



**Universidad Tecnológica Centroamericana**

**Facultad de Ingeniería Mecatrónica**

**Proyecto de Investigación**

**DISEÑO DE UNA PLACA DE IMU MÚLTIPLES CON FUSIÓN DE  
SENSORES PARA LA ESTIMACIÓN DE ORIENTACIÓN**

**Previo a la obtención del título:**

**Ingeniero en Mecatrónica**

**Presentado por:**

**21641009**

**Juan Elías Kattan Urrutia**

**Asesor: Ing. José Luis Ordoñez**

**San Pedro Sula, Cortés**

**Julio 2021**

**Derechos de Autor**

## EPÍGRAFE

*“El experto es una persona que ha cometido todos los errores que se pueden cometer en un campo muy limitado”.*

*- Niels Bohr*

## Resumen Ejecutivo

Una unidad de medida inercial, o IMU, es un dispositivo electrónico, especialmente un dispositivo MEMS, que fusiona las salidas de un giroscopio, magnetómetro y acelerómetro para medir la orientación, el rumbo, la velocidad angular de un cuerpo o cualquier otro valor escalar o vectorial relacionado con lo antes mencionado. Un problema común con la unidad de medición inercial comercializada producida en masa es su falta de estabilidad en su salida, especialmente con los sesgos relacionados con el giroscopio y los problemas de calibración del magnetómetro. La deriva del giroscopio, también conocida como inestabilidad de polarización, es el error acumulado debido a la integración de la velocidad angular de su salida bruta. La salida del magnetómetro también es esencial para las aplicaciones de fusión de sensores. Un magnetómetro ideal mide la fuerza del campo magnético a través de los tres ejes en el espacio euclidiano, trazando una esfera perfecta centrada en el origen. Las fuentes de hierro duro y hierro dulce distorsionan la esfera en un elipsoide desplazado del origen. El alcance de este proyecto final es eliminar o minimizar los efectos del sesgo de deriva del giroscopio del sistema de referencia de actitud y rumbo (AHRS) para tener información en tiempo real más precisa y estable sobre el rumbo y la orientación de la placa electrónica. El algoritmo de Kalman extendido elimina el problema de errores acumulados por deriva en el ángulo de guiñada en las aplicaciones de fusión de sensores. Las mejoras en la estabilidad de la salida en ángulos en función al número de sensores MPU-9250 no son lineales o exponenciales cuando se utilizan cuatro sensores. Sin embargo, hay una mejora en la reducción de ruido.

Palabras clave – fusión de sensores, AHRS, filtro extendido de Kalman

## **Abstract**

An inertial measurement unit, or IMU, is an electronic device, especially a MEMS device, that fuses the outputs of a gyroscope, magnetometer, and accelerometer to measure a body's orientation, heading, angular velocity, or any other scalar or vector value related to the aforementioned. A commonplace problem with mass-produced commercialized inertial measurement unit is their lack of stability in their output, especially with gyroscope-related biases and magnetometer calibration issues. The gyroscope drift, also known as the bias instability, is the accumulated error due to the integration of the angular velocity of its raw output. The magnetometer output is also essential for sensor fusion applications. An ideal magnetometer measures the magnetic field strength across the three axes in the Euclidean space, plotting a perfect sphere centered at the origin. Hard iron and soft iron sources distort the sphere into an ellipsoid displaced from the origin. The scope of this capstone project is to eliminate or minimize the effects of the gyroscope drift bias from the attitude and heading reference system (AHRS) to have more accurate and stable real-time information about the electronic board's heading and orientation.

Keywords – sensor fusion, AHRS, extended Kalman filter

# Índice De Contenido

1	Introducción.....	1
2	Planteamiento del problema.....	2
2.1	Precedentes del problema.....	2
2.2	Definición del problema.....	2
2.3	Justificación.....	3
2.4	Preguntas de Investigación.....	3
2.5	Objetivos.....	3
2.5.1	Objetivos Generales.....	3
2.5.2	Objetivos Específicos.....	3
3	Marco Teórico.....	4
3.1	Fusión de sensores.....	4
3.1.1	Concepto de Fusión de Sensores.....	4
3.1.2	Unidad de Medición Inercial.....	4
3.2	Sistemas de Referencia de Actitud y Rumbo.....	6
3.2.1	Navegación por Estima.....	7
3.2.2	Otros Usos de Fusión de Datos.....	8
3.3	Filtros para Datos de Salida del Sensor.....	10
3.3.1	Filtro de Kalman.....	10
3.3.2	Filtro de Kalman Extendido.....	14
3.3.3	Filtro de Kalman <i>Unscented</i> .....	17
3.4	Modelos matemáticos para descripción espacial.....	19
3.4.1	Álgebra lineal.....	19
3.4.2	Ángulos de Euler.....	20

3.4.3	Cuaterniones .....	21
4	Metodología .....	23
4.1	Enfoque .....	23
4.2	Variables de investigación.....	23
4.2.1	Variables independientes .....	23
4.2.2	Variables dependientes.....	23
4.3	Técnicas e Instrumentos aplicados .....	24
4.4	Materiales .....	24
4.5	Población y muestra .....	24
4.6	Metodología de estudio.....	25
4.7	Metodología de validación .....	27
4.8	Cronograma de actividades.....	29
5	Análisis y Resultados.....	31
5.1	Diseño de la Placa .....	31
5.2	Fusión de los cuatro IMU.....	32
5.2.1	Acelerómetro.....	32
5.2.2	Magnetómetro .....	35
5.2.3	Giroscopio .....	39
5.2.4	Elección del algoritmo .....	42
5.2.5	Modelado del Espacio de Estados .....	43
5.3	Resultados.....	44
5.3.1	MPU-9250 singular con ejemplo integrado MATLAB.....	44
5.3.2	MÚLTIPLES MPU-9250 ACOPLADOS CON FILTRO EXTENDIDO DE KALMAN ...	46
6	Conclusiones.....	54
7	Recomendaciones.....	54

8	Futuros Trabajos.....	54
9	Bibliografía.....	55
10	Anexos.....	59
10.1	Configuración del Arduino y de los cuatro MPU-9250.....	59
10.2	Calibración e Implementación del Filtro de Kalman Extendido .....	59
10.3	Gráficas .....	61

## Índice De Ilustraciones

Ilustración 1. Unidad de medición inercial de cápsula Apollo, parte del sistema de guía que se exhibe en Draper Labs (Cambridge, MA), donde se desarrolló.....	5
Ilustración 2. Seguimiento del filtro de la desviación del sesgo del giroscopio.....	6
Ilustración 3. Se muestra una persona corriendo en línea recta en un pasillo (a) persona moviéndose en línea recta en el pasillo en el marco de la cámara, (b) robot siguiendo la trayectoria.....	8
Ilustración 4. Metodología de minería y fusión de datos propuesta para predecir el rendimiento de los estudiantes a partir de múltiples fuentes de datos.....	10
Ilustración 5. Flujograma del filtro de Kalman y sus ecuaciones para el cálculo de sus parámetros.....	14
Ilustración 6. Flujograma del filtro de Kalman extendido y las ecuaciones para el cálculo de sus parámetros.....	15
Ilustración 7. Problema de bloqueo del cardán para ángulos de Euler (A: sin bloqueo del cardán, B: los ángulos de guiñada y balanceo están bloqueados.....	20
Ilustración 8. Visualizando cuaterniones, una serie de videos explorables de 3blue1brown..	22
Ilustración 9. Estructura jerárquica de una unidad inercial multisensor modelada linealmente controlada por filtro. Cada bloque describe el estado de cada eje (i.e., x,y,z).....	27
Ilustración 10. Media del error en las estimaciones de sesgo del acelerómetro y giroscopio para un error de sincronización de tiempo de 100 ms. Simulación MC: línea roja del eje X (adelante) con círculos; Eje Y (de lado) - línea verde con cuadrados.....	28
Ilustración 11. Cronograma, primera captura de pantalla.....	29
Ilustración 12. Cronograma, segunda captura de pantalla.....	29
Ilustración 13. Cronograma, tercera captura de pantalla.....	30
Ilustración 14. Cronograma, cuarta captura de pantalla.....	30
Ilustración 15. Esquemático de la PCB diseñada en Fritzing (izquierda) y el dispositivo fabricado (derecha).....	32



Ilustración 16. Distancias entre los sensores MPU-9250 para la simulación de salidas.....	34
Ilustración 17. Gráficos trazados por las salidas IMU ideales simuladas.....	35
Ilustración 18. Gráficos trazados por la salida del magnetómetro del IMU ideal. ....	35
Ilustración 19. Salidas del magnetómetro del IMU ideal luego de girar 360° por los tres ejes. .....	36
Ilustración 20. Salida del magnetómetro del IMU ideal, agregando corrupciones por hierro suave y hierro duro.....	39
Ilustración 21. Salida de la trayectoria fusionada de IMUs ideales. ....	40
Ilustración 22. Algoritmo de combinación de las cuatro señales de los MPU-9250. ....	41
<i>Ilustración 23. Compensación de error del ángulo de salida de un IMU no ideal virtual. ....</i>	<i>41</i>
Ilustración 24. Algoritmo de fusión de sensores con filtro extendido de Kalman. Leyenda: entradas del sensor (azul), EKF (naranja), ángulos de salida (verde). ....	43
Ilustración 25. Salida de fusión de sensor MPU-9250 singular con filtro de Kalman de estado de error interno.....	45
<i>Ilustración 26. Gráfica de valores de ángulos sacados con filtro extendido de Kalman, un solo sensor MPU-9250. ....</i>	<i>48</i>
Ilustración 27. Gráfica de valores de ángulos sacados con filtro extendido de Kalman, dos sensores MPU-9250.....	49
Ilustración 28. Gráfica de valores de ángulos sacados con filtro extendido de Kalman, tres sensores MPU-9250. ....	50
Ilustración 29. Gráfica de valores de ángulos sacados con filtro extendido de Kalman, cuatro sensores MPU-9250.....	51
Ilustración 30. Relación del ángulo de guiñada con la diferencia cresta-vaguada.....	52
Ilustración 31. Relación del ángulo de cabeceo con la diferencia cresta-vaguada. ....	53
Ilustración 32. Relación del ángulo de balanceo con la diferencia cresta-vaguada.....	53

## Índice De Tablas

Tabla 1. Comparación de la naturaleza del modelo matemático, distribución asumida y el costo de implementación del filtro de Kalman y sus mejoras. ....	17
Tabla 2.....	46
Tabla 3.....	46
Tabla 4.....	48
Tabla 5.....	50
Tabla 6.....	51
Tabla 7.....	52

## Índice De Ecuaciones

Ecuación 1.....	11
Ecuación 2.....	11
Ecuación 3.....	12
Ecuación 4.....	12
Ecuación 5.....	12
Ecuación 6.....	12
Ecuación 7.....	12
Ecuación 8.....	13
Ecuación 9.....	13
Ecuación 10.....	13
Ecuación 11.....	13
Ecuación 12.....	16
Ecuación 13.....	16
Ecuación 14.....	16
Ecuación 15.....	16
Ecuación 16.....	16
Ecuación 17.....	17
Ecuación 18.....	19
Ecuación 19.....	19
Ecuación 20.....	21
Ecuación 21.....	21
Ecuación 22.....	21
Ecuación 23.....	21
Ecuación 24.....	21

Ecuación 25. Matriz de posición de los sensores para la simulación de las salidas. ....	33
Ecuación 26. Cuaternión a ángulo de Euler, balanceo.....	33
Ecuación 27. Cuaternión a ángulo de Euler, cabeceo .....	33
Ecuación 28. Cuaternión a ángulo de Euler, guiñada.....	33
Ecuación 29. Elipsoide creado por las salidas del magnetómetro. ....	37
Ecuación 30. Esfera de las salidas de un magnetómetro ideal. ....	37
Ecuación 31 .....	37
Ecuación 32 .....	37
Ecuación 33 .....	38
Ecuación 34 .....	38
Ecuación 35 .....	38
Ecuación 36 .....	38
Ecuación 37 .....	38
Ecuación 38 .....	38
Ecuación 39 .....	44
Ecuación 40 .....	44
Ecuación 41 .....	44
Ecuación 42 .....	44
Ecuación 43 .....	44

# 1 INTRODUCCIÓN

Una unidad de medida inercial, o IMU, es un dispositivo electrónico, especialmente un dispositivo MEMS, que fusiona las salidas de un giroscopio, magnetómetro y acelerómetro para medir la orientación, el rumbo, la velocidad angular de un cuerpo o cualquier otro valor escalar o vectorial relacionado con lo antes mencionado. Son ideales para proyectos de bajo costo que involucran robótica, vehículos aéreos y submarinos no tripulados y controladores de realidad virtual. Un problema común con la unidad de medición inercial comercializada, es decir, la producida en masa para aplicaciones de bajo costo, es su falta de estabilidad en su salida, especialmente con los sesgos relacionados con el giroscopio y los problemas de calibración del magnetómetro. La deriva del giroscopio, también conocida como inestabilidad de polarización, es el error acumulado debido a la integración de la velocidad angular de su salida bruta. La salida del magnetómetro también es esencial para las aplicaciones de fusión de sensores. Un magnetómetro ideal mide la fuerza del campo magnético a través de los tres ejes en el espacio euclidiano, trazando una esfera perfecta centrada en el origen. Las fuentes de hierro duro y hierro dulce distorsionan la esfera en un elipsoide desplazado del origen.

Se desarrollará el algoritmo de filtrado por medio de MATLAB y los resultados serán validados por medios cuantitativos y cualitativos en la metodología jerárquica. Se hará un prototipo de una placa multisensorial IMU (MIMU) compacta para su futura implementación en proyectos de robótica u otra aplicación.

El alcance de este proyecto final será eliminar o minimizar los efectos del sesgo de deriva del giroscopio del sistema de referencia de actitud y rumbo (AHRS) para tener información en tiempo real más precisa y estable sobre el rumbo y la orientación de la placa electrónica a desarrollar.

Las diferentes secciones del presente informe consisten en la introducción, planteamiento del problema, marco teórico, análisis y resultados, y conclusiones y recomendaciones.

## **2 PLANTEAMIENTO DEL PROBLEMA**

### **2.1 PRECEDENTES DEL PROBLEMA**

Un desafío con las IMU existentes basadas en microsensores es su resolución y estabilidad limitadas que las hace poco confiables después de períodos cortos (pocos segundos) y normalmente sufren errores acumulados, lo que es un problema para proyectos de vehículos autónomos de bajo costo.

### **2.2 DEFINICIÓN DEL PROBLEMA**

Un problema común con los IMUs comerciales es la falta de estabilidad en su salida, especialmente con los sesgos relacionados con el giroscopio y los problemas de calibración del magnetómetro. La deriva del giroscopio, también conocida como inestabilidad de polarización, es el error acumulado debido a la integración de la velocidad angular de su salida bruta.

Los giroscopios están sujetos a inestabilidades de sesgo, en las que la lectura cero iniciales del giroscopio provocará una deriva con el tiempo debido a la integración de imperfecciones inherentes y ruido dentro del dispositivo. La repetibilidad de sesgo se puede calibrar en el rango de temperatura conocido de la IMU. Sin embargo, la integración constante de la velocidad angular de la salida bruta del giroscopio provocará un error angular. Estos errores se acumularán a medida que la rotación basada en el giroscopio o las estimaciones de ángulos se desvíen a largo plazo. El resultado indeseable de la deriva es que el error de un rumbo calculado aumenta continuamente sin cesar.

El eje de guiñada es más sensible a esta deriva. Una buena parte de la desviación del giroscopio del eje de cabeceo (actitud) y balanceo se puede eliminar dentro de una IMU mediante el uso de retroalimentación de acelerómetro para monitorear la posición relativa a la gravedad. El filtrado de la salida del giroscopio dentro de una IMU utilizando un filtro de paso bajo o de Kalman también es un método ampliamente utilizado para cancelar una parte del error de deriva.

## **2.3 JUSTIFICACIÓN**

Los sistemas de navegación inercial equipados con IMU forman la base para la navegación y el control de vehículos comerciales y militares como aviones tripulados, misiles, barcos, submarinos y satélites. Las IMU también son componentes esenciales en la guía y el control de sistemas no tripulados como UAV (vehículos aéreos no tripulados), UGV (vehículos terrestres no tripulados) y UUV (vehículos submarinos no tripulados).

Teniendo en cuenta la gran importancia de la precisión y confiabilidad de las unidades de medición inercial en aplicaciones vehiculares, el control dinámico de un vehículo autónomo requiere sensores con respuesta dinámica, y una IMU precisa hace un trabajo satisfactorio al rastrear la actitud dinámica y los cambios de posición con precisión.

Un sensor IMU defectuoso, así como salidas inválidas o irreales del mismo, pueden tener consecuencias fatales o costosas.

## **2.4 PREGUNTAS DE INVESTIGACIÓN**

¿Puede solucionarse el problema de deriva de un sensor mediante programación?

¿Qué algoritmo puede resolver este problema presente en librerías genéricas de *Sensor Fusion*?

¿Qué efecto tiene agregar sensores IMU adicionales al algoritmo?

## **2.5 OBJETIVOS**

### **2.5.1 OBJETIVOS GENERALES**

Diseño de un prototipo de una unidad de medida inercial que proporcione varios minutos de navegación confiable en ausencia de otras señales de referencia (por ejemplo, GPS), así como mejoras en las cualidades del sensor.

### **2.5.2 OBJETIVOS ESPECÍFICOS**

Diseño de un algoritmo de fusión de sensores que eliminen o minimicen el error por deriva.

Determinar la relación entre el número de sensores MPU-9250 y las mejoras en la calidad de las salidas del sensor.

## **3 MARCO TEÓRICO**

### **3.1 FUSIÓN DE SENSORES**

#### **3.1.1 CONCEPTO DE FUSIÓN DE SENSORES**

La fusión de sensores, también conocido como fusión multisensorial, es la combinación de datos sensoriales o datos derivados de los mismos, utilizando algoritmos de software para sinérgicamente producir datos con mayor robustez, cobertura espacial y temporal ampliada, mayor confianza, menos ambigüedad e incertidumbre comparado a la utilización individual de cada sensor (Elmenreich, 2002, p. 7) . El nivel de fusión del sensor se define según el tipo de entrada al algoritmo de fusión. La fusión de sensores ejecuta combinando datos sin procesar provenientes de diversas fuentes y parámetros estimados o extrapolados.

Menciona Bashir et al. (2020) "las aplicaciones del algoritmo de fusión de sensores incluyen la determinación de la trayectoria vertical en la navegación, el control de sistemas robóticos, la detección de caídas y los deportes."

La combinación de parámetros generalmente consta de dos pasos: la acción de recolectar información de los transductores, y la de optimizar la fiabilidad de la información mediante optimización del coeficiente de distribución, redundancia de seguridad y predicción de estado (C. Liu et al., 2020). Existen diversos algoritmos, softwares y herramientas con la cual se puede dar este proceso. Menciona Yu et al. (2021) que:

"Primero, el conjunto de datos óptimo se obtiene después del procesamiento de datos de dos pasos del criterio de Dixon y el filtro de Kalman, lo que evita efectivamente la incertidumbre del valor de observación original. Luego, se obtiene el factor de peso mejorado de acuerdo con el peso dinámico y el peso sólido, y luego se obtienen los mejores datos de fusión." (p. 5)

#### **3.1.2 UNIDAD DE MEDICIÓN INERCIAL**

Una unidad de medida inercial (IMU) es un transductor multisensorial que mide la aceleración de un cuerpo, su velocidad angular y la orientación de este, utilizando una combinación, respectivamente, de acelerómetros, giroscopios y magnetómetros. Las IMU se utilizan



generalmente vehículos aéreos, especialmente los autónomos (UAV, o *unmanned aerial vehicle*) y astronaves; entre ellos, satélites y módulos de aterrizaje (Kok et al., 2017, p. 3) .

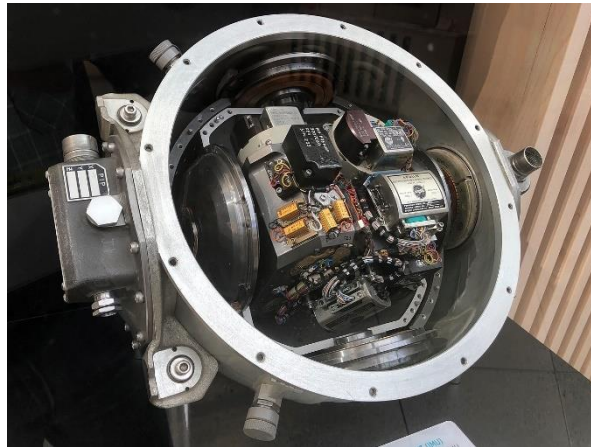


Ilustración 1. Unidad de medición inercial de cápsula Apollo, parte del sistema de guía que se exhibe en Draper Labs (Cambridge, MA), donde se desarrolló.

Fuente: (Erickson, 2021)

Las IMU generalmente se incorporan en los sistemas de navegación inercial (INS) que usan las salidas del transductor no procesadas para calcular la actitud, las velocidades angulares, la velocidad lineal y la posición del dispositivo respecto al planeta. Menciona Larey et al. (2020) que un sistema de navegación inercial (INS) consta de la unidad IMU algoritmos para calcular los ejes de posición ya establecidos.

Una unidad de IMU comercial de bajo costo se basa en tecnología MEMS (sistemas microelectromecánicos). Varias unidades de sensores IMU son denominadas MIMU, o IMU múltiples. El siguiente artículo resume perfectamente el problema con estos dispositivos de bajo costo que se tiene como alcance resolver en este proyecto de investigación. Menciona Larey et al. (2020):

“Las mediciones de IMU no son precisas y tienen errores de medición significativos que hacen que la solución de navegación se desvíe con el tiempo. Para evitar la deriva, los INS se fusionan con sensores externos. Sin embargo, en algunas situaciones, el sensor externo no está disponible (por ejemplo, GPS en un entorno interior), la solución de navegación dependerá únicamente del INS, una solución de navegación inercial pura. Uno de los

enfoques para mitigar la deriva mientras se mantiene un sistema de bajo costo es utilizar una arquitectura de múltiples IMU (MIMU).” (p. 1)

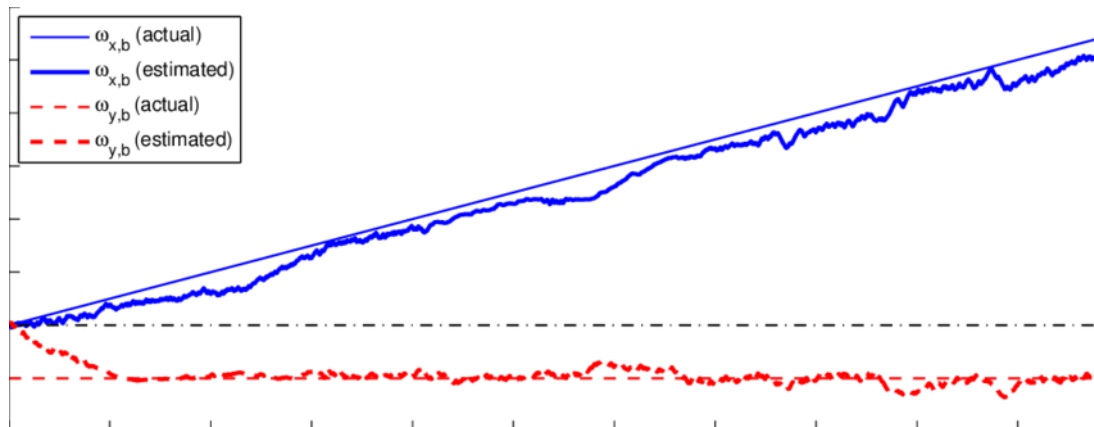


Ilustración 2. Seguimiento del filtro de la desviación del sesgo del giroscopio.

Fuente: (Madgwick et al., 2011)

### 3.2 SISTEMAS DE REFERENCIA DE ACTITUD Y RUMBO

Un sistema de referencia de actitud y rumbo (AHRS) consiste en sensores para la orientación en cada eje en el espacio euclidiano que proporcionan información de actitud para la aeronave. La diferencia principal entre una unidad de medida inercial (IMU) y un AHRS es que el IMU envía los datos del sensor a un dispositivo adicional, mientras que el AHRS adiciona un sistema de procesamiento a bordo para proporcionar información de actitud y rumbo. Un algoritmo de estimación no lineal (e.g. filtro de Kalman extendido) se utiliza normalmente para encontrar una solución en base a diversas entradas. Menciona García et al. (2020) que, utilizando un procesador con software de control PixHawk, integrando un AHRS, se pueden implementar diversos algoritmos no lineales como el EKM para estimar la actitud del vehículo y crear un vector de dirección utilizado como referencia para el desplazamiento del vehículo.

A pesar de su extenso uso en la aeronautica y robótica, los AHRS tienen aplicaciones en procesos mucho más diversos. En las grúas torre, por ejemplo, aunque la grúa es estable, el mástil y el brazo se balancean y se doblan por el peso de las cargas y fuerzas de fenómenos meteorológicos. Para ello, se crearon los sistemas de control de balanceo para grúas torre mejoran la productividad, la precisión de la colocación de la carga y reducen el daño material causado por el contacto accidental de cargas oscilantes. Mencionan Kruk y Sulowicz (2019) que, aplicando simultáneamente otros algoritmos y funciones matemáticas como método de

estabilización basado en el sistema de sensores AHRS y lógica difusa se pudo obtener un mejor método para la estabilización de la carga de una torre grúa, manteniendo la estabilidad de la carga y amortiguando sus oscilaciones.

Así mismo, demostraron Kim y Lee (2019) que se puede implementar un dispositivo de detección remota por medio de reconocimiento de movimiento basado en AHRS para controlar el avatar en el juego VR (realidad virtual) que lleva el HMD (pantalla montada en la cabeza).

### **3.2.1 NAVEGACIÓN POR ESTIMA**

La navegación por estima, o *dead reckoning*, es el proceso de calcular la posición en tiempo real de un cuerpo en movimiento usando una posición fija previamente determinada y luego incorporando estimaciones de velocidad y rumbo en función del tiempo.

La navegación por estima está sujeta a errores acumulativos. Los avances en las ayudas a la navegación que brindan información precisa sobre la posición, en particular la navegación por satélite utilizando el Sistema de Posicionamiento Global (GPS), han hecho que la navegación por estima sea obsoleta para muchas de sus aplicaciones originales. Sin embargo, los sistemas de navegación inercial utilizan la navegación a estima y se siguen ampliamente aplicando.

Menciona (Tan et al., 2018) que se puede determinar la posición de un solo robot móvil en un enjambre utilizando técnicas de navegación por estima, siendo la integral de Riemann utilizada para dividir la trayectoria de un robot en una distancia muy pequeña.

Otro notable ejemplo de navegación por estima es un sistema de posicionamiento basado en el modelo de movimiento del vehículo cuyo sistema utiliza información de estados anteriores, salidas de velocidad y rumbo del vehículo para estimar la posición en tiempo real. La información del vehículo es procesada por un filtro de Kalman velocidad-aceleración para corregir el error del odómetro, mejorando la precisión del sistema de posicionamiento local. (Zhang et al., 2019)

El último ejemplo que se muestra es un robot móvil que sigue a una persona que utiliza la navegación a estima de peatones con datos inerciales de teléfonos inteligentes. La ruta y la orientación de la persona a seguir se calcula utilizando la estimación del anfitrión, utilizando

el proceso de tres pasos de detección de pasos, estimación de longitud de zancada y estimación de orientación. (Dam et al., 2020)

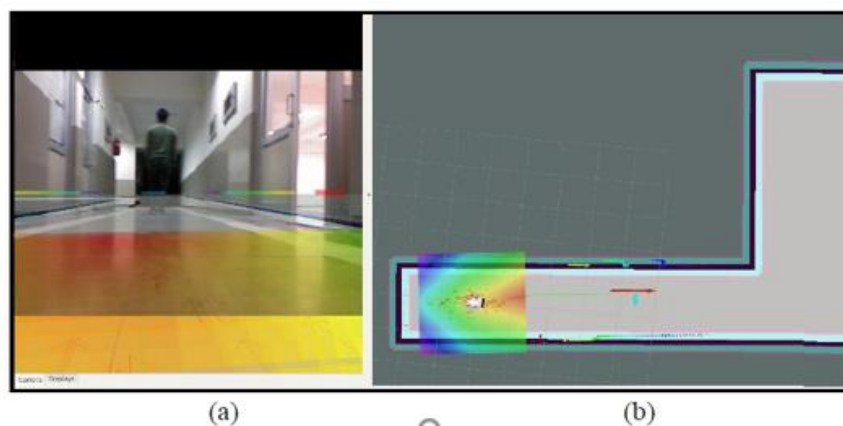
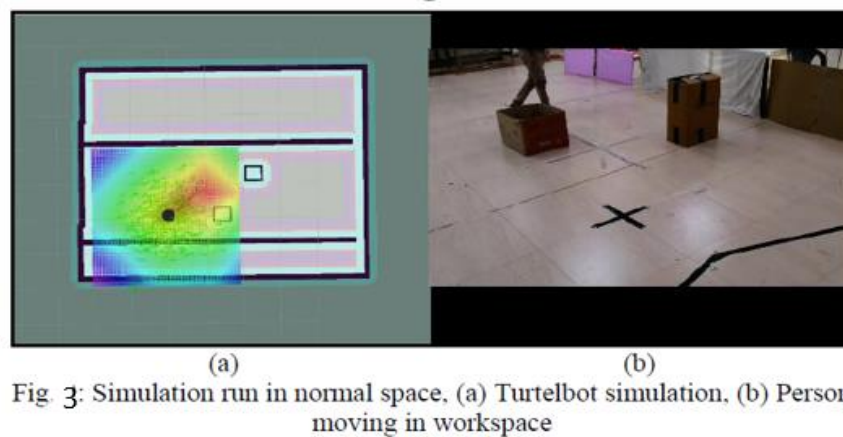


Ilustración 3. Se muestra una persona corriendo en línea recta en un pasillo (a) persona moviéndose en línea recta en el pasillo en el marco de la cámara, (b) robot siguiendo la trayectoria.

Fuente: (Dam et al., 2020)

### 3.2.2 OTROS USOS DE FUSIÓN DE DATOS

Si bien las siguientes aplicaciones de fusión de datos no serán incorporadas en este proyecto de tesis, éstos han tenido algoritmos y métodos similares o idénticos a los de fusión de sensores, por lo cual los posteriores resultados de esta investigación podrían aportar conocimiento o herramientas nuevas a los siguientes campos de aplicación. La fusión de datos de múltiples sensores es solamente una rama de la teoría de fusión de datos. Tiene áreas de aplicación en sistemas de información geoespacial, quimioinformática, biometría, inteligencia

empresarial, cartografía de suelos, oceanografía, redes de sensores inalámbricos y muchos otros enfoques científicos, de ingeniería y humanidades.

Si bien la fusión de sensores geospaciales es la razón de ser de esta disertación de tesis, el siguiente ejemplo muestra una aplicación de fusión de datos para el mapeo de la calidad del aire utilizando observaciones de sensores de bajo costo. El siguiente artículo describe cómo se utiliza el monitoreo de la calidad del aire mediante observaciones de sensores de bajo costo, implementando fusión de datos. Menciona Gressent et al. (2020) que:

“Se tiene como objetivo utilizar la gran cantidad de observaciones proporcionadas por los sensores para el mapeo de la calidad del aire a escala urbana con el fin de mostrar el valor agregado potencial con respecto a los cálculos del modelo de dispersión (ADMS Urban). Se aplica un procesamiento previo al conjunto de datos del sensor sin procesar para eliminar las observaciones no confiables basadas en una técnica de detección de valores atípicos y para compensar la deriva de la medición ajustando la variación diaria estimada de las concentraciones de frecuencia de fondo subyacentes con las estaciones de referencia. Luego, la fusión de datos se realiza combinando las observaciones de sensores de bajo costo fijos y móviles preprocesados.” (pág. 1)

La siguiente ilustración demuestra un flujograma de fusión de datos para predecir el rendimiento académico final de los estudiantes universitarios utilizando datos multimodales de múltiples fuentes de entornos de aprendizaje combinados. Explica Chango et al., (2021) que “se aplica un procesamiento al conjunto de datos del sensor sin procesar para eliminar las observaciones no confiables basadas en una técnica de detección de valores atípicos y para compensar la deriva de la medición ajustando la variación diaria estimada.”

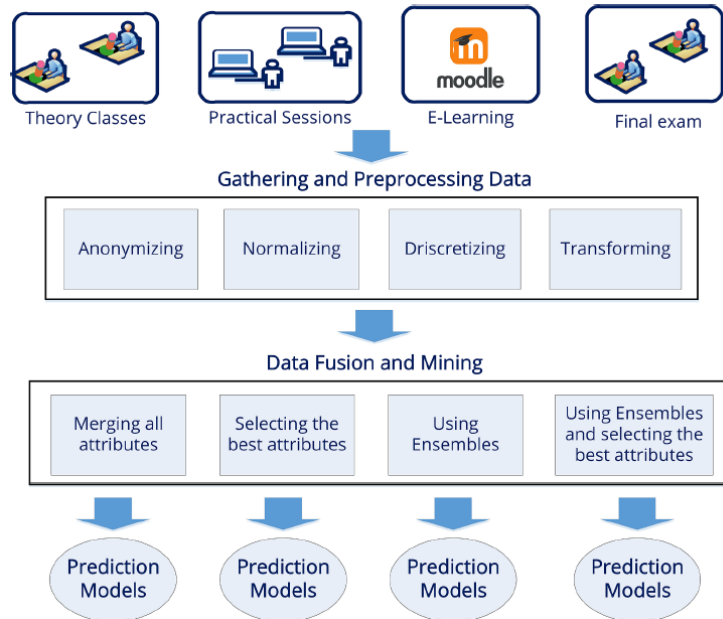


Ilustración 4. Metodología de minería y fusión de datos propuesta para predecir el rendimiento de los estudiantes a partir de múltiples fuentes de datos.

Fuente: (Chango et al., 2021)

### 3.3 FILTROS PARA DATOS DE SALIDA DEL SENSOR

#### 3.3.1 FILTRO DE KALMAN

##### Filtro de Kalman lineal o estándar

En los campos de estadística y teoría de control, el filtrado de Kalman (o estimación cuadrática lineal) es un algoritmo que utiliza mediciones observadas (generalmente discretas) a través del tiempo, que tienen ruido estadístico y otras impurezas, y produce estimaciones de variables desconocidas que tienden a ser más exactos que los basados en una sola medición, mediante la estimación de una distribución de probabilidad conjunta sobre las variables para cada período de tiempo (Hun et al., 2016). Hay que tener en cuenta las limitaciones de los modelos lineales en aplicaciones de ingeniería. Menciona Sabet et al. (2017) que “debido a las simplificaciones del modelado, los modelos matemáticos disponibles adolecen de la incertidumbre de sus parámetros y generalmente necesitan una fase de identificación para mejorar la precisión del modelado.”

El filtro Kalman fue desarrollado por el ingeniero eléctrico y matemático Rudolf Emil Kalman en 1960. Desde entonces, se desarrollaron numerosas aplicaciones con la implementación del filtro de Kalman, entre ellos en los campos de navegación y seguimiento de objetos por visión artificial. El filtro de Kalman consta de dos procesos separados: el proceso de predicción y el proceso de medición, que funcionan de forma recursiva.

La literatura de los usos del filtro de Kalman sigue siendo extensa. En su trabajo de sistemas de transporte inteligentes, menciona Al Malki et al. (2021) que “un RSSI (indicador de fuerza de señal recibida) fue usado para estimar la ubicación del nodo por falta de información GPS. El vehículo dentro del alcance recibía señales para ser localizados con mayor precisión de la que era posible dada la sensibilidad del receptor.”

Ambos procesos se modelan mediante grupos de ecuaciones en el modelo de espacio de estados para lograr resultados de estimación óptimos.

Los filtros de Kalman combinan dos fuentes de información, los estados pronosticados por medios matemáticos y las mediciones imperfectas (i.e. con ruido) para producir estimaciones óptimas e insesgadas de los estados del sistema. Menciona G. X. Liu et al. (2018) que “a fusión de información de múltiples fuentes es un área de investigación en rápida evolución y requiere conocimiento interdisciplinario en teoría de control, procesamiento de señales, inteligencia artificial, probabilidad y estadística, etc.”. Menciona Hun et al. (2016) que el modelo de predicción del filtro de Kalman multidimensional y el modelo de medición de un sistema en tiempo real, expresados en forma discreta, son los siguientes:

$$x_k = \phi x_{k-1} + B u_k + w_k$$

Ecuación 1.

$$z_k = H x_k + v_k$$

Ecuación 2.

Donde  $x_k$  es la salida prevista,  $z_k$  es la salida de medición,  $\phi$  denota la matriz de transición de estado,  $B$  es la matriz de entrada de control y  $u$  es la matriz de entrada de control opcional.  $H$  es la matriz de transformación de medición, mientras que  $w_k$  y  $v_k$  son la matriz de ruido del modelo y la matriz de ruido de medición, respectivamente.

Con estructura similar a la ecuación 1, la expresión del estado de predicción se describe a continuación:

$$\tilde{x}_{\bar{k}} = \phi \tilde{x}_{k-1} + Bu$$

Ecuación 3

donde  $\tilde{x}_{\bar{k}}$  se define como el estado a priori estimado en el instante discreto k, y  $\tilde{x}_k$  se define como el estado a posteriori ilustrado en el instante discreto k dada la medida  $z_k$ . Hay que tener en cuenta que, a partir de la ecuación (3), el estado a priori  $\tilde{x}_{\bar{k}}$  puede elaborarse como un estado hipotético predicado a partir de las ecuaciones de transición de estados del sistema, mientras que el estado a posteriori  $\tilde{x}_k$  puede elaborarse como el estado medido obtenido por la observación o salidas sensoriales del sistema. Al hacer que  $x_k$  sea el valor real de la medición del estado, el error de predicción a priori  $e_{\bar{k}}$  y el error de estimación a posteriori  $e_k$  se pueden expresar como:

$$e_{\bar{k}} = x_k - \tilde{x}_{\bar{k}}$$

Ecuación 4.

$$e_k = x_k - \tilde{x}_k$$

Ecuación 5.

De la ecuación (4), la covarianza del error de predicción a priori se puede expresar como:

$$P_{\bar{k}} = E(e_{\bar{k}}e_{\bar{k}}^T) = E[(x_k - \tilde{x}_{\bar{k}})(x_k - \tilde{x}_{\bar{k}})^T]$$

Ecuación 6

Usando todas las ecuaciones establecidas anteriormente, la covarianza del error de predicción a priori puede también expresarse como:

$$P_{\bar{k}} = \phi E[(x_{k-1} - \tilde{x}_{k-1})(x_{k-1} - \tilde{x}_{k-1})^T] \phi^T + E[(w_k w_k^T)] = \phi P_{k-1} \phi^T + Q_k$$

Ecuación 7.

La ecuación (7) dio un paso importante en el proceso de predicción del algoritmo de filtrado de Kalman al obtener la covarianza del error de predicción a priori utilizando la matriz de transición de estado del sistema  $\phi$ , la covarianza del error de medición a posteriori de la estimación anterior  $P_{k-1}$  y la covarianza del ruido del proceso  $Q_k = E[(w_k w_k^T)]$ .



La siguiente etapa del algoritmo de Kalman es el proceso de medición (con señales reales del sensor). La ecuación (2) describe la ecuación de observación, o la ecuación de medición real, del sistema. La salida de medición  $z_k$  se obtiene normalmente de los sensores o dispositivos de medición del sistema. A partir de aquí, es posible expresar la medida a posteriori de la siguiente manera:

$$\tilde{x}_k = \tilde{x}_{\bar{k}} + K_k(z_k - H\tilde{x}_{\bar{k}})$$

Ecuación 8.

$$\tilde{x}_k = \tilde{x}_{\bar{k}} + K_k(Hx_k + v_k - H\tilde{x}_{\bar{k}})$$

Ecuación 9.

Dada la covarianza del error de medición a posteriori, con referencia a la Ecuación (5):

$$P_k = E[e_k e_k^T] = E[(x_k - \tilde{x}_{\bar{k}})(x_k - \tilde{x}_{\bar{k}})^T]$$

Ecuación 10.

Se organizan todas las ecuaciones anteriores en la versión simplificada de la ecuación de actualización de la covarianza de error expresada en términos de la ganancia de Kalman óptima y la covarianza de error de predicción a priori obtenida de la Ecuación (7), expresada en la siguiente ecuación:

$$P_k = P_{\bar{k}} - P_{\bar{k}}H^T(HP_{\bar{k}}H^T + R_k)^{-1}HP_{\bar{k}} = (I - K_kH)P_{\bar{k}}$$

Ecuación 11.

En resumen, el algoritmo de filtrado de Kalman se puede resumir y se muestra en la Figura 1. El proceso de predicción, como se muestra en la Figura 1, cubre la predicción del estado a priori y la covarianza de error a priori. El proceso de medición, por el contrario, cubre el cálculo de la ganancia de Kalman óptima, actualizando el estado de estimación a posteriori y la covarianza del error a posteriori. Ambos procesos se ejecutan de forma recursiva, formando el conocido algoritmo de filtrado de Kalman.

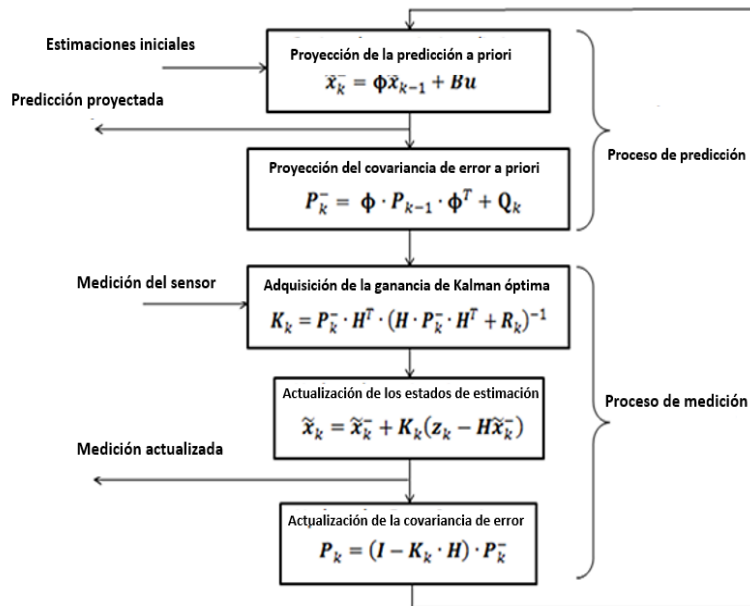


Ilustración 5. Flujograma del filtro de Kalman y sus ecuaciones para el cálculo de sus parámetros.

Fuente: (Hun et al., 2016)

### 3.3.2 FILTRO DE KALMAN EXTENDIDO

El filtro de Kalman extendido (EKF) es una versión no lineal, no idealizada del filtro de Kalman. El EKF se linealiza (principalmente a partir de series de Taylor multivariables) sobre una estimación de la media y la covarianza actuales. En los modelos de transición bien definidos, el EKF es estándar de facto en la teoría de la estimación de estado no lineal (la mayoría de los modelos de sistemas en los campos de la ingeniería son no lineales), sistemas de navegación y GPS. Los sistemas no lineales tienen un ruido blanco autónomo tanto en el sistema como en el modelo de estimación.

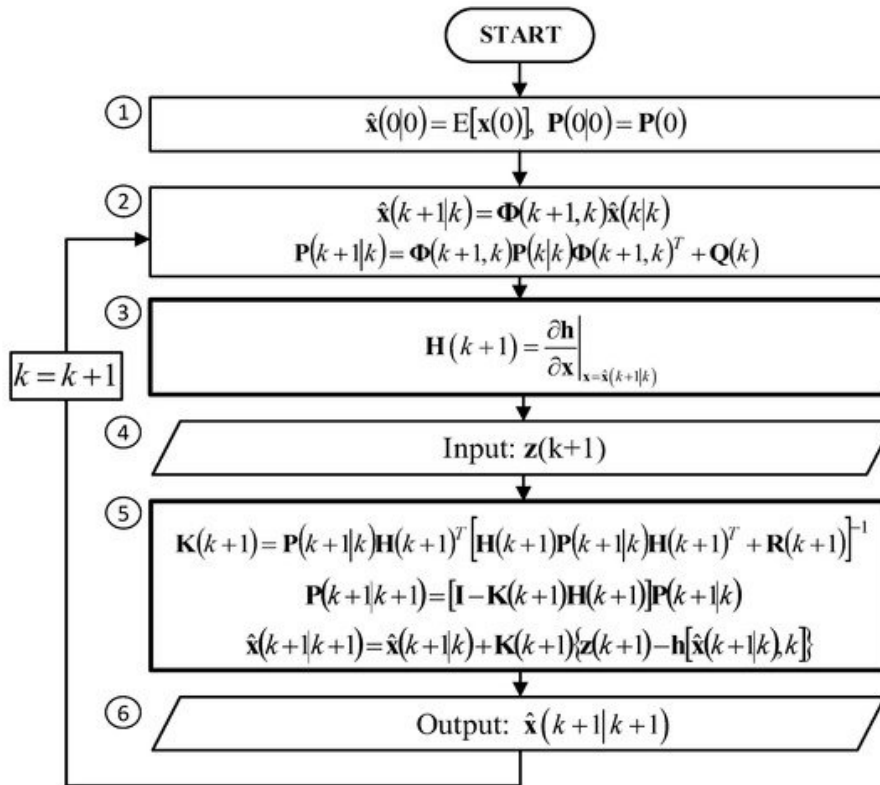


Ilustración 6. Flujograma del filtro de Kalman extendido y las ecuaciones para el cálculo de sus parámetros.

Fuente: (Kaniewski, 2020)

Ha recibido una atención relevante con la creciente demanda de localización robótica. Por lo tanto, el EKF atrae más atención que el Filtro de Kalman (KF). En su algoritmo de localización, por ejemplo, menciona Ullah et al. (2020) que "utilizó un algoritmo de localización, con objetivo de lograr un alto nivel de precisión y una cobertura más amplia, basado en EKF mediante computación de borde, y se utilizó un robot móvil para actualizar su ubicación con respecto al hito. Si el sistema se modela considerando no linealidades, entonces la función de transición de estado se vuelve no lineal. Aquí, el ruido ingresa al sistema de manera lineal, pero puede haber lugares donde el ruido no es aditivo. En un sistema general, la transición de estado o la función de medición, o ambas, pueden ser no lineales. Para todos estos casos, se aplicará un estimador de estado no lineal como el filtro de Kalman extendido.

En el siguiente artículo, se propone un filtro de Kalman extendido tolerante a fallas adaptativo. Cao et al. (2021) propuso un algoritmo de filtro de Kalman extendido a prueba de fallas que elimina los datos de interferencia roja a tiempo real, y posteriormente hace una estimación de

actitud, estas siendo a menudo es inexactas durante el movimiento altamente dinámico. Ha recibido una atención relevante con la creciente demanda de localización robótica. Por lo tanto, el EKF atrae más atención que el Filtro de Kalman (KF). En su algoritmo de localización, por ejemplo, menciona Ullah et al. (2020) que "utilizó un algoritmo de localización, con objetivo de lograr un alto nivel de precisión y una cobertura más amplia, basado en EKF mediante computación de borde, y se utilizó un robot móvil para actualizar su ubicación con respecto al hito. Si el sistema se modela considerando no linealidades, entonces la función de transición de estado se vuelve no lineal. Aquí, el ruido ingresa al sistema de manera lineal, pero puede haber lugares donde el ruido no es aditivo. En un sistema general, la transición de estado o la función de medición, o ambas, pueden ser no lineales. Para todos estos casos, se aplicará un estimador de estado no lineal como el filtro de Kalman extendido.

$$x_k = f(x_{k-1}, u_k) + w_k$$

Ecuación 12.

$$y_k = g(u_k) + v_k$$

Ecuación 13.

La matriz jacobiana se utiliza para analizar la estabilidad de señales pequeñas del sistema. Esta matriz jacobiana se deriva de la matriz de estado y los elementos de esta matriz jacobiana se utilizarán para realizar el resultado de sensibilidad.

$$F = \frac{\delta f}{\delta x} [\hat{x}_{k-1}, u_k]$$

Ecuación 14.

$$G = \frac{\delta g}{\delta x} [\hat{x}_k]$$

Ecuación 15.

El EKF linealiza una estimación de la media y la covarianza actuales, como se aprecia en las ecuaciones a continuación:

$$\Delta x_k \approx F \Delta x_{k-1} + w_k$$

Ecuación 16.

$$\Delta y_k \approx G \Delta x_k + v_k$$

Ecuación 17.

(Wan & Van Der Merwe, 2000)

El filtro de Kalman extendido tiene los siguientes inconvenientes:

1. Puede ser difícil calcular analíticamente a los jacobianos debido a las complicadas derivadas.
2. Puede haber un alto costo computacional para calcularlos numéricamente.
3. No se puede aplicar a sistemas con modelos discontinuos, ya que ese tipo de sistema no se puede diferenciar y no habría jacobianos.
4. Los sistemas altamente no lineales no se pueden representar con precisión mediante la linealización porque la covarianza se propaga a través de la linealización del modelo no lineal subyacente.

Los inconvenientes señalados han llevado al desarrollo de otra herramienta matemática basada en el filtro de Kalman, el filtro de Kalman *unscented* (inoloro), o UKF.

### 3.3.3 FILTRO DE KALMAN *UNSCENTED*

Cuando los modelos de observación y transición de estado son excesivamente no lineales, el filtro de Kalman extendido puede dar un rendimiento no deseado. El filtro de Kalman *unscented* (UKF) utiliza una técnica de muestreo determinista conocida como transformación *unscented* (UT), que se aproxima a una distribución de probabilidad utilizando una pequeña cantidad de puntos de prueba cuidadosamente elegidos, para elegir un conjunto mínimo de puntos de muestra alrededor de la media. Posteriormente, los puntos de sigma se extienden a través de las funciones no lineales, a partir de las cuales se logra calcular un nuevo valor de media y covarianza. La salida del filtro depende de variables como los puntos sigma a utilizar, así como de la metodología de los cálculos estadísticos y elección de filtros.

Tabla 1. Comparación de la naturaleza del modelo matemático, distribución asumida y el costo de implementación del filtro de Kalman y sus mejoras.

Estimador de estado	Modelo	Distribución asumida	Costo computacional
Filtro de Kalman	Lineal	Gaussiana	Bajo
EKF	Localmente lineal	Gaussiana	Bajo (si las jacobianas se necesitan computar analíticamente); medio (si las jacobianas se computan numéricamente)
UKF	No lineal	Gaussiana	Medio

Fuente: (Astroza et al., 2019)

Menciona Bonyan Khamseh et al. (2019) que:

En UKF, se utiliza una técnica de muestreo determinista, es decir, transformada *unscented* para generar un conjunto mínimo de puntos de muestra a partir de la media y la covarianza a priori del estado. Estos puntos muestreados, también conocidos como puntos sigma, se propagan a través del modelo de proceso no lineal y la media posterior y la covarianza se recuperan numéricamente. Los resultados de la investigación también destacan las ventajas de UKF sobre EKF en algunas aplicaciones, de la siguiente manera: menos problemas de implementación y costos computacionales, más precisión, estabilidad, convergencia y robustez y facilidad de ajuste. (pág. 449)

El siguiente artículo estudia la optimización de la transferencia de energía de las baterías de los automóviles. Menciona Ben Sassi et al. (2020): "Este documento propone una teoría de múltiples innovaciones para mejorar la precisión de estimación del UKF. Al expandir un único valor de voltaje de innovación a múltiples innovaciones que consisten en los valores anteriores y actuales de la salida de la batería, la precisión del UKF es drásticamente mejorado."

Uno de los usos más novedosos de UKF incluye el seguimiento de objetivos bajo el agua mediante mediciones de sonda pasiva. En su artículo, menciona Ravi Kumar (2021) que:

"Se desarrolla un estimador novedoso llamado UKF híbrido (HUKF) para abordar el problema del seguimiento pasivo de objetivos en escenarios submarinos utilizando mediciones solo de rumbo (capturadas por una matriz remolcada). Como su nombre indica, el algoritmo es un híbrido obtenido mediante la combinación de tres algoritmos existentes: UKF, técnica de integración y mecanismo de preprocesamiento para producir un rendimiento mucho mejor que cualquiera de ellos individualmente. La reducción de ruido del sensor en las mediciones espaciales causada por la técnica de integración junto

con la reducción de ruido en las mediciones de tiempo causada por el mecanismo de preprocesamiento en sucesivas iteraciones de tiempo contribuye acumulativamente a la mejora del rendimiento.” (pág. 1)

### 3.4 MODELOS MATEMÁTICOS PARA DESCRIPCIÓN ESPACIAL

#### 3.4.1 ÁLGEBRA LINEAL

Una matriz de rotación describe la transformación que sufre una entidad en el espacio euclidiano cuando cambia su orientación. Una función de rotación básica (o rotación elemental) es una rotación alrededor de uno de los ejes de un sistema de coordenadas. Por ejemplo, la rotación sobre el eje x se puede describir de la siguiente manera:

$$R_x(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

Ecuación 18.

A partir de esta matriz base, así como también las matrices bases aplicadas en los otros ejes (y y z), se pueden obtener matrices de rotación para la transformación con la cual se desea imponer a una función en el espacio. Por ejemplo, el producto tiene el siguiente aspecto, y cada columna representa la guiñada, el cabeceo y el balanceo, respectivamente.

$$R = \begin{pmatrix} \cos(\alpha) * \cos(\beta) & \cos(\alpha) * \sin(\beta) * \sin(\gamma) - \sin(\alpha) * \cos(\gamma) & \cos(\alpha) * \sin(\beta) * \cos(\gamma) + \sin(\alpha) * \sin(\gamma) \\ \sin(\alpha) * \sin(\beta) & \sin(\alpha) * \sin(\beta) * \sin(\gamma) + \cos(\alpha) * \cos(\gamma) & \sin(\alpha) * \sin(\beta) * \cos(\gamma) - \cos(\alpha) * \sin(\gamma) \\ -\sin(\beta) & \cos(\beta) * \sin(\gamma) & \cos(\beta) * \cos(\gamma) \end{pmatrix}$$

Ecuación 19.

Las matrices de rotación describen rotaciones de un cuerpo tridimensional en el espacio euclidiano sobre el origen p(0,0,0). Esto se debe a que la multiplicación de matrices no tiene efecto sobre las coordenadas del origen (también conocido como vector cero). Las matrices de rotación proporcionan una descripción algebraica de la transformación rotacional que sufre el cuerpo en el espacio, herramienta poderosa en los gráficos tridimensionales por computadora, como las animaciones, los datos sensoriales de tres parámetros. Otro parámetro inherente a toda matriz es la determinante. Geométricamente hablando, la determinante de una matriz es el volumen con signo de un cubo unitario después de aplicar la transformación definida por la matriz. Un determinante negativo significa que la orientación del espacio se invierte. Si se asignan dimensiones a sus dedos y si después de la transformación esas asignaciones aún se mantienen, significa que la orientación del espacio no se cambia y que el determinante es positivo. Si después de la transformación la asignación se mantiene en

otra mano, entonces la orientación espacial se invierte y significa que el determinante es negativo (Meckes & Meckes, 2018).

### 3.4.2 ÁNGULOS DE EULER

Los ángulos de Euler son tres ángulos ortogonales que describen la orientación de un cuerpo rígido con respecto a un sistema de coordenadas fijo, generalmente el origen. En modelación tridimensional se usa comúnmente la convención (z; y; x) en la cual primero gira el objeto respecto al eje z, luego al eje y y finalmente al eje x. Se pueden usar otras convenciones dependiendo de la aplicación y la estandarización de esta.

Un problema que se presenta en este modelo matemático es el problema del bloqueo del cardán, frecuentado por desarrolladores de programas informáticos que tratan con modelado tridimensional y en sistemas de navegación. Ocurre cuando se desea aislar un ángulo de Euler de la matriz de rotación, operación que implica la pérdida de un grado de libertad cuando dos ejes se estacionan en paralelo por un instante en el tiempo, trabando el sistema tridimensional en uno bidimensional.

Aunque se describe popularmente como candado, esta descripción no es del todo precisa, ya que no se restringe ningún cardán. Los tres cardanes aún pueden girar libremente en sus tres grados de libertad. El problema surge cuando no hay ningún cardán disponible para acomodar la rotación alrededor del eje.

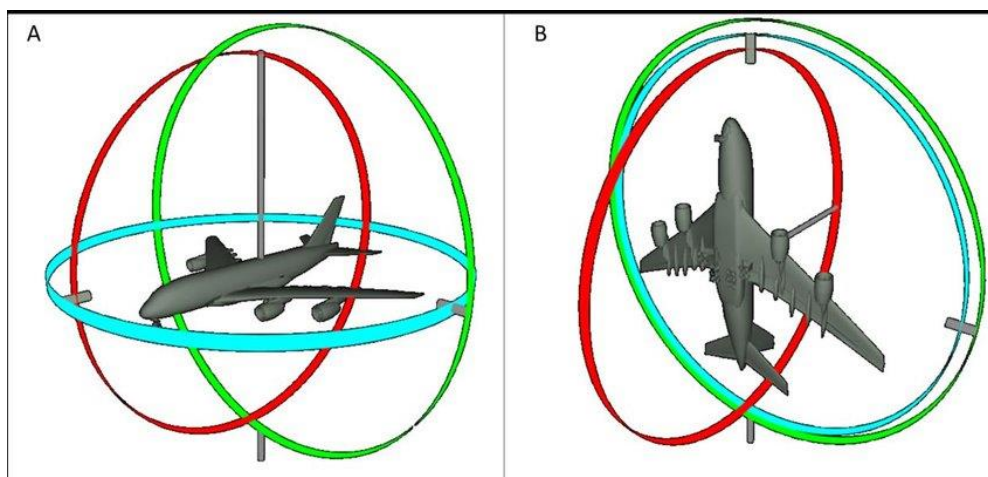


Ilustración 7. Problema de bloqueo del cardán para ángulos de Euler (A: sin bloqueo del cardán, B: los ángulos de guiñada y balanceo están bloqueados).

Fuente: (Tannous, 2018)



### 3.4.3 CUATERNIONES

Los cuaterniones unitarios, también llamados versores, describen la orientación de un cuerpo rígido en el espacio y sus acciones de rotación. Estos contienen la información necesaria para describir la rotación del ángulo alrededor de un eje arbitrario. Los cuaterniones tienen extensas aplicaciones en animación computarizada, robótica y navegación, siendo la visión robótica y el rastreo espacial de la máquina sus mayores exponentes (Bayro-Corrochano et al., 2018). Un cuaternión de unidad usa una representación de 4 dimensiones de la orientación de la siguiente manera:

$$q = (q_0 \ q_1 \ q_2 \ q_3)^T = \begin{pmatrix} q_0 \\ q_v \end{pmatrix}; q \in R^4$$

Ecuación 20.

Una rotación se puede definir usando cuaterniones de unidad de la siguiente manera:

$$\bar{x}^u = q^{uv} \times \bar{x}^v \times q^{(uv)c}$$

Ecuación 21.

donde  $q^{(uv)c} = q^{uv}$  denota el conjugado de cuaternión, definido de la siguiente manera:

$$q^c = (q_0 \ -q_v^T)$$

Ecuación 22.

y  $\bar{x}^v$  denota la representación de cuaternión de  $x^v$  como

$$\bar{x}^v = (0 \ (x^v)^T)^T$$

Ecuación 23.

La notación  $\times$  denota la multiplicación de cuaterniones dada por

$$p \times q = \frac{p_0 q_0 - p_v \cdot q_v}{p_0 q_v + q_0 p_0 + p_v * q_v} = p^L q = q^R p$$

Ecuación 24.

(Maclachlan & Reid, 2003)

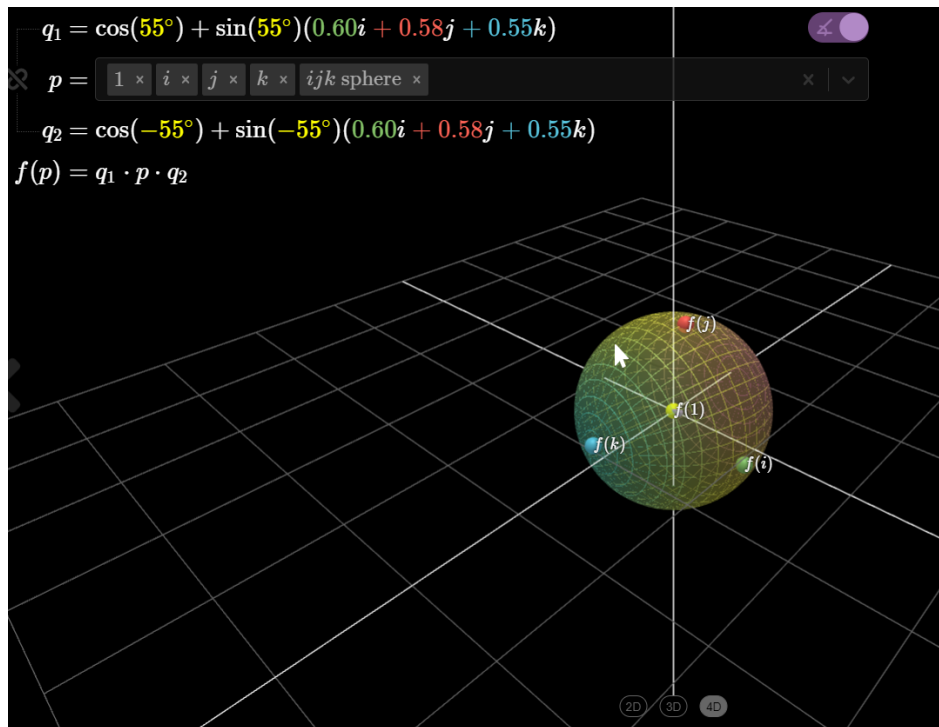


Ilustración 8. Visualizando cuaterniones, una serie de videos explorables de 3blue1brown.

Más allá de su uso como descriptor de rotación, se utiliza esta herramienta matemática en la robótica hasta en los cálculos de redes neuronales. Las redes neuronales de cuaterniones son modelos de neuronas y del procesamiento de datos en el cerebro humano, pero cuyos cálculos en los nodos se basan en cuaterniones. En el artículo posteriormente referido, se exploran técnicas geométricas para robótica. Menciona Bayro-Corrochano et al. (2018):

“La interpolación puede manejar la interpolación y dilatación en 3D de puntos, líneas, planos, círculos y esferas. Con este procedimiento, interpolamos trayectorias del instrumento quirúrgico. Usando cuaterniones, formulamos la red neuronal de picos de cuaterniones para el control. Esta nueva estructura de red neuronal se basa en *Spike Neural Networks* y se desarrolló utilizando el álgebra de cuaterniones. El algoritmo de entrenamiento de valor real se amplió para que pudiera hacer ajustes de los pesos de acuerdo con las propiedades y el producto del álgebra de cuaterniones.” (pág. 1)

## **4 METODOLOGÍA**

### **4.1 ENFOQUE**

Para este proyecto de investigación, se empleará un enfoque experimental, lo que implica recopilar datos cuantitativos y realizar análisis estadísticos sobre ellos durante la investigación. Este enfoque generalmente es un método cuantitativo, ya que se hacen conclusiones a partir de valores numéricos de las variables. Sin embargo, en esta investigación se sacarán conclusiones también tomando en cuenta variables cualitativas como el ruido, ya que los parámetros de cuantificación del ruido no es lo suficientemente sencillo y directo como para estar al alcance en esta investigación.

### **4.2 VARIABLES DE INVESTIGACIÓN**

En la investigación experimental, el experimentador manipula o cambia la variable independiente para medir el efecto de este cambio en la variable dependiente. Se han definido las siguientes variables para esta investigación.

#### **4.2.1 VARIABLES INDEPENDIENTES**

Las variables independientes establecidas son el número de sensores, así como su tipo y modelo; y el filtro o algoritmo a utilizar fusionar las mediciones de los sensores y dar salida a una señal más favorable.

#### **4.2.2 VARIABLES DEPENDIENTES**

Las variables dependientes son todas aquellas afectadas por la elección de las variables independientes. El número y la calidad de los sensores a fusionar, así como la programación del microcontrolador, afectan directamente los parámetros con que se evaluará la efectividad del algoritmo. Entre estos se encuentran:

Sesgos. El sesgo es un desplazamiento constante del valor de salida del valor de entrada. Existen dos tipos: estabilidad de sesgo de activación y estabilidad de sesgo. En los sensores IMU, la lectura cero iniciales del giroscopio provocará una desviación con el tiempo debido a la integración de imperfecciones inherentes y ruido dentro del dispositivo. Esto se controlará con las variables independientes y se calificará cualitativamente si el sesgo es aceptable para

la operación del sensor final. Sin embargo, Un giróscopo con sesgo más bajo dará lugar a menos errores en las estimaciones de posición para una unidad de medida inercial, por lo que se puede indirectamente medir este parámetro de forma cuantitativa.

Ruido. La calidad de una señal a menudo se expresa cuantitativamente como la relación señal/ruido, que es la relación entre la verdadera amplitud de la señal subyacente (por ejemplo, la amplitud media o la altura máxima) y la desviación estándar del ruido.

### **4.3 TÉCNICAS E INSTRUMENTOS APLICADOS**

Se hizo una simulación de hardware por medio de Proteus de Arduino Uno, microcontrolador de mi elección, conectándolo por un puerto virtual a MATLAB, programa donde se estará desarrollando la programación. En MATLAB, se instalaron dos adiciones esenciales, *Sensor Fusion and Tracking Toolbox™* (caja de herramientas de seguimiento y fusión de sensores), que incluye algoritmos base incorporados y herramientas didácticas para diseñar, simular y probar sistemas de control que usan fusión de datos para sinérgicamente procesar la información de estos; y conjuntamente *Simulink® Support Package for Arduino®* (paquete de soporte Simulink para hardware de Arduino) y *MATLAB® Support Package for Arduino® Hardware* (paquete de soporte MATLAB para Arduino). El primero permite programar por medio de un interfaz gráfico intuitivo los algoritmos que se ejecutan en la placa; el segundo paquete permite la conexión exitosa entre la placa Arduino Uno y MATLAB.

El programa Proteus permite la simulación del hardware, así como también el diseño de la placa que se imprimiría posteriormente.

### **4.4 MATERIALES**

Sensores IMU (cantidad determinada en la metodología), placa Arduino Uno y todos los componentes para crear la conexión y comunicación adecuada en la PCB, como los resistores del bus I2C y multiplexores.

### **4.5 POBLACIÓN Y MUESTRA**

Para poder lograr los objetivos y sacar conclusiones, es menester hacer comparación con los datos brutos del singular sensor IMU con la librería genérica de *Sensor Fusion* y con los datos filtrados de la placa a desarrollar. Como se definió en la sección de variables, estos datos serían

validados por medio de herramientas estadísticas que rigen el control de calidad de los sensores.

#### **4.6 METODOLOGÍA DE ESTUDIO**

Para el desarrollo de la placa se ha decidido regirse por la metodología jerárquica de Hehenberger, caracterizada por la ausencia de ciclos y presencia de jerarquías. Los elementos externos, como la capacidad sensorial de la placa, son características que existen en las expectativas del cliente. Los elementos internos, como el algoritmo a ser implementado, son los que el desarrollador implementa en el producto mecatrónico para cumplir con las expectativas del cliente.

Mencionan Hehenberger & Zeman (2004) que:

“Las tareas de diseño específicas del dominio no necesitan integrarse como un todo en el nivel mecatrónico de la tarea de diseño. En consecuencia, los modelos deben cubrir las diferentes vistas de un sistema, así como los diferentes grados de detalle que lleva de una jerarquía de modelos a una jerarquía de parámetros de diseño. Las vistas específicas del objeto representan los fenómenos/efectos relevantes de interés como la geometría, la dinámica, la estabilidad o el material y se utilizan como interfaces para el ingeniero de diseño durante su trabajo. Su descripción por cantidades significativas se utiliza para analizar diferentes conceptos de diseño mecatrónico.” (pág. 1)

En nuestro contexto mecatrónico, las disciplinas que se tomarán en consideración son la ingeniería de control y la electrónica. Los modelos consisten en las ecuaciones que describen el estado de espacios que describen la orientación de la placa, así como también el diagrama de bloques donde se transformarán los estados de espacio por medio de filtros. La jerarquía de los parámetros de diseño se desarrolla para cada dominio individualmente. En la metodología jerárquica, se definen los parámetros de diseño y los requerimientos funcionales. Es necesario cambiar entre el enfoque funcional y el físico durante el proceso de desarrollo del producto, ya que hay que probar cíclicamente el prototipo o modelo hasta que los requerimientos o el alcance sea satisfecho junto con sus parámetros de diseño. Los sensores IMU MEMS son conocidos por su versatilidad, bajo costo y bajo peso; sin embargo, si no existe la posibilidad de controlarlos y emplearlos efectivamente pierden su funcionalidad. Por este hecho, es posible separar el módulo electrónico con la ingeniería de control.

Una manera directa y elegante de construir una relación jerárquica es por medio de un modelo analítico simple. El cliente requiere cierta reducción de errores de escala en el sensor final, así como menor tiempo de calibración, ruido e inestabilidad por integración del giroscopio. Las limitaciones estrictas y los algoritmos de control más sofisticados requieren una visión más detallada de los requisitos funcionales que un sistema mecatrónico más directo, como la de controlar un motor. El cliente puede que quiera utilizar la placa para un robot autónomo en una mina donde haya mucha perturbación por imanes fuertes. Por ende, las funciones del magnetómetro y su función como sensor individual deben adaptarse a sus requerimientos. Los algoritmos de fusión utilizan lecturas de magnetómetro que deben compensar las distorsiones magnéticas, como la distorsión de hierro duro. Las distorsiones de hierro duro son producidas por materiales que crean un campo magnético, lo que da como resultado un cambio de origen en la superficie de respuesta. Estas distorsiones se pueden corregir restando los valores de corrección de las lecturas del magnetómetro para cada eje.

La relación entre la velocidad angular y posición existen en un estado de espacios idealizado. Por la naturaleza de los sensores, que recopilan datos sensoriales según una frecuencia dada, este espacio estado es de naturaleza discreta.

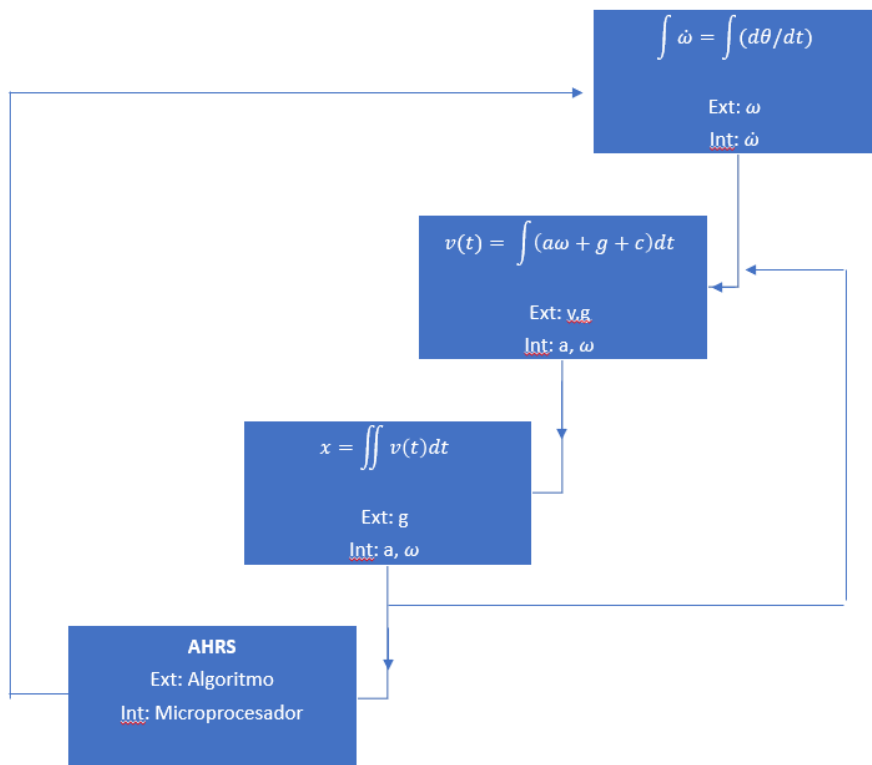


Ilustración 9. Estructura jerárquica de una unidad inercial multisensor modelada linealmente controlada por filtro. Cada bloque describe el estado de cada eje (i.e., x,y,z).

#### 4.7 METODOLOGÍA DE VALIDACIÓN

Para validar la metodología empleada, hay que probar la salida bruta de un sensor IMU individual en la matriz de sensores. Se medirán en tiempo real la velocidad angular y orientación y se evaluarán las variables dependientes y los requerimientos funcionales por los medios cualitativos y cuantitativos mencionados anteriormente. Se identificarán imperfecciones en el proceso de mediciones, i.e., sesgos y ruido. Posteriormente se hará lo mismo para la salida de la placa con el algoritmo final. Se compararán los dos resultados. La validación se hará por una comparativa de los datos obtenidos con datos estadísticos establecidos por otros indicadores científicos. Existen varias herramientas estadísticas que también pueden servir como algoritmos de filtrado para la fusión de sensores, como el filtro de Kalman, que utiliza una serie de medidas observadas a lo largo del tiempo, que contienen ruido y otras inexactitudes, y produce estimaciones de variables desconocidas. Con esta herramienta, podemos predecir con precisión el estado futuro de la actitud y orientación de la PCB.

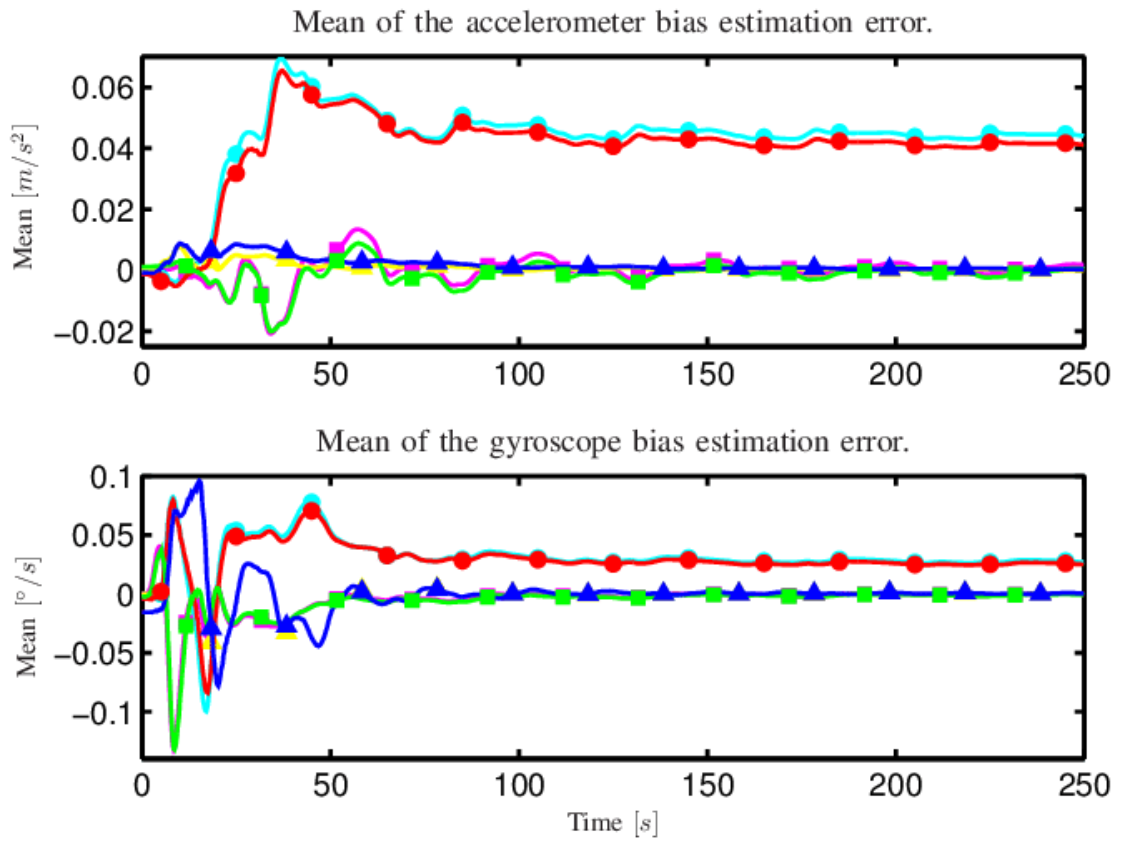


Ilustración 10. Media del error en las estimaciones de sesgo del acelerómetro y giroscopio para un error de sincronización de tiempo de 100 ms. Simulación MC: línea roja del eje X (adelante) con círculos; Eje Y (de lado) - línea verde con cuadrados.

Fuente: (Skog & Handel, 2011)



## 4.8 CRONOGRAMA DE ACTIVIDADES

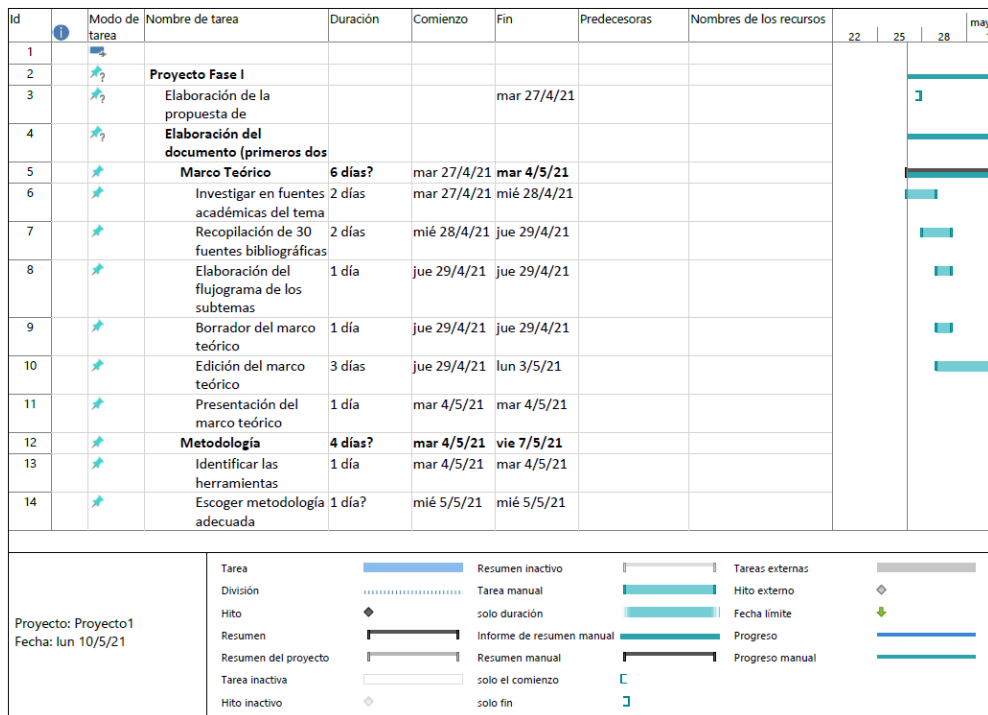


Ilustración 11. Cronograma, primera captura de pantalla.



Ilustración 12. Cronograma, segunda captura de pantalla.

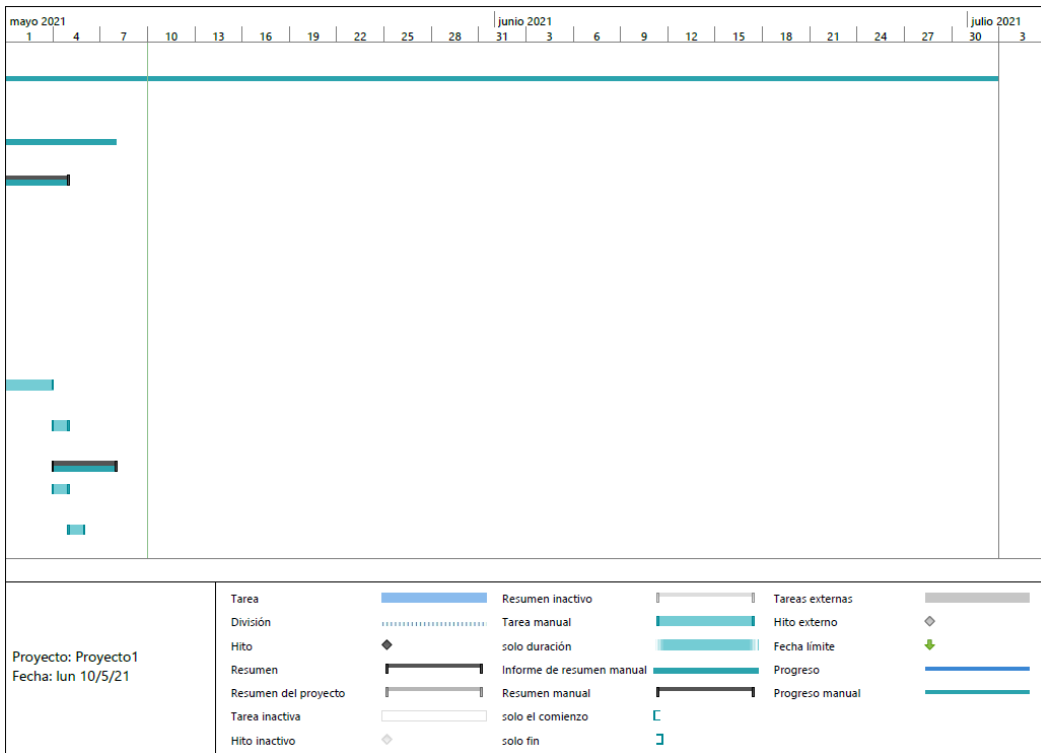


Ilustración 13. Cronograma, tercera captura de pantalla.

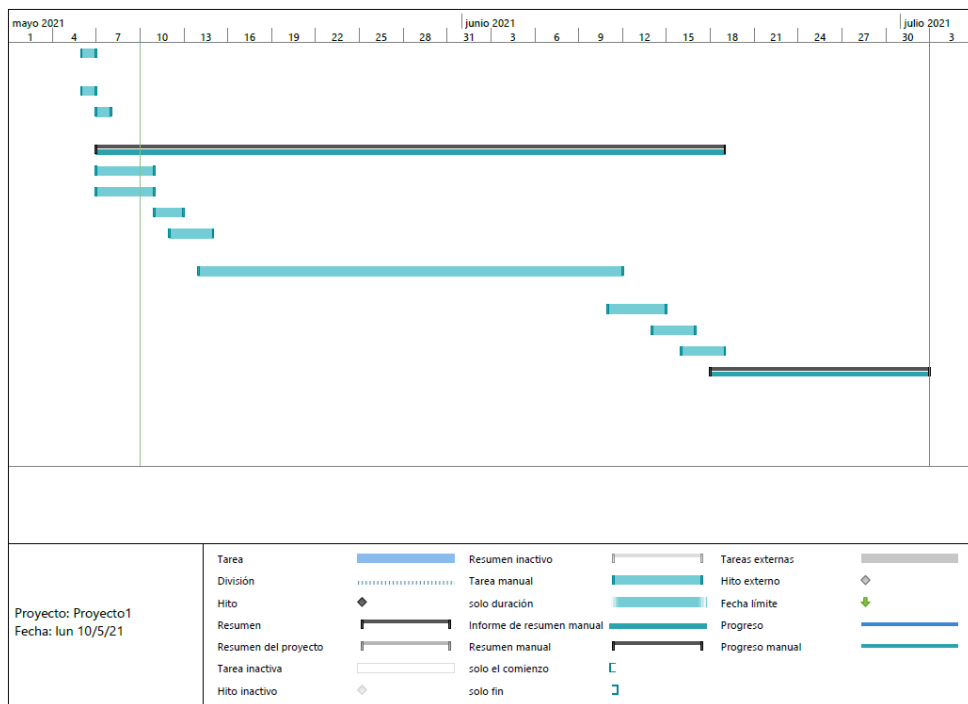


Ilustración 14. Cronograma, cuarta captura de pantalla.

## 5 ANÁLISIS Y RESULTADOS

En el presente capítulo se desarrolla por completo el algoritmo de filtrado y fusión de las salidas de la matriz de sensores IMU, así como el diseño de la PCB y el análisis de gráficas trazadas por la salida final de la placa. El objetivo de este proyecto es crear un prototipo de una unidad de medida inercial que proporcione varios minutos de navegación confiable en ausencia de otras señales de referencia (por ejemplo, GPS), así como mejoras en las cualidades del sensor.

En el marco teórico se presentaron diversos algoritmos que sirven para la fusión y el filtrado de las salidas de los giroscopios, magnetómetros y acelerómetros, que permiten dar como salida una estimación aceptable de la orientación del sensor.

### 5.1 DISEÑO DE LA PLACA

Las pruebas que dieron los resultados fueron hechas con una placa de prueba sin soldar con distancias semejantes a las de la placa que se muestra a continuación. En la ilustración 15 se muestra el diseño de la PCB y su renderizado. Los cuatro sensores tienen una configuración de matriz cuadrada 2 x 2, para que la corrección del centro de masa sea más fácil, además de que esta configuración hace la placa más compacta y fácil de implementar en un proyecto como dron o robot autónomo. La placa consta de un multiplexor TCA9548A, que actúa en forma de buffer intermedio con diferentes salidas y se controla también a su vez con el bus I2C del Arduino Uno. Los cuatro sensores pueden conectarse a cualquiera de los ocho buses con SDA y SCL disponibles en el multiplexor. La placa diseñada los tiene en los buses 2, 4, 5 y 7 por motivos de proximidad de los sensores a los pines del TCA9548A. Los cuatro pines del Arduino Uno que se conectan a la placa son el EMF de 5 voltios, tierra, pin de datos en serie (SDA) y el pin de reloj en serie (SCL).

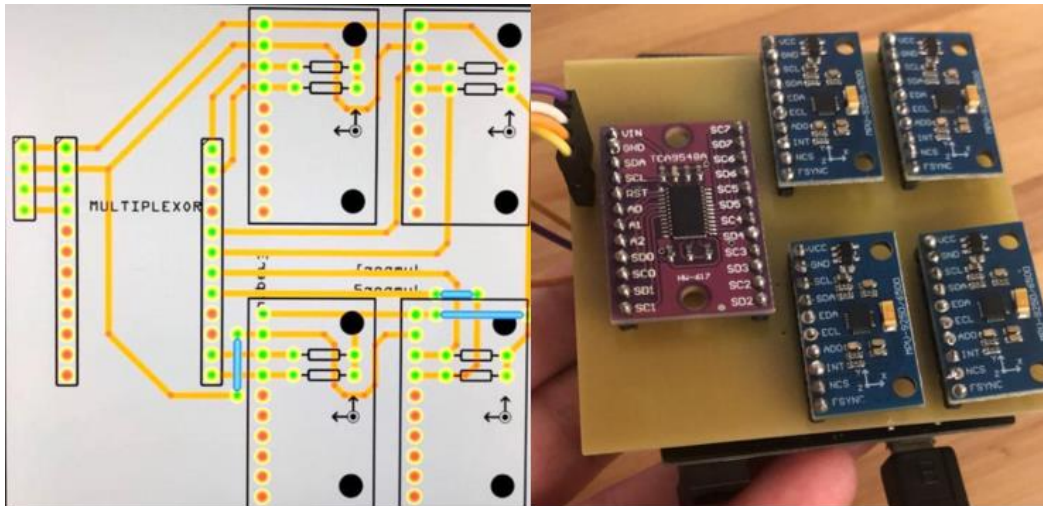


Ilustración 15. Esquemático de la PCB diseñada en Fritzing (izquierda) y el dispositivo fabricado (derecha)

Se requieren dos cables, datos en serie (SDA) y reloj en serie (SCL) para la comunicación entre hasta casi 128 (112) dispositivos cuando se usa direccionamiento de 7 bits y hasta casi 1024 (1008) dispositivos cuando se usa direccionamiento de 10 bits. MPU-9250 cuenta con tres convertidores de analógico a digital (ADC) de 16 bits para digitalizar las salidas del giroscopio, tres ADC de 16 bits para digitalizar las salidas del acelerómetro y tres ADC de 16 bits para digitalizar las salidas del magnetómetro. Las dos líneas son de "drenaje abierto", lo que significa que se deben conectar resistencias de extracción para que las líneas sean altas porque los dispositivos en el bus I2C están activos en baja. Se eligieron resistencias pull-up de 10 k $\Omega$ ; de lo contrario, no habría comunicación en el bus I2C porque la MCU no podría transmitir la dirección I2C.

## 5.2 FUSIÓN DE LOS CUATRO IMU

### 5.2.1 ACCELERÓMETRO

A continuación, se documenta el proceso de simulación de cuatro IMUs ideales posicionados de una manera idéntica (es decir, a distancias entre sí) a la de la placa diseñada. Se simulan las salidas de cuatro unidades IMU a través de una trayectoria generada por la función `waypointTrajectory`, que genera trayectorias usando puntos de ruta (*waypoints*) especificados. Cuando se crea el objeto sistema, se especifica el tiempo de llegada, la velocidad y la orientación en cada punto de ruta, que en este caso son los sensores IMU. Los puntos P1,

P2, P3 y P4 representan los cuatro sensores, cuya lectura asume que los puntos de masa respecto a ellos se encuentran a una distancia nula.

La función `transformMotion` permite calcular las magnitudes de movimiento del marco de P1 en relación con el marco de PN (N siendo un subíndice que indica la posición del último elemento o sensor) con una función que depende de la posición, orientación, velocidad, aceleración y velocidad angular de un marco a otro. Con esta función, se asume que la posición y la orientación entre los marcos son fijas, lo que es ideal para nuestra simulación porque la distancia entre los componentes de la PCB obviamente permanece estática. A continuación, se muestra la distancia entre el sensor MEMS de la unidad de medición inercial. Con este dato, se formula la matriz de posición de los sensores, el cuaternión que describe la orientación de este.

Ecuación 25. Matriz de posición de los sensores para la simulación de las salidas.

$$\begin{bmatrix} posP1 \\ posP2 \\ posP3 \\ posP4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0.0254 \\ 0 & 0 & 0 & 0.03556 \\ 0 & 0.0254 & 0.03556 & 0 \end{bmatrix}$$

Según la conveniencia del marco de referencia de Euler ZXY, cada sensor MPU-9250 tiene una orientación de 90° en el eje y. Todas las rotaciones en 3-D pueden describirse mediante un eje de rotación y un ángulo alrededor de ese eje. Una ventaja de los cuaterniones sobre las matrices de rotación es que el eje y el ángulo de rotación son fáciles de interpretar. Con el vector de ángulos de Euler ( $\varphi \theta \psi$ ) podemos sacar el cuaternión que introduciremos a la función `transformMotion`.

Ecuación 26. Cuaternión a ángulo de Euler, balanceo

$$balanceo = \varphi = \left( \frac{2(q_0q_1 + q_2q_3)}{1 - 2(q_1^2 + q_2^2)} \right)$$

Ecuación 27. Cuaternión a ángulo de Euler, cabeceo

$$cabeceo = \theta = (2(q_0q_2 - q_3q_1))$$

Ecuación 28. Cuaternión a ángulo de Euler, guiñada

$$guiñada = \psi = \left( \frac{2(q_0q_3 + q_1q_2)}{1 - 2(q_2^2 + q_3^2)} \right)$$

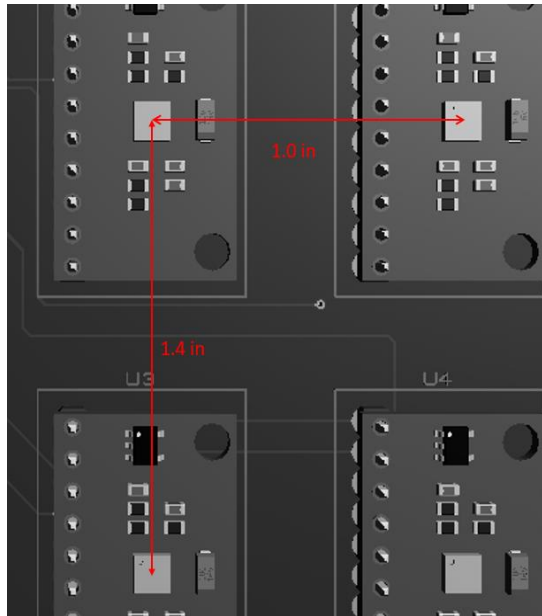


Ilustración 16. Distancias entre los sensores MPU-9250 para la simulación de salidas.

Cuando se habla de orientación, se refiere a qué tan lejos se gira un objeto de algún marco de referencia conocido. Para definir una orientación, debemos elegir el marco de referencia contra el que queremos describir la orientación y luego especificar la rotación desde ese marco usando alguna representación. Como se explicó ya en el marco teórico, los ángulos de balanceo, cabeceo y guiñada son susceptibles al bloqueo del cardán. Por esta razón, se prefiere usar cuaterniones y matrices de coseno directo.

Estos grupos de números representan una rotación tridimensional entre dos marcos de coordenadas diferentes. El marco de coordenadas propio del objeto que se fija al cuerpo y gira con él y algún marco de coordenadas externo y es esta rotación o estos conjuntos de números los que estaban tratando de estimar por midiendo alguna cantidad con sensores.

Un acelerómetro que no esté en su centro de masa va a detectar una aceleración cuando la placa gire. Esto se debe a que un par neto hará que un objeto gire con una aceleración angular. Está dentro del alcance de esta investigación encontrar una manera de lidiar con estas corrupciones. Como se está lidiando con múltiples sensores, adicionar una masa correctiva no daría directamente una solución, ya que es imposible tener más de dos centros de masas. Lo que es posible es, por medio de programación, fusionar todas las entradas de los sensores promediando las funciones de la salida, lo que nos permite obtener el valor promedio de la función en su dominio. De esta manera, es posible obtener una sola salida y un solo centro de masa.

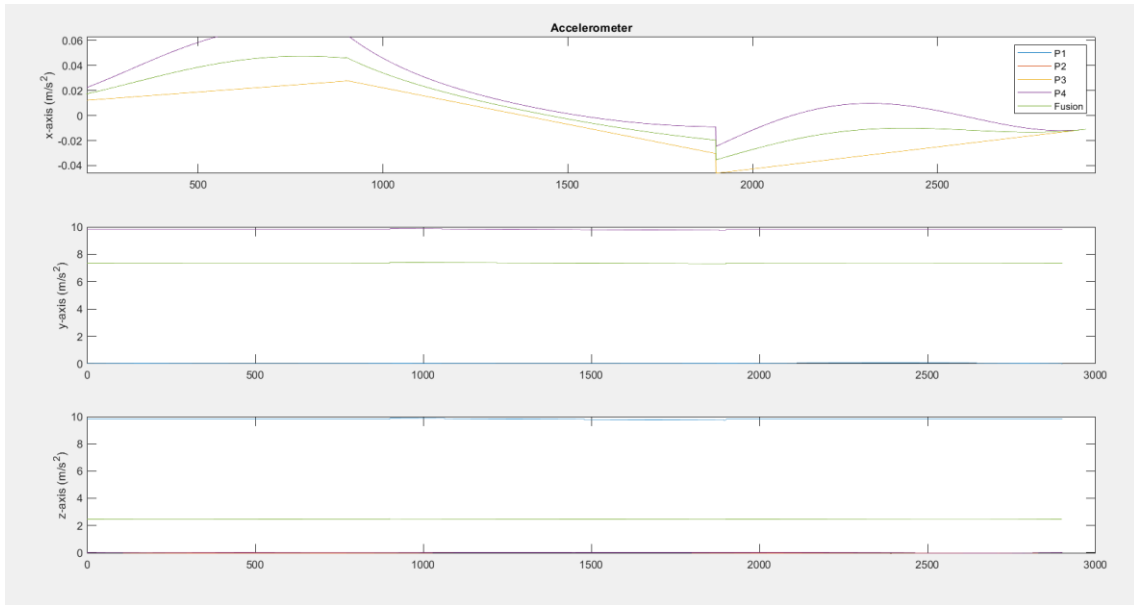


Ilustración 17. Gráficos trazados por las salidas IMU ideales simuladas.

### 5.2.2 MAGNETÓMETRO

Un magnetómetro mide campos magnéticos. Puede detectar fluctuaciones en el campo magnético de la Tierra midiendo la densidad del flujo magnético del aire en el punto del sensor en el espacio. A través de esas fluctuaciones, se encuentra el vector hacia el norte magnético de la Tierra. Esto se puede fusionar junto con los datos del acelerómetro y el giroscopio para determinar el rumbo absoluto de la placa.

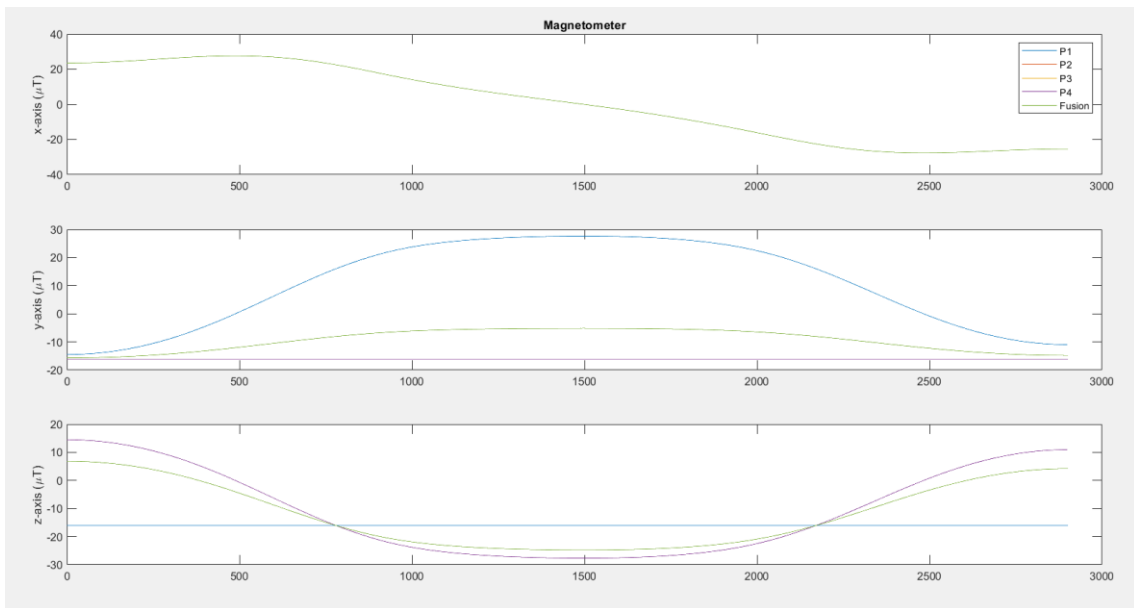


Ilustración 18. Gráficos trazados por la salida del magnetómetro del IMU ideal.

Un magnetómetro ideal de tres ejes mide la intensidad del campo magnético a lo largo de los ejes ortogonales X, Y y Z. En ausencia de cualquier interferencia magnética, las lecturas del magnetómetro miden el campo magnético de la Tierra. Si las mediciones del magnetómetro se toman mientras el sensor gira en todas las orientaciones posibles, las mediciones deben estar en una esfera. El radio de la esfera es la fuerza del campo magnético. A continuación, simulamos las muestras de un magnetómetro ideal con 480 muestras.

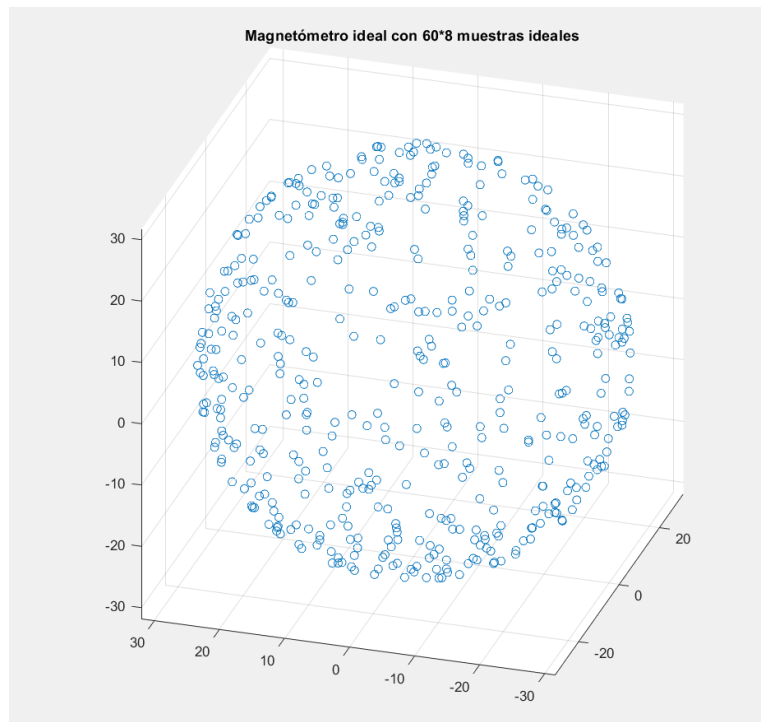


Ilustración 19. Salidas del magnetómetro del IMU ideal luego de girar 360° por los tres ejes.

El mayor contaminante de los magnetómetros son las interferencias magnéticas locales; por lo tanto, su calibración es muy esencial antes de su uso. Los métodos de calibración tradicionales requieren información de referencia y requisitos específicos, como mantener el nivel de la brújula magnética durante la calibración, que es muy difícil de manejar en exteriores.

Fang et al. (2011) presenta un método eficaz basado en el hecho de que el modelo de error de la brújula magnética es un elipsoide, y se adopta un método de restricción de mínimos cuadrados para estimar los parámetros de un elipsoide girando la brújula magnética en varias orientaciones (aleatorias) sin necesidad de mantener los requisitos especiales mencionados. El elipsoide que resulta del magnetómetro no calibrado se puede describir con la siguiente ecuación:



Ecuación 29. Elipsoide creado por las salidas del magnetómetro.

$$(x - b)R(x - b)^T = \beta^2$$

R es una matriz de 3 x 3, **b** es un vector de 1 x 3 que define el centro del elipsoide, **x** es un vector de 1 x 3 de medidas de magnetómetro no calibrado y  $\beta$  es un escalar que indica la intensidad del campo magnético. La ecuación anterior es la forma general de una cónica. Para un elipsoide, R debe ser definida positiva.

Una función esférica indica un magnetómetro libre de errores de hierro suave. En MATLAB, una matriz real A de 3 x 3 y un vector b de 1 x 3 con las mismas variables de la ecuación anterior describen la esfera del magnetómetro calibrado.

Ecuación 30. Esfera de las salidas de un magnetómetro ideal.

$$m = (x - b)A$$

La matriz A tiene un determinante de 1 y es la raíz cuadrada de la matriz de R. Además, A tiene la misma forma que R: la identidad, una diagonal o una matriz simétrica.

Wu et al. (2020) propuso en su investigación un método mejorado de compensación y calibración del magnetómetro basado en el algoritmo de Levenberg-Marquardt. El algoritmo de Levenberg-Marquardt (o método de mínimos cuadrados amortiguados, DLS) se utiliza para resolver problemas de mínimos cuadrados no lineales. Estos problemas de minimización surgen especialmente en el ajuste de curvas de mínimos cuadrados. Usando la convención de las variables anteriores, podemos traducir las ecuaciones de la investigación de Wu para esta aplicación. En este caso,  $x = [x_x \ x_y \ x_z]^T$  y  $m = [m_x \ m_y \ m_z]^T$  para que la computación matemática sea posible. **S** describe el vector de error por el hierro suave y **H** describe el error por hierro duro.

Ecuación 31

$$x = S^{-1}m - H$$

La intensidad total del campo magnético de la Tierra que se mide con un magnetómetro ideal debe ser una constante. Este se describe matemáticamente de la siguiente manera:

Ecuación 32

$$|m|^2 = m^T m = (x + H)^T S^T S (x + H)$$

La ecuación anterior está limitada por una esfera o elipsoide positivo, que se puede escribir de la siguiente forma:

Ecuación 33

$$|x|^T = \frac{S^T S}{|m|^2} x - \frac{2H^T S^T S}{|m|^2} x + \frac{H^T S^T S H}{|m|^2} = 1$$

La distancia de la esfera o elipsoide desde el origen indica el error por hierro duro. Entonces, podemos hacer la siguiente relación:

Ecuación 34

$$x = m - H$$

Sea R el radio de la esfera.

Ecuación 35

$$(x + H)^T (x + H) = m^T m = R^2$$

Ecuación 36

$$R^2 - (x^T x + 2x^T H + H^T H) = 0$$

El algoritmo de Levenberg-Marquardt funciona de la siguiente manera: dado un conjunto de N pares de datos empíricos m de variables independientes y dependientes, este calcula los parámetros (R, H) de la curva modelo  $R - f(\mathbf{x}, H)$  de modo que la suma de los cuadrados de las desviaciones S (R, H) se minimiza.

Ecuación 37

$$f(x, H) = |x^T x + 2x^T H + H^T H|^{\frac{1}{2}}$$

El error residual es la suma de las distancias entre los datos calibrados y una esfera de radio está dada por:

Ecuación 38

$$E = \frac{1}{2\beta^2} \sqrt{\frac{\sum ||(x - b)A||^2 - \beta^2}{N}}$$

Sea  $\hat{R}$  el radio óptimo de la intensidad del campo magnético, calculado con la fórmula de ajuste de esfera. Para que el ajuste del elipsoide basado en el algoritmo de Levenberg-Marquardt funcione, se toman en cuenta los errores por hierro suave S y hierro duro H.

A continuación, se simula la salida de un magnetómetro ideal corrompido por contaminantes por hierro suave y hierro duro, con 480 muestras.

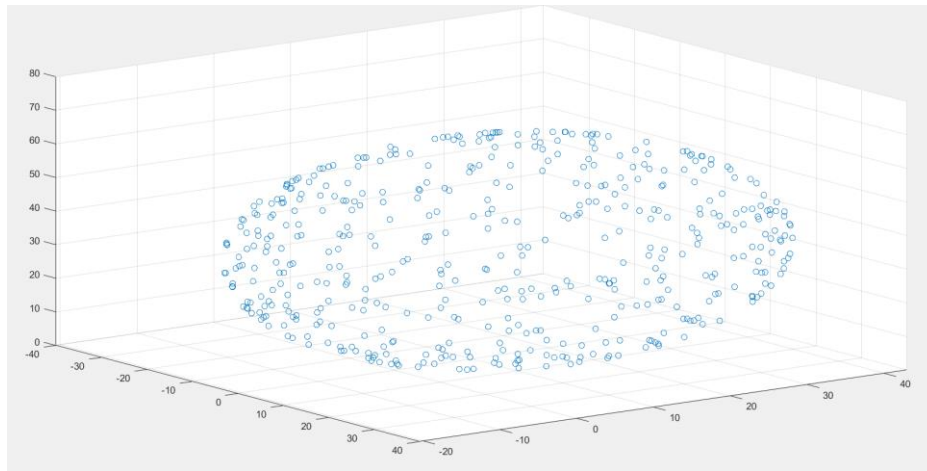


Ilustración 20. Salida del magnetómetro del IMU ideal, agregando corrupciones por hierro suave y hierro duro.

Para mejorar la precisión de la estimación del ángulo de guiñada, los magnetómetros deben calibrarse nuevamente antes del despegue (o cualquier otro uso del MPU-9250). El algoritmo de calibración toma en consideración que todas las mediciones tienen cierto grado de incertidumbre que puede provenir de una variedad de fuentes, como las mencionadas anteriormente. Tendríamos que promediar un número infinito de mediciones para acercarnos al valor medio verdadero, e incluso entonces, no se nos garantiza que el valor medio sea exacto porque todavía hay algún error sistemático de la herramienta de medición, que nunca se puede calibrar perfectamente.

Se introduce en el código un bucle con tiempo suficiente para una rotación de un eje antes de la fusión del sensor. El MPU-9250 puede ajustarse a cualquier frecuencia de muestreo. Por razones pragmáticas, la frecuencia de muestreo en Herz se establece en 100. Para cada eje, digamos  $x$ , el valor del valor máximo de  $x$  se suma con el valor mínimo de  $x$ , dividido por dos, en el tiempo discreto de  $t$ , y así sucesivamente para el eje- $y$  y el eje- $z$ . El número de promedios calculados es el valor de la frecuencia de muestreo.

### 5.2.3 GIROSCOPIO

El objetivo principal de esta sección es describir el mecanismo que elimina y corrige la deriva de manera significativa para optimizar el giroscopio MEMS para la estimación de guiñada y

rumbo, la justificación e implementación de fusión de múltiples giroscopios y la corrección del error de estimación.

La adición y el promedio de múltiples salidas de giroscopio se basa en el concepto de que el rendimiento colectivo de múltiples dispositivos nominalmente idénticos puede ser mejor que el de uno de los dispositivos considerados por sí mismo. El método permitiría sintetizar las lecturas de una única señal más precisa de los giroscopios fusionados de las unidades MPU-9250.

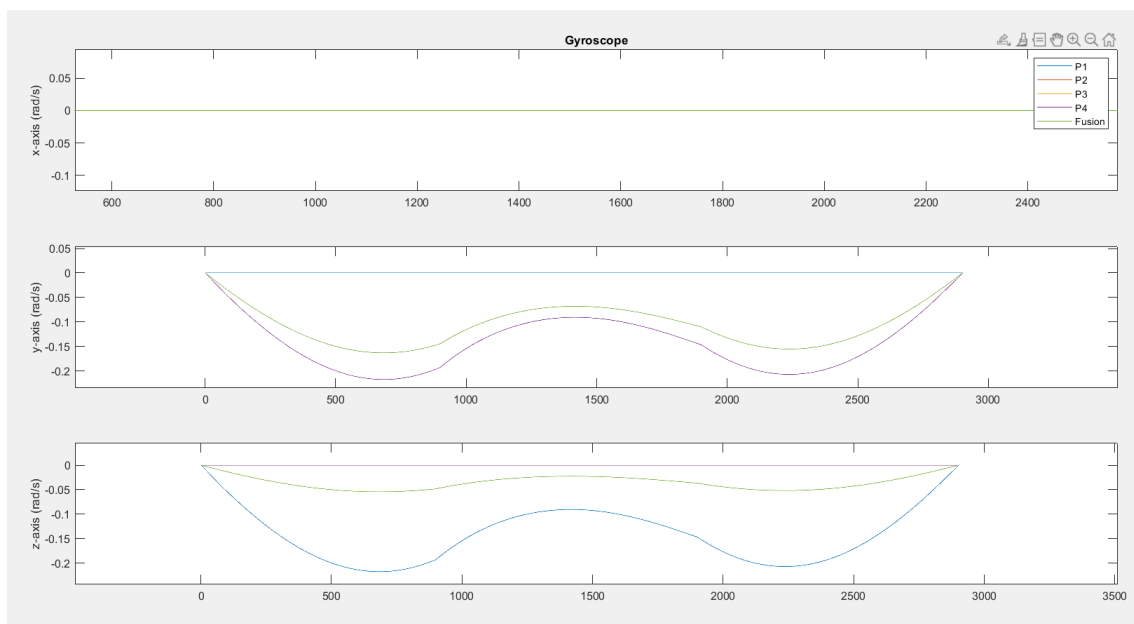


Ilustración 21. Salida de la trayectoria fusionada de IMUs ideales.

El algoritmo de fusión del giroscopio consiste en combinar y filtrar las salidas digitalizadas de múltiples giroscopios para obtener estimaciones de tasa de varianza mínima, definida en la sección anterior como un estimador insesgado que tiene una varianza más baja que cualquier otro estimador insesgado para todos los valores posibles del parámetro. En las operaciones de combinación y filtrado, los datos de medición de los giroscopios se ponderan y suavizan entre sí de acuerdo con la matriz de ganancia de un filtro de varianza mínima, en contraste con el algoritmo de Levenberg-Marquardt utilizado en la sección del magnetómetro. El siguiente esquema de control muestra el algoritmo de combinación. Se agregan cuatro salidas de giroscopio sin procesar a un bloque de promediado, para que sea procesado por un filtro de variación mínima. El modelo de verdad es generado por las señales de IMU ideales utilizando las funciones de MATLAB descritas en las secciones anteriores.

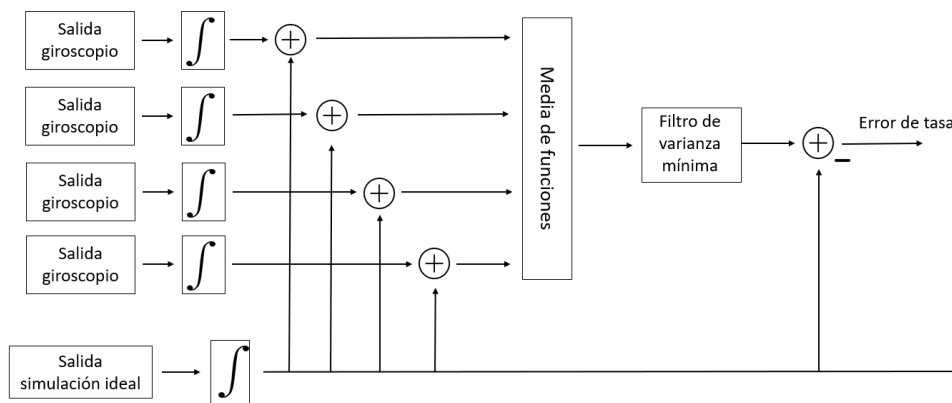


Ilustración 22. Algoritmo de combinación de las cuatro señales de los MPU-9250.

Posteriormente, se expone la gráfica de un IMU no ideal simulado en MATLAB con la función `imuSensor`, donde los ángulos de la cabeceo, balanceo y guiñada con compensados respecto a su valor real.

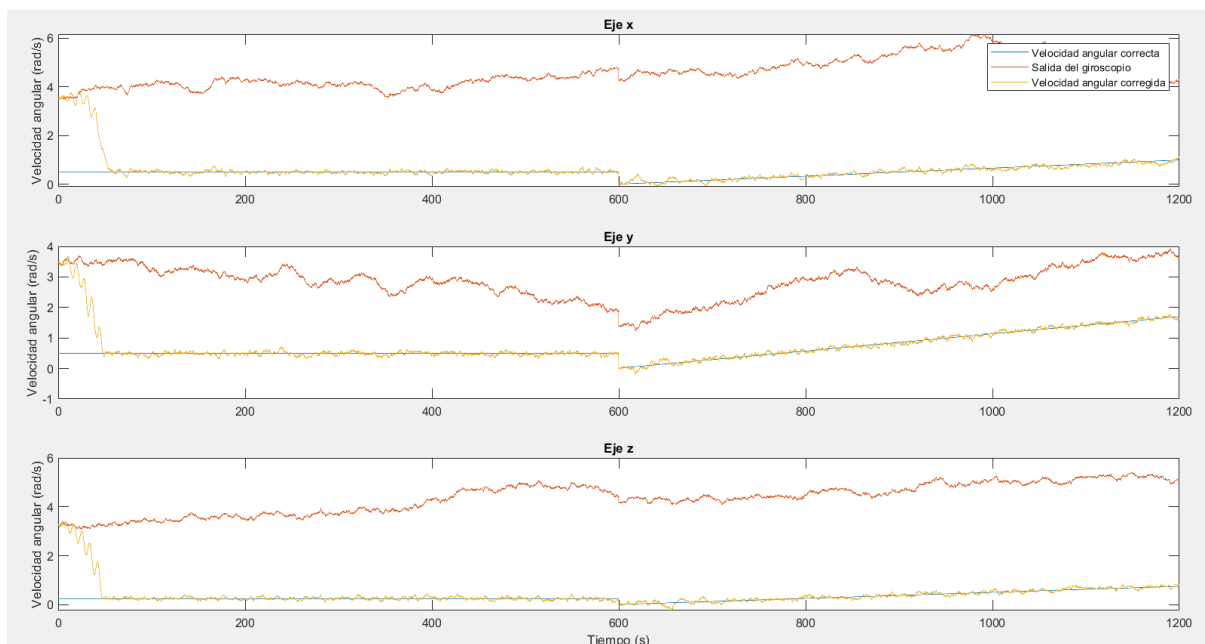


Ilustración 23. Compensación de error del ángulo de salida de un IMU no ideal virtual.

La deriva del giroscopio se debe principalmente a la integración de dos componentes: una variable de cambio lento, cercana a la CC, llamada inestabilidad de polarización, y una variable de ruido de frecuencia más alta llamada caminata angular aleatoria (ARW). Estos parámetros se miden en grados de rotación por unidad de tiempo. El eje de guiñada es más sensible a

esta deriva. Una buena parte de la desviación del giroscopio del eje de cabeceo (actitud) y balanceo se puede eliminar dentro de una IMU mediante el uso de la retroalimentación del acelerómetro para monitorear la posición relativa a la gravedad.

#### **5.2.4 ELECCIÓN DEL ALGORITMO**

Las dos métricas de rendimiento son la exactitud y precisión de la fusión y el tiempo computacional para realizar la fusión.

Menciona St-Pierre & Gingras (2004) que el filtro de Kalman *Unscented* tiene un rendimiento ligeramente mejor que el filtro Kalman extendido cuando se utiliza como método de fusión en un módulo de posicionamiento de un sistema de información de navegación integrado. Se toma en cuenta que el estudio referido es muy anticuado para sacar conclusiones para las otras métricas como poder de procesamiento y el tiempo para computar el filtro. El principal escollo al trabajar con algoritmos tan costosos desde el punto de vista computacional es la potencia de procesamiento limitada disponible. Esta sección explorará si vale la pena intercambiar la potencia computacional por precisión y precisión.

Los sistemas integrados como los sensores MEMS se dirigen principalmente a un bajo consumo de energía, ya que la mayoría de ellos funcionan con baterías y apuntan a bajos costos de fabricación. Por lo tanto, estos sistemas a menudo se diseñan en torno a arquitecturas de microcontroladores de un solo núcleo con bajas frecuencias de funcionamiento. El estudio de Valade et al. (2017) concluye que los modelos de sistemas grandes son costosos de calcular y no se pueden abordar a altas frecuencias con objetivos de baja potencia.

Tomando en cuenta que nuestro microprocesador está alimentado por un procesador Atmega328 que funciona a 16 MHz, que incluye 32 KB de memoria de programa, 1 KB de EEPROM, 2 KB de RAM, con operaciones de flotación de precisión única de aproximadamente 100,000, es necesario ser ahorrativo con el poder de procesamiento del Arduino Uno.

El filtro de Kalman extendido muestra mejores resultados que los otros dos métodos probados en la investigación. También escogió el filtro de Kalman extendido por la naturaleza no lineal de los movimientos tridimensionales de un vehículo autónomo que pudiera usar esta placa.

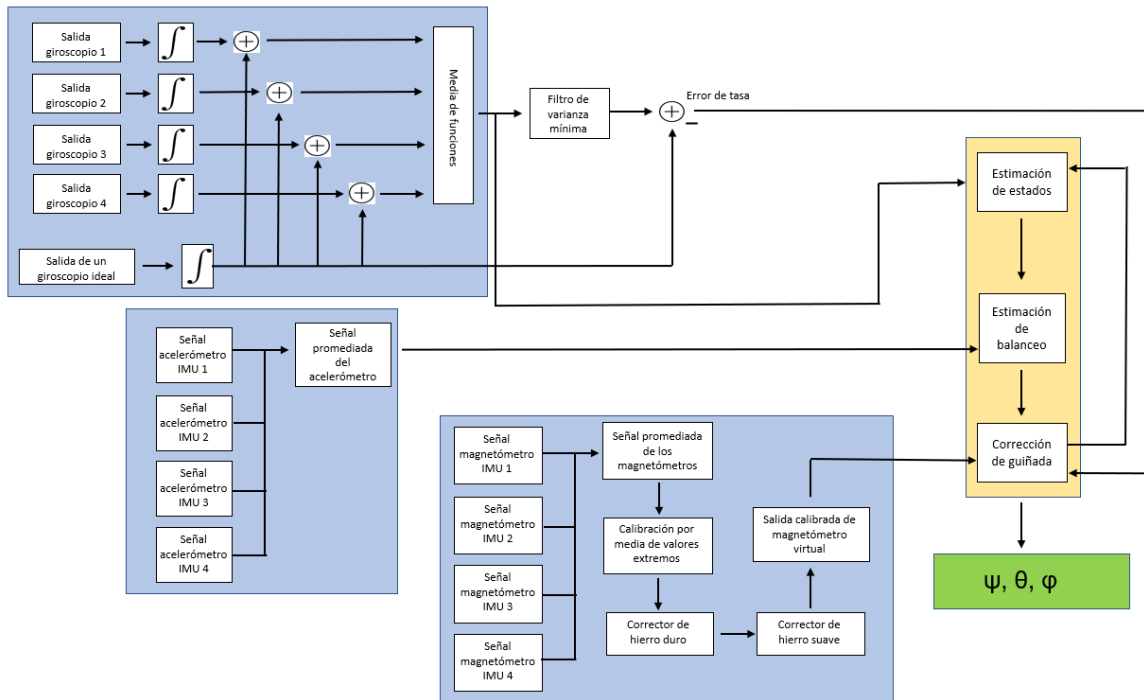


Ilustración 24. Algoritmo de fusión de sensores con filtro extendido de Kalman. Leyenda: entradas del sensor (azul), EKF (naranja), ángulos de salida (verde).

### 5.2.5 MODELADO DEL ESPACIO DE ESTADOS

A continuación, se presenta el modelo de espacio de estados que describe los cuatro sensores MPU-9250. Para definir un modelo para la plataforma del sensor, las siguientes variables se definen de la siguiente manera:  $y$  es la tasa angular medida,  $\omega$  es la tasa angular verdadera,  $b$  es la desviación del sesgo,  $S$  es el error del factor de escala y  $v$  es el ángulo de caminata aleatoria (ARW) ruido blanco. El error del factor de escala (qué tan bien la salida del sensor corresponde a una entrada de fuerza o velocidad) afectará las salidas del giroscopio MEMS en una situación con cambios repentinos en la orientación y el término de deriva de sesgo también es un error dominante cuando el sensor funciona durante mucho tiempo. hora. Uno de los principales alcances de este proyecto es mitigar o eliminar los efectos del sesgo de deriva, por lo que se hace énfasis en este error en particular.

La salida del giroscopio se filtra mediante el filtro de Kalman extendido. Los filtros de Kalman extendidos operan linealizando el modelo no lineal alrededor de la trayectoria de referencia actual y luego diseñando la ganancia del filtro de Kalman para el modelo linealizado, lo cual nos permite hacer el modelado de manera lineal por simplicidad.

Se estableció en el marco teórico las ecuaciones para el filtro de Kalman. Como se muestra a continuación, este filtro consiste en un ciclo que contiene dos pasos: actualización de tiempo, que es un proceso de predicción, y actualización de medidas, que es un proceso de corrección, donde  $\hat{x}_k$  es el estado estimado a posteriori,  $P_k$  es el estado a priori,  $u$  es el vector de control,  $z$  es la señal de medición,  $k$  es un punto discreto en el tiempo,  $A$  es el modelo de transición de estado,  $K$  es la ganancia de Kalman,  $B$  es el modelo de entrada de control,  $H$  es la matriz de medición,  $R$  es la covarianza del ruido medido,  $P$  es la covarianza del error,  $Q$  es la covarianza del ruido del proceso e  $I$  es la matriz de identidad Las primeras dos ecuaciones calculan los estados futuros.

Ecuación 39

$$\hat{x}_k = A\hat{x}_{k-1} + Bu_{k-1}$$

Ecuación 40

$$P_k = AP_{k-1}A^T + Q$$

Las siguientes tres ecuaciones representan el proceso de actualización y corrección de la medición.

Ecuación 41

$$K_k = \frac{P_k H^T}{HP_k H^T + R} = P_k H^T (HP_k H^T + R)^{-1}$$

Ecuación 42

$$\hat{x}_{k+1} = \hat{x}_k + K_k(z_k - H\hat{x}_k)$$

Ecuación 43

$$P_{k+1} = (I - K_k H)P_k$$

## 5.3 RESULTADOS

### 5.3.1 MPU-9250 SINGULAR CON EJEMPLO INTEGRADO MATLAB

A continuación, se muestran los resultados del sensor singular MPU-9250 en reposo en una superficie plana, con ángulos de guiñada, cabeceo y balanceo de cero ( $0^\circ$ ); con la herramienta



de fusión de sensores de MATLAB, sin calibración del magnetómetro, filtro de Kalman para el giroscopio y el promedio de las salidas de la matriz de sensores.

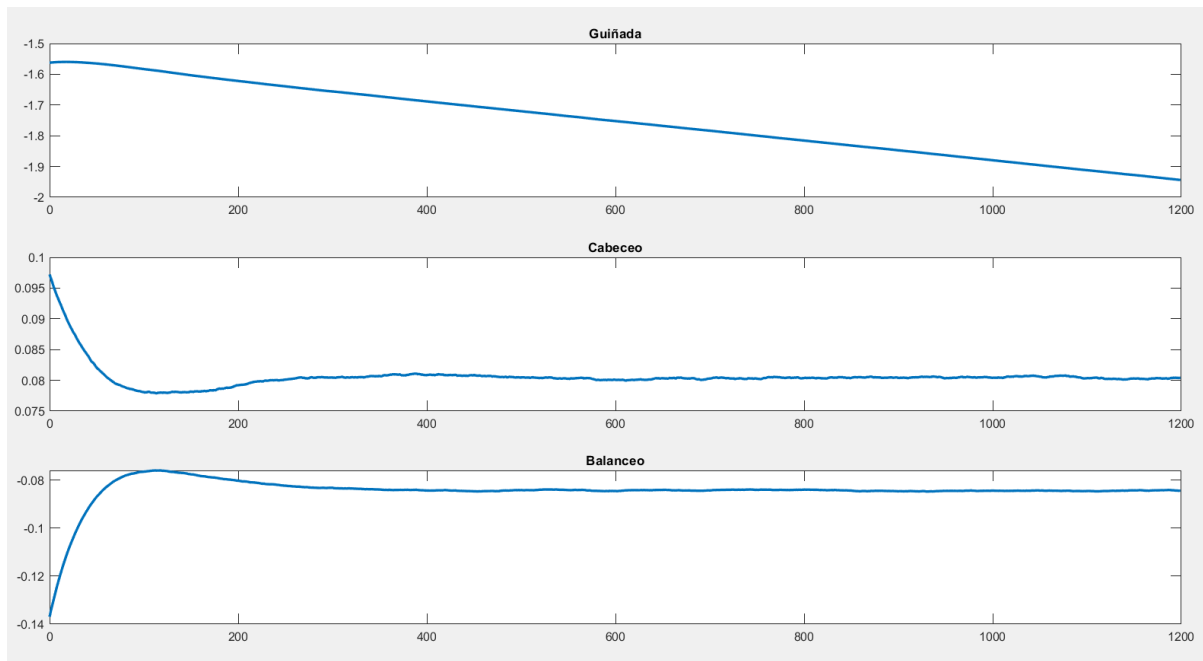


Ilustración 25. Salida de fusión de sensor MPU-9250 singular con filtro de Kalman de estado de error interno.

Se aprecia que el ángulo de guiñada es el que más sufre de deriva. Esto es esperado, ya que la deriva de los ángulos de cabeceo y balanceo se reduce dentro de una IMU mediante el uso de la retroalimentación del acelerómetro para monitorear la posición relativa a la gravedad. El ángulo de guiñada tuvo un proceso de desviación aproximadamente lineal, con una deriva de  $0.1820^\circ/\text{s}$  durante 120 segundos.

Los siguientes valores de ángulo se asemejan a un sistema de repuesta de estado estable y transitorio. El tiempo de asentamiento ocurre entre el intervalo de tiempo de 20 y 40 segundos (los cálculos exactos están fuera del alcance de esta investigación), intervalo en el cual los ángulos de cabeceo y balanceo se estabilizan. El promedio de las salidas del sensor después de 40 segundos es el error de estado estable. El ángulo de cabeceo, dentro del intervalo de tiempo de 40 segundos a 120 segundos, promedia un valor de 0.08043 radianes, o  $4.608^\circ$ , lo cual en un sensor ideal sería de  $0^\circ$ . Dentro del mismo intervalo, el ángulo de balanceo promedió -0.0843 radianes, o  $-4.831^\circ$ . Igualmente, en un sensor ideal, el ángulo sería  $0^\circ$ .

Tabla 2

Ángulo	Dif. final-inicial (rad)	Dif. final-inicial (grad)
Guiñada	-0.3814	-21.85
Cabeceo	-0.0168	-0.9643
Balanceo	-0.1816	-10.41

Se aprecia que el ángulo de guiñada comienza en  $-1.563$  radianes, o  $-89.52^\circ$ . Eso se debe a la naturaleza del algoritmo de fusión de sensores `imufilter` que *Sensor Fusion and Tracking Toolbox* ofrece como ejemplo para el sensor MPU-9250 singular. Cuando el eje x del sensor del dispositivo apunta hacia el norte, el eje y del dispositivo apunta hacia el este y el eje z del dispositivo apunta hacia abajo. Por lo tanto, el ángulo  $0^\circ$  real es computado como  $-90^\circ$  en el ángulo de Euler del código.

La función devuelve un objeto de sistema de filtro de Kalman indirecto para la fusión de datos de acelerómetro y giroscopio para estimar la orientación del dispositivo. El filtro utiliza un vector de estado de nueve elementos para rastrear el error en la estimación de orientación, la estimación de sesgo del giroscopio y la estimación de aceleración lineal. Como se observó en las pruebas anteriores, la estimación del sesgo del giroscopio no cumple con los requisitos de error mínimo que necesitaría un dispositivo de navegación y orientación, de ahí este proyecto de investigación.

Tabla 3

Ángulo	Error inicial	Error inicial (grad)	Error final	Error final (grad)	Diferencia errores
Guiñada	-1.563	-89.52	-1.944	-111.4	-21.85
Cabeceo	0.09722	5.570	0.08039	4.606	-0.9643
Balanceo	-0.1369	-7.845	-0.08439	-4.835	3.010

### 5.3.2 MÚLTIPLES MPU-9250 ACOPLADOS CON FILTRO EXTENDIDO DE KALMAN

En esta sección se plasman los resultados obtenidos de las cuatro pruebas realizadas con la placa, primero permitiendo el funcionamiento de un solo sensor, seguido de dos sensores, hasta completar la matriz de los cuatro sensores. El propósito de esto es conseguir una relación o función entre el número de MPU-9250 en una matriz multisensorial y la precisión o mejora en la medición de orientación o rumbo.

A continuación, se muestra la gráfica trazada por la salida de un sensor MPU-9250 en reposo. Idealmente, los ángulos de guiñada, cabeceo y balanceo se mantienen en un valor constante. Siempre habrá una desviación debido a la naturaleza imperfecta de los sensores reales, especialmente los de bajo costo como este.; por eso, es menester interpretar los errores para mejoras futuras. Se puede apreciar que existe un pico, o valor máximo para cada ángulo al inicializar el computador de ángulos con el filtro extendido de Kalman. Esto es similar al resultado de la prueba realizada en la sección pasada con filtro de Kalman de estado de error interno con un único sensor, donde los ángulos de cabeceo y balanceo pasan por un proceso de estabilización en el estado transitorio. En la prueba monosensorial con el algoritmo diseñado en esta investigación, el pico lo sufren el ángulo de guiñada y de balanceo, aunque sea por un mucho menor tiempo. Luego de aproximadamente 20 segundos, la salida del sensor se asemeja a una onda senoidal, con características como cresta, amplitud, vaguada y frecuencia.

Aunque sería ideal obtener el error de ángulo real ( $<0^\circ, 0^\circ, 0^\circ>$ ) respecto al ángulo medido, esto no fue posible debido a que el ángulo inicial según el programa no es el vector de ceros, y no hay dispositivos de instrumentación prácticos para compensar por ese error. Por esta razón, para poder medir cuantitativamente las mejoras (o la falta de ella) de cada prueba, se medirán la estabilidad del sistema, determinadas por la amplitud (o la mitad de la distancia entre la vaguada y la cresta), y cualitativamente evaluar el ruido. Se aprecia que el ángulo de cabeceo sufre menos ruido que los otros ángulos, seguido del ángulo de guiñada. El ángulo de cabeceo se mantuvo suave y libre de ruido y otras perturbaciones durante la prueba de 120 segundos, además de no tener un pico o impulso al principio. Los tres ángulos se disparan al finalizar el bucle, asemejándose a una función de pulso. Es posible que esto último no sea fuente de errores o inconvenientes en aplicaciones reales, ya que solo se da al finalizar la aplicación.

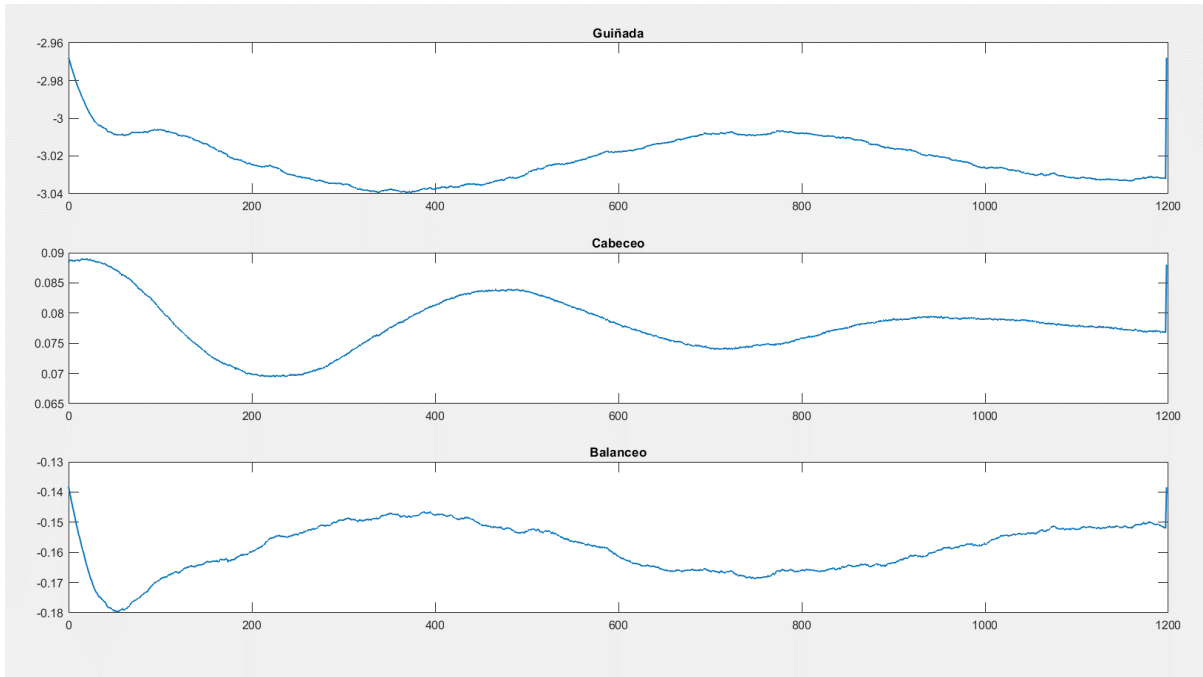


Ilustración 26. Gráfica de valores de ángulos sacados con filtro extendido de Kalman, un solo sensor MPU-9250.

A continuación, se muestra una tabla comparativa de los tres ángulos posibles en el espacio tridimensional, con su valor promedio en radianes y en grados en los 120 segundos de prueba, así como los valores máximo y mínimo del estado estable o permanente del sistema, i.e., la cresta y la vaguada. Los promedios de todos los ángulos son negativos. Para calcular la cresta y la vaguada, solo se tomaron en cuenta los valores después de estabilizarse el sistema, i.e., luego de 10 segundos, y antes de la función pulso al finalizar la aplicación, siendo establecido este límite en 110 segundos. Confirmando lo dicho anteriormente respecto al ángulo de cabeceo, este fue el que menos diferencia cresta-vaguada, o amplitud, presentó, lo que indica una salida constante y estable.

Tabla 4

Ángulo	Average	Average (grad)	Cresta	Vaguada	Diferencia C-V
Guiñada	-3.0211329	-173.098226	-2.96774654	-3.03929188	0.071545341
Cabeceo	-1.47157754	-84.3152124	0.08899042	0.06946623	0.01952419
Balanceo	-0.15864023	-9.08941903	-0.13827063	-0.17955417	0.041283534

A continuación, se muestra la gráfica trazada por las salidas del algoritmo extendido de Kalman utilizando las entradas de dos sensores. Se puede apreciar la falta de impulsos o de estados

transitorios al inicializar el programa, lo que es un hito logrado para futuras aplicaciones, especialmente en aquellas donde no hay 20 segundos de sobra para inicializar la aplicación. Siempre existe una función pulso al finalizar el bucle; sin embargo, este se extiende unos pocos segundos más, precedido por lo que se asemeja a una función escalera. El ángulo de balanceo fue el que más ruido presentó, seguido del ángulo de guiñada y luego el ángulo de cabeceo.

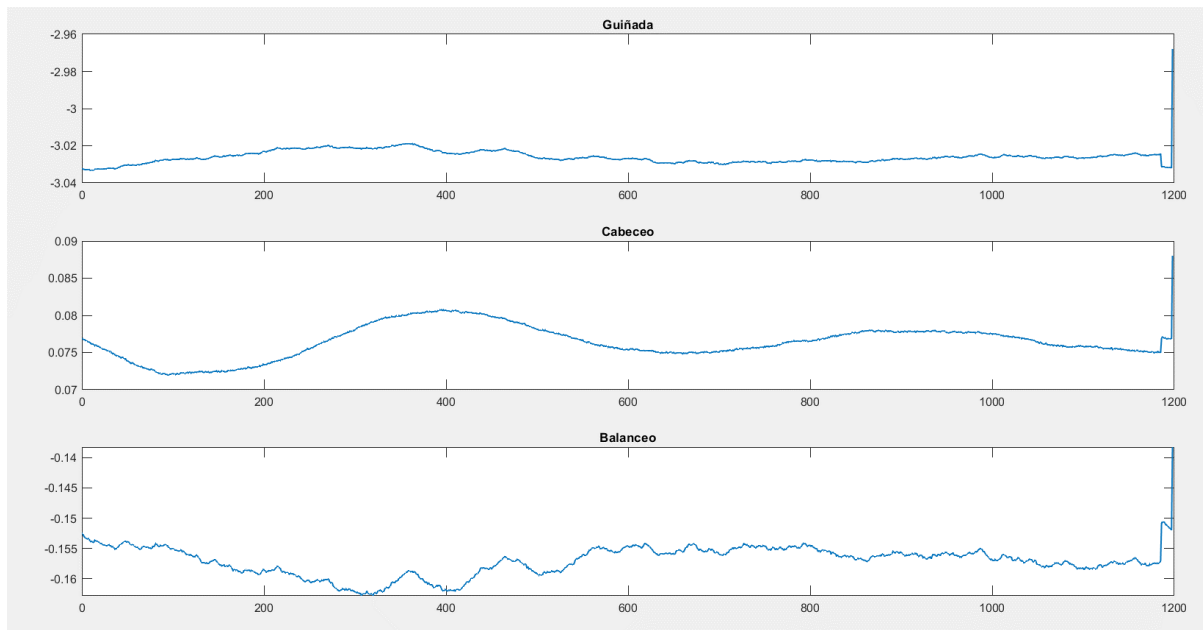


Ilustración 27. Gráfica de valores de ángulos sacados con filtro extendido de Kalman, dos sensores MPU-9250.

A continuación, se presenta la tabla de valores con el valor promedio, la cresta y la vaguada y la diferencia entre ellos. A diferencia de la prueba anterior, aquí únicamente el ángulo de cabeceo tiene una forma sinusoidal. El gráfico trazado por el ángulo de guiñada se asemeja a una función de variable independiente constante; el gráfico trazado por el ángulo de balanceo no es suave y presenta impurezas debido al ruido. Por esta razón es posible que llamar a los valores máximos y mínimos cresta y vaguada sea erróneo; no obstante, la diferencia entre estos valores sigue siendo útil para determinar la falta de constancia del valor de estimación del ángulo. Hubo una disminución considerable en la diferencia cresta-vaguada.

Tabla 5

Ángulo	Average	Average (grad)	Cresta	Vaguada	Diferencia C-V
Guiñada	-3.02601166	-173.377759	-2.96781926	-3.03321859	0.065399329
Cabeceo	-1.47480344	-84.5000428	0.08806461	0.07193352	0.016131082
Balaneo	-0.15705081	-8.99835195	-0.13827063	-0.16281894	0.024548301

A continuación, se muestran las gráficas de los valores trazados por la salida del filtro extendido de Kalman utilizando tres sensores MPU-9250. Se puede apreciar que se el ángulo de guiñada no tuvo mucha mejora (evaluándola según el criterio establecido), por lo menos no una mejora lineal o exponencial. Se amortiguó la gráfica trazada por el ángulo de cabeceo, corrigiendo la diferencia de las amplitudes por 50%. El ángulo de balanceo no tuvo mejoras en cuanto a la eliminación del ruido. Persiste el impulso al finalizar la aplicación.

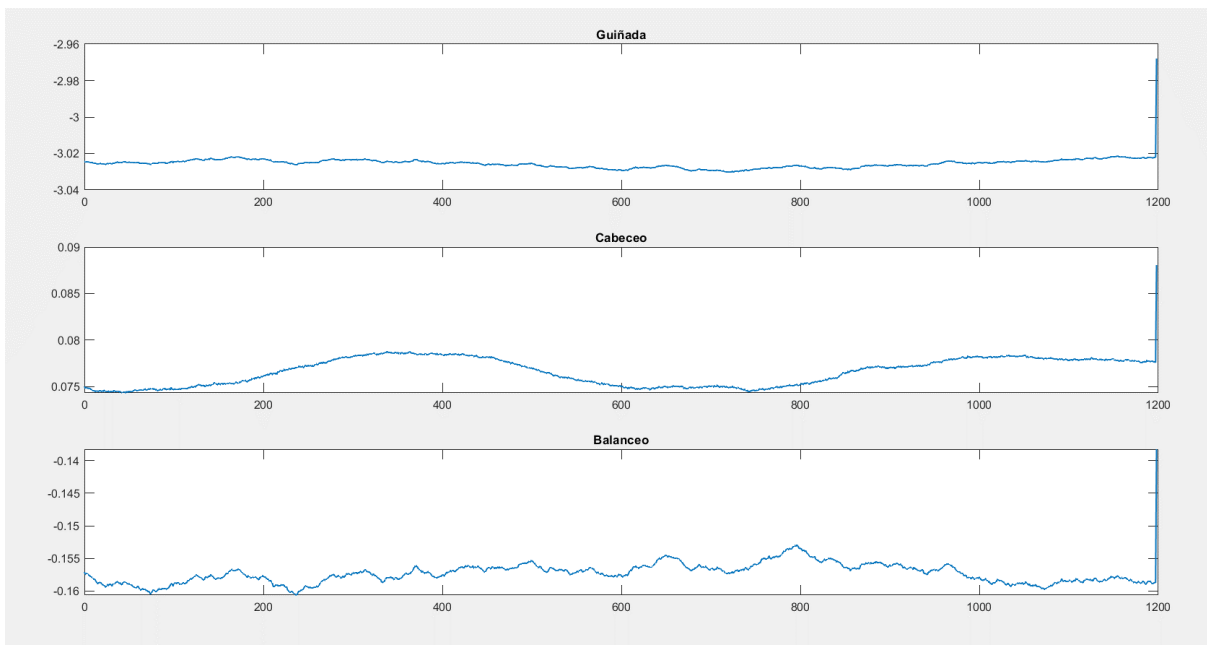


Ilustración 28. Gráfica de valores de ángulos sacados con filtro extendido de Kalman, tres sensores MPU-9250.

Tabla 6

Ángulo	Average	Average (grad)	Cresta	Vaguada	Diferencia C-V
Guiñada	-3.0255439	-173.350958	-2.96781926	-3.03021464	0.062395383
Cabeceo	-1.47447574	-84.4812673	0.08806461	0.07431518	0.013749424
Balaneo	-0.1572955	-9.01237175	-0.13827063	-0.16061441	0.022343776

Para finalizar, se muestran las gráficas trazadas por los ángulos de guiñada, cabeceo y balanceo de un filtro de Kalman extendido utilizando cuatro sensores MPU-9250 como entradas. El ruido en el ángulo de balanceo se redujo. Persiste la función impulso en los tres gráficos al finalizar la aplicación. Hubo una reducción mínima en la diferencia cresta-vaguada.

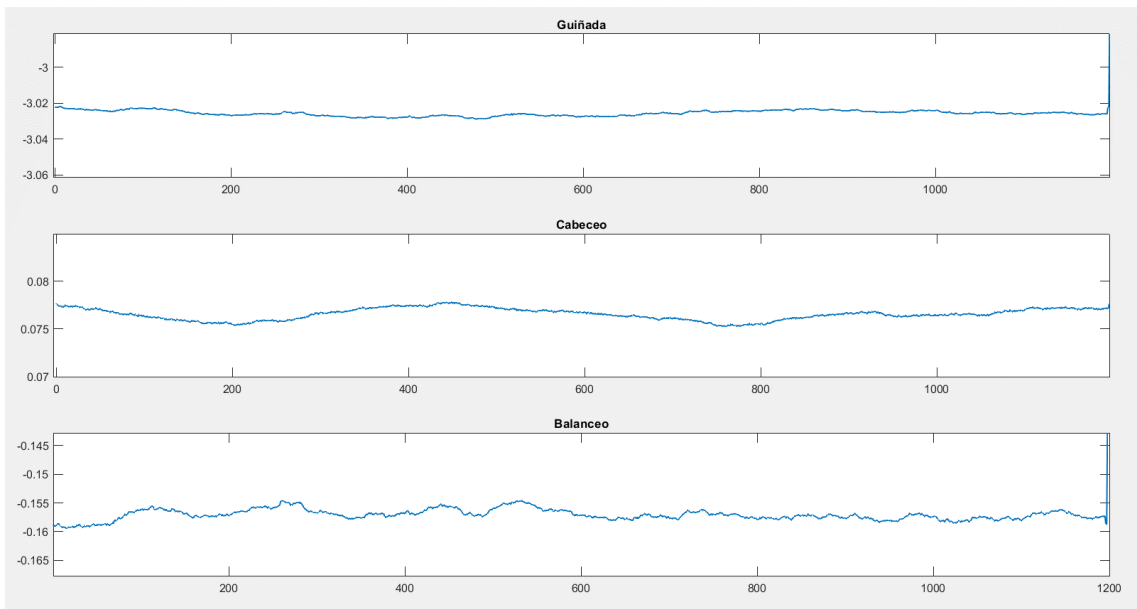


Ilustración 29. Gráfica de valores de ángulos sacados con filtro extendido de Kalman, cuatro sensores MPU-9250.

Se presenta la última tabla sacada de los gráficos generados.

Tabla 7

Ángulo	Average	Average (grad)	Cresta	Vaguada	Diferencia C-V
Guiñada	-3.02545379	-173.345795	-2.96781926	-3.02872791	0.060908651
Cabeceo	-1.47444124	-84.4792906	0.08806461	0.07526464	0.012799966
Balanceo	-0.15704774	-8.9981759	-0.13827063	-0.15943327	0.021162631

A continuación, se demuestra un gráfico que demuestra la tendencia no lineal o no exponencial en la relación entre el número de sensores MPU-9250 con la diferencia cresta-vaguada de la salida de los ángulos. Esta tendencia, que se asemeja a una relación logarítmica, es menos pronunciada en el ángulo de cabeceo, que se asemeja a más una relación lineal. La diferencia entre el primer punto (un sensor IMU en el eje independiente) y el segundo punto es de 0.006146012 radianes. Entre el segundo y el tercero es de 0.003003946 radianes y entre el tercero y el cuarto es de 0.001486732.

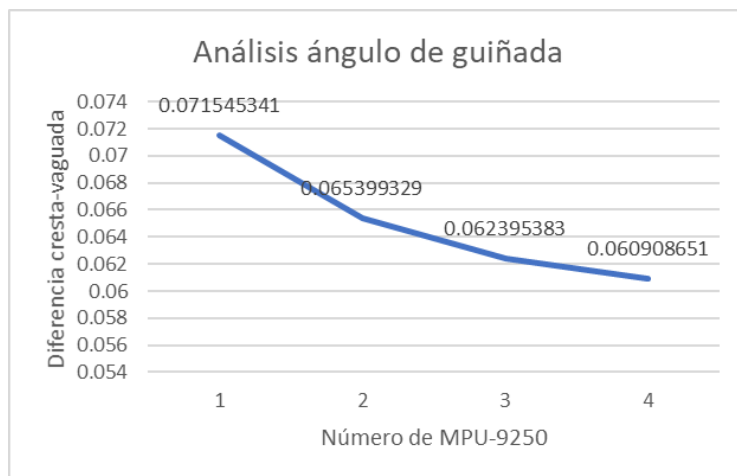


Ilustración 30. Relación del ángulo de guiñada con la diferencia cresta-vaguada.

Para el ángulo de cabeceo, la relación es casi lineal, con las mínimas diferencias de 0.003393108 radianes entre el primer y segundo punto, 0.002381658 radianes entre los puntos dos y tres puntos, y 0.000949458 radianes entre el tercer y cuarto punto.



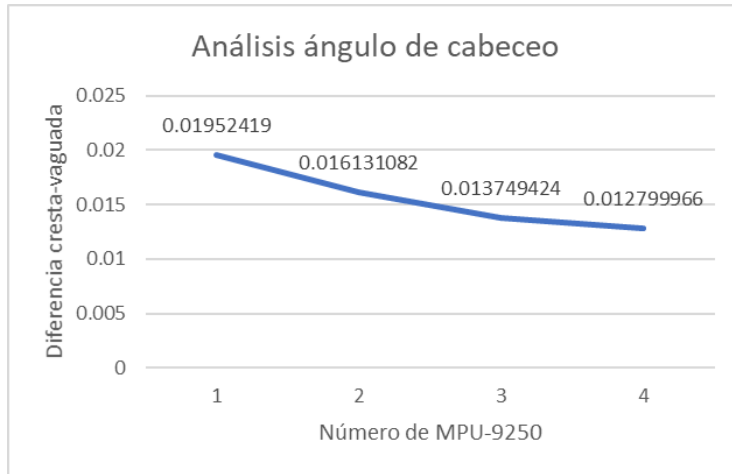


Ilustración 31. Relación del ángulo de cabeceo con la diferencia cresta-vaguada.

Para el ángulo de balanceo, existe una relación aproximadamente lineal después del uso del segundo sensor. Existe una mejora inmediata de 0.016735234 radianes entre el uso de uno y dos sensores en la prueba realizada. En el intervalo entre el punto dos y el punto tres, existe la mínima mejora de 0.002204525 radianes, y 0.001181145 radianes entre el tercer y cuarto punto.

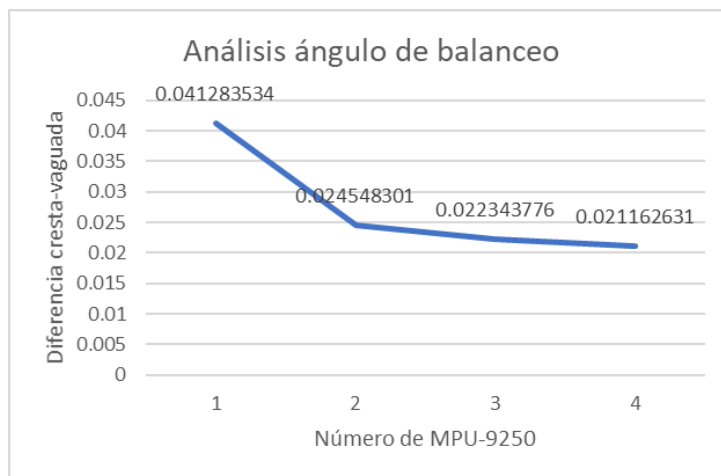


Ilustración 32. Relación del ángulo de balanceo con la diferencia cresta-vaguada.

## **6 CONCLUSIONES**

El algoritmo de Kalman extendido elimina el problema de errores acumulados por deriva en el ángulo de guiñada en las aplicaciones de fusión de sensores.

Las mejoras en la estabilidad de la salida en ángulos en función al número de sensores MPU-9250 no son lineales o exponenciales cuando se utilizan cuatro sensores. Sin embargo, hay una mejora en la reducción de ruido.

Para el ángulo de guiñada, hubo una mejora en la diferencia cresta-vaguada de 0.0106367 radianes. Para el ángulo de cabeceo, la mejora fue de 0.0067242 radianes y para el ángulo de balanceo, 0.0201209 radianes.

## **7 RECOMENDACIONES**

Para la robótica en general, múltiples IMU son más útiles en situaciones en las que sus partes se mueven entre sí (por ejemplo, un brazo de robot con diferentes sensores para cada dedo). Si los sensores se colocan en diferentes ubicaciones de un vehículo o robot, terminará con más mediciones que estados del sistema (es decir, un espacio de observación sobredefinido). Con el filtro de Kalman extendido, se pueden obtener resultados mucho más precisos con esta configuración.

## **8 FUTUROS TRABAJOS**

Un sistema de medición inercial se puede acoplar a un sistema de posicionamiento global para mayor precisión y confiabilidad; es decir, se puede diseñar un algoritmo de fusión multisensorial GPS-IMU. El alcance del proyecto fue crear un algoritmo de filtro que elimine o reduzca los sesgos de deriva y otras inexactitudes del sistema en ausencia de un GPS u otras señales externas, ya que una señal de GPS a menudo es débil o no está disponible en aplicaciones de robótica autónoma, como en vehículos submarinos que van demasiado profundo.

## 9 BIBLIOGRAFÍA

1. Al Malki, H. H., Moustafa, A. I., & Sinky, M. H. (2021). An Improving position method using Extended Kalman filter. *Procedia Computer Science*, 182, 28-37. <https://doi.org/10.1016/j.procs.2021.02.005>
2. Astroza, R., Ebrahimian, H., & Conte, J. P. (2019). Performance comparison of Kalman-based filters for nonlinear structural finite element model updating. *Journal of Sound and Vibration*, 438, 520-542. <https://doi.org/10.1016/j.jsv.2018.09.023>
3. Bashir, M. A., Malik, F. M., Akbar, Z. A., & Uzair, M. (2020). Kalman Filter Based Sensor Fusion for Altitude Estimation of Aerial Vehicle. *IOP Conference Series: Materials Science and Engineering*, 853, 012034. <https://doi.org/10.1088/1757-899X/853/1/012034>
4. Bayro-Corrochano, E., Lechuga-Gutiérrez, L., & Garza-Burgos, M. (2018). Geometric techniques for robotics and HMI: Interpolation and haptics in conformal geometric algebra and control using quaternion spike neural networks. *Robotics and Autonomous Systems*, 104, 72-84. <https://doi.org/10.1016/j.robot.2018.02.015>
5. Ben Sassi, H., Errahimi, F., & ES-Sbai, N. (2020). State of charge estimation by multi-innovation unscented Kalman filter for vehicular applications. *Journal of Energy Storage*, 32, 101978. <https://doi.org/10.1016/j.est.2020.101978>
6. Bonyan Khamseh, H., Ghorbani, S., & Janabi-Sharifi, F. (2019). Unscented Kalman filter state estimation for manipulating unmanned aerial vehicles. *Aerospace Science and Technology*, 92, 446-463. <https://doi.org/10.1016/j.ast.2019.06.009>
7. Cao, Y., Bu, X., Xu, M., & Yang, H. (2021). Infrared focal plane array attitude measurement method based on adaptive fault-tolerant extended Kalman filter. *Measurement*, 176, 109172. <https://doi.org/10.1016/j.measurement.2021.109172>
8. Chango, W., Cerezo, R., & Romero, C. (2021). Multi-source and multimodal data fusion for predicting academic performance in blended learning university courses. *Computers & Electrical Engineering*, 89, 106908. <https://doi.org/10.1016/j.compeleceng.2020.106908>
9. Dam, A., Verma, A., Pangi, C. T., Raviteja, R., & Prasad, C. S. (2020). Person Following Mobile Robot using Pedestrian Dead-Reckoning with Inertial data of Smartphones.

2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 1-4. <https://doi.org/10.1109/ICCCNT49239.2020.9225292>

10. Elmenreich, W. (2002). *Sensor Fusion in Time-Triggered Systems*.
11. Erickson, A. S. (2021). Joseph G. Gavin, Jr. And MIT's contribution to aerospace in the Apollo era and beyond. *Acta Astronautica*, 181, 167-189. <https://doi.org/10.1016/j.actaastro.2020.06.032>
12. García, J., Molina, J. M., & Trincado, J. (2020). Real evaluation for designing sensor fusion in UAV platforms. *Information Fusion*, 63, 136-152. <https://doi.org/10.1016/j.inffus.2020.06.003>
13. Gressent, A., Malherbe, L., Colette, A., Rollin, H., & Scimia, R. (2020). Data fusion for air quality mapping using low-cost sensor observations: Feasibility and added-value. *Environment International*, 143, 105965. <https://doi.org/10.1016/j.envint.2020.105965>
14. Hehenberger, P., & Zeman, K. (2004). Hierarchical Structuring of Mechatronic Design Models. *IFAC Proceedings Volumes*, 37, 675-680. [https://doi.org/10.1016/S1474-6670\(17\)31181-3](https://doi.org/10.1016/S1474-6670(17)31181-3)
15. Hun, L. C., Yeng, O. L., Sze, L. T., & Chet, K. V. (2016). Kalman Filtering and Its Real-Time Applications. En *Real-time Systems*. IntechOpen. <https://doi.org/10.5772/62352>
16. Kaniewski, P. (2020). Extended Kalman Filter with Reduced Computational Demands for Systems with Non-Linear Measurement Models. *Sensors*, 20(6), 1584. <https://doi.org/10.3390/s20061584>
17. Kim, J. Y., & Lee, W. H. (2019). Walking Simulation for VR Game Character Using Remote Sensing Device Based on AHRS-Motion Recognition. *IEEE Access*, 7, 19423-19434. <https://doi.org/10.1109/ACCESS.2018.2878237>
18. Kok, M., Hol, J. D., & Schön, T. B. (2017). Using Inertial Sensors for Position and Orientation Estimation. *Foundations and Trends® in Signal Processing*, 11(1-2), 1-153. <https://doi.org/10.1561/20000000094>
19. Kruk, D., & Sulowicz, M. (2019). AHRS Based Anti-Sway Tower Crane Controller. *2019 20th International Conference on Research and Education in Mechatronics (REM)*, 1-6. <https://doi.org/10.1109/REM.2019.8744117>

20. Larey, A., Akin, E., & Klein, I. (2020). Multiple Inertial Measurement Units—An Empirical Study. *IEEE Access*, 8, 75656-75665. <https://doi.org/10.1109/ACCESS.2020.2988601>
21. Liu, C., Cheng, S., & Li, K. (2020). Position and Speed Measuring Method of Maglev Train Based on Federal Kalman Filter and Information Fusion. *Journal of Physics: Conference Series*, 1621, 012067. <https://doi.org/10.1088/1742-6596/1621/1/012067>
22. Liu, G.-X., Shi, L.-F., Xun, J.-H., Chen, S., Zhao, L., & Shi, Y.-F. (2018). An orientation estimation algorithm based on multi-source information fusion. *Measurement Science and Technology*, 29(11), 115101. <https://doi.org/10.1088/1361-6501/aadc4c>
23. Maclachlan, C., & Reid, A. W. (2003). Quaternion Algebras I. En C. Maclachlan & A. W. Reid (Eds.), *The Arithmetic of Hyperbolic 3-Manifolds* (pp. 77-109). Springer. [https://doi.org/10.1007/978-1-4757-6720-9\\_3](https://doi.org/10.1007/978-1-4757-6720-9_3)
24. Madgwick, S., Harrison, A., & Vaidyanathan, R. (2011). Estimation of IMU and MARG orientation using a gradient descent algorithm. *IEEE ... International Conference on Rehabilitation Robotics: [proceedings], 2011*, 5975346. <https://doi.org/10.1109/ICORR.2011.5975346>
25. Meckes, E. S., & Meckes, M. W. (2018). *Linear Algebra*. Cambridge University Press. <https://doi.org/10.1017/9781316823200>
26. Ravi Kumar, D. V. A. N. (2021). Hybrid Unscented Kalman Filter with Rare features for Underwater Target tracking using Passive Sonar Measurements. *Optik*, 226, 165813. <https://doi.org/10.1016/j.ijleo.2020.165813>
27. Sabet, M., Daniali, H., Fathi, A., & Alizadeh, E. (2017). Identification of an Autonomous Underwater Vehicle Hydrodynamic Model Using the Extended, Cubature, and Transformed Unscented Kalman Filter. *IEEE Journal of Oceanic Engineering*, PP, 1-11. <https://doi.org/10.1109/JOE.2017.2694470>
28. Skog, I., & Handel, P. (2011). Time Synchronization Errors in Loosely Coupled GPS-Aided Inertial Navigation Systems. *IEEE Transactions on Intelligent Transportation Systems*, 12(4), 1014-1023. <https://doi.org/10.1109/TITS.2011.2126569>

29. Tan, W., Anglea, T., & Wang, Y. (2018). Analysis of Dead Reckoning Accuracy in Swarm Robotics System. *2018 13th World Congress on Intelligent Control and Automation (WCICA)*, 860-864. <https://doi.org/10.1109/WCICA.2018.8630531>
30. Tannous, H. (2018). *Interactive and Connected Rehabilitation Systems for E-Health: A Ph.D Manuscript*.
31. Ullah, I., Qian, S., Deng, Z., & Lee, J.-H. (2020). Extended Kalman Filter-based localization algorithm by edge computing in Wireless Sensor Networks. *Digital Communications and Networks*. <https://doi.org/10.1016/j.dcan.2020.08.002>
32. Wan, E. A., & Van Der Merwe, R. (2000). The unscented Kalman filter for nonlinear estimation. *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, 153-158. <https://doi.org/10.1109/ASSPCC.2000.882463>
33. Yu, Y., Liu, X., Xu, C., & Cheng, X. (2021). Multi-sensor data fusion algorithm based on the improved weighting factor. *Journal of Physics: Conference Series*, 1754(1), 012227. <https://doi.org/10.1088/1742-6596/1754/1/012227>
34. Zhang, M., Yang, J., Zhao, J., & Dai, Y. (2019). A Dead-Reckoning Based Local Positioning System for Intelligent Vehicles. *2019 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS)*, 513-517. <https://doi.org/10.1109/ICPICS47731.2019.8942565>

## 10 ANEXOS

### 10.1 CONFIGURACIÓN DEL ARDUINO Y DE LOS CUATRO MPU-9250

```
a = arduino('COM3', 'Uno', 'Libraries', 'I2C');
fs = 100; % Frecuencia de muestreo (100 por segundo)

% Registrar los IMUs al arduino "a". Las direcciones
serán asignadas automáticamente
imu1 = mpu9250(a);
imu2 = mpu9250(a);
imu3 = mpu9250(a);
imu4 = mpu9250(a);
```

### 10.2 CALIBRACIÓN E IMPLEMENTACIÓN DEL FILTRO DE KALMAN EXTENDIDO

```
tic;
timer = 60;
magReadings=[];
while(toc < timer)
    % Damos rotaciones de 0° a 360° en el eje x durante
este tiempo
    [accel1,gyro1,mag1] = read(imu1);
    [accel2,gyro2,mag2] = read(imu2);
    [accel3,gyro3,mag3] = read(imu3);
    [accel4,gyro4,mag4] = read(imu4);
    mag = (mag1 + mag2 + mag3 + mag4)/4;
    magReadings = [magReadings;mag];
end

% Media de valores máximos y mínimos de la lectura del
magnetómetro, eje x
magx_min = min(magReadings(:,1));
magx_max = max(magReadings(:,1));
magx_correction = (magx_max+magx_min)/2;
```

```
tic;
stopTimer = 100;
magReadings=[];
while(toc<stopTimer)
```

```

    % Damos rotaciones de 0° a 360° en el eje y durante
este tiempo
    [accel1,gyro1,mag1] = read(imu1);
    [accel2,gyro2,mag2] = read(imu2);
    [accel3,gyro3,mag3] = read(imu3);
    [accel4,gyro4,mag4] = read(imu4);
    mag = (mag1 + mag2 + mag3 + mag4)/4;
    magReadings = [magReadings;mag];
end

```

```

% Media de valores máximos y mínimos de la lectura del
magnetómetro, eje y
magy_min = min(magReadings(:,2));
magy_max = max(magReadings(:,2));
magy_correction = (magy_max+magy_min)/2;

```

```

tic;
stopTimer = 60;
magReadings=[];
while(toc<stopTimer)
    % Damos rotaciones de 0° a 360° en el eje z durante
este tiempo
    [accel1,gyro1,mag1] = read(imu1);
    [accel2,gyro2,mag2] = read(imu2);
    [accel3,gyro3,mag3] = read(imu3);
    [accel4,gyro4,mag4] = read(imu4);
    mag = (mag1 + mag2 + mag3 + mag4)/4;
    magReadings = [magReadings;mag];
end

```

```

% Media de valores máximos y mínimos de la lectura del
magnetómetro, eje z
magz_min = min(magReadings(:,3));
magz_max = max(magReadings(:,3));
magz_correction = (magz_max+magz_min)/2;

```

```

% CORRECCIÓN DE DISTORCIONES POR HIERRO DURO Y SUAVE.
tic;
stopTimer = 60;
magReadings=[];
while(toc<stopTimer)
    [accel1,gyro1,mag1] = read(imu1);
    [accel2,gyro2,mag2] = read(imu2);
    [accel3,gyro3,mag3] = read(imu3);
    [accel4,gyro4,mag4] = read(imu4);

```



```

    D = (mag1 + mag2 + mag3 + mag4)/4;
end
[A,B, expmfs] = magcal(D);

% Ingresar los siguientes valores según hoja técnica del
sensor
RuidoGiroscopio = 3.046e-06; % Ruido en rad/s (varianza)
RuidoAcelerometro = 0.006; % Ruido en m/s^2 (varianza)
viewer = HelperOrientationViewer('Title', {'EKF
MULTISENSOR'});

% Configuración Filtro Extendido de Kalman
FiltroDeKalmanExtendido =
ahrsfilter('SampleRate', imu.SampleRate, 'GyroscopeNoise',
RuidoGiroscopio, 'AccelerometerNoise', RuidoAcelerometro);

tic;
stopTimer = 120;
i = 1;
while(toc < stopTimer)
    [accel1,gyro1,mag1] = readSensorDataMPU9250(imu1);
    [accel2,gyro2,mag2] = readSensorDataMPU9250(imu2);
    [accel3,gyro3,mag3] = readSensorDataMPU9250(imu3);
    [accel4,gyro4,mag4] = readSensorDataMPU9250(imu4);
    accel = (accel1 + accel2 + accel3 + accel4)/4;
    mag = (mag1 + mag2 + mag3 + mag4)/4;
    gyro = (gyro1 + gyro2 + gyro3 + gyro4)/4;
    rotators = FiltroDeKalmanExtendido(accel,gyro,mag);
    for j = numel(rotators)
        viewer(rotators(j));
    end
    rotators_eul = quat2eul(rotators);
    yaw(i) = rotators_eul(1,1);
    pitch(i) = rotators_eul(1,2);
    roll(i) = rotators_eul(1,3);
    i = i+1;
end

```

### 10.3 GRÁFICAS

```

% GRÁFICAS
figure(1)
subplot(3,1,1)
plot(time, yaw, 'LineWidth',1)
title('Guiñada')

subplot(3,1,2)

```

```
plot(time, pitch, 'LineWidth',1)
title('Cabeceo')
```

```
subplot(3,1,3)
plot(time, roll, 'LineWidth',1)
title('Balanceo')
```