiene solución única si y solo si la matriz del sistema tiene rango n

$$\text{rank} \begin{bmatrix} C_d \\ C_d A_d \\ C_d A_d^2 \\ \vdots \\ C_d A_d^{n-1} \end{bmatrix} = n$$

2.9. Que es un estimador de Kalman

El filtro de Kalman es un algoritmo de estimación utilizado en sistemas dinámicos para combinar mediciones ruidosas con estimaciones previas para obtener estimaciones más precisas del estado del sistema. (Nakhmani, 2020, pág. 105)

2.10. Filtro de Kalman:

Los filtros de Kalman requieren un modelo matemático del sistema que describe cómo evoluciona el estado del sistema en el tiempo. Este modelo incluye las ecuaciones de estado que relacionan el estado actual con el estado anterior y las ecuaciones de medición que relacionan las mediciones con el estado del sistema.

El estimador de Kalman consta de dos etapas principales: la etapa de predicción y la etapa de actualización.

- A. Etapa de predicción: En esta etapa, se utiliza el modelo de evolución del sistema para predecir el estado futuro del sistema. Se estima el estado y su incertidumbre utilizando la estimación previa del estado y la matriz de covarianza del estado anterior
- B. Etapa de actualización: En esta etapa, se incorporan las mediciones obtenidas del sistema para corregir la estimación previa. Se calcula la ganancia de Kalman, que determina cómo se ponderan las mediciones y las estimaciones previas. Luego, se actualiza la estimación del estado y su matriz de covarianza utilizando la ganancia de Kalman y la diferencia entre las mediciones y las predicciones. El modelo matemático utilizado en el filtro de Kalman

describe cómo evoluciona el estado del sistema en el tiempo. En el caso más simple, se utiliza un modelo lineal que se puede representar mediante un sistema de ecuaciones de primer orden. El modelo puede escribirse en forma discreta utilizando la notación de tiempo discreto (S.Grewal & P.Andrews, 2008, pág. 138)

2.11. Modelo y Predicción del estado.

El modelo de evolución del sistema se puede expresar de la siguiente manera:

Ecuación de predicción del estado:

$$X_K = A_{K-1} X_{K-1} + B_{K-1} U_{K-1} + W_{K-1}$$

Donde

1. X_{K-1} -- Es el vector de estado en el instante de tiempo
2. A_{K-1} -- Es la matriz de transición de estado que describe cómo evoluciona el estado del sistema desde el instante de tiempo $k-1$ al instante de tiempo
3. B_{K-1} -- Es la matriz de control que representa la influencia de la entrada de control U_{K-1} en el estado del sistema. U_{K-1} es el vector de entrada de control en el instante de tiempo
4. W_{K-1} -- Es el vector de ruido de proceso que representa las perturbaciones o incertidumbres en el modelo de evolución del sistema. (Nakhmani, 2020, pág. 125)

2.12. Ecuación de predicción de la matriz de covarianza del estado:

$$P_K = A_{K-1} P_{K-1} A_{K-1}^T + Q_{K-1}$$

Donde:

1. P_K - Es la matriz de covarianza del estado en el instante de tiempo k .
2. Q_{K-1} -- Es la matriz de covarianza del ruido de proceso que modela la incertidumbre en el modelo de evolución del sistema.

Una vez que se ha realizado la predicción del estado y la matriz de covarianza, se procede a la etapa de actualización utilizando las mediciones disponibles. En esta

etapa, se calcula la ganancia de Kalman, que determina cómo se combinan las mediciones y las estimaciones previas para obtener la estimación corregida del estado (S.Grewal & P.Andrews, 2008, pág. 144).

2.13. La ecuación de la ganancia de Kalman se puede expresar de la siguiente manera:

$$K_k = P_k H_k^T (H_k P_k H_k^T + R_k)^{-1}$$

Donde:

- K_k Es la ganancia de Kalman en el instante de tiempo k.
- H_k - Es la matriz de observación que relaciona el estado del sistema con las mediciones disponibles en el instante de tiempo k-1.
- R_k Es la matriz de covarianza del ruido de medición que representa las incertidumbres en las mediciones disponibles en el instante de tiempo k.

Finalmente, la estimación actualizada del estado del sistema y la matriz de covarianza se calculan utilizando la ganancia de Kalman y las mediciones disponibles. (S.Grewal & P.Andrews, 2008, pág. 397) .

2.14. Ecuación de actualización del estado:

$$X_k = X_k + K_k (Z_k - H_k * X_k)$$

Donde:

- Z_k - Es el vector de mediciones disponibles en el instante de tiempo k.
- Ecuación de actualización de la matriz de covarianza del estado:

$$P_k = (I - K[k] * H[k]) * P[k]$$

Donde: I es la matriz de identidad. - Estas ecuaciones forman la base matemática del filtro de Kalman y son utilizadas en cada iteración del

proceso de predicción y actualización para obtener estimaciones más precisas del estado del sistema.

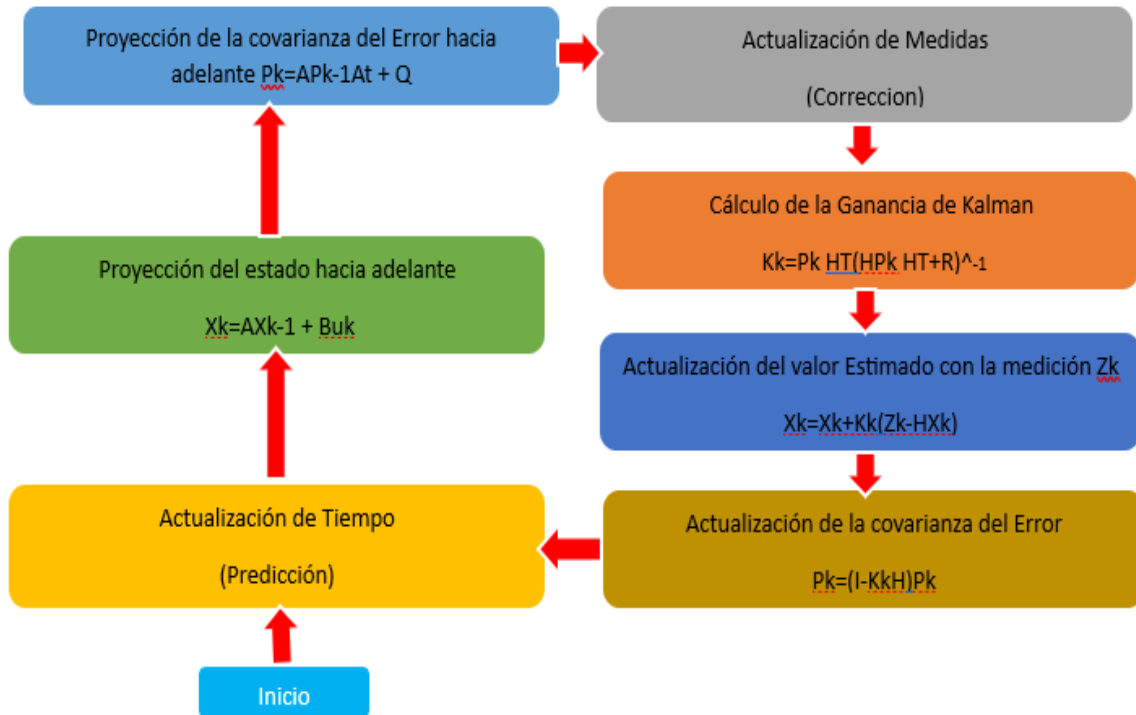


Figura: 8 Diagrama de flujo de la ejecución del algoritmo

2.15. Algoritmos empleados en la estimación de estados.

La implementación de los algoritmos demanda sistemas robustos de procesamiento, lo cual pone en riesgo la operación en tiempo real del estimador cuando se desea modelar sistemas dinámicos en plena operación. Se propone una metodología para la estimación en tiempo real de parámetros de sistemas dinámicos lineales mediante el método del Filtro de Kalman. (Bravo, 2013)

2.15.1. Información a priori: Al determinar su ubicación, un robot tiene acceso a dos tipos de información.

- Primero, tiene información a priori recopilada por el propio robot o proporcionada por una fuente externa en una fase de inicialización.

En segundo lugar, el robot obtiene información sobre el entorno a través de cada observación y acción que realiza durante la navegación.

En general, la información a priori suministrada al robot describe el entorno en el que se desplaza el robot. Especifica ciertas características que no varían en el tiempo y, por lo tanto, se pueden usar para determinar una ubicación. (Chen., 1999). La información a priori puede venir en diferente información que se sepan como ubicación geográfica del entorno, conocimientos previos en base a la experiencia del programador etc.

- Otra forma de proporcionar información a priori al robot es en términos de relaciones causa-efecto. Dada la entrada de las observaciones, estas las relaciones le dicen al robot dónde está. Posiblemente el robot pueda ajustar estas relaciones causa-efecto mientras navega por el entorno.

2.16. Información a posteriori.

El segundo tipo de información a la que tiene acceso un robot es la de navegación. información, es decir, información que el robot recopila de sus sensores mientras navegando por el entorno.

Un robot normalmente realiza dos tipos de acciones alternas cuando navega; conduce o actúa en el ambiente, por un lado, y siente el ambiente por el otro. Estos dos tipos de acciones dan lugar a dos tipos diferentes de información de posición.

Para poder moverse en un entorno, un vehículo robótico tiene un sistema de guía o conducción. Un sistema de guía puede consistir en ruedas, pistas en principio cualquier cosa que haga moverse al vehículo, estos componentes se denominan actuadores. Obviamente, el sistema de guía juega un papel importante en el físico posición de un robot. (John C. Doyle, 1992)

El sistema de guía cambia directamente la ubicación del vehículo. Sin un sistema de guía, el robot no se desplaza, lo que hace que localizarse sea mucho más fácil.

Suponiendo que un robot tiene un sistema de guía, la forma en que el sistema de guía cambia la ubicación contiene información valiosa para estimar la locación, conocer los efectos de las acciones ejecutadas por el sistema de conducción da una indicación directa de la ubicación del vehículo después de la ejecución de estos comportamientos.

Al monitorear lo que realmente hace el sistema de conducción usando sensores, se puede estimar el desplazamiento del vehículo robot. Esto da como resultado mediciones de posición relativa, o también a veces denominadas mediciones propioceptivas.

Las medidas de posición relativa son medidas que se realizan mirando únicamente al robot en sí. No se utiliza información externa y, por lo tanto, estas mediciones solo pueden proporcionar información relativa al punto donde se iniciaron las mediciones.

El robot detecta el entorno por medio de sus sensores, estos sensores dan información de la situación momentánea, llamadas observaciones o mediciones. Esta información describe cosas sobre el entorno del robot en un momento determinado.

Las observaciones realizadas desde el entorno proporcionan información sobre la ubicación del robot que es independiente de cualquier estimación de ubicación anterior. Proporcionan mediciones de posición absolutas, a veces también llamadas mediciones exteroceptivas, para enfatizar que la información de estas mediciones proviene de mirar el entorno en lugar del robot sí mismo.

2.17. Problema de localización del robot

Para el problema de localización de un robot móvil, el estado del sistema corresponde a la posición del robot. La entrada depende de la configuración del robot móvil, en este caso se tiene como comandos de entrada a la velocidad lineal y a la velocidad angular del robot. El modelo de movimiento propuesto es el modelo de movimiento de velocidad. (Anota, 2020).

El problema de la localización de robots consiste en responder a la pregunta ¿Dónde estoy? desde el punto de vista de un robot. Esto

significa que el robot tiene que averiguar su referencia a las coordenadas X Y la dirección de avance de un robot en un sistema de coordenadas global. (Gustavo Velasco Hernandez, 2019)

2.18. Mediciones de posición relativa del robot.

La adquisición de mediciones relativas también se conoce como cálculo de cálculo muerto, que se ha utilizado durante mucho tiempo, ya que las personas comenzaron a viajar

Originalmente, este es el proceso de estimar la posición de un avión o un barco, basado únicamente en la velocidad y dirección de viaje y el tiempo que pasó desde la última posición conocida. Dado que las estimaciones de posición se basan en posiciones anteriores, el error en las estimaciones aumenta con el tiempo. En aplicaciones robóticas, las mediciones de posición relativa se adquieren mediante odometría o navegación inercial, más adelante abordaremos de una manera más detallada estos conceptos

CAPÍTULO 3.

MODELADO DEL ROBOT DIFERENCIAL DE NAVEGACION.

3.1. Construcción del robot diferencial.

La construcción de un robot diferencial de navegación autónoma implica la creación de un vehículo robótico capaz de moverse y navegar de forma autónoma en su entorno. El robot diferencial es un tipo común de robot móvil que utiliza una configuración de dos ruedas accionadas independientemente, lo que le permite girar y moverse en diversas direcciones.

A continuación, se presenta una breve explicación de los componentes clave en la construcción de un robot diferencial de navegación autónoma:

3.2. Componentes del robot diferencial

1. **Plataforma móvil:** La base del robot es una plataforma móvil que proporciona soporte estructural y espacio para alojar los componentes electrónicos y mecánicos. Puede ser una estructura de chasis rígida o una plataforma móvil modular.
2. **Ruedas:** El robot diferencial utiliza dos ruedas colocadas en los lados opuestos de la plataforma. Estas ruedas son accionadas por motores independientes, lo que permite controlar la velocidad y dirección del robot.
3. **Sensores:** Los sensores son elementos clave para la navegación autónoma. Los robots diferenciales de navegación autónoma suelen utilizar una combinación de sensores como:
 - **Sensores de posición:** Estos pueden incluir encoders en las ruedas para medir la distancia recorrida y calcular la posición y velocidad del robot.
 - **Sensores de distancia:** Pueden ser sensores de ultrasonido o infrarrojos para detectar obstáculos y mapear el entorno del robot.
 - **Sensores de orientación:** Pueden ser brújulas electrónicas (magnetómetros) o unidades de medición inercial (IMU) para medir la orientación del robot en relación con el campo magnético terrestre o utilizando acelerómetros y giroscopios.
 - **Sensores de Odometría:** Pueden utilizar técnicas como la medición de la velocidad de las ruedas y el deslizamiento para estimar la posición del robot.
4. **Unidad de control:** La unidad de control es el cerebro del robot. Puede ser una computadora embarcada, un microcontrolador o un sistema embebido que procesa los datos de los sensores, ejecuta algoritmos de control y toma decisiones para guiar

el robot en su navegación autónoma.

5. **Algoritmos de control y planificación de trayectorias:** Los algoritmos de control y planificación de trayectorias permiten al robot tomar decisiones sobre cómo moverse y evitar obstáculos en su entorno. Estos algoritmos pueden basarse en técnicas como control PID, algoritmos de seguimiento de trayectorias o algoritmos de planificación de caminos.
6. **Actuadores:** Además de las ruedas motorizadas, los actuadores pueden incluir sistemas de dirección (como servomotores) y otros dispositivos dependiendo de las funciones específicas del robot, como brazos robóticos para la manipulación de objetos.

3.3. Diseño del robot

Se desarrollo el diseño del robot diferencial en el software SolidWorks, con las características para colocar los mecanismos físicos necesarios para el diseño diferencial

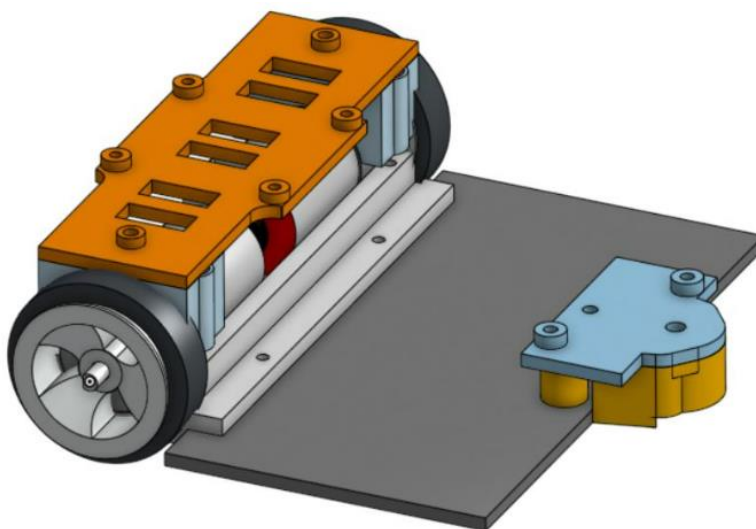


Figura: 9 Diseño en 3D del robot diferencial

Seleccionar el sistema embebido de control y la parte técnica de programación y escoger los componentes como el controlador, los actuadores y los elementos de instrumentación, el sistema de energía, los componentes son los siguientes:

1. Puente H
2. Tarjeta de control, ES32
3. Actuadores

4. Encoder
5. Rueda Loca
6. Batería

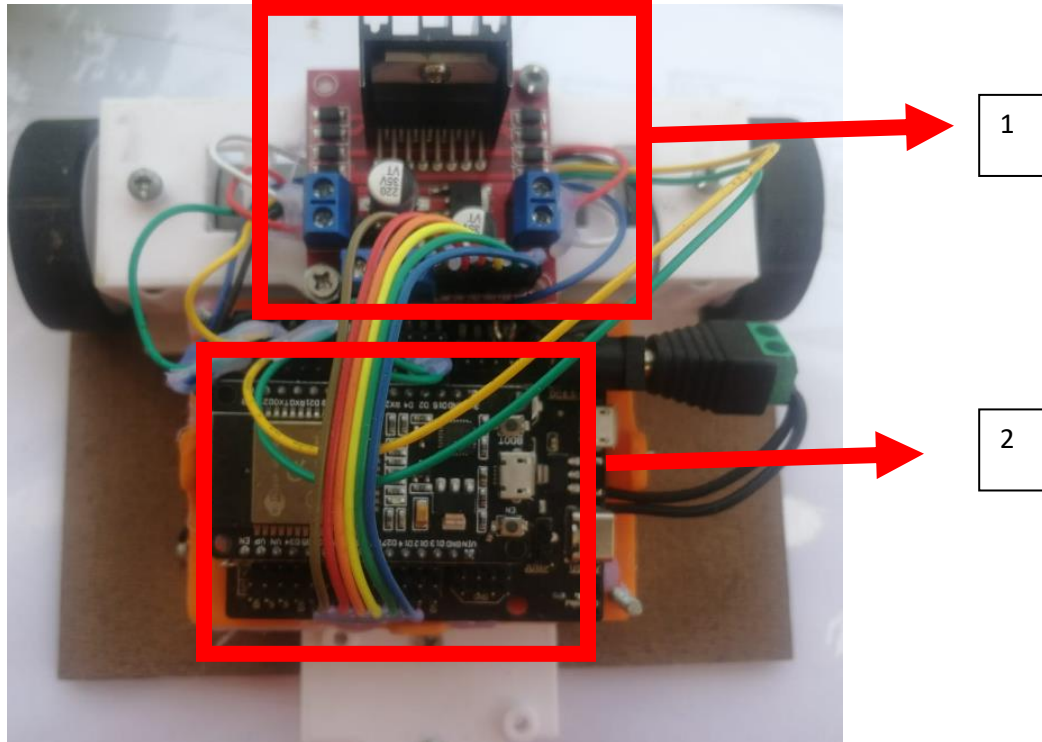


Figura: 10 Vista frontal del robot.

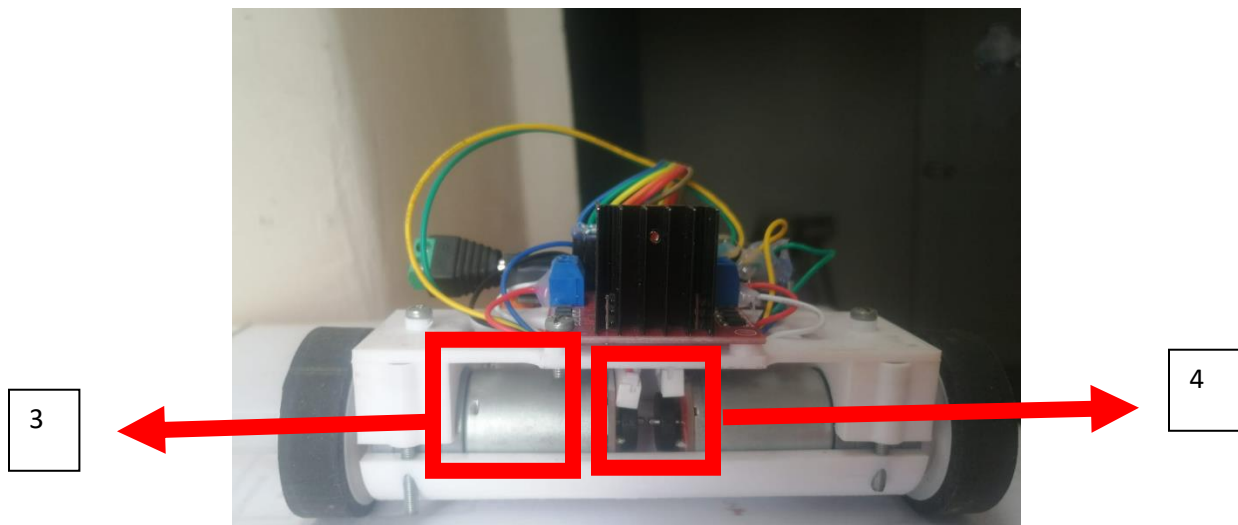


Figura: 11 Vista de atrás del robot diferencial

El robot con tracción diferencial es un vehículo que utiliza un sistema de transmisión de dos ruedas independientes, cada rueda está unida a su propio motor y tiene encoder implementados.

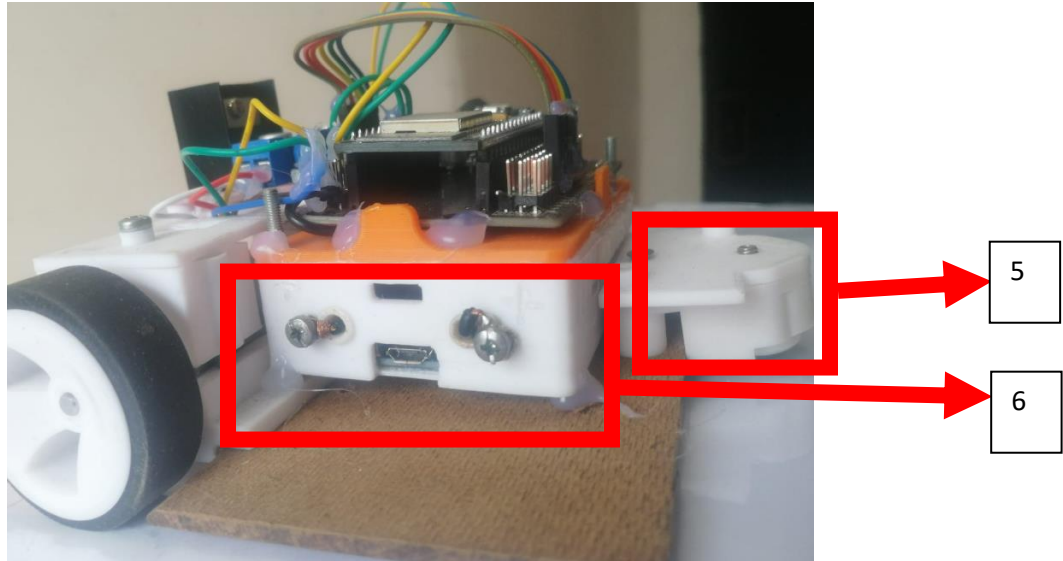


Figura: 12 Vista lateral del robot

3.4. Diagrama de conexión del robot diferencial de navegación.

Se presentan los componentes electrónicos que conforman el Robot Móvil Diferencial, compuesto por baterías, los encoders izquierdo y derecho, motores DC derecho e izquierdo, la adaptación del puente H, un controlador ESP32, y la armadura de plástico que soportan todas las componentes incluidas las ruedas

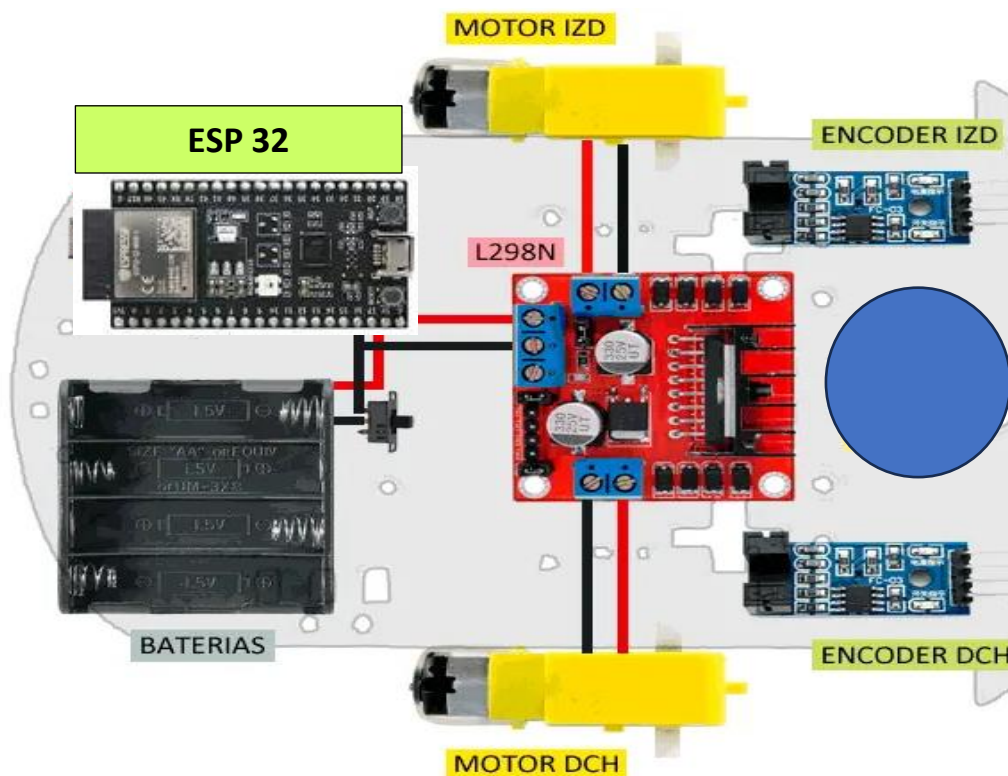


Figura: 13 Esquema electrónico

3.5. Componentes electrónicos, sensores y actuadores del robot diferencial de navegación autónoma.

- **Microcontrolador ESP32:** El microcontrolador es un dispositivo versátil y potente que combina un procesador de alto rendimiento y conectividad inalámbrica. Es ampliamente utilizado en proyectos de IoT y aplicaciones embebidas, ofreciendo una amplia compatibilidad y un bajo costo.
- **Motores DC:** Controla el movimiento en proyectos basados en Arduino. Pueden ser controlados en términos de velocidad y dirección y requieren un circuito de control adicional para conectarse a Arduino y aprovechar las capacidades de control del microcontrolador.
- **Encoder:** Involucran control de motores, robótica, sistemas de posicionamiento y más. Proporcionan una manera precisa de obtener información sobre el movimiento angular, lo que permite un control más preciso y una retroalimentación en tiempo real.

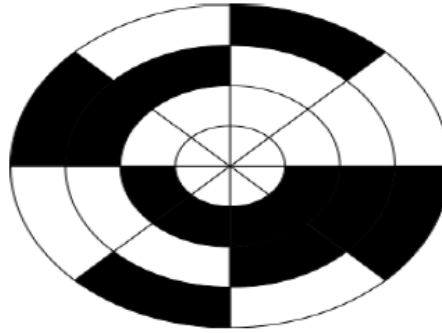


Figura: 14 Disco del encoder

¿Cómo procesar la información el encoder?

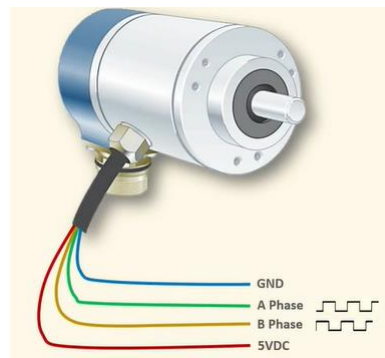


Figura: 15 Líneas de datos del Encoder

- Encoder: Tiene cuatro líneas una de tierra y una de 5v, 1 fase A y una de fase B ambas están desfasadas 90 grados. Estas fases nos ayudan a identificar el tipo de giro del encoder que puede ser horario antihorario envían ondas cuadradas de señal, estas son señales cuadradas tipos tacómetros de pulsos envían una señal cuadrada desfasadas 90°.
- Puente H: El uso de un puente H implican el control de motores DC, como robots móviles, brazos robóticos, vehículos autónomos, entre otros, ya que brinda un control preciso y flexible sobre el motor en términos de dirección y velocidad.

3.6. Modelado matemático del Robots Móviles de navegación

El modelo dinámico de un vehículo terrestre 2D de navegación autónoma es una representación matemática que describe el movimiento y comportamiento del vehículo en un entorno

bidimensional. Está compuesto por varias partes clave:

- **Estado del vehículo:** El estado del vehículo se define por su posición (x, y), orientación (θ), velocidad lineal (v) y velocidad angular (ω). Estas variables describen la posición y orientación del vehículo en el plano xy , así como su movimiento.
- **Dinámica del vehículo:** La dinámica del vehículo se rige por las leyes de la física, como las ecuaciones de Newton. Estas ecuaciones relacionan las fuerzas y torques que actúan sobre el vehículo con su aceleración lineal y angular. Las fuerzas pueden incluir la fuerza de tracción de las ruedas, la resistencia del aire y la fricción, mientras que los torques pueden provenir de la dirección de las ruedas.
- **Restricciones y características del vehículo:** El modelo también tiene en cuenta las características del vehículo, como su masa (m), momento de inercia (I) y las propiedades de los actuadores, como los motores o las ruedas. Estos parámetros influyen en la respuesta del vehículo ante las acciones de control y ayudan a predecir su comportamiento.

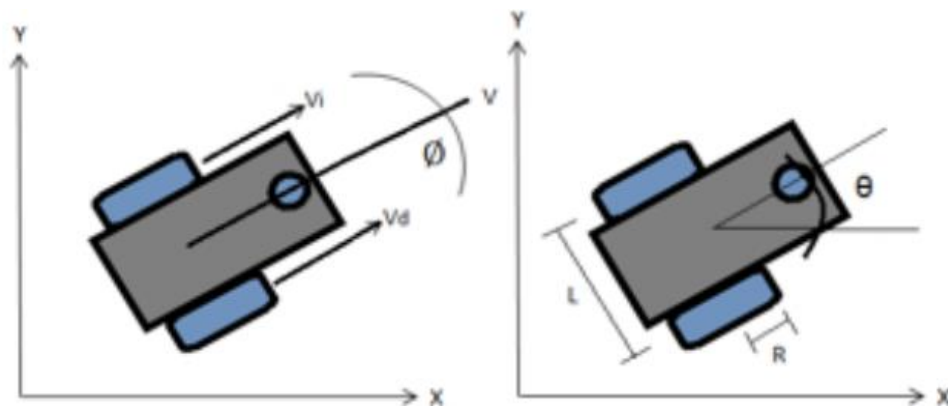


Figura: 16 Variables de la característica del vehículo

Control: El modelo dinámico se utiliza para desarrollar algoritmos de control que permiten al vehículo tomar decisiones y realizar ajustes en tiempo real para lograr una navegación autónoma. Estos algoritmos pueden utilizar técnicas como el control PID (Proporcional, Integral, Derivativo) o métodos de control más avanzados para generar las

fuerzas y torques necesarios para seguir una trayectoria deseada.

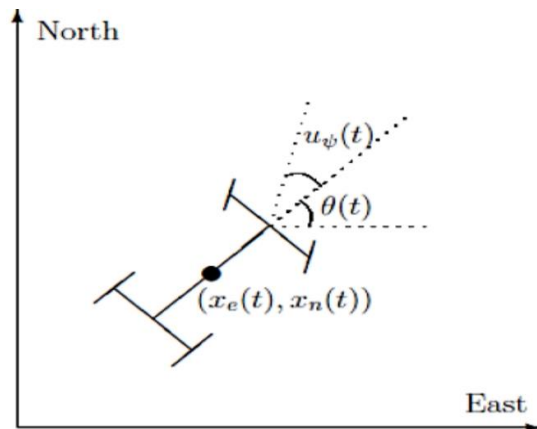


Figura: 17 Esquema del robot de navegación.

Leyes de Newton

$$\sum \vec{F}_K = m \vec{a} \rightarrow F_m - f_k = m a \quad \therefore \quad a = \frac{F_m - f_k}{m}$$

Donde

F_m → Es la fuerza aplicada por el motor eléctrico del vehículo

f_k → Es la Fuerza de fricción o de arrastre

A su vez estas fuerzas la podemos calcular de la siguiente manera:

$$F_m = \frac{P u(t)}{v} \quad \& \quad f_k = \frac{C_d}{L} A v^2$$

Donde

P → Potencia del motor eléctrico

$u(t)$ → Posición del acelerador este coeficiente varía entre 0 a 1

C_d → Coeficiente de arrastre o de fricción

A → Área frontal del vehículo más terrestre

L → Longitud entre las distancias entre los ejes de las ruedas

v → Velocidad de vehículo terrestre

Por lo que las ecuaciones de aceleración se pueden expresar de la siguiente manera: $a = \frac{\frac{P u(t)}{v} - \frac{C_d}{L} A v^2}{m}$, también podemos obtener las ecuaciones de las velocidades del vehículo terrestre a partir centro de masa del sistema

$$v_x = v \cos \theta$$

$$v_y = v \sin \theta$$

$$a = \frac{\frac{P u(t)}{v} - \frac{C_d}{L} A v^2}{m}$$

$$\omega = \dot{\theta} = \tan^{-1} \left(\frac{v_y}{v_x} \right)$$

Estas ecuaciones representan la dinámica del robot navegación la cual podemos colocar de la siguiente manera

$$\begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \\ \frac{dv}{dt} \\ \frac{d\theta}{dt} \end{bmatrix} = \begin{bmatrix} v \cos \theta \\ v \sin \theta \\ \frac{P u(t)}{mv} - \frac{C_d A v^2}{mL} \\ \tan^{-1} \left(\frac{v_y}{v_x} \right) \end{bmatrix} \rightarrow \text{Sistema no lineal}$$

Donde el vector de estados es $\vec{X} = \begin{bmatrix} x \\ y \\ v \\ \theta \end{bmatrix}$, las cuales son las posiciones en $\{x, y\}$, la velocidad lineal v y la posición angular θ

Observando las ecuaciones dinámicas de robot podemos apreciar que el sistema es no lineal lo cual hace que no podamos utilizar el modelo en espacios de estados, pero este problema lo podemos solucionar implementando un observable o también llamado modelo lineal de estimación, para resolver este problema e implementar el filtro de Kalman vamos a utilizar las ecuaciones de cinemática del movimiento acelerado uniforme

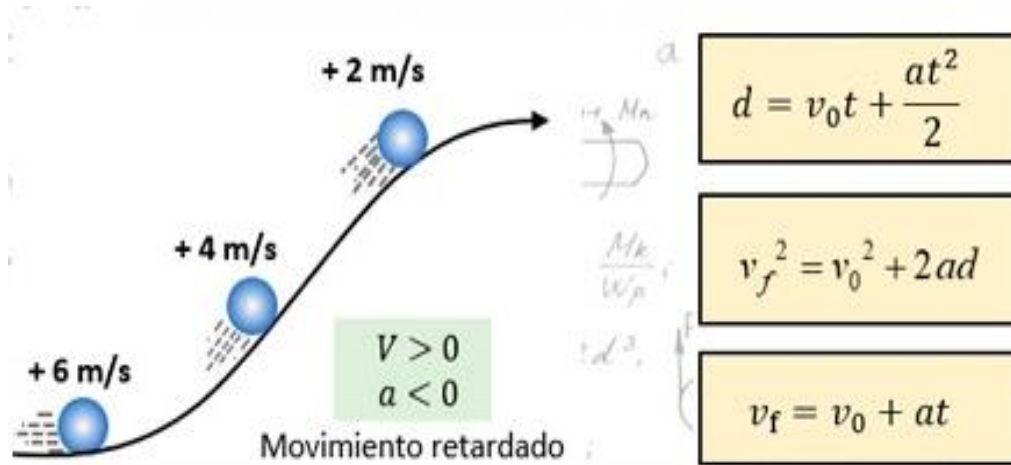


Figura: 18 Movimiento Acelerado Uniforme

3.7. Ecuaciones para el observable o estimador de Kalman

El modelo para el estimador de Kalman esta descrito por las siguientes ecuaciones las cuales se obtienen analizando las ecuaciones de movimiento acelerado uniforme (1), donde se discretizo el modelo para un tiempo de muestreo igual T_s y las variables de estados X_{k+1} , Y_{k+1} , $V_{X_{k+1}}$ y $V_{Y_{k+1}}$ se actualizarán en cada instante de tiempo T_s

$$X_{k+1} = X_k + v_x k T_s + \frac{1}{2} a_x T_s^2$$

$$Y_{k+1} = Y_k + V_y k T_s + \frac{1}{2} a_y T_s^2$$

$$V_{X_{k+1}} = V_x k + a_x T_s$$

$$V_{Y_{k+1}} = V_y k + a_y T_s \quad (1)$$

Forma matricial del estimador de Kalman (2)

$$\begin{bmatrix} X_{k+1} \\ Y_{k+1} \\ V_{X_{k+1}} \\ V_{Y_{k+1}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_k \\ Y_k \\ V_{xk} \\ V_{yk} \end{bmatrix} + \begin{bmatrix} \frac{T_s^2}{2} & 0 \\ 0 & \frac{T_s^2}{2} \\ T_s & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} a_x \\ a_y \end{bmatrix}$$

$Z_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \rightarrow$ Matriz de medicion de estados la cual también es igual la matriz H

$$P_k = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix} \rightarrow \text{La matriz de covarianza del error inicial } P \text{ se}$$

coloca de manera aleatoria y con un valor muy cercano a cero, para asignar el valor de Q, que representa la covarianza de ruido en el proceso, se hace una búsqueda heurística y se selecciona un valor muy cercano a cero.

$$Q = 0,00001$$

El ruido en las medidas está representado por la matriz R. La desviación está indicada por el valor de la precisión. La covarianza es igual al cuadrado de la desviación estándar. La matriz es diagonal.

$$R = \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 \\ 0 & 0 & \sigma_{v_x}^2 & 0 \\ 0 & 0 & 0 & \sigma_{v_y}^2 \end{bmatrix} \rightarrow \text{Donde } \sigma^2 \text{ representa la exactitud de la}$$

medición en ese instante de tiempo, lo que significa que por cada iteración el valor de R cambiaría

Entonces resumiendo tenemos todas matrices y variables para inicializar nuestro filtro de Kalman para estimar las trayectorias de robot diferencial de navegación las cuales son:

$$A_{k-1} = \begin{bmatrix} 1 & 0 & Ts & 0 \\ 0 & 1 & 0 & Ts \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \rightarrow \text{Es la matriz de transición de estado que describe}$$

cómo evoluciona el estado del sistema desde el instante de tiempo k-1 al instante de tiempo

$$B_{k-1} = \begin{bmatrix} \frac{Ts^2}{2} & 0 \\ 0 & \frac{Ts^2}{2} \\ Ts & 0 \\ 0 & 0 \end{bmatrix} \rightarrow \text{Es la matriz de control que representa la influencia de la}$$

entrada de control

$$Z_k = HK_{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \rightarrow \text{Es la matriz de observación que relaciona el estado}$$

del sistema con las mediciones disponibles en el instante de tiempo k-1.

$$P_k = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix} \rightarrow \text{Es la matriz de covarianza del estado en el}$$

instante de tiempo k

$Q = 0,00001 \rightarrow$ Es la matriz de covarianza del ruido de proceso que modela la incertidumbre en el modelo de evolución del sistema

$$R_k = \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 \\ 0 & 0 & \sigma_{v_x}^2 & 0 \\ 0 & 0 & 0 & \sigma_{v_y}^2 \end{bmatrix} \rightarrow \text{Es la matriz de covarianza del ruido de}$$

medición que representa las incertidumbres en las mediciones disponibles en el instante de tiempo k.

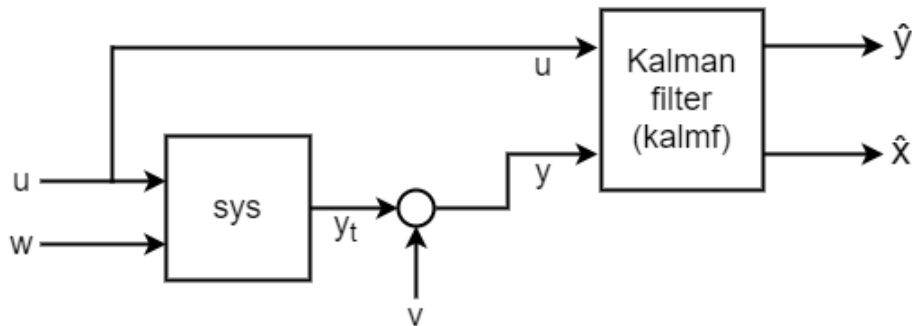


Figura: 19 Diagrama de bloques de conexión del modelo dinámico del robot de navegación y del estimador de Kalman

- Explicación más detallada de lo que hará el estimador de Kalman en nuestro proyecto
 - **Algoritmo:** Este algoritmo resuelve el problema de manera óptima tomando en cuenta el ruido (o incertidumbre) presente tanto en las variables medidas (valores entregados por los sensores) como en las variables de control (variables aplicadas por los actuadores)
 - **El filtro de Kalman está basado en las siguientes suposiciones:**

El sistema es lineal y se tiene una ecuación de estado que describe su comportamiento dinámico y una ecuación de medición que describe la relación entre las variables de estado y la salida de sensores. Ambas ecuaciones están descritas en tiempo discreto.

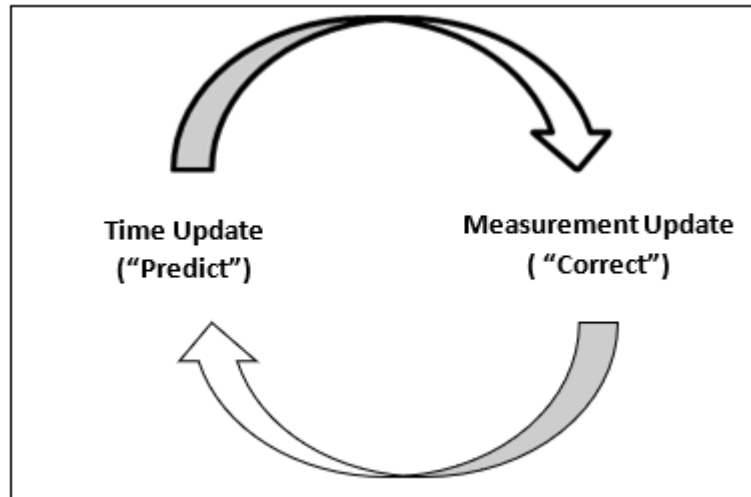


Figura: 20 Predicción y Corrección

Se conoce la estadística del ruido de medición y del ruido del sistema. Ambos son independientes entre sí, con media igual a cero y distribución de probabilidad normal (Gaussiana: La distribución tiene la propiedad unimodal un solo máximo y ese máximo concentrada la media).

3.8. Etapas del algoritmo del filtro de Kalman

1. **Estado del sistema:** El estado del sistema es una descripción matemática de las variables que caracterizan completamente el sistema en un momento dado. Puede ser una combinación de variables físicas, como posición, velocidad y aceleración, o variables abstractas que representan características internas del sistema.
2. **Modelo de evolución del sistema:** El modelo de evolución del sistema describe cómo cambia el estado del sistema en el tiempo. Se basa en ecuaciones que representan las relaciones dinámicas entre las variables de estado. En el caso más simple, se puede utilizar un modelo lineal, como un sistema de ecuaciones de primer orden. Sin embargo, también es posible utilizar modelos no lineales para sistemas más complejos.
3. **Mediciones:** Las mediciones son observaciones del sistema que se obtienen a través de sensores o instrumentos de medición. Estas mediciones pueden contener ruido y errores debido a limitaciones del sensor o perturbaciones externas.
4. **Matriz de covarianza:** La matriz de covarianza es una representación de la

incertidumbre asociada con la estimación del estado del sistema. Proporciona información sobre la dispersión de las estimaciones y cómo están correlacionadas entre sí. Es una herramienta clave para el filtro de Kalman, ya que permite tener en cuenta la incertidumbre en las estimaciones y las mediciones.

5. **Predicción:** La etapa de predicción del filtro de Kalman utiliza el modelo de evolución del sistema y la estimación previa del estado para predecir el estado futuro del sistema. Utiliza las ecuaciones de evolución del sistema y actualiza la estimación del estado junto con la matriz de covarianza asociada.
6. **Actualización:** La etapa de actualización del filtro de Kalman utiliza las mediciones disponibles para corregir la estimación previa del estado. Se calcula una ganancia de Kalman que pondera la confiabilidad de las mediciones y las estimaciones previas. Luego, se actualiza la estimación del estado y la matriz de covarianza utilizando la ganancia de Kalman y la diferencia entre las mediciones reales y las predicciones.
7. **Ganancia de Kalman:** La ganancia de Kalman determina cómo se ponderan las mediciones y las estimaciones previas en la actualización del filtro. Está diseñada para minimizar el error de estimación y tiene en cuenta la incertidumbre de las mediciones y las estimaciones previas.
8. **Proceso de iteración:** El filtro de Kalman es un proceso iterativo, donde la predicción y la actualización se repiten continuamente a medida que se obtienen nuevas mediciones. Esto permite una mejora gradual y continua en la estimación del estado del sistema a medida que se incorpora nueva información.

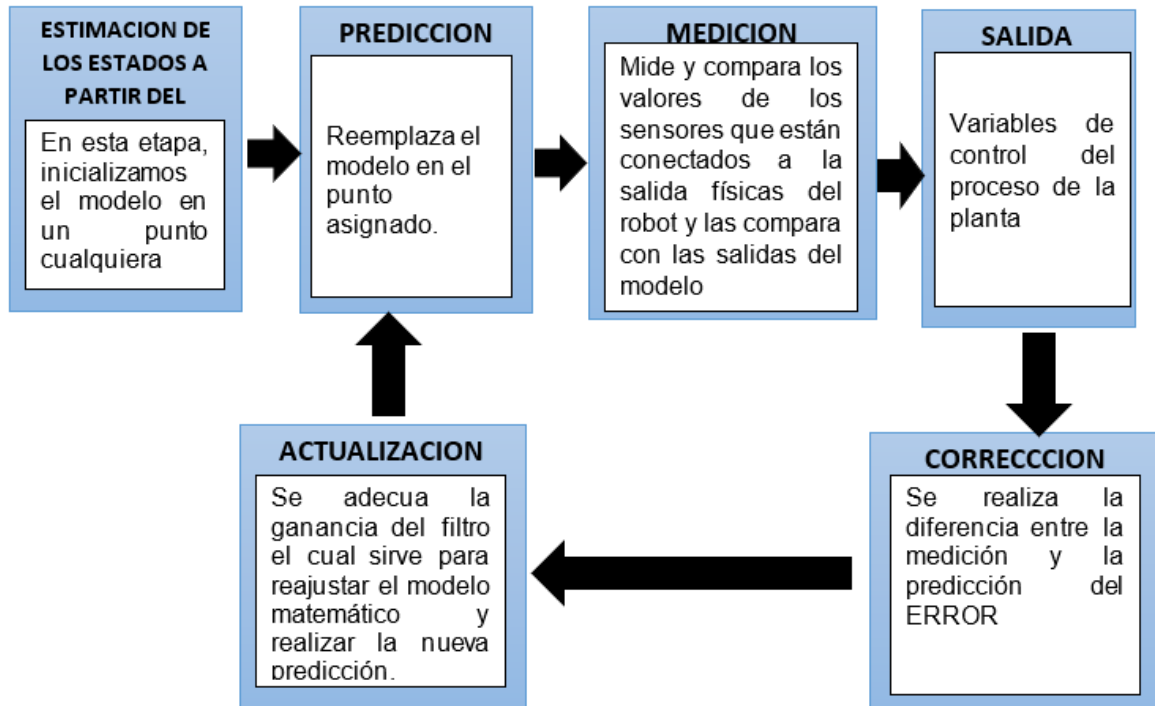


Figura: 21 Secuencia de ejecución de algoritmo.

CAPÍTULO 4

SIMULACION Y COMPARACION DEL FILTRO

4.1. Diseño y Simulación del Filtro de Kalman

La modelación y simulación de un filtro de Kalman en MATLAB Simulink implica el desarrollo de un sistema que utiliza este algoritmo para estimar el estado de un sistema dinámico a partir de mediciones ruidosas. El filtro de Kalman es una técnica de estimación óptima ampliamente utilizada en aplicaciones de control y procesamiento de señales.

4.2. Simulación del filtro de Kalman en MATLAB Simulink

La Simulación del filtro de Kalman en MATLAB Simulink generalmente sigue los siguientes pasos:

1. **Definición del sistema dinámico:** Se define el sistema dinámico que se desea estimar. Esto incluye la especificación de las ecuaciones de estado y las ecuaciones de observación, que describen cómo evoluciona el estado del sistema y cómo se relaciona con las mediciones.
2. **Diseño del filtro de Kalman:** Se implementa el algoritmo del filtro de Kalman en MATLAB Simulink. Esto implica la configuración de bloques y conexiones en el entorno de Simulink que representan las etapas del filtro de Kalman, como la predicción del estado, la actualización basada en las mediciones y la estimación del estado.
3. **Generación de señales de entrada:** Se generan señales de entrada para el sistema dinámico y se agregan niveles de ruido simulado a estas señales. Estas señales de entrada serán utilizadas para estimar el estado del sistema mediante el filtro de Kalman.
4. **Simulación del sistema:** Se lleva a cabo la simulación del sistema completo en MATLAB Simulink. Durante la simulación, el filtro de Kalman estima el estado del sistema en función de las mediciones ruidosas y se generan las salidas correspondientes.
5. **Análisis de resultados:** Se analizan los resultados de la simulación para evaluar la efectividad del filtro de Kalman en la estimación del estado del sistema. Esto

puede incluir la comparación entre el estado verdadero y el estado estimado, la evaluación del error de estimación y otros análisis relevantes.

En resumen, la modelación y simulación de un filtro de Kalman en MATLAB Simulink permite estimar el estado de un sistema dinámico a partir de mediciones ruidosas utilizando este algoritmo. MATLAB Simulink proporciona una plataforma conveniente y visualmente intuitiva para implementar y simular el filtro de Kalman, lo que facilita el análisis y la evaluación de su desempeño en diferentes escenarios.

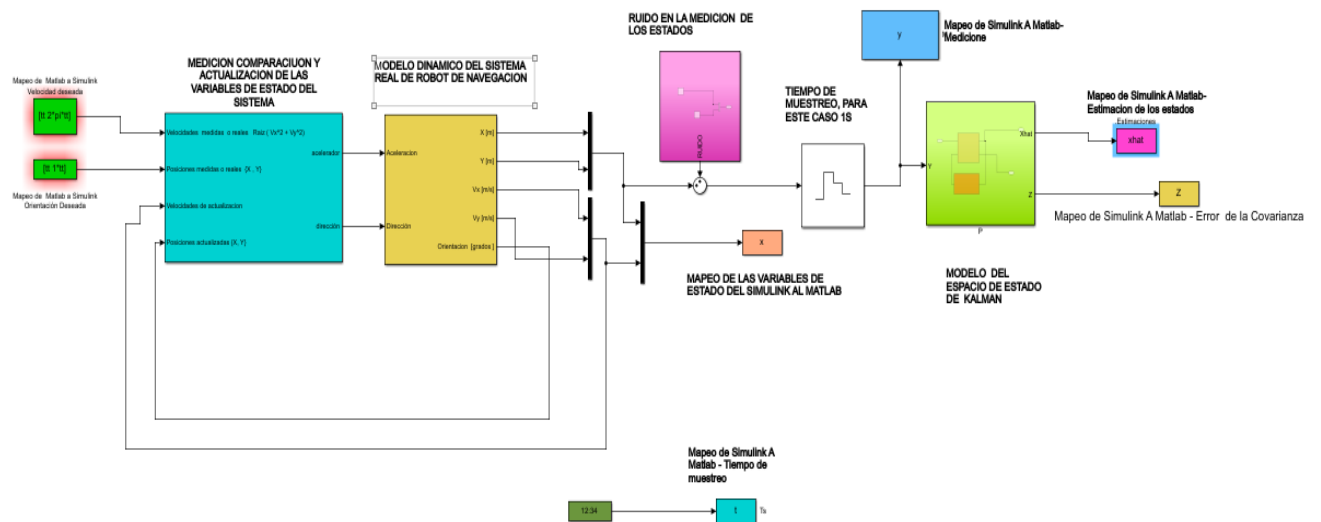


Figura 22: Diagrama de bloque general del Simulink del filtro de Kalman

4.3. Descripción de cada bloque del código

En esta sesión de código lo que realizo fue la medición de las variables de estado tanto en las posiciones $\{X, Y\}$ del sistema y la velocidad del robot, y comparamos con la estimación dada por el modelo de estados de nuestro estimador de Kalman, donde se busca que en cada actualización el error de la comparación entre estas dos respuestas tienda a Cero.

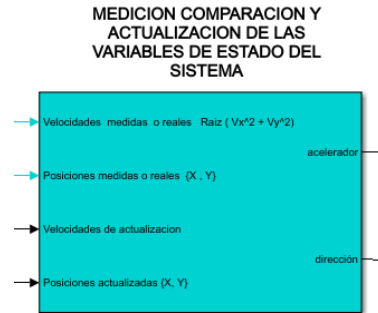


Figura: 23 Bloques de variables de estado del sistema

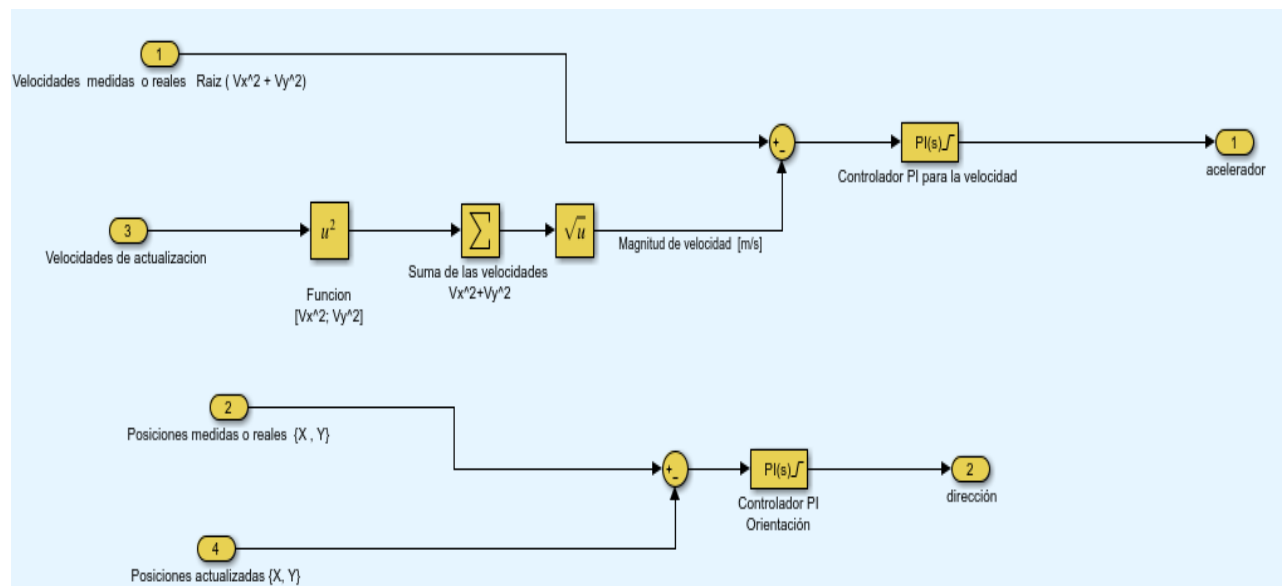


Figura: 24: Actualización y medición de las variables de estado del sistema

En esta sesión de código lo que se realizó fue la implementación de las cuatro ecuaciones que describen el modelo real del robot de navegación autónoma, las cuales son las siguientes.

1. $v_x = v \cos \theta$
2. $v_y = v \sin \theta$
3. $a = \frac{\frac{P u(t)}{v} - \frac{C_d}{L} A v^2}{m}$
4. $\omega = \tan (u_p(t))$

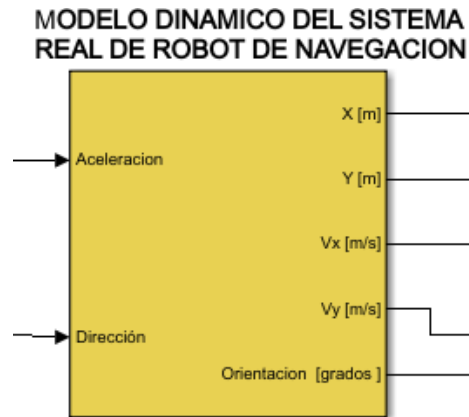


Figura: 25 Bloque del modelo dinámico del robot

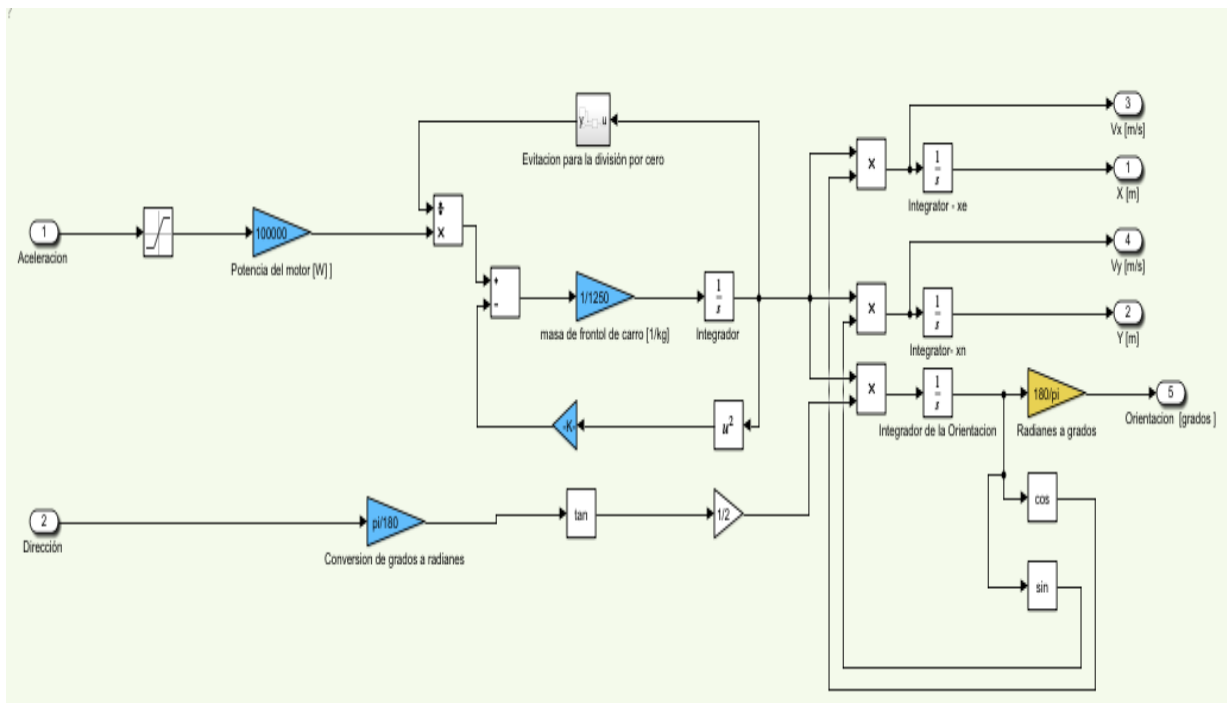


Figura: 26 Ecuación dinámica del robot de navegación autónoma

En esta parte del código lo que representamos es el modelo en espacios de estados del estimador de Kalman considerando el tiempo de muestreo $T_s = 1$ segundo

$$A_{k-1} = \begin{bmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, B_{k-1} = \begin{bmatrix} \frac{T_s^2}{2} & 0 \\ 0 & \frac{T_s^2}{2} \\ T_s & 0 \\ 0 & 0 \end{bmatrix}$$

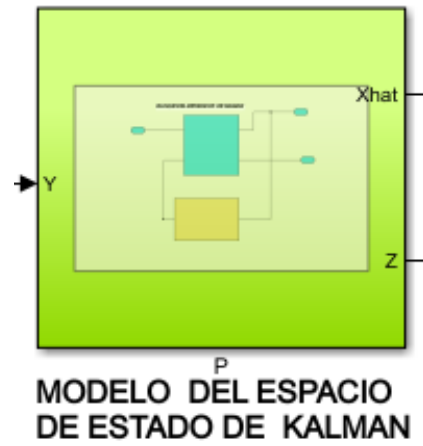


Figura: 27 Bloque del modelo del espacio de estado KALMAN

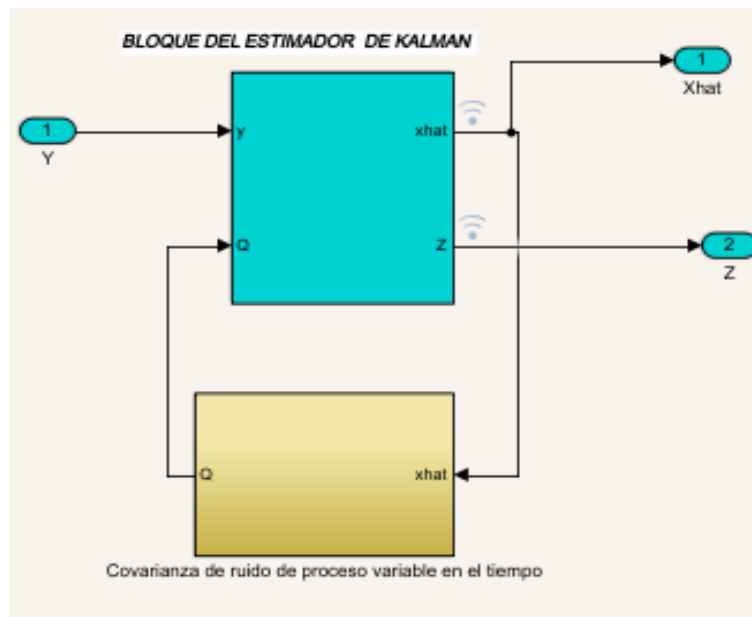


Figura: 28 Modelo de estado del estimador de Kalman

Modelo del estimador de Kalman , covarianza de ruidos de acuerdo a las variables Y y Q.

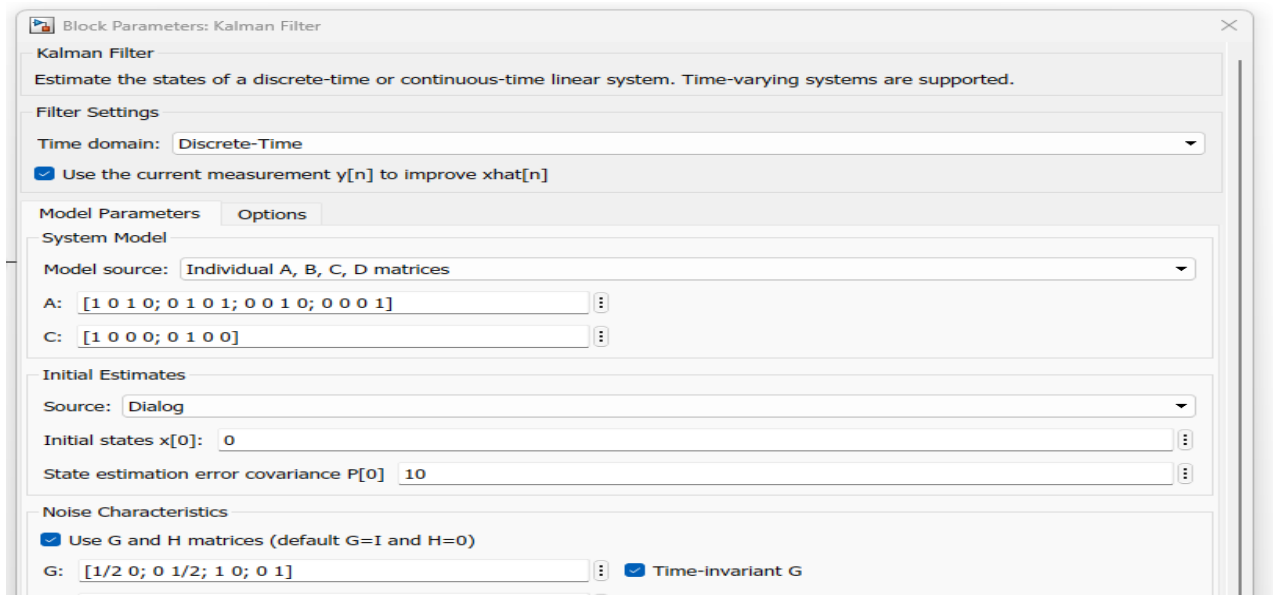


Figura: 29 Bloque de programación de filtro de Kalman

En este bloque lo que se realizó fue colocar las matrices de covarianza y desviación estándar

$$P_k = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix}, R_k = \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 \\ 0 & 0 & \sigma_{Vx}^2 & 0 \\ 0 & 0 & 0 & \sigma_{Vy}^2 \end{bmatrix}$$

Bloque de covarianza de ruido de proceso variable en el tiempo .

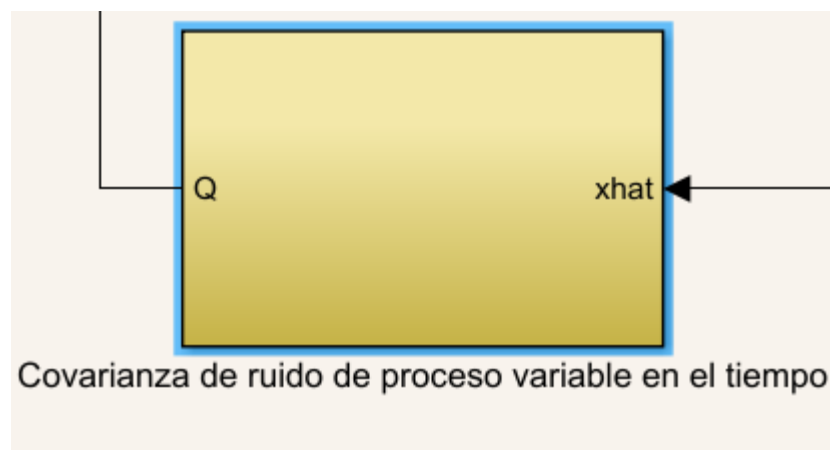


Figura: 30 Bloque de covarianza de ruidos del proceso en el tiempo

Ecuaciones de matriz de covarianza de ruidos de las mediciones y de los estados

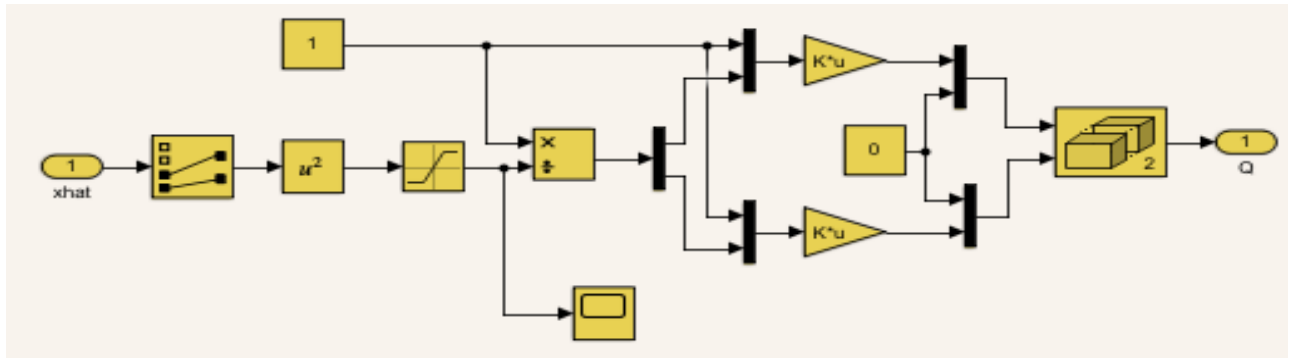


Figura: 31 Matriz de covarianza de ruidos de las mediciones y de los estados

CAPÍTULO 5.

ANÁLISIS DE RESULTADOS.

5. Simulación y resultados obtenidos

Los resultados del experimento con un robot de navegación autónoma, se logró realizar una trayectoria circular utilizando un filtro de Kalman para estimar la posición y la trayectoria del robot. Sin embargo, durante las pruebas, se encontró que los sensores del robot presentaban errores en las mediciones debido a varios factores, como el ruido mecánico y las interferencias eléctricas, lo cual introdujo cierta incertidumbre en el conocimiento de la trayectoria real del robot, que era una circunferencia o un círculo.

Al ejecutarse la trayectoria circular, se observó que la fuerza de fricción del suelo generaba ruido mecánico que afectaba las mediciones de los sensores. Además, las vibraciones mecánicas resultantes del movimiento de las partes y componentes del robot causaban interferencias eléctricas adicionales, aumentando aún más el nivel de error en las mediciones.

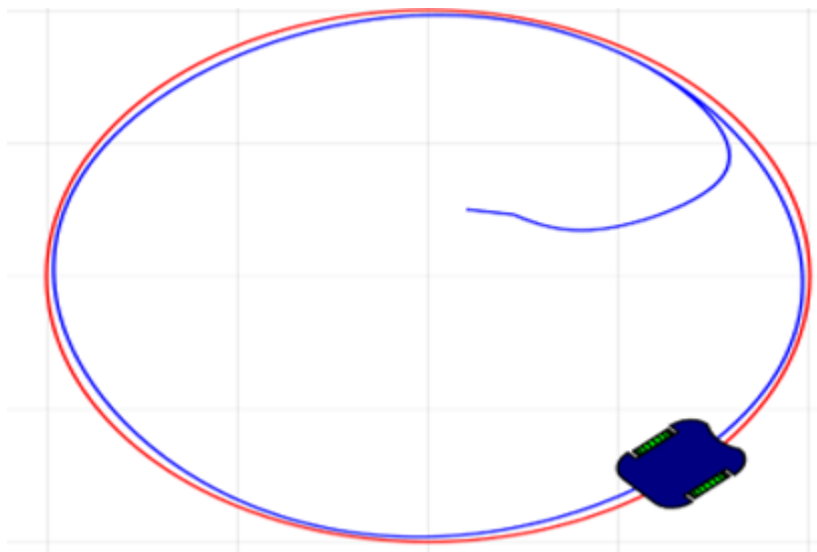


Figura: 32 Trayectoria real del robot diferencial de navegación autónoma.

5.1. Navegación autónoma

Durante el experimento con el robot de navegación autónoma, se encontró con desafíos significativos al tratar con los errores en las mediciones provenientes de los sensores. Estos errores afectaron directamente la capacidad de predecir con precisión las posiciones en X y Y del robot, así como sus velocidades lineales y angulares. Por lo tanto, predecir la trayectoria real del robot resultó ser una tarea difícil.

Uno de los principales problemas fue el ruido mecánico presente en las mediciones de los sensores. La fuerza de fricción del suelo generaba vibraciones y fluctuaciones en las mediciones, lo que introducía errores significativos en la estimación de la posición y la velocidad del robot. Estos errores se sumaban con cada nueva medición, lo que dificultaba la tarea de predecir la trayectoria

precisa del robot.

Además, las interferencias eléctricas también jugaron un papel importante en la incidencia de errores de las mediciones. Las vibraciones mecánicas causadas por el movimiento de las partes y componentes del robot generaban ruido en las señales eléctricas captadas por los sensores, lo que resultaba en mediciones imprecisas. Estas interferencias agravaban aún más la incertidumbre asociada con la estimación de la trayectoria real del robot.

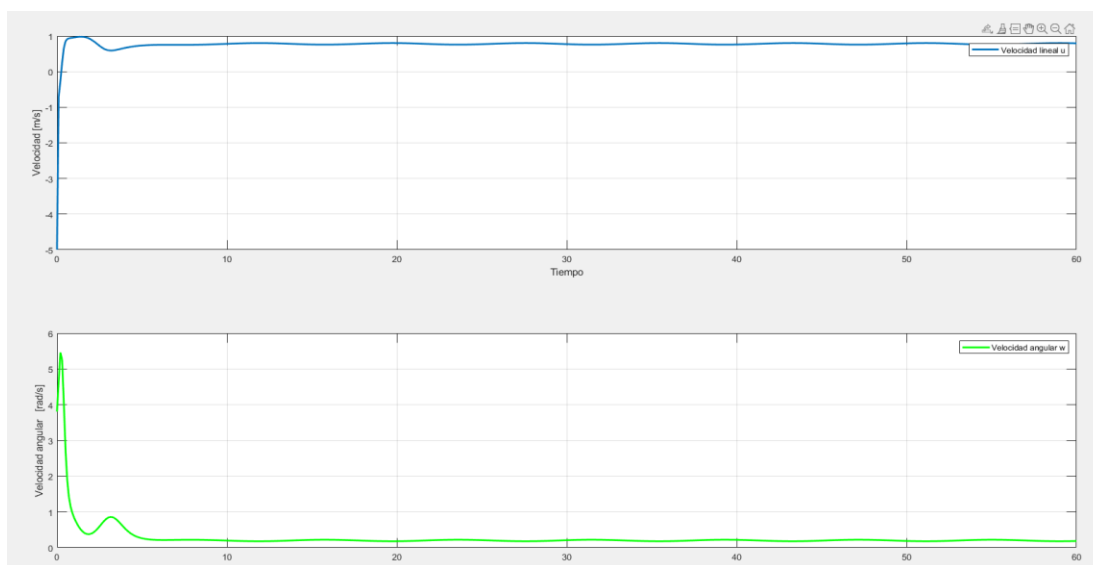


Figura: 33 Curvas de las velocidades lineal y angular del robot diferencial de navegación autónoma

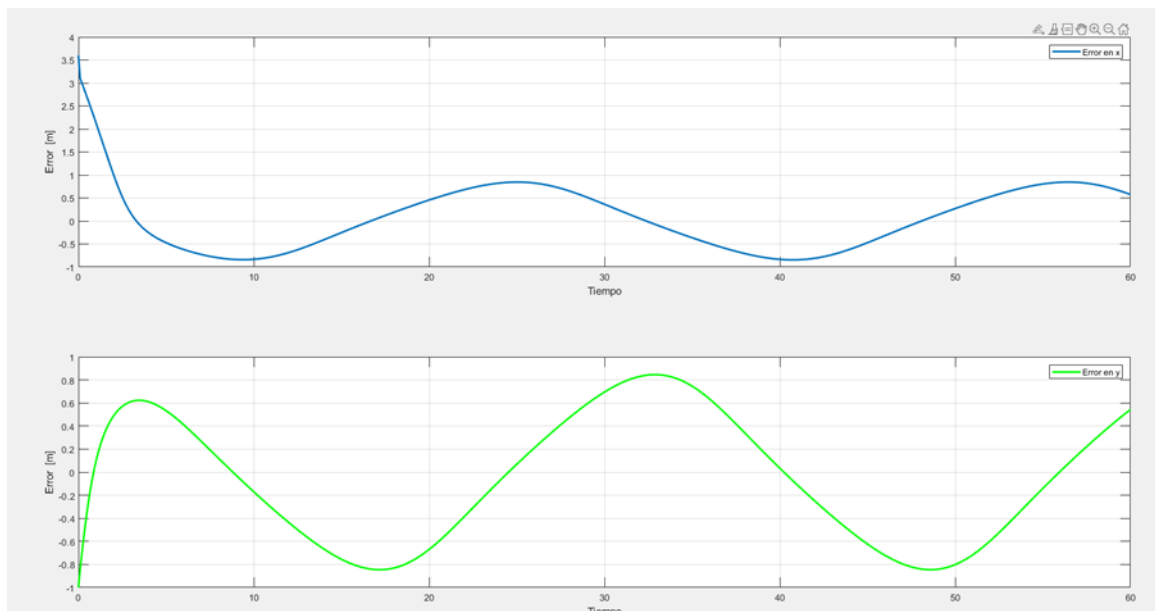


Figura: 34 Curvas de error en la medición de las señales del vector posición $\{x, y\}$ en cada instante de tiempo

Sin embargo, para abordar estos desafíos y mejorar la precisión en la estimación de la trayectoria del robot, se implementó un filtro de Kalman. Este filtro permitió combinar las mediciones ruidosas de los sensores con las estimaciones previas y los modelos del sistema para obtener una estimación más precisa del estado del robot.

5.2. Resultados de las estimaciones predichas por el filtro de Kalman

Inicialmente, al estimar la trayectoria del robot basándose en pocas mediciones de los sensores, el filtro de Kalman presentó una estimación inicial con cierto nivel de error. Sin embargo, a medida que se incorporaron más señales de los sensores y el filtro recibió más información, se observó una mejora significativa en la estimación del error de medición y en la trayectoria real del robot.

En particular, cuando el robot siguió una trayectoria circular, el filtro de Kalman pudo reducir gradualmente el error de medición inicial y proporcionar estimaciones más precisas y confiables de la posición y la trayectoria reales del robot. A medida que se recibían más mediciones de los sensores, el filtro pudo ajustar y refinar continuamente la estimación del estado del robot, teniendo en cuenta tanto la información actual como las estimaciones previas.

El resultado final fue una estimación cada vez más cercana a la trayectoria real circular del robot. El filtro de Kalman demostró su capacidad para compensar los errores de medición y reducir la incertidumbre, brindando una estimación más precisa y confiable de la trayectoria del robot en

tiempo real.

El filtro de Kalman se mostró efectivo al estimar la trayectoria de un robot de navegación autónoma en forma de una circunferencia. A medida que recibía más señales de los sensores, el filtro mejoró la estimación del error de medición y proporcionó una trayectoria estimada más cercana a la trayectoria real del robot, lo que permitió un seguimiento más preciso y confiable en tiempo real.

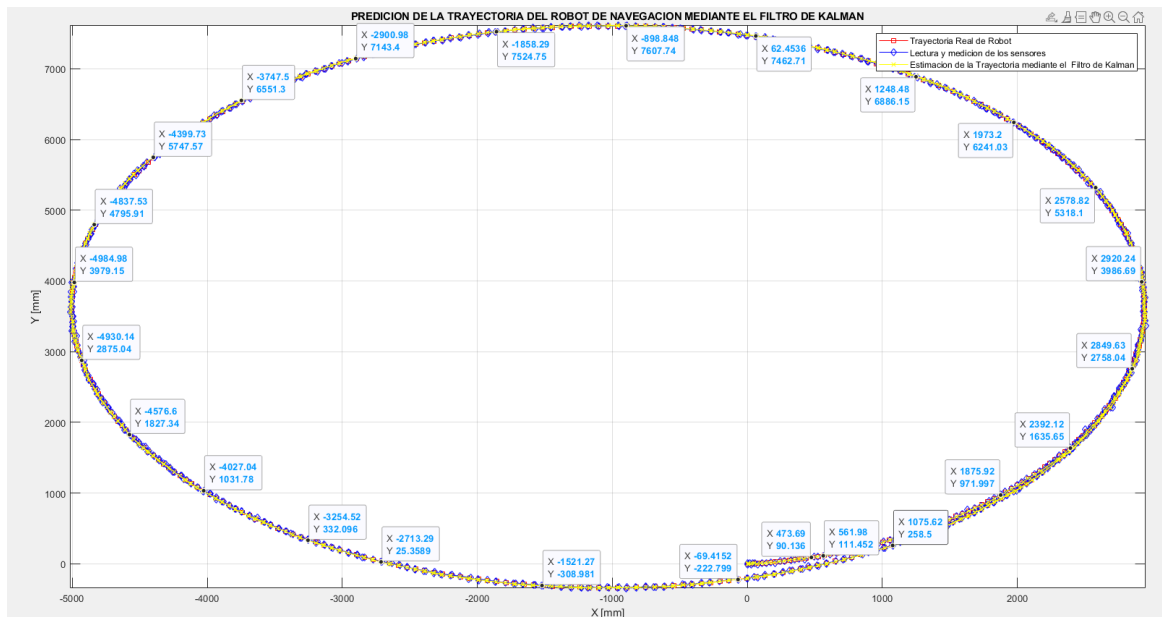


Figura: 35 Estimación de la trayectoria del robot mediante el filtro de Kalman

El filtro de Kalman actuó como un filtro de señal inteligente que tuvo en cuenta tanto las mediciones actuales como las estimaciones previas del estado del robot. A medida que recibía nuevas mediciones, el filtro combinaba esta información con los modelos del sistema y las mediciones anteriores para obtener una estimación más precisa del estado actual del robot.

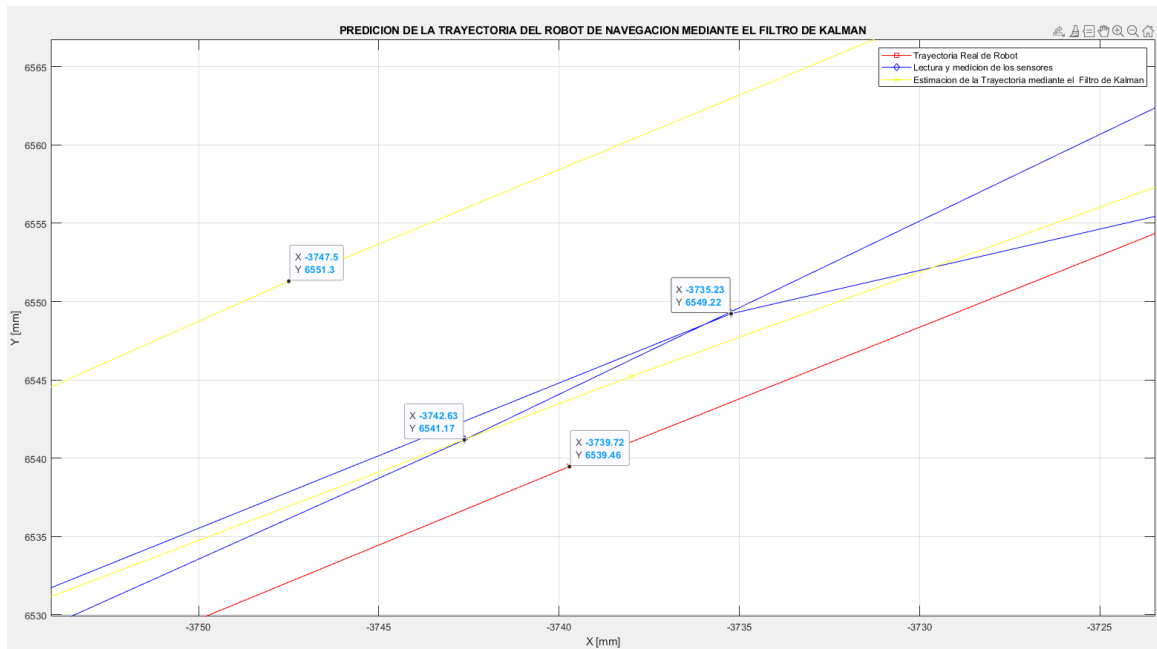


Figura: 36 Trayectoria del robot

En cuanto a las vibraciones de las partes móviles del robot, estas generaban interferencias eléctricas que también afectaban las mediciones de los sensores. El filtro de Kalman, al aprovechar su capacidad de adaptarse a las condiciones cambiantes, pudo mitigar gradualmente los efectos de estas interferencias y mejorar la calidad de las mediciones.

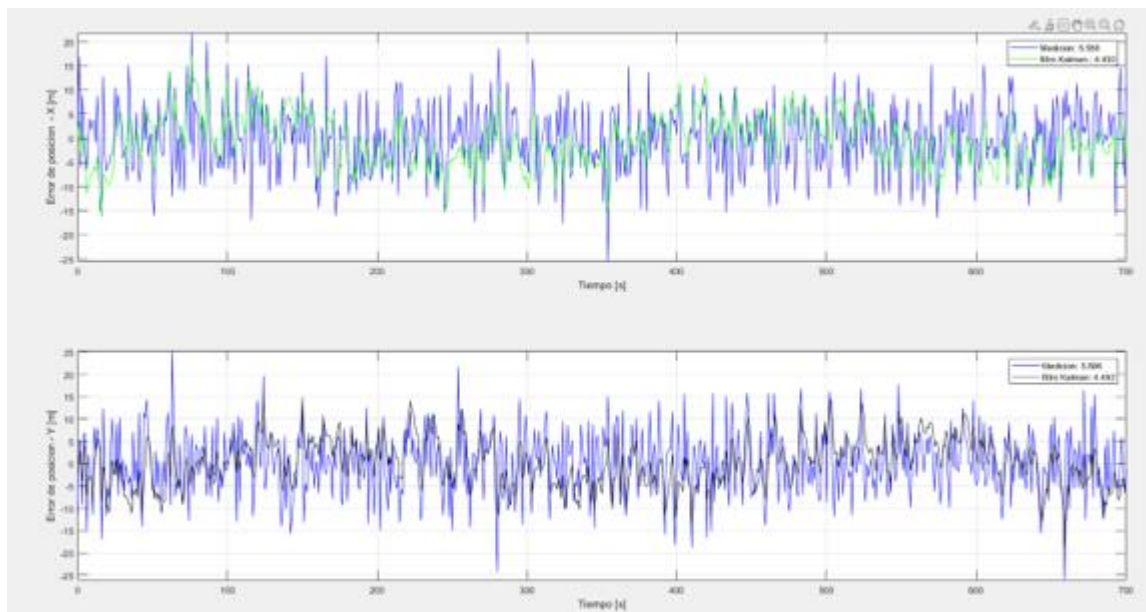


Figura: 37 Error en la medición señales de las posiciones en $\{X, Y\}$

A medida que avanzaba el experimento, el filtro de Kalman filtraba eficazmente todas las señales de ruido no deseadas. La estimación del estado del robot se volvió más precisa y estable a medida que el filtro aprendía a reconocer y eliminar los errores sistemáticos causados por el ruido mecánico y las interferencias eléctricas.

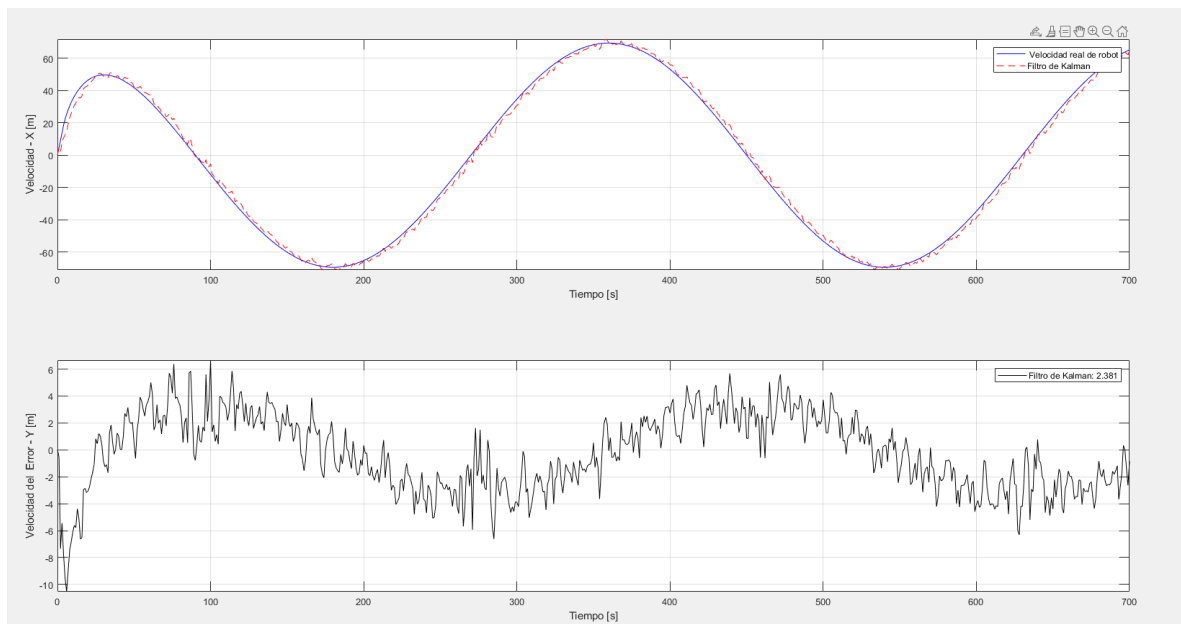


Figura: 38 Error en la medición de los sensores de velocidad del robot. Velocidad real del robot (Superior), Título (Inferior)

CONCLUSIONES

1. Como resultado el filtro de Kalman demostró ser óptimo al estimar la trayectoria real de nuestro robot móvil diferencial, Si se presentaron los errores en las mediciones de los sensores, producto del ruido mecánico y vibración de las partes móviles, el algoritmo corrigió estos errores y se pudo predecir con precisión la trayectoria del robot.
2. El modelo del algoritmo del filtro de Kalman desarrolló una solución al problema de filtrado por la presencia de ruidos, esta permitió actualizar los estados a partir de los datos anteriores logrando resultados con mayor confiabilidad. Con cada nueva medición realizada, el filtro de Kalman traza una nueva trayectoria que se estima con precisión a la trayectoria real del robot.
3. El modelo cinemático se utilizó para obtener la posición de (x, y) del robot, dada la velocidad angular de cada motor DC. la velocidad lineal y angular propuestas en el robot diferencial, todas estas variables se utilizaron en el Filtro de Kalman.
4. Se realizó un prototipo de robot móvil para validar los resultados obtenidos del algoritmo y simulación a través de Matlab-Simulink realizando varias pruebas con los encoders para conseguir la medida de velocidad angular incorporados en el robot durante movimiento del robot.
5. El desarrollo de este algoritmo permitió abordar futuras aplicaciones de navegación autónoma y mediante el uso de la revisión bibliográfica realizada y el diseño del robot se facilitará aplicar conocimientos de control y comparativas con otros tipos de filtro.
6. El modelo de control utilizado ayudó a resolver la trayectoria del recorrido que se ha trazado en el robot diferencial, se corrige la trayectoria y se comprueba a través de las gráficas obtenidas en el capítulo IV.

RECOMENDACIONES Y TRABAJOS FUTUROS

1. Se recomienda utilizar otro tipo de algoritmo para el filtrado de datos de entrada en el robot móvil, tal es el caso del filtro de partículas o un filtro a través de un algoritmo genético para obtener mayor precisión en la lectura de señales.
2. Se recomienda utilizar una amplia gama de sensores que se acoplen más a la precisión, que ayuden obtener información en la predicción de trayectorias, y permitan obtener una mejor resolución en la estimación de parámetros del robot.
3. Se recomienda hallar el modelo del motor DC en función de la potencia, la inercia y el torque, que son las características propias del motor para obtener mejores resultados con los parámetros en los espacios de estado.
4. Se recomienda utilizar un medidor de la velocidad real en el robot móvil para que al ser comparado con la velocidad que proporciona el algoritmo del filtro de Kalman, puedan ser ajustados a través de las velocidades angulares en función del tiempo.

BIBLIOGRAFÍA

- . Van Der Merwe, R. W. (2004). *Sigma-Point Kalman Filters for Integrated Navigation*. . Berlin: Proceedings of the AIAA Guidance, and Control Conference and Exhibit.
- Angeles, J. (2003). *FUNDAMENTALS OF ROBOTIC MECHANICALS SYSTEMS* (SEGUNDA ed.). (TleFE, Ed.) MONTREAL, CANADA: FREDERICK F.
- Anota, R. B. (2020). Localizacion de un robot movil mediante el filtro de Kalman Extendido y el simulados Gazebo. *Research in computing science* 149, 10.
- Arasaratnam, I. &. (2009). Cubature Kalman Filters. *IEEE Transactions on Automatic Control*, 4(6), 1254-1269.
- Armada, E. G. (2015). *Robots*. Madrid: CATARATA.
- Bar-Shalom, Y. L. (2001). *Estimation with Applications to Tracking and Navigation*. Boston: John Wiley & Sons.
- Bonilla, M. (2018). Navegacion Autonoma de un robot con tecnicas de localizacion y ruteo. *INAOE*, 8-9.
- Bravo, v. A. (2013). Metodologia para la estimacion de parametros en tiempo real mediante filtro de Kalman minimos cuadrados. *Iteckne volumen 10*, 37-44.
- Campoverde, J. (2018). Implementacion de un sistema de navegacion autonomo basado en slam y navegacion reactiva. *Universidad de las fuerzas Armadas*, 15-20.
- Chen., C.-T. (1999). *Linear System Theory and Design*. Oxford University Press.; 3rd edition.
- Cusidó, A. R. (2014). *Control Automatico de procesos Industriales*. España: Diaz de Santos.
- Díaz, J. (2013). *Guía de recomendaciones para diseño de software centrado en el usuario* . Buenos Aires , Argentina: edup.
- Dominguez, S., & Campoy, P. (2006). *Control en el Espacio de Estado*. España: Pearson Educacion.
- Dorf, R. C., & H.Bishop, R. (2022). *Modern Control Systems*. Reino Unido: Pearson Education.
- Fritsche, C. &. (2015). Applications of the Kalman Filter for Navigation and Mapping. *In Robotic Mapping and Exploration*, 209-234.
- G.C. Goodwin, S. G. (2000-). *Control System Design*. Prentice Hall: Jersey.
- GUERRERO, M. (2016). Evaluacion de Rendimiento de Estimadores No Lineales basados en la Aplicacion del Filtro de Kalman a Señales Biomedicas. *TUNJA:UNIVERSIDAD PEDAGOGICA Y TECNOLOGIA DE COLOMBIA*.
- Gustavo Velasco Hernandez, S. C. (2019). *Autonomous Navigation of mobile robots in factory environment*. Dublin - Irlanda: ScienceDirect.

- Gutierrez, S. (2014). Observador de velocidad y aceleracion para sistemas mecanicos a partir de lectura encoder. *Universidad Martha Abreu de las Villas*, 4-7.
- Haykin, S. (. (2001). *Kalman Filtering and Neural Networks*. London: John Wiley & Sons.
- Hernandez Gaviño, R. (2010). Introduccion a los sistemas de control. En R. Hernandez Gaviño, *Introduccion a los sistemas de control:Conceptos aplicaciones y simulacion con Matlab* (pág. 2). Naucalpan: Pearson Educacion de Mexico.
- Hilario López García, R. G. (1996). *Programación de robots industriales*. España: Universidad de oviedo.
- John C. Doyle, B. A. (1992). *Feedback control theory*. Texas - Estados Unidos: Macmillan.
- Julier, S. &. (2004). nscented Filtering and Nonlinear Estimation. . *Proceedings of the IEEE*,, 92(3), 401-422.
- Li, L. &. (2016). Kalman Filter-Based Indoor Pedestrian Navigation Using Smartphone Sensors. *Sensors. sing Smartphone Sensors. Sensors*, 16(2), 175.
- Li, M. &. (2019). Tightly Coupled Visual-Inertial Navigation System for Autonomous Aerial Robots. *IEEE Transactions on Robotic*, 35(3), 747-764.
- MARTÍ, A. (12 de Marzo de 2018). XATARA. Obtenido de [//www.xataka.com/espacio/opportunity-versus-curiosity-un-cara-a-cara-de- los-rovers-de-la-nasa-repasando-su-exploracion-de-marte](http://www.xataka.com/espacio/opportunity-versus-curiosity-un-cara-a-cara-de-los-rovers-de-la-nasa-repasando-su-exploracion-de-marte)
- Nakhmani, A. (2020). *Modern Control: State-Space Analysis and Design Methods*. New York: McGraw-Hill.
- Negenborn, R. (2003). *Robot Localization and Kalman Filters. On finding your position in a noisy world*. California - Estados Unidos : ResearchGate.
- Peña, R. S. (1992). *Introduccion a la teoria de control robusto*. Buenos Aires Argentina .
- Roos, M. O. (2014). Visual-Inertial Sensor Fusion: A Kalman Filter-Based Approach for UAV Navigation. *Journal of Intelligent & Robotic Systems*, , 7 4(1-2), 729-744.
- Rubin, B. F. (5 de Junio de 2019). *CNET*. Obtenido de <https://www.cnet.com/es/noticias/amazon-prime-air-dron-silencioso/>
- S.Grewal, M., & P.Andrews, A. (2008). *Kalman Filtering: Theory and Practice Using MATLAB*. New Jersey: Wiley.
- Santos, M. (2019). Localizacion y mapeo simultaneo robusto aplicado a la planeacion de trayectorias en robots moviles. *Centro de Investigacion de estudios avanzados del instituto politecnico nacional*, 1-4.
- SICMA21. (25 de 10 de 2021). *Robots industriales: tecnología y aplicaciones*. Barcelona, Barcelona, España.

- Simon, D. J. (2006). *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Chicago.
- Thrun, S. B. (2005). *Probabilistic Robotics*. MIT Press.
- Villa Lopez, R. (2018). Desarrollo y evaluacion del algoritmo de filtro de Kalman para mejorar la precision del posicionamiento GPS. *Escuela Superior Politecnica de Chimborazo*, 38.
- Wang, J. X. (2019). Unscented Kalman Filter for Integrated Navigation of Autonomous Underwater Vehicles. . *Journal of Marine Science and Engineering*, 77, 198.
- Welch, G. &. (2006). *An Introduction to the Kalman Filter*. Carolina : University of North Carolina at Chapel Hill.
- You, S. X. (2018). *daptive Hybrid Kalman Filter-Based Inertial Navigation System for Unmanned Aerial Vehicles. Sensors*. Miami.
- Zhang, Y. X. (2016). Adaptive Unscented Kalman Filter for Autonomous Navigation of Underwater Vehicles. . *Journal of Marine Science and Application*, 15(1), 43-52.

ANEXOS

1.- CODIGO DE PROGRAMACION DEL ROBOT NAVEGACION EN EL IDE ARDUNIO

```

#include "BluetoothSerial.h" //llamar a la libreria de bluetooth
#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
///propia de la ESP32
#error Bluetooth is not enabled! Please run `make menuconfig` to and
enable it
#endif
BluetoothSerial SerialBT;//CONSTRUCTOR
//LECTURA DE ENCODER DE CUADRATURA CON ESP32 DEV KIT V1
String inputString = "";
bool stringComplete = false;
const char separator = ',';
const int dataLength = 2;//indica el numero de datos a recibir
double data[dataLength]; // Valor regula ciclo de trabajo (PWM)
const int C1 = 23; // Entrada de la señal A del encoder M1.
const int C2 = 22; // Entrada de la señal B del encoder M1.
const int C3 = 19; //entrada de seales A encoder M2.
const int C4 = 21; //entrada de seales B encoder M2.
int PWM1 = 32;
int IN_M1A = 33;
int IN_M1B = 25;
int PWM2 = 14;
int IN_M2A = 27;
int IN_M2B = 26;
const int frequency = 500;
const int PWM1_CANAL = 0;
const int PWM2_CANAL = 1;
const int resolution = 8; //0 AL 255
volatile int n = 0; ///numero de pulsos encoder 1
volatile int n1 = 0;///numero de pulsos encoder 2
volatile byte ant = 0;
volatile byte act = 0;
volatile byte ant1 = 0;///motor 2
volatile byte act1 = 0;///motor 2
unsigned long lastTime = 0; // Tiempo anterior
unsigned long sampleTime = 100; // Tiempo de muestreo
unsigned long lastTime1 = 0; // Tiempo anterior m2
unsigned long sampleTime1 = 100; // Tiempo de muestreo m2
double u=0.0;//variable para almacenar velocidad m1
double u1=0.0;//variable para almacenar velocidad m2
double V=0.0;//variable para almacenar velocidad LINEAL
double w=0.0;//variable para almacenar velocidad ANGULAR
int cv=0;//variable de control M1
int cv1=0;//variable de control M2
const int R=193.6;//RESOLUCION DE ENCODER
const double constValue=0.6815;///constante para hallar velocidad lineal
float D_RUEDAS=0.155;///15,5cm-->0.155m
float Diametro_Ruedas=0.0424;///0.0424m-->42.4mm
void IRAM_ATTR encoder(){
    ant=act;
    if(digitalRead(C1)==0 && digitalRead(C2)==0)act=0;
    if(digitalRead(C1)==0 && digitalRead(C2)==1)act=1;
    if(digitalRead(C1)==1 && digitalRead(C2)==0)act=2;

```

```

    if(digitalRead(C1)==1 && digitalRead(C2)==1) act=3;
    //Serial.print("act: ");Serial.println(act);
    //act=PIN_D & 12;
    if(ant==0 && act== 1) n++;
    if(ant==1 && act==3) n++;
    if(ant==2 && act== 0) n++;
    if(ant==3 && act== 2) n++;
    if(ant==0 && act==2) n--;
    if(ant==1 && act==0) n--;
    if(ant==2 && act==3) n--;
    if(ant==3 && act==1) n--;
}
void IRAM_ATTR encoder1(){
    ant1=act1;
    if(digitalRead(C3)==0 && digitalRead(C4)==0) act1=0;
    if(digitalRead(C3)==0 && digitalRead(C4)==1) act1=1;
    if(digitalRead(C3)==1 && digitalRead(C4)==0) act1=2;
    if(digitalRead(C3)==1 && digitalRead(C4)==1) act1=3;
    //Serial.print("act: ");Serial.println(act);
    //act=PIN_D & 12;
    if(ant1==0 && act1== 1) n1++;
    if(ant1==1 && act1==3) n1++;
    if(ant1==2 && act1== 0) n1++;
    if(ant1==3 && act1== 2) n1++;
    if(ant1==0 && act1==2) n1--;
    if(ant1==1 && act1==0) n1--;
    if(ant1==2 && act1==3) n1--;
    if(ant1==3 && act1==1) n1--;
}
void setup() {
    pinMode(C1, INPUT);
    pinMode(C2, INPUT);
    pinMode(C3, INPUT);
    pinMode(C4, INPUT);
    attachInterrupt(digitalPinToInterrupt(C1), encoder, CHANGE);
    attachInterrupt(digitalPinToInterrupt(C2), encoder, CHANGE);
    attachInterrupt(digitalPinToInterrupt(C3), encoder1, CHANGE);
    attachInterrupt(digitalPinToInterrupt(C4), encoder1, CHANGE);
    ///PINES CONTROL M1
    pinMode(PWM1, OUTPUT);
    pinMode(IN_M1A, OUTPUT);
    pinMode(IN_M1B, OUTPUT);
    ///pines control M2
    pinMode(PWM2, OUTPUT);
    pinMode(IN_M2A, OUTPUT);
    pinMode(IN_M2B, OUTPUT);
    //INICIO APAGANDO LOS MOTORES
    digitalWrite(IN_M1A, LOW);
    digitalWrite(IN_M1B, LOW);
    digitalWrite(IN_M2A, LOW);
    digitalWrite(IN_M2B, LOW);
    //delay(1000);///TIEMPO DE ESPERA
    //CONFIGURACION DE PWMS
    ledcSetup(PWM1_CANAL, frequency, resolution);///CONFIGURAR EL CANAL DEL
    PWM CON PARAMETROS FRECUENCIA Y RESOLUCION M1
    ledcSetup(PWM2_CANAL, frequency, resolution);///CONFIGURAR EL CANAL DEL
    PWM CON PARAMETROS FRECUENCIA Y RESOLUCION M2

```

```

ledcAttachPin(PWM1, PWM1_CANAL); //ASIGNO EL CANAL CONFIGURADO AL M1
ledcAttachPin(PWM2, PWM2_CANAL); //ASIGNO EL CANAL CONFIGURADO AL M2
SerialBT.begin("LEONELA"); //CONFIGURO BLUETOOTH CON NOMBRE "LEONELA 1"
Serial.begin(115200); //CONFIGURO VELOCIDAD DE COMUNICACION SERIAL

}
void loop() {
  if(SerialBT.available()){
    //Serial.print("cadena>");
    //Serial.println((char)SerialBT.read());
    delay(50);
  }
  //RECIBIR DATOS POR SERIAL////////////////////////////////////
  while (SerialBT.available()) {////SerialBT.available()<<BLUETOOTH,
Serial.available()<COM
    char inChar = (char)SerialBT.read();
    inputString += inChar;
    if (inChar == '\n') {
      stringComplete = true;
    }
  }
  ///////////////////////////////////////////////////

  //////////////// SI RECIBE DATOS ///////////////////
  if (stringComplete){

    for (int i = 0; i < dataLength ; i++){
      int index = inputString.indexOf(separator);
      data[i] = inputString.substring(0, index).toInt();
      inputString = inputString.substring(index + 1);
    }
    cv = data[0]; //almaceno la variable de control PARA EL M1 recibida
en CV
    ////////////////prueba recepcion 2 datos////////////////////////////////////
    cv1 = data[1]; //almaceno la variable de control PARA EL M2 recibida
en CV1
    //imprimir datos recibidos.
    Serial.print("cv>");
    Serial.println(cv);
    Serial.print("cv1>");
    Serial.println(cv1);
    //delay(1000);
    ///////////////////////////////////////////////////
    if (cv > 0) anticlockwise(cv); else clockwise(abs(cv)); //comparacion
valores positivos o negativos para M1
    if (cv1 > 0) anticlockwise1(cv1); else
clockwise1(abs(cv1)); //comparacion valores positivos o negativos para M2
    inputString = "";
    stringComplete = false;
  }
  if (millis() - lastTime >= sampleTime){ // Se actualiza cada sampleTime
(miliseundos)
    computeU(); //calcular velocidad M1
    computeU1(); //calcular velocidad M2
    V = (abs(u) + abs(u1)) / 2;
    w = (abs(u) - abs(u1)) / D_RUEDAS;
    //imprimir por bluetooth

```

```

        //SerialBT.print("Velocidad M1: ");
        SerialBT.print(V);//IMPRIMO VELOCIDAD LINEAL
        SerialBT.print(",");
        SerialBT.println(w);//IMPRIMO VELOCIDAD ANGULAR
        //SerialBT.print("Velocidad M2:
");SerialBT.print(u1);SerialBT.println(" pulsos");
        //SerialBT.print("Velocidad lineal
");SerialBT.print(V);SerialBT.println(", m/s");
        //SerialBT.print("Velocidad angular
");SerialBT.print(w);SerialBT.println(", rad/s");
        //IMPRIMIR POR SERIAL
        //Serial.print("Velocidad M1: ");Serial.print(u);Serial.println("
pulsos");
        //Serial.print("Velocidad M2: ");Serial.print(u1);Serial.println("
pulsos");
        //Serial.print("Velocidad M2: ");Serial.print(u1);Serial.println("
pulsos");
        //SerialBT.println(u);
        //Serial.print(u);Serial.print(",");Serial.println("2.01");
        //SerialBT.println(u);
        delay(20);
    }
}
void computeU(){//MOTOR 1
    //u=n;//ASIGANDO PULSOS M1 A u
    u= (constValue*n)/(millis() - lastTime);//PULSOS CONVERTIDOS A
VELOCIDAD LINEAL M1
    lastTime = millis();
    n=0;
}
void computeU1(){//MOTOR 2
    //u1=n1;//ASIGANDO PULSOS M2 A U1
    u1= (constValue*n1)/(millis() - lastTime1);//PULSOS CONVERTIDOS A
VELOCIDAD LINEAL M2
    lastTime1 = millis();
    n1=0;
}
void clockwise(int vel_M1){//HORARIO M1
    digitalWrite(IN_M1A, LOW);
    digitalWrite(IN_M1B, HIGH);
    ledcWrite(PWM1_CANAL,vel_M1);
}
void anticlockwise(int vel_M1){//ANTIHORARIO M1
    digitalWrite(IN_M1A, HIGH);
    digitalWrite(IN_M1B, LOW);
    ledcWrite(PWM1_CANAL,vel_M1);
}
void anticlockwise1(int vel_M2){//HORARIO M2
    digitalWrite(IN_M2A, LOW);
    digitalWrite(IN_M2B, HIGH);
    ledcWrite(PWM2_CANAL,vel_M2);
}
void clockwise1(int vel_M2){//ANTIHORARIO M2
    digitalWrite(IN_M2A, LOW);
    digitalWrite(IN_M2B, HIGH);
    ledcWrite(PWM2_CANAL,vel_M2);
}
}

```

2. CODIGO DE PROGRAMACION EM MATLAB Y RAFICAS DEL FILTRO DE KALMAN

✓ FUNCIONES CREADA PARA GRAFICAR EN 3D EL ROBOT MÓVIL

```
function Mobile_Graph=MobilePlot(dx,dy,angz)
global Mobile;
dz=0;

% Matriz de rotación z
Rz=[ cos(angz), -sin(angz) 0 0; sin(angz) cos(angz) 0 0; 0 0 1 0;0 0 0
1];
Rot_TheWholeArm=Rz;
Rot_TheWholeArm(1:3,4) = [dx dy dz]';

WheelColor='g';
BodyColor=[0 0 0.5];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CUERPO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tam=0;
for ii = 1:(length(Mobile.Base))
    robotPatch = Rot_TheWholeArm*Mobile.Base{ii};
    Mobile_Graph(tam+ii) =
    patch(robotPatch(1,:),robotPatch(2,:),robotPatch(3,:),BodyColor,'LineWid
th',1.8);
end

Rot_TheWholeArm(1:3,4) = [dx-0.38*sin(angz) dy+0.38*cos(angz) dz]';

tam=tam+ii;

for ii =1:(length(Mobile.Wheel))
    robotPatch = Rot_TheWholeArm*Mobile.Wheel{ii};
    Mobile_Graph(tam+ii) =
    patch(robotPatch(1,:),robotPatch(2,:),robotPatch(3,:),WheelColor,'LineWi
dth',1.8);
end

Rot_TheWholeArm(1:3,4) = [dx+((0.38+0.1)*sin(angz)) dy-
((0.38+0.1)*cos(angz)) dz]';

tam=tam+ii;
for ii =1:(length(Mobile.Wheel))
    robotPatch = Rot_TheWholeArm*Mobile.Wheel{ii};
    Mobile_Graph(tam+ii) =
    patch(robotPatch(1,:),robotPatch(2,:),robotPatch(3,:),WheelColor,'LineWi
dth',1.8);
end.
```


3. FUNCIÓN PARA GRAFICAR AL ROBOT EN ANIMACIÓN MÓVIL

```

function MobileRobot
width=1.2;
lth=1;
high=0.3;
radio=0.25;

global Mobile;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Cuerpo %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

base=[19.35 -12.39 0,
18.366 -14.764 0,
16.802 -16.802 0,
14.764 -18.366 0,
12.39 -19.35 0,
9.843 -19.685 0,
9.843 -14.764 0,
-9.843 -14.764 0,
-9.843 -19.685 0,
-12.39 -19.35 0,
-14.764 -18.366 0,
-16.802 -16.802 0,
-18.366 -14.764 0,
-19.35 -12.39 0,
-19.685 -9.843 0,
-19.685 9.843 0,
-19.35 12.39 0,
-18.366 14.764 0,
-16.802 16.802 0,
-14.764 18.366 0,
-12.39 19.35 0,
-9.843 19.685 0,
-9.843 14.764 0,
9.843 14.764 0,
9.843 19.685 0,
12.39 19.35 0,
14.764 18.366 0,
16.802 16.802 0,
18.366 14.764 0,
19.35 12.39 0,
19.685 9.843 0,
18.973 8.297 0,
18.384 6.699 0,
17.922 5.061 0,
17.59 3.391 0,
17.39 1.701 0,
17.323 -0 0,
17.39 -1.701 0,
17.59 -3.391 0,
17.922 -5.061 0,
18.384 -6.699 0,
18.973 -8.297 0,
19.685 -9.843 0]/39.37;

```

```

base(:,1)=width*base(:,1);
base(:,2)=lth*base(:,2);
zs=high;

for k=1:length(base)-1
    Mobile.Base{k}=[base(k,1) base(k,2) base(k,3) 1;base(k+1,1)
base(k+1,2) base(k,3) 1;base(k+1,1) base(k+1,2) zs 1 ;base(k,1)
base(k,2) zs 1]';
    if k==length(base)-1
        Mobile.Base{k+1}=[base(length(base),1) base(length(base),2)
base(1,3) 1; base(1,1) base(1,2) base(1,3) 1;base(1,1) base(1,2) zs
1;base(length(base),1) base(length(base),2) zs 1]';
    end
end

Mobile.Base{k+2}=[base'; ones(1,length(base))];
Mobile.Base{k+3}=[[base(:,1) base(:,2) base(:,3)+zs]';
ones(1,length(base))];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% LLANTAS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
th=0:0.3:2*pi;
x=radio*cos(th);
z=radio*sin(th);
y=zeros(1,length(th));

wheel=[x' y' z'+high/2];

zs=0.1;

for k=1:length(wheel)-1
    Mobile.Wheel{k}=[wheel(k,1) wheel(k,2) wheel(k,3) 1;wheel(k+1,1)
wheel(k,2) wheel(k+1,3) 1;wheel(k+1,1) zs wheel(k+1,3) 1 ;wheel(k,1) zs
wheel(k,3) 1]';
    if k==length(wheel)-1
        Mobile.Wheel{k+1}=[wheel(length(wheel),1) wheel(length(wheel),2)
wheel(length(wheel),3) 1; wheel(1,1) wheel(length(wheel),2) wheel(1,3)
1;wheel(1,1) zs wheel(1,3) 1;wheel(length(wheel),1) zs
wheel(length(wheel),3) 1]';
    end
end

Mobile.Wheel{k+2}=[wheel'; ones(1,length(wheel))];
Mobile.Wheel{k+3}=[[wheel(:,1) wheel(:,2)+zs wheel(:,3)]]';
ones(1,length(wheel))];

```

✓ CODIGO PARA SIMULAR EL ROBOT DIFERENCIAL DE NAVEGACION AUTONOMO

```

clear all
close all
clc

% 1) Tiempo
ts=0.1;
t=0:ts:60;

```

```

% 2) Condiciones iniciales
xc(1)=0.6;      %Posicion del centro de eje de las ruedas en X[m]
yc(1)=1;      %Posicion del centro de eje de las ruedas en Y[m]
phi(1)=pi;    %orientacion inicial con respecto al eje x [rad]

% 3) Referencias deseadas
% Trayectoria de un circulo
xrd = 4*cos(0.2*t);
yrd= 4*sin(0.2*t);
%
%

% % Trayectoria de 5 puntas
div=10;

% % a=1/div; b=1/div; c=1/div; d=1/div; j=3; k=2.5;
% % xrd = cos(a*t)-cos(b*t).^j;
% % yrd= sin(c*t)-sin(d*t).^k;

xrdp= diff([0 xrd]); %Posición x1
yrdp= diff([0 yrd]); %Posición y1

a=0.2;

xr(1)=xc(1)+a*cos(phi(1));
yr(1)=yc(1)+a*sin(phi(1));

for k=1:length(t)

% 4) Control

%a) Errores de control
xre(k) = xrd(k) - xr(k);
yre(k) = yrd(k) - yr(k);
e = [xre(k);yre(k)];

%b) Matriz Jacobiana
J=[cos(phi(k)) -a*sin(phi(k));...
   sin(phi(k))  a*cos(phi(k))];

%c) Matriz de ganancia
K = [1 0;...
     0 1];

%d) Ley de control
hdp=[xrdp(k);yrdp(k)];

v = inv(J)*(hdp+K*tanh(e));

u(k)=v(1); %Velocidad lineal de entrada al robot
w(k)=v(2); % Velocidad angular de entrada al robot

```

```

% 5) Aplicar acciones de control al robot

xrp(k)=u(k)*cos(phi(k))-a*w(k)*sin(phi(k));
yrp(k)=u(k)*sin(phi(k))+a*w(k)*cos(phi(k));

% Hallar posiciones
xr(k+1)=xr(k)+ts*xrp(k);
yr(k+1)=yr(k)+ts*yrp(k);
phi(k+1)=phi(k)+ts*w(k);

xc(k+1)=xr(k+1)-a*cos(phi(k+1));
yc(k+1)=yr(k+1)-a*sin(phi(k+1));

end

% Simulacion

pasos=10; fig=figure('Name','Simulacion');
set(fig,'position',[60 60 980 600]);
axis square; cameratoolbar
axis([-6 6 -6 6 0 1]);
grid on
MobileRobot;
M1=MobilePlot(xr(1),yr(1),phi(1));hold on
M2=plot(xr(1),yr(1),'b','LineWidth',2);
plot(xrd,yrd,'r','LineWidth',2);

for i=1:pasos:length(t)
    delete (M1)
    delete (M2)
    M1=MobilePlot(xc(i),yc(i),phi(i)); hold on
    M2=plot(xr(1:i),yr(1:i),'b','LineWidth',2);
    pause (ts)
end

% Graficas
figure('Name','Errores')
subplot(211)
plot(t,xre,'linewidth',2), grid on
legend('Error en x')
xlabel('Tiempo'), ylabel('Error [m]')
subplot(212)
plot(t,yre,'g','linewidth',2), grid on
legend('Error en y')
xlabel('Tiempo'), ylabel('Error [m]')

figure('Name','Acciones de control')
subplot(211)
plot(t,u,'linewidth',2), grid on
legend('Velocidad lineal u')
xlabel('Tiempo'), ylabel('Velocidad [m/s]')
subplot(212)
plot(t,w,'g','linewidth',2), grid on
legend('Velocidad angular w')

```

```
xlabel('Tiempo'), ylabel('Velocidad angular [rad/s]')
```

- ✓ CODIGO PARA GRAFICAR LAS RESPUESTA DEL FILTRO DE KALMAN EN EL WORKSPACE DE MATLAB

```
close all
clc

% Graficas de los resultados
figure
plot(x(:,1),x(:,2),'k', y(:,1),y(:,2),'gs',...
      xhat(:,1),xhat(:,2),'rx',...
      'LineStyle','-');
grid on
title('PREDICION DE LA TRAYECTORIA DEL ROBOT DE NAVEGACION MEDIANTE EL
FILTRO DE KALMAN');
xlabel('X [mm]');
ylabel('Y [mm]');
legend('Posicion Real de Robot','Medicion Actualizada ','Estimacion
del Filtro de Kalman ','Localizacion','Mejor señal');
axis tight;

%Errores de medicion en la posicion X [m]
n_xe = y(:,1)-x(:,1);
% Errores de medicion en la posicion [m]
n_xn = y(:,2)-x(:,2);

% Error de posición en X del filtro de Kalman [m]

e_xe = xhat(:,1)-x(:,1);

% Error de posición en Y del filtro de Kalman [m]

e_xn = xhat(:,2)-x(:,2);

figure;
% Errores de posición X
subplot(2,1,1);
plot(t,n_xe,'b',t,e_xe,'g');
grid on
ylabel('Error de posicion - X [m]');
xlabel('Tiempo [s]');
legend(sprintf('Medicion:
%.3f',norm(n_xe,1)/numel(n_xe)),sprintf('filtro Kalman.:
%.3f',norm(e_xe,1)/numel(e_xe)));
axis tight;
% Errores de posición Y
subplot(2,1,2);
plot(t,y(:,2)-x(:,2),'b',t,xhat(:,2)-x(:,2),'k');
grid on

ylabel('Error de posicion - Y [m]');
xlabel('Tiempo [s]');
```

```

legend(sprintf('Mediccion:
%.3f',norm(n_xn,1)/numel(n_xn)),sprintf('filtro Kalman:
%.3f',norm(e_xn,1)/numel(e_xn)));
axis tight;

%GRAFICAS DE LAS VELOCIDADES
e_ve = xhat(:,3)-x(:,3); % [m/s] Error de velocidad en X del filtro de
Kalman
e_vn = xhat(:,4)-x(:,4); % [m/s] Error de velocidad en Y del filtro de
Kalman
figure;

% velocidad en direcci3n en X y su estimaci3n
subplot(2,1,1);
plot(t,x(:,3),'b',t,xhat(:,3),'g');
grid on
ylabel('Velocidad - X [m]');
xlabel('Tiempo [s]');
legend('Actual','Filtro de Kalman','Localizacion','Mejor senal');
axis tight;
subplot(2,1,2);
% Estimacion de error
plot(t,e_ve,'k');
grid on
ylabel('Velocidad del Error - Y [m]');
xlabel('Tiempo [s]');
legend(sprintf('Filtro de Kalman: %.3f',norm(e_ve,1)/numel(e_ve)));
axis tight;

```

3. ETAPA DE CONSTRUCCION DE ROBOT DIFERENCIAL DE NAVEGACION AUTOMANA.

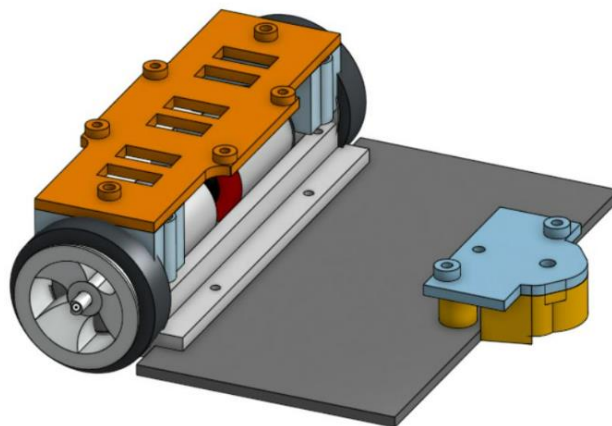


Figura: 39 Dise1o del robot 3D

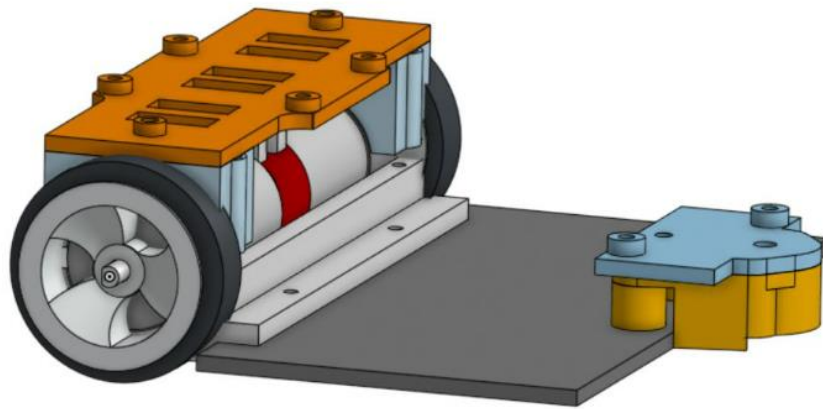


Figura: 40 Diseño lateral del robot 3D

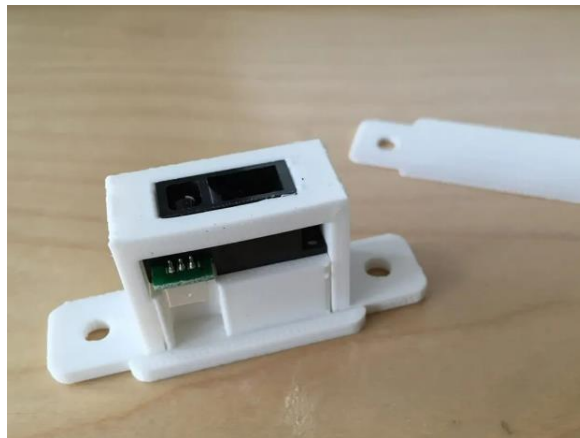


Figura: 41 Estructura para sensores impreso 3D

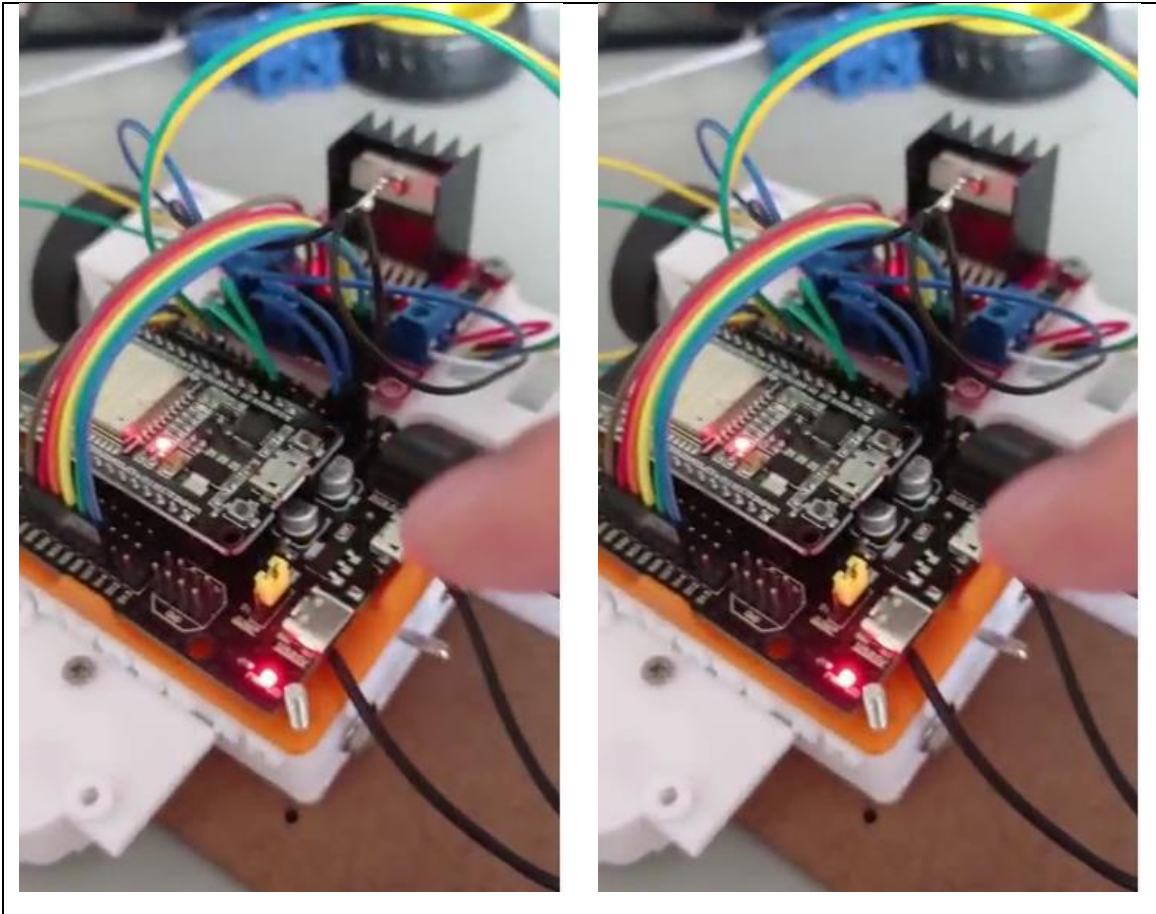


Figura: 42 Robot con sus componentes.

4. MEDICION Y LECTURA DE LAS SEÑALES ENTREGADAS POR LOS SENSORES, ENCODERS, IMU.

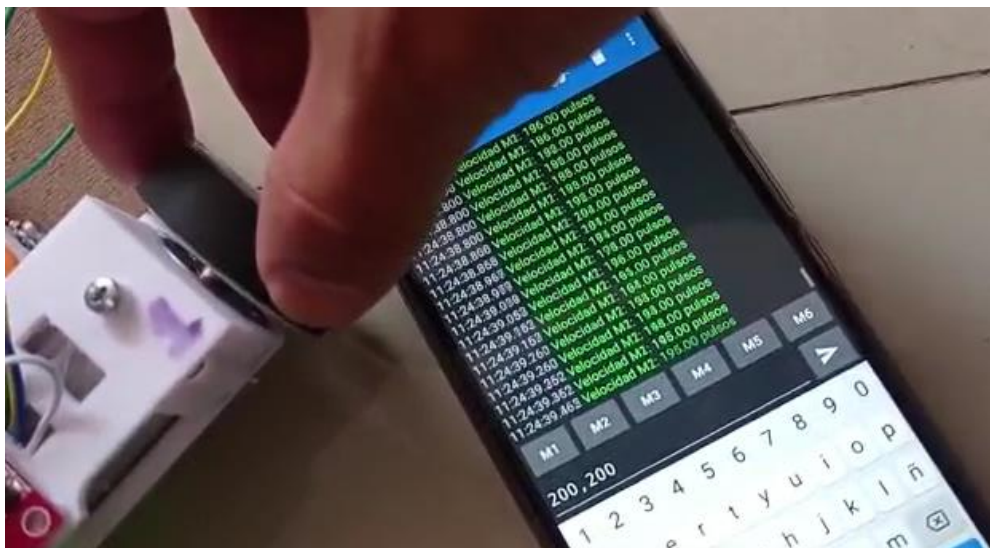


Figura: 43 Prueba de datos Obtenidos del robot vía Bluetooth

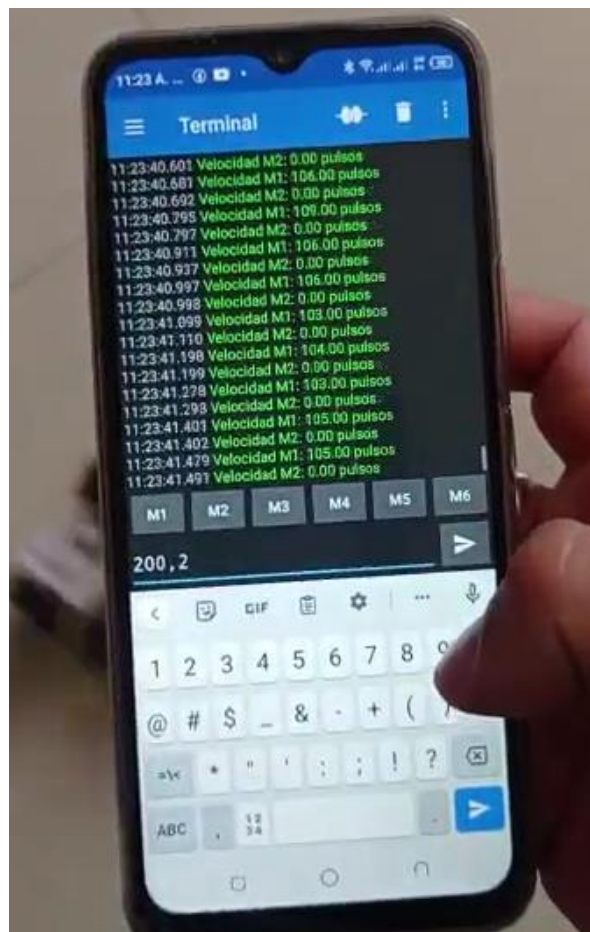


Figura: 44 Visualización de datos obtenidos del robot vía Bluetooth

- La medición y lectura de datos entregados por los sensores en nuestro robot diferencial de navegación autónoma.
- Estos sensores, encargados de captar la velocidad lineal y angular en tiempo real, nos brindan información crucial para estimar las trayectorias reales del robot.
- Gracias a la aplicación óptima de un filtro de Kalman, nuestras estimaciones se perfeccionan, dando lugar a una trayectoria circular impecable.
- Estas lecturas de datos, impregnadas de precisión y exactitud científica, son el cimiento fundamental que nos permite explorar los confines de la autonomía robótica.

Entrada de datos (desde Serie estándar sobre el vínculo Bluetooth (C

Los datos procedentes del origen de datos actual aparecerán abajo según se reciban.

Datos actuales

TIME	CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8	CH9
19:10:34,77	0,77	2,11							

Información histórica

TIME	CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8	CH9
19:10:34,77	0,77	2,11							
19:10:34,67	0,77	2,02							
19:10:34,56	0,8	2,07							
19:10:34,46	0,81	2,24							
19:10:34,38	0,78	1,98							
19:10:34,27	0,78	2,07							
19:10:34,16	0,8	2,02							
19:10:34,06	0,8	2,11							
19:10:33,98	0,82	2,15							
19:10:33,86	0,82	2,15							
19:10:33,77	0,82	2,11							
19:10:33,67	0,81	2,07							
19:10:33,56	0,82	2,02							
19:10:33,47	0,84	2,15							
19:10:33,38	0,86	2,15							

Figura: 45 Datos obtenidos Serial

A	D
VELOCIDAD LINEAL	VELOCIDAD ANGULAR
1,02	2,11
1,02	2,07
1,02	2,11
1,03	2,07
1,04	2,02
1,03	2,02
1,04	2,02
1,04	2,02
1,02	2,02
1,02	2,02
1,01	2,15
1,02	2,2
1,02	2,2
1,02	2,11
1,02	2,11
1,03	2,15
1,04	2,11
1,04	2,11
1,04	2,02
1,04	2,15
1,05	2,07
1,05	2,07
1,05	2,07
1,05	2,07
1,05	2,07
1,05	2,07
1,06	2,02
1,06	2,02
1,05	2,02
1,05	2,11
1,04	2,11
1,04	2,2
1,04	2,2
1,05	2,15
1,05	2,2
1,05	2,15
1,05	2,15
1,05	2,15
1,05	2,15
1,05	2,2
1,05	2,2

Figura: 46 Datos obtenidos de velocidad Lineal y Angular.