

Universidad Nacional de Educación a Distancia

Escuela Técnica Superior de Ingeniería Informática

Máster Universitario en Ingeniería de Sistemas y de Control

Prácticas de Instrumentación y Control

---

# CALIBRACIÓN DE UNA IMU PARA LA IMPLEMENTACIÓN DE ALGORITMOS DE FUSIÓN DE DATOS.

---

*Autor:*

Enrique FLORES MONTOYA

*Directoras:*

Raquel DORMIDO

Natividad DURO

28 de mayo de 2023



# Resumen

---

El presente trabajo describe detalladamente el proceso de calibración una Unidad Inercial de Medida (IMU por sus siglas en inglés). El documento constituye un trabajo preliminar previo a la implementación de algoritmos de fusión de datos tales como el filtro de Kalman aplicados a la estimación de la actitud de vehículos autónomos. El capítulo 1 presenta el contexto del trabajo, el capítulo 2 recoge las base teórica fundamental, el capítulo 3 describe el montaje experimental y, finalmente, en 4 se describe detalladamente el proceso de calibración de los diferentes sensores que integran la IMU.

---



# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Avances en UAVs and MAVs . . . . .	1
1.2. El filtro de Kalman . . . . .	3
1.3. Microcontroladores . . . . .	5
<b>2. Fundamentos Teóricos</b>	<b>9</b>
2.1. Sistemas de referencia . . . . .	9
2.1.1. Sistema de referencia inercial: $(O_I, x_I, y_I, z_I)$ . . . . .	9
2.1.2. Sistema de referencia ejes cuerpo: $(O_b, x_b, y_b, z_b)$ . . . . .	10
2.2. Formalismos para describir la actitud . . . . .	10
2.2.1. Ángulos de Euler . . . . .	10
2.2.2. Cuaternios . . . . .	12
<b>3. Dispositivo experimental</b>	<b>17</b>
3.1. Raspberry Pi Pico . . . . .	17
3.1.1. Especificaciones . . . . .	17
3.1.2. Librerías . . . . .	18
3.1.3. Picoprobe . . . . .	19
3.2. Sensor MPU 9250 . . . . .	19
<b>4. Caracterización de los sensores</b>	<b>21</b>
4.1. Calibración del giróscopo. . . . .	21
4.2. Calibración del acelerómetro . . . . .	23
4.3. Calibración del magnetómetro. . . . .	26



# Capítulo 1

## Introducción

### 1.1. Avances en UAVs and MAVs

Los vehículos autónomos o Unmanned Aerial Vehicles (UAVs) han experimentado un intenso desarrollo en las últimas décadas. Su avance se ha visto impulsado por las mejoras en las baterías y los motores sin escobillas, así como por el desarrollo de nuevos materiales compuestos y el abaratamiento y la miniaturización de la electrónica [1, 2, 3]. Los UAVs se emplean en diferentes aplicaciones, tanto civiles como militares: dragaminas (identificación de minas marinas), cobertura inalámbrica, mantenimiento de carreteras, planificación urbanística, prevención de desastres naturales y usos agrícolas. Se trata de un sector creciente cuyo valor de mercado se esperaba que alcanzase los 127 billones de dólares en 2020 [3]. Existe una amplia variedad de configuraciones de UAV que van desde las aeronaves de ala fija hasta los vehículos de ala batida pasando por dispositivos de ala rotatoria y de tipo globo. Los vehículos de ala fija se utilizan para cubrir grandes áreas de terreno dada su mayor autonomía y alcance. En contraste, los vehículos de ala rotatoria basados en el uso de hélices son más adecuados para interiores y zonas de baja accesibilidad gracias a su mayor maniobrabilidad.

Independientemente de su configuración alar, los UAVs comparten una serie de elementos comunes que se resumen en el diagrama de bloques de la figura 1.1. Todo sistema de vuelo autónomo consta de un vehículo no pilotado, una estación de control en tierra que puede ser autónoma u operada manualmente y un sistema de telemetría y telecomandos que mantiene una comunicación entre la estación en suelo y el vehículo. Dentro de los elementos que conforman el vehículo, podemos distinguir los siguientes subsistemas,

1. **Planta de potencia y sistema de gestión de energía.** La mayoría de los vehículos de tamaño mediano y pequeño son alimentados por baterías. Los vehículos alimentados por baterías se caracterizan por una gran fiabilidad, baja firma térmica y nulas emisiones de contaminantes. Los recientes avances en baterías de Ión Litio-Polímero

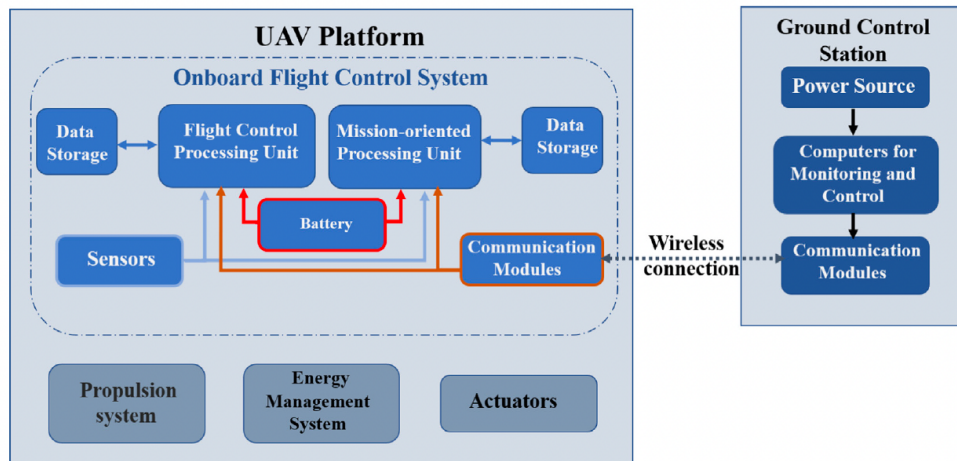


Figura 1.1: Diagrama de bloques de un UAV. (Fuente: [1]).

(LiPo) han permitido lograr autonomías de hasta 90 minutos. Alternativamente, los UAVs eléctricos pueden ser alimentados por pilas de combustible. Las pilas de combustible producen energía eléctrica a partir de hidrógeno y se caracterizan por una mayor densidad energética que las baterías. No obstante, presentan una menor potencia específica [1]. Otra de las ventajas es que la recarga del depósito de combustible es mucho más rápida que la de las baterías. Finalmente, los vehículos de mayor tamaño emplean motores térmicos: motores alternativos o turbinas de gas como fuente de potencia. Un análisis más detallado sobre las fuentes de potencia y las estrategias de gestión de energía puede encontrarse en [1]. De ahora en adelante, nos centraremos en el análisis vehículos eléctricos alimentados por baterías.

2. **Sistema de propulsión y actuadores.** En los UAVs, el sistema de propulsión transforma la energía eléctrica proporcionada por las baterías en energía mecánica que permite al UAV desplazarse y desempeñar su misión. Generalmente, los UAVs utilizan un sistema de propulsión a hélice para mantenerse en vuelo. En las aeronaves de ala rotatoria, el sistema de propulsión proporciona la fuerza de sustentación que permite al vehículo elevarse del suelo. Por el contrario, en las aeronaves de ala fija, la fuerza de sustentación se genera en las alas y el sistema de propulsión se limita a compensar la resistencia aerodinámica permitiendo al vehículo mantener una velocidad de avance. La transformación de la energía eléctrica en energía mecánica se realiza mediante motores sin escobillas de corriente continua. Las aeronaves de ala fija requieren además de un sistema de actuadores que permita accionar las superficies de control aerodinámico como los alerones y el timón de cola. En los UAVs eléctricos de pequeño tamaño esto puede lograrse mediante el uso de servomotores.
3. **Sistema de control a bordo.** El sistema de control de vuelo a bordo consta de, como mínimo, una unidad de procesamiento y control de vuelo, un módulo de comunicación



que permite al vehículo recibir órdenes de la estación en tierra y una serie de sensores que proporcionan al ordenador de a bordo información sobre el entorno y el estado del UAV. Además, el sistema de vuelo puede incluir una unidad de procesamiento de misión, un sistema de almacenamiento de datos y sensores adicionales. Estos últimos suelen estar relacionados con la misión del UAV y son independientes del control de vuelo *i. e.*, cámaras termográficas, medidores de la composición del aire, etc.

Dependiendo del objetivo de su misión, los UAVs modernos pueden ser generalmente operados tanto en modo manual como de manera completamente autónoma. Independientemente de como se realice la planificación de la trayectoria del vehículo a alto nivel, el control y la estabilización en vuelo requieren estimar su posición y su actitud para poder enviar las señales de control a los actuadores. Para ello, estos sistemas integran diferentes sensores tales como medidores de altitud barométricos, IMUs (Inertial Measurement Unit), GPSs y cámaras de baja resolución. No obstante, en general, se necesita procesar los datos de los sensores para estimar el estado del sistema. En particular, para la estimación de la actitud, la mayoría de los vehículos utilizan filtros de Kalman [4, 5, 6].

## 1.2. El filtro de Kalman

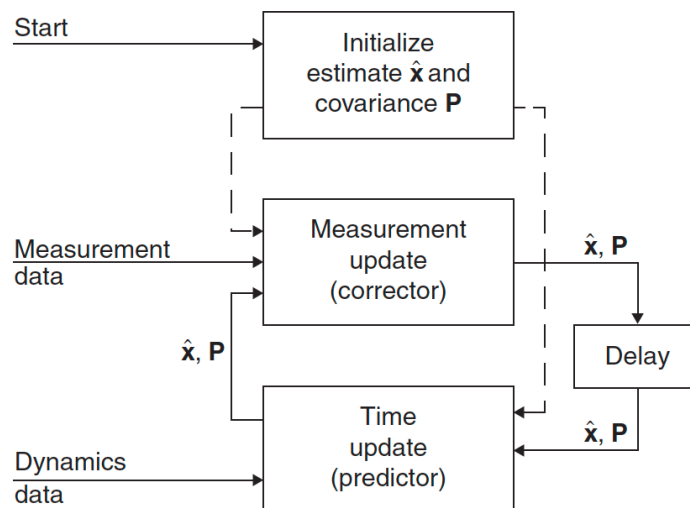


Figura 1.2: Diagrama de flujo de un filtro de Kalman. (Fuente: [7]).

El filtro de Kalman [8] fue desarrollado por Rudolf E. Kalman en 1960. En [9], Kalman describe una solución recursiva para el problema del filtrado lineal de datos discretos. Un filtro de Kalman es un estimador en tiempo real que se utiliza para trazar el estado de un

sistema dinámico con perturbaciones aleatorias a partir de mediciones ruidosas. El filtro es un procedimiento matemático que opera por medio de un mecanismo de predicción y corrección (<https://www.kalmanfilter.net/default.aspx>). En esencia, este algoritmo pronostica el nuevo estado a partir de su estimación previa añadiendo un término de corrección proporcional al error de predicción, de tal forma que este último es minimizado estadísticamente. La inicialización del filtro requiere asignar unos valores iniciales al vector de estados  $x$  y a su matriz de covarianza  $P$ , que representa la incertidumbre de la estimación inicial. Tras la inicialización, el algoritmo itera en bucle entre la fase de predicción y corrección. En gran medida, su popularidad emerge de su buen comportamiento en un gran número de aplicaciones reales.

La utilización de filtros de Kalman esta muy extendida tanto en el campo de la electrónica industrial como en la robótica [10, 11]. Uno de los primeros usos del filtro de Kalman fue el procesamiento de los datos para el guiado y el control de trayectoria de las naves en el programa Apollo. Desde ese momento, debido en gran parte al avance en el cálculo digital, el filtro de Kalman ha sido objeto de una extensiva investigación y aplicación, particularmente en el área de la navegación autónoma y asistida, en rastreo de misiles y en economía. A día de hoy, los filtros de Kalman constituyen una parte esencial de los sistemas de navegación modernos tanto inerciales como por satélite [7, 12]. El filtro se caracteriza por su bajo coste computacional y sus buenas propiedades recursivas. Es uno de los algoritmos de fusión de datos y sensores más comunes y más utilizados.

Debido al extenso volumen de trabajo dedicado a la investigación y el desarrollo de los filtros de Kalman, existe un gran número de variantes con diferentes dominios de aplicación. Entre ellas, podemos citar el filtro de Kalman extendido (EKF por sus siglas en inglés), el filtro de Kalman Unscented (UKF), el filtro de Kalman robusto (RKF) y el filtro de Kalman adaptativo (AKF). Las diferencias entre algunas de estas variantes serán detalladas más adelante. Una revisión sistemática de los diferentes tipos de filtros de Kalman puede encontrarse en el trabajo de Emer y Özbek [13].

No obstante, además del filtro de Kalman, existen otros algoritmos de fusión de datos que también se utilizan en la estimación de la actitud de vehículos. Entre ellos, cabe mencionar el filtro complementario. Este algoritmo realiza un filtrado pasa bajos de la estimación de la actitud obtenida de los datos del acelerómetro y combina estas medidas con un filtrado pasa altos de la estimación de la actitud obtenida mediante la integración directa de las medidas del giróscopo.

El filtro complementario es más simple, ya que no considera ninguna descripción estadística del ruido de las señales y se basa únicamente en un análisis en el dominio de la frecuencia. Por ello, involucra menos operaciones, permitiendo una mayor frecuencia de iteración. Tal y como se demuestra en [14], en algunos escenarios el filtro complementario se comporta como un filtro de Kalman estacionario con ganancia constante. En la práctica, una gran número de aplicaciones emplean filtros complementarios lineales SISO. Por

ejemplo, en [15] se presenta un filtro complementario no lineal para la estimación de la actitud de una aeronave de ala fija no tripulada. El trabajo de Euston *et al.* [15] desarrolla un modelo de la aceleración no inercial de la nave que se utiliza para compensar la salida del acelerómetro y obtener unas lecturas del vector de gravedad sin sesgo. Finalmente, en [16] se compara el comportamiento de un filtro de Kalman extendido y de un filtro complementario para estimar la inclinación de un sistema integrado las medidas del acelerómetro y del giróscopo de una Unidad Inercial de Medida con sensores MEMS.

La unidad de procesado y control de vuelo es la encargada de estimar el estado de la aeronave. Para ello, esta unidad integra los datos recogidos por los distintos sensores mediante un algoritmo de fusión de datos como un filtro de Kalman o un filtro complementario. La información sobre la actitud del UAV es utilizada por el algoritmo de control para mantener la estabilidad y seguir la trayectoria prevista. En un sistema de navegación inercial, los datos recogidos por la IMU (Inertial Measurement Unit) son procesados por el ordenador de vuelo para estimar la actitud del vehículo. La comunicación con los sensores y la implementación del algoritmo de fusión se realizan en la unidad de procesado cuyo soporte físico (hardware) es una computadora o un microcontrolador.

### 1.3. Microcontroladores

Un microcontrolador es un sistema embebido [17] que está programado de manera secuencial y que, a diferencia de un microprocesador, no requiere de un sistema operativo (OS) [18]. Los microcontroladores consisten en un circuito digital síncrono que incorpora una unidad central de procesado (CPU), una memoria que almacena el programa (memoria de programa) y una memoria que contiene los datos que se generan durante la ejecución del código (memoria de datos).

El uso de microcontroladores esta muy extendido en todo tipo de aplicaciones desde dispositivos médicos implantables, sistemas de procesado de control, dispositivos de instrumentación industrial y electrodomésticos [19]. Su utilización industrial se ha visto impulsada por su bajo consumo de potencia y por el abaratamiento de la electrónica debido al desarrollo de la tecnología de impresión de placas de circuitos.

Además, su gran versatilidad ha potenciado el uso de los microcontroladores para el prototipado y la educación [18]. A día de hoy, en el campo de la robótica existen un gran número de herramientas de propotipado accesibles al gran público tales como Arduino [19] o Raspberry pi [20]. Estos dispositivos se caracterizan por su flexibilidad, bajo coste y accesibilidad. Los microcontroladores de uso extendido como el Arduino Uno incorporan conectividad USB para la alimentación, la comunicación y la carga del código (bootloader). Todos estos aspectos, junto con la existencia de bibliotecas *ready to use* han facilitado su extensión y fomentado la filosofía del DIY (Do It Yourself).

De entre todos los microcontroladores de bajo coste, Arduino es la tecnología más extendida, con una mayor comunidad de usuarios, soporte y bibliotecas. Existen diferentes modelos de circuitos impresos Arduino: Uno, Due, Mega, Nano, etc. El modelo más común es el Arduino Uno, que monta un procesador ATmega328 con una frecuencia de reloj de 16 MHz. La placa Arduino Uno cuenta con una memoria de programa de 32 kB, una memoria RAM de 2kB y una memoria EEPROM de 1kB. El microcontrolador posee 14 pines digitales de salida y entrada (I/O digital pins) y 6 pines de analógicos, además de dos pines de potencia a 5 V y 3.3 V. Las otras versiones de Arduino ofrecen un número diferente de entradas y salidas y/o niveles de memoria de programa y RAM marginalmente superiores [19].

Debido a sus limitadas prestaciones de memoria y capacidad de procesamiento, la tecnología Arduino no permite ni el uso extensivo de bibliotecas ni de grandes frecuencias de ejecución. Esto limita enormemente la extensión, y por tanto, la complejidad y la sofisticación del código principal. La falta de memoria interna impide, entre otros, la inclusión de bibliotecas para la implementación de un filtro de Kalman tales como (`Kalman_Filter_Library.h`). La utilización de cámaras de baja resolución que permitan, mediante el procesamiento de imágenes con herramientas de flujo óptico, la determinación de las velocidades de desplazamiento horizontal del vehículo queda totalmente descartada.

Otra de las fuertes limitaciones del microcontrolador Arduino Uno es que su chip, el ATmega328, que pertenece a la familia de microcontroladores AVR desarrollados por ATMEL [21], tiene una arquitectura de 8 bits. Esto implica que todas las operaciones básicas se realizan en paquetes de 8 bits. En aras de la comparación, las computadoras modernas y los celulares tienen típicamente arquitecturas de 64 y 32 bits respectivamente. Además, la mayoría de los microcontroladores del mercado carecen de Unidad de Procesado de Números de coma Flotante (FPU por sus siglas en inglés). Las operaciones con números de coma flotante están implementadas mediante software.

Por todas estas razones, la utilización de microcontroladores Arduino para el prototipado en vehículos autónomos condiciona enormemente sus prestaciones. En consecuencia, es necesario considerar el uso de microcontroladores con mayor capacidad de procesamiento y memoria interna. Además de Arduino, existen otros muchos microcontroladores disponibles en el mercado que se utilizan en aplicaciones de robótica tales como el ESP8266, el MSP430, el ESP32, el BBCmicrobit y el STM32. La Raspberry pi Pico es un microcontrolador recientemente desarrollado y comercializado por Raspberry. Las especificaciones detalladas pueden encontrarse en la sección 3. Su uso está menos extendido que el de Arduino pero su potencia y capacidad de procesamiento son mucho mayores. Esto lo convierte en un microcontrolador más adecuado para aplicaciones más exigentes donde se requieren mayores frecuencias de ejecución y un procesamiento de datos más complejo.

El objetivo del presente trabajo es abordar la sensorización de un sistema de navegación inercial. Para ello, se busca implementar un filtro de Kalman para la estimación de la

actitud de un vehículo autónomo utilizando un microcontrolador Raspberry Pi Pico. La programación de este dispositivo requiere un mayor esfuerzo inicial debido a,

- su mayor complejidad y flexibilidad,
- la necesidad de construir un entorno para el *bootloading*,
- su menor implantación, que se traduce en un menor nivel de documentación.

No obstante, la Raspberry pi Pico es un microcontrolador con prestaciones suficientes para alojar un algoritmo de control, un protocolo de telecomandos-telemetría y una sensorización avanzada. Esto permitirá extender las funcionalidades de un futuro vehículo autónomo desarrollado en base a este microcontrolador. El trabajo busca abarcar todo el proceso de implementación desde el más bajo nivel (acceso a los registros de memoria del sensor) hasta un nivel medio (implementación y comparación entre diferentes filtros). El presente documento se estructura de la siguiente forma: el capítulo 2 presenta los fundamentos teóricos, el capítulo 3 describe el montaje experimental, el capítulo 4 aborda la calibración de los sensores. La implementación del filtro de Kalman para la integración de las lecturas de los sensores se abordará en futuras entregas.



# Capítulo 2

## Fundamentos Teóricos

### 2.1. Sistemas de referencia

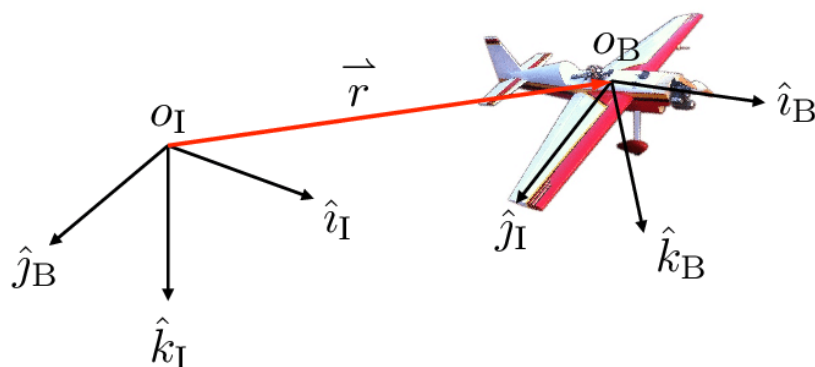


Figura 2.1: Sistemas de referencia inercial y ejes cuerpo. Fuente: [22].

En esta sección se presentan los sistemas de referencia empleados para definir la actitud de un vehículo aéreo. Los sistemas de referencia se denotan, de forma genérica por,  $F(O, x, y, z)$  donde  $(O)$  indica el origen y  $(x, y, z)$  son los ejes del sistema de referencia. En la definición de la actitud de un vehículo es necesario considerar dos sistemas de referencia relevantes.

#### 2.1.1. Sistema de referencia inercial: $(O_I, x_I, y_I, z_I)$

Es un sistema de referencia cartesiano inercial centrado en un punto arbitrario de la superficie terrestre,  $O_I$ . El eje  $x_I$  apunta en la dirección del norte geodésico, el eje  $y_I$  apunta en la dirección del este geodésico y el eje  $z_I$  apunta en la dirección del centro de la tierra. Este sistema de referencia, en ocasiones denotado por NED (North-East-Down) se utiliza

a menudo en para la navegación en aeronaves de pequeño tamaño. Generalmente, se toma el punto de despegue del vehículo como origen del sistema de coordenadas y se utiliza este sistema de referencia para describir la posición, la velocidad y la aceleración del vehículo con respecto a este punto.

### 2.1.2. Sistema de referencia ejes cuerpo: $(O_b, x_b, y_b, z_b)$

El sistema de referencia ejes cuerpo  $(O_b, x_b, y_b, z_b)$  está centrado en el centro de gravedad del vehículo. El eje  $x_b$  apunta hacia adelante y está contenido en el plano de simetría del vehículo, el eje  $y_b$  apunta a estribor (hacia la derecha desde el punto de vista de un observador que mira en la dirección del eje  $x_b$ ) y el eje  $z_b$  apunta hacia abajo, en dirección al suelo, y esta también contenido en el plano de simetría del vehículo.

En los vehículos aéreos, las rotaciones alrededor de los ejes cuerpo se conocen respectivamente como balance, cabeceo y guiñada o *roll*, *pitch* y *yaw* por sus nombres en inglés. Estos ángulos describen la orientación del vehículo con respecto a los ejes inerciales  $(O_I, x_I, y_I, z_I)$ , lo que se conoce comúnmente como actitud del vehículo.

## 2.2. Formalismos para describir la actitud

Las rotaciones permiten pasar de unos sistemas de referencia a otros. Existen diferentes formalismos que se emplean para representar las transformaciones entre sistemas de referencia. El método más intuitivo para describir la orientación o actitud de un cuerpo en el espacio son los ángulos de Euler según la convención de Tait-Bryan. Sin embargo, los ángulos de Euler tienen ciertas limitaciones que hacen que los cuaternios sean el formalismo adoptado en los sistemas de estimación de la actitud. La descripción de la actitud mediante cuaternios puede traducirse a ángulos de Euler, lo que permite obtener un resultado más fácilmente interpretable.

### 2.2.1. Ángulos de Euler

#### Fundamentos

La forma más intuitiva de describir la actitud de un cuerpo en el espacio es mediante los ángulos de Euler. La orientación de un sistema de referencia con respecto a otro puede describirse mediante tres rotaciones sucesivas. Las rotaciones se representan mediante matrices ortonormales  $3 \times 3$ <sup>1</sup>. La composición de rotaciones se obtiene multiplicando matrices

<sup>1</sup>Una matriz  $M$  es ortogonal si y solo si  $M^T = M^{-1}$  y además es ortonormal si cumple que su norma es la unidad, es decir, que el módulo de cada vector que la forma es la unidad.



de rotación individuales y la transformación inversa viene dada por la matriz transpuesta de la matriz de rotación. El orden en el que se realizan las rotaciones alrededor de los diferentes ejes se denomina convención. Nótese que las rotaciones finitas no son conmutativas. Existen diferentes convenciones de ángulos de Euler. La convención utilizada para describir la orientación de aeronaves y vehículos espaciales es la convención de Tait-Bryan  $zyx$ . En otros campos científicos como la Física de Partículas o la Astronomía, son usuales convenciones distintas como la  $zyz$  o la  $zxz$ . En aviónica, los ángulos de Euler se utilizan para describir la relación entre los ejes inerciales  $-I$  y los ejes cuerpo  $-b$ ,

- **Yaw**  $\psi$ : ángulo de rotación alrededor del eje  $z_I$  del sistema de referencia inercial. El rumbo del vehículo viene dado por el ángulo formado entre la proyección el eje  $x_b$  del vehículo sobre el plano formado por los ejes  $x_I$  y  $y_I$  y el eje  $x_I$ . La matriz de rotación asociada es,

$$C_\psi = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

- **Pitch**  $\theta$ : ángulo formado entre el plano  $x_I - y_I$  y el eje  $x_b$ . La matriz de rotación correspondiente a este giro es,

$$C_\theta = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (2.2)$$

- **Roll**  $\phi$ : ángulo de rotación alrededor del eje  $x_b$ . La matriz de rotación asociada a esta transformación viene dada por,

$$C_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \psi & \sin \psi \\ 0 & -\sin \psi & \cos \psi \end{bmatrix} \quad (2.3)$$

La matriz de rotación global se obtiene mediante la composición de las tres rotaciones *yaw*, *pitch*, *roll* y viene dada por la multiplicación de las matrices  $C_\psi$ ,  $C_\theta$  y  $C_\phi$ . De esta manera, la matriz de rotación que permite transformar las componentes de un vector expresadas en ejes inerciales a ejes cuerpo viene dada por,

$$C_I^b = C_\phi C_\theta C_\psi \quad (2.4)$$

donde el orden de las rotaciones tiene lugar de derecha a izquierda. La expresión de la matriz de rotación resultante es,

$$C_I^b = \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \sin \phi \cos \theta \\ \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi & \cos \phi \cos \theta \end{bmatrix} \quad (2.5)$$

dado que la matriz  $C_I^b$  es ortonormal, la transformación de un vector expresado en ejes cuerpo  $(O_b, x_b, y_b, z_b)$  a ejes inerciales  $(O_I, x_I, y_I, z_I)$  se realiza multiplicando por la matriz  $C_b^I$ , que satisface,

$$C_b^I = (C_I^b)^{-1} = (C_I^b)^T \quad (2.6)$$

### Propagación de ángulos de Euler

El vector de velocidad angular del cuerpo con respecto al sistema de referencia inercial expresado en ejes cuerpo viene dado por,

$$\omega = [\omega_x \quad \omega_y \quad \omega_z] = [p \quad q \quad r] \quad (2.7)$$

La relación entre la velocidad angular y la variación temporal de los ángulos de Euler viene dada por,

$$p = \dot{\phi} - \dot{\psi} \sin \theta \quad (2.8)$$

$$q = \dot{\theta} \cos \phi + \dot{\psi} \cos \theta \sin \phi \quad (2.9)$$

$$r = -\dot{\theta} \sin \phi + \dot{\psi} \cos \theta \cos \phi \quad (2.10)$$

y la relación inversa se expresa como,

$$\dot{\phi} = p + (q \sin \phi + r \cos \phi) \tan \theta \quad (2.11)$$

$$\dot{\theta} = q \cos \phi - r \sin \phi \quad (2.12)$$

$$\dot{\psi} = (q \sin \phi + r \cos \phi) \frac{1}{\cos \theta} \quad (2.13)$$

Nótese que la ecuación (2.13) es singular cuando el ángulo de asiento (pitch) es  $\theta = \pm\pi/2$ . Esto limita el uso de los ángulos de Euler para representar la actitud de un cuerpo haciendo que los cuaternios unitarios sean la alternativa escogida por la mayoría de sistemas de integración de la actitud.

## 2.2.2. Cuaternios

### Fundamentos

Los cuaternios son un formalismo que permite describir la actitud de un cuerpo y las rotaciones entre diferentes sistemas de referencia. Un cuaternio esta formado por cuatro componentes,

$$\mathbf{q} = [q_1 \quad q_2 \quad q_3 \quad q_4] \quad (2.14)$$

El cuaternio también puede expresarse como un número complejo con una componente real y cuatro componentes imaginarias,

$$\mathbf{q} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} \quad (2.15)$$

El producto entre dos cuaternios  $\mathbf{q}_1 = [q_0 \ q_1 \ q_2 \ q_3]$  y  $\mathbf{q}_2 = [a \ b \ c \ d]$  puede expresarse como el producto de matrices,

$$\mathbf{q}_1 \cdot \mathbf{q}_2 = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \quad (2.16)$$

Los cuaternios pueden usarse para describir las rotaciones entre sistemas de referencia. Un cuaternio de componentes  $[q_0 \ q_1 \ q_2 \ q_3]$  representa un giro de un ángulo  $\alpha$  alrededor de un eje  $\mathbf{t} = [t_x \ t_y \ t_z]$ ,

$$\mathbf{q} = \left[ \cos \frac{\alpha}{2}, \mathbf{t} \sin \frac{\alpha}{2} \right] \quad (2.17)$$

Sea  $\mathbf{r}_b$  un vector cualquiera expresado en ejes cuerpo, este puede escribirse como un cuaternio cuya componente escalar es nula,

$$\mathbf{r}_b = 0 + r_x\mathbf{i} + r_y\mathbf{j} + r_z\mathbf{k} \quad (2.18)$$

Para expresar el vector  $\mathbf{r}_b$  en ejes inerciales se aplica,

$$\mathbf{r}_I = \mathbf{q} \cdot \mathbf{r}_b \cdot \mathbf{q}^* \quad (2.19)$$

donde  $\mathbf{q}^*$  denota el cuaternio conjugado de  $\mathbf{q}$ . Esta operación también puede expresarse de forma matricial como,

$$\mathbf{r}_I = \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{C} \end{bmatrix} \mathbf{r}_b \quad (2.20)$$

donde  $\mathbf{C}$  es la matriz de rotación entre el sistema de referencia ejes cuerpo y el sistema inercial  $C_b^I$ . La matriz  $\mathbf{C}$  expresada en función de las componentes de  $\mathbf{q}$  viene dada por,

$$\mathbf{C} = \begin{bmatrix} (q_0^2 + q_1^2 - q_2^2 - q_3^2) & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & (q_0^2 - q_1^2 + q_2^2 - q_3^2) & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & (q_0^2 - q_1^2 - q_2^2 + q_3^2) \end{bmatrix} \quad (2.21)$$

de nuevo, la matriz  $\mathbf{C}$  es ortonormal y su transpuesta puede utilizarse para rotar un vector expresado en el sistema de referencia inercial al sistema de referencia ejes cuerpo.

### Propagación de cuaternios

Dado un sistema de ejes cuerpo  $-b$  que rota con un vector velocidad angular  $\omega = [p \ q \ r]$  con respecto al sistema de referencia inercial  $-I$  y sea  $\mathbf{q}$  el cuaternio que describe la orientación del sistema de referencia ejes cuerpo con respecto al sistema de referencia inercial, la variación temporal de  $\mathbf{q}$  viene dada por,

$$\dot{\mathbf{q}} = \frac{1}{2} \mathbf{q} \cdot \boldsymbol{\Omega} \quad (2.22)$$

donde  $\boldsymbol{\Omega}$  se define como,

$$\boldsymbol{\Omega} = \begin{bmatrix} 0 \\ p \\ q \\ r \end{bmatrix} \quad (2.23)$$

La ecuación (2.22) puede expresarse en forma matricial como,

$$\dot{\mathbf{q}} = \frac{1}{2} \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \begin{bmatrix} 0 \\ p \\ q \\ r \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \frac{1}{2} \tilde{S}(\mathbf{q}) \omega \quad (2.24)$$

Esta expresión puede reescribirse para expresarla como el producto entre una matriz  $S(\omega)$  y el cuaternio  $\mathbf{q}$ ,

$$\dot{\mathbf{q}} = \frac{1}{2} \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \frac{1}{2} S(\omega) \mathbf{q} \quad (2.25)$$

La integración de la ecuación (2.25) en el tiempo proporciona la evolución de la orientación del cuerpo descrita por el cuaternio  $\mathbf{q}$  en función de la velocidad angular del sistema  $-b$  con respecto al sistema inercial  $-I$ ,  $\omega$ .

### Relación entre los cuaternios y los ángulos de Euler

El cuaternio  $\mathbf{q}$  que representa la orientación del sistema ejes cuerpo  $-b$  con respecto al sistema inercial  $-I$  puede expresarse en función de los ángulos de Euler  $(\phi, \theta, \psi)$  como,

$$q_0 = \cos \frac{\phi}{2} \cos \frac{\theta}{2} \cos \frac{\psi}{2} + \sin \frac{\phi}{2} \sin \frac{\theta}{2} \sin \frac{\psi}{2} \quad (2.26)$$

$$q_1 = \sin \frac{\phi}{2} \cos \frac{\theta}{2} \cos \frac{\psi}{2} - \cos \frac{\phi}{2} \sin \frac{\theta}{2} \sin \frac{\psi}{2} \quad (2.27)$$

$$q_2 = \cos \frac{\phi}{2} \sin \frac{\theta}{2} \cos \frac{\psi}{2} + \sin \frac{\phi}{2} \cos \frac{\theta}{2} \sin \frac{\psi}{2} \quad (2.28)$$

$$q_3 = \cos \frac{\phi}{2} \cos \frac{\theta}{2} \sin \frac{\psi}{2} - \sin \frac{\phi}{2} \sin \frac{\theta}{2} \cos \frac{\psi}{2} \quad (2.29)$$

Igualmente, a partir del cuaternio  $\mathbf{q}$  que describe la orientación del sistema  $-b$  podemos obtener los ángulos de Euler que definen la actitud del cuerpo,

$$\phi = \arctan \left[ \frac{2(q_2q_3 + q_0q_1)}{1 - 2(q_1^2 + q_2^2)} \right] \quad (2.30)$$

$$\theta = \arcsin [2(q_0q_2 - q_1q_3)] \quad (2.31)$$

$$\psi = \arctan \left[ \frac{2(q_1q_2 + q_0q_3)}{1 - 2(q_2^2 + q_3^2)} \right] \quad (2.32)$$

Las funciones arcotangente implementadas en los ordenadores solo producen resultados entre  $\pm\pi/2$ . Para poder generar todas las orientaciones es necesario sustituir la función `atan` por `atan2`, que tiene en cuenta el signo del numerador y del denominador. De nuevo, estas expresiones son singulares para  $\theta \rightarrow \pm\pi/2$ .



# Capítulo 3

## Dispositivo experimental

El dispositivo experimental utilizado para la implementación de un filtro de Kalman para la estimación de la actitud se representa en la figura 3.1. El circuito se monta sobre una placa de prototipado e incluye dos microcontroladores Raspberry Pi pico y la Unidad de Medición Inercial (IMU por sus siglas en inglés) MPU9250. A continuación se describen en detalle las características de los diferentes elementos del circuito.

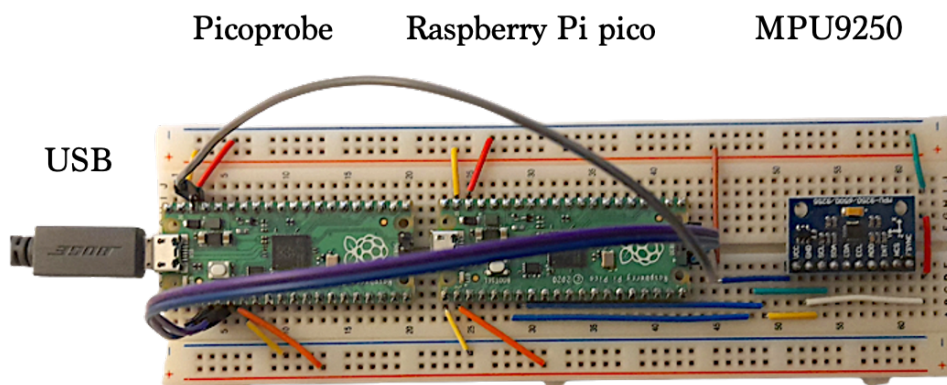


Figura 3.1: Montaje experimental

### 3.1. Raspberry Pi Pico

#### 3.1.1. Especificaciones

La Raspberry pi Pico (<https://www.raspberrypi.com/products/raspberry-pi-pico/>) es un microcontrolador de alto rendimiento de tipo placa basado en el chip RP2040 diseñado por Raspberry Pi [23]. El microcontrolador Raspberry pi Pico (RP $\pi$  Pico) embarca un

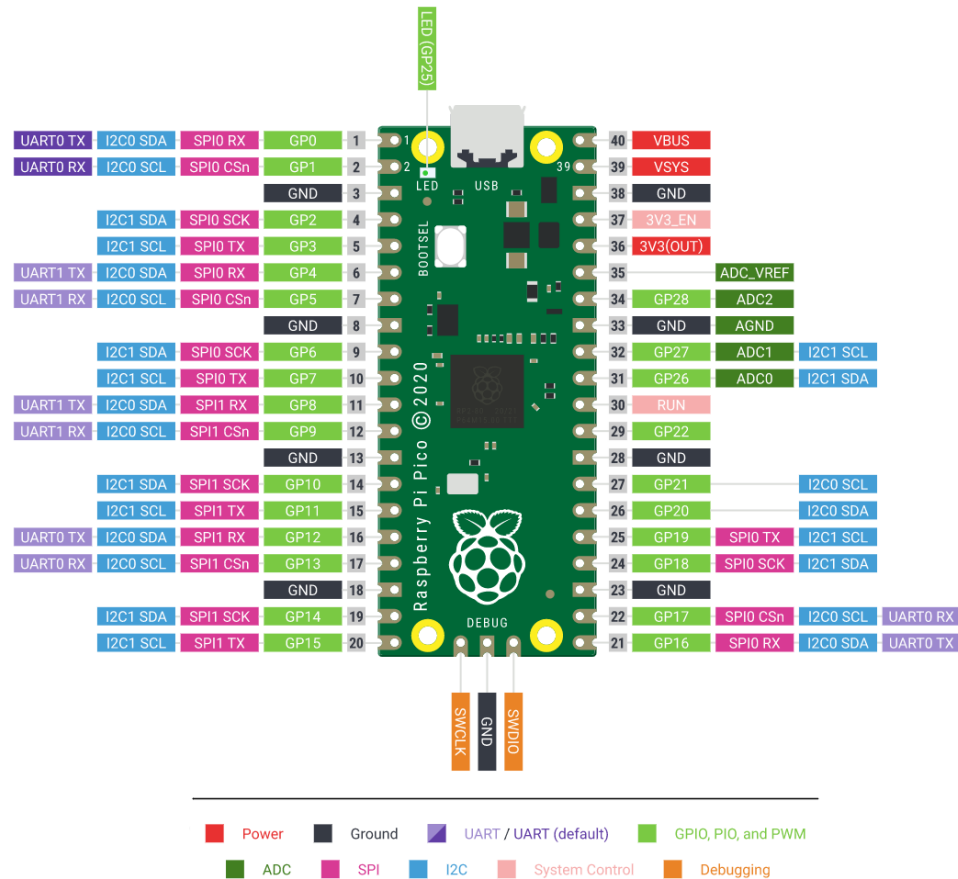


Figura 3.2: Raspberry pi Pico Pinout

procesador de doble núcleo dual-core ARM-CORTEX-M0+ (hasta 133 MHz) con una memoria SRAM interna de 264kB. La arquitectura del procesador ARM-CORTEX-M0+ es de 32 bits. La RP $\pi$  Pico cuenta con 2MB de memoria flash y un chip de *power supply* que soporta voltajes de entrada de entre 1.8 y 5.5 V. El microcontrolador (ver figura 3.2) presenta 26 pines GPIO (General Purpose In-Out) incluyendo pines programables y soportando los protocolos de comunicación I<sup>2</sup>C y SPI.

### 3.1.2. Librerías

La Raspberry pi Pico puede programarse en MicroPython [24] o en C/C++ [25]. En el presente trabajo se ha optado por la programación en C/C++. Para empezar a programar la Raspberry pi Pico en C/C++ es necesario definir un *environment* e instalar una serie de librerías tales como SDK, OpenOCD, GDB y picoprobe. El proceso de instalación y los pasos a seguir para configurar el *environment* están descritos en [25].

La librería más importante es la librería SDK del chip RP2040 (<https://raspberrypi>).



[github.io/pico-sdk-doxygen/](https://github.io/pico-sdk-doxygen/)) [25, 26]. La librería SDK (Software Development Kit) proporciona *headers*, funciones y métodos para escribir programas en dispositivos basados en el chip RP2040 tales como la Raspberry pi Pico. Estos métodos permiten controlar distintos elementos del hardware de la Raspberry pi Pico tales como los buses I2C, SPI y los *timers*.

Un único programa se ejecuta en el dispositivo cada vez bajo el método convencional `main()`. El sistema soporta las librerías estándar de C/C++ e incluye múltiples APIs (Application Programming Interface) para acceder al hardware del RP2040, incluyendo DMA, IRQs y una amplia variedad de pines programables y periféricos de función definida. Así mismo, la SDK provee librerías de alto nivel para manejar temporizadores, USB, sincronizaciones y programación multi-núcleo. La librería SDK utiliza CMake para compilar.

### 3.1.3. Picoprobe

Los archivos compilados pueden cargarse en el microcontrolador de dos formas. En primer lugar, es posible conectar la Raspberry pi Pico como si fuese un dispositivo de almacenamiento USB manteniendo pulsado el botón de BOOTSEL y conectando el cable USB. De esta manera podremos cargar los archivos compilados en la Raspberry copiándolos en la carpeta correspondiente. A continuación, deberemos desconectar el dispositivo y volver a enchufarlo para que el código cargado se ejecute. Este proceso es simple pero tedioso si se desean realizar pequeños cambios del código de manera constante.

La segunda opción es utilizar un segundo controlador Raspberry pi Pico conectado al ordenador por el puerto USB y a la Raspberry por el puerto UART. Esto permite utilizar `openOCD` para cargar los archivos compilados y resetear el microcontrolador desde la línea de comandos del ordenador. Este método, denominado `picoprobe` permite reprogramar el dispositivo de manera mucho más rápida y eficiente y es el que se ha adoptado en el presente trabajo. Los detalles pueden encontrarse en el Apéndice A de [25]. El circuito de la figura 3.1 muestra dos microcontroladores Raspberry pi Pico conectados en la configuración de `picoprobe`.

## 3.2. Sensor MPU 9250

El sensor MPU 9250 [27] es un módulo multi-chip que consiste en en dos *dies* integrados en un único paquete QFN (ver figura 3.3). El primero de los *dies* alberga un giróscopo y un acelerómetro de tres ejes. El segundo *die* contiene el magnetómetro de tres ejes AK8963. Por tanto, el sensor MPU 9250 es un dispositivo de *Motion Tracking* de 9 ejes que combina un giróscopo de 3 ejes, un acelerómetro de 3 ejes y un magnetómetro de 3 ejes así como un Procesador de Movimiento Digital (DMP).

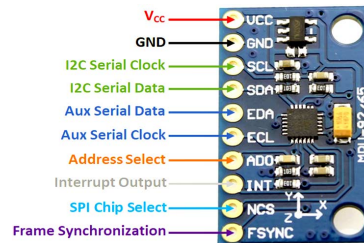


Figura 3.3: Sensor MPU 9250

El MPU9250 puede conectarse a otros dispositivos mediante los buses de comunicación  $I^2C$  (frecuencia de acceso a registros de memoria de 400 kHz) y SPI (frecuencia de acceso a registros de memoria de 1 MHz). El voltaje de operación del dispositivo es de 2.4 – 3.6 V. El sensor cuenta con nueve conversores analógico-digitales de 16 bits para muestrear las lecturas del giróscopo, el acelerómetro y el magnetómetro. Para la detección precisa de movimientos rápidos y lentos, los diferentes sensores cuentan con distintos rangos de medida programables:  $\pm 250$ ,  $\pm 500$ ,  $\pm 1000$ ,  $\pm 2000^\circ/\text{sec}$  (dps) para el giróscopo, y  $\pm 2$  g,  $\pm 4$  g,  $\pm 8$  g,  $\pm 16$  g para el acelerómetro. Si no se especifica lo contrario, en lo que sigue los rangos de medida programados para el giroscopio y el acelerómetro son de  $\pm 1000^\circ/\text{sec}$  y  $\pm 4$  g respectivamente. El magnetómetro tiene un rango de medida fijo de  $\pm 4800\mu T$ . El sensor cuenta con filtros digitales programables integrados, un reloj de precisión con una deriva del 1 % en el rango de temperatura  $40^\circ\text{C} - 85^\circ\text{C}$ , un sensor de temperatura e *interrupts* programables. Se trata de un sensor de bajo coste (a partir de 1.59 euros) lo que lo convierte en un buen candidato para la estimación de la actitud en vehículos autónomos de pequeño tamaño y en aplicaciones académicas y de prototipado.

La comunicación entre el MPU9250 y el microcontrolador se realiza mediante el protocolo de comunicación  $I^2C$  [28]. En el contexto del presente trabajo se ha desarrollado una librería que permite acceder a las lecturas de los sensores que integran la Unidad de Medida Inercial. Esta librería se denomina `IMU.h` y permite inicializar los parámetros de acelerómetro y el giróscopo, así como aplicar la calibración determinada en el capítulo 4 a las medidas de los sensores. El acceso a los registros de memoria del sensor se realiza mediante las funciones `I2CwriteByte` e `I2Cread`.

# Capítulo 4

## Caracterización de los sensores

### 4.1. Calibración del giróscopo.

El MPU9250 utiliza la tecnología Sistema Micro-Electro-Mecánico (MEMS por sus siglas en inglés) para medir la velocidad angular y la aceleración [27]. El término sensor MEMS se aplica a dispositivos miniaturizados fabricados mediante técnicas propias de la industria de la microelectrónica y que permiten medir una fuerza o una magnitud derivada como una aceleración, una presión o una velocidad angular [29, 30]. La transducción mecánica se consigue mediante materiales piezorresistivos, materiales piezoeléctricos o técnicas capacitivas. La medida de la velocidad angular se obtiene gracias al efecto Coriolis que consiste en la aparición de una fuerza ficticia cuando un cuerpo se desplaza con respecto a un sistema de referencia en rotación (ver figura 4.1).

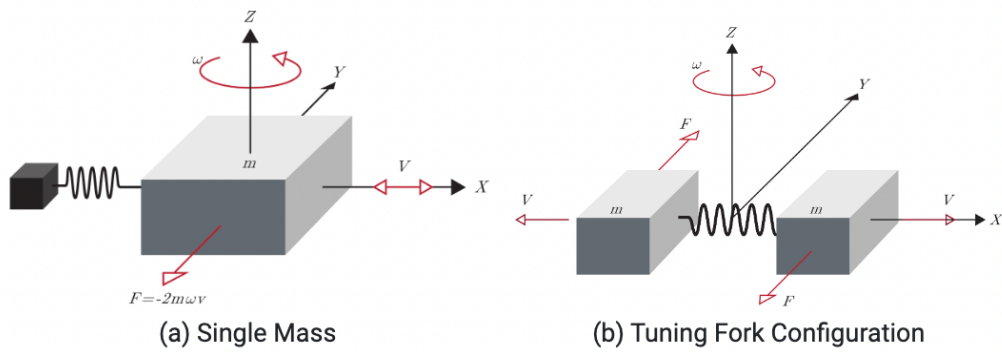


Figura 4.1: Modelo de un giróscopo. Fuente: VECTORNAV: MEMS Operation

El giróscopo proporciona las componentes del vector de velocidad angular,  $\omega$ , en ejes cuerpo,  $[p \ q \ r]$ . En una situación ideal, la orientación del cuerpo expresada en forma de ángulos de Euler podría obtenerse integrando las ecuaciones (2.11) – (2.13) a partir de unas

condiciones iniciales. Sin embargo, debido al ruido presente en las medidas del gir6scopo, la integraci6n directa de la velocidad angular presenta una deriva temporal, esto es, el valor de la integral diverge con respecto al valor real de la magnitud que se busca estimar.

La calibraci6n del gir6scopo consiste en determinar los valores del bias, lo que permite reducir la deriva sistemática de las medidas del sensor. Existen otras t6cnicas de calibraci6n m1s sofisticadas tales como las descritas en [31, 32]. No obstante, estos m6todos de calibraci6n m1s avanzados que permiten corregir la desalineaci6n entre los ejes y los factores de escala est1n fuera del alcance del presente trabajo. Para obtener los valores de bias del gir6scopo se miden los valores de la velocidad angular en los tres ejes con el sensor est1tico. El bias en cada direcci6n se determina calculando el valor medio de la se1al. En este caso, la calibraci6n del gir6scopo se ha realizado utilizando un tiempo de muestreo de 50 ms y un rango de medida de  $\pm 1000^\circ/\text{s}$ . La figura 4.2 representa los valores de la velocidad angular medidos durante un intervalo de tiempo de 30 s con y sin bias.

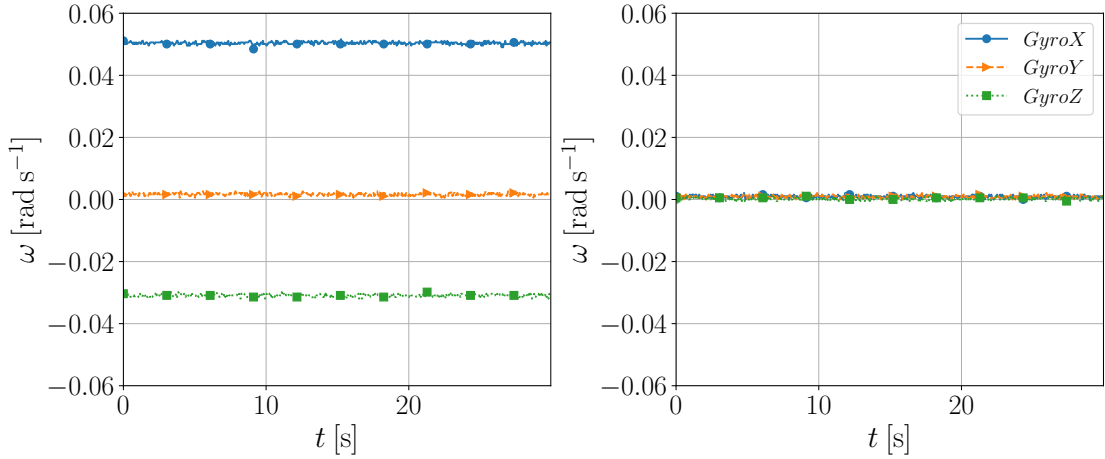


Figura 4.2: Mediciones est1ticas del gir6scopo: con bias (izquierda) y sin bias (derecha)

La tabla 4.1 representa los valores del bias y de la desviaci6n t6pica de las lecturas del gir6scopo en cada eje. La desviaci6n t6pica de las medidas del gir6scopo se utilizar1 m1s adelante para estimar la incertidumbre de la etapa de predicci6n del filtro de Kalman.

Eje	Bias [rad/s]	STD [rad/s]
$\omega_x$	0.0504	$4.199 \times 10^{-4}$
$\omega_y$	0.0017	$4.204 \times 10^{-4}$
$\omega_z$	-0.0309	$4.391 \times 10^{-4}$

Tabla 4.1: Caracterizaci6n del gir6scopo.

## 4.2. Calibración del acelerómetro

El acelerómetro proporciona los valores de la aceleración en las tres direcciones del espacio. El MPU9250 utiliza la tecnología MEMS para medir el vector de aceleración. Debido a su construcción, en reposo, el acelerómetro mide una aceleración de módulo  $g$  según el sentido de la gravedad mientras que, en caída libre, se mide una aceleración nula. En ausencia de aceleraciones externas, la medida del vector de gravedad permite tener una referencia de en qué dirección se encuentra el suelo. Las medidas del acelerómetro están sometidas a errores de sesgo (bias), factores de escala y desalineación de los ejes.

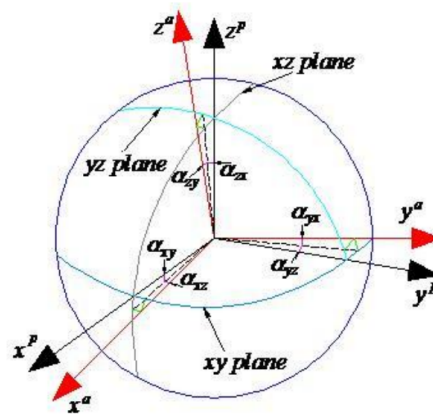


Figura 4.3: Errores de alineamiento entre los ejes del acelerómetro y el sistema de ejes cuerpo. Fuente: [33].

$k_x$	$k_y$	$k_z$	$a_{yz}$ [rad]	$a_{zy}$ [rad]	$a_{zx}$ [rad]	$b_x$ [m s <sup>-2</sup> ]	$b_y$ [m s <sup>-2</sup> ]	$b_z$ [m s <sup>-2</sup> ]
0.9957	0.9898	0.9666	0.01423	0.05225	0.00074	-0.9497	-1.5212	-1.5564

Tabla 4.2: Parámetros de calibración del acelerómetro

Idealmente, las direcciones en las que el sensor mide la aceleración deberían de ser ortogonales entre sí. Sin embargo, debido a las imprecisiones en la construcción de la IMU, a menudo existen desajustes en el alineamiento de los ejes. Esto se traduce en una falta de ortogonalidad entre los ejes del sensor que causa errores en las lecturas del acelerómetro. La figura 4.3 esquematiza los errores de desalineamiento entre los ejes del sensor  $-a$  y el sistema de ejes cuerpo  $-b$ . En esta sección se presenta un método para corregir los errores de alineamiento mediante de una calibración estática del acelerómetro. Los detalles del método pueden encontrarse en [33]. En principio, se requiere conocer un total de seis ángulos de desviación,  $(a_{xy}, a_{xz}, a_{yx}, a_{yz}, a_{zx}, a_{zy})$ , entre los ejes del sensor y los ejes cuerpo para corregir los errores de ortogonalidad. Sin embargo, en [34] se demuestra que el modelo

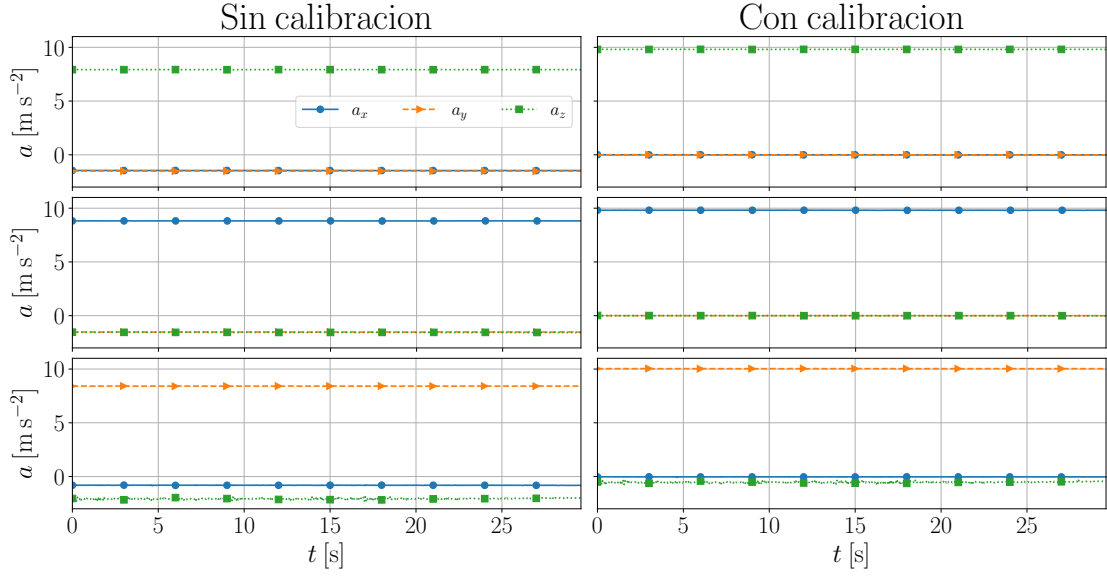


Figura 4.4: Lecturas estáticas del acelerómetro para tres orientaciones diferentes previas a la calibración (izquierda) y posteriores a la calibración (derecha).

de error puede simplificarse si se asume que el eje  $x_b$  coincide con el eje  $x_a$ . Las lecturas del sensor pueden modelarse de acuerdo al siguiente modelo de error,

$$v_i = ST^{-1}g_i + b + \epsilon \quad (4.1)$$

donde  $v_i$  es una lectura bruta,  $i$ , del acelerómetro,  $S$  es una matriz diagonal que contiene los factores de escala,  $T$  es la matriz de desalineación,  $b$  es el bias constante de cada uno de los ejes del sensor y  $\epsilon$  es el ruido. El vector  $g_i$  es el vector de gravedad. La matriz  $T$  permite transformar un vector expresado en ejes acelerómetro  $-a$  a ejes cuerpo  $-b$ . Su expresión es,

$$T = \begin{bmatrix} 1 & -a_{yz} & -a_{zy} \\ 0 & 1 & -a_{zx} \\ 0 & 0 & 1 \end{bmatrix} \quad (4.2)$$

A partir de la ecuación (4.1), un total de nueve parámetros recogidos en (4.3) deben ser determinados para corregir las lecturas del sensor.

$$[k_x \quad k_y \quad k_z \quad a_{yz} \quad a_{zy} \quad a_{zx} \quad b_x \quad b_y \quad b_z] \quad (4.3)$$

La ecuación (4.1) puede expandirse para obtener,

$$\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}_i = \begin{bmatrix} b_x + k_x g_{xi} + a_{yz} k_x g_{yi} - k_x g_{zi} (a_{zy} - a_{yz} a_{zx}) \\ b_y + k_y g_{yi} + a_{zx} k_y g_{zi} \\ b_z + k_z g_{zi} \end{bmatrix}_i \quad (4.4)$$

El subíndice  $-i$  denota la muestra  $i$ . Por tanto, utilizando un número  $i$  de muestras correspondiente al número de incógnitas en cada eje, la ecuación (4.4) puede descomponerse en tres sistemas lineales de ecuaciones,

$$\begin{bmatrix} v_{z1} \\ v_{z2} \end{bmatrix} = \begin{bmatrix} 1 & g_{z1} \\ 1 & g_{z2} \end{bmatrix} \begin{bmatrix} b_z \\ k_z \end{bmatrix} \quad (4.5)$$

$$\begin{bmatrix} v_{y1} \\ v_{y2} \\ v_{y3} \end{bmatrix} = \begin{bmatrix} 1 & g_{y1} & g_{z1} \\ 1 & g_{y2} & g_{z2} \\ 1 & g_{y3} & g_{z3} \end{bmatrix} \begin{bmatrix} b_y \\ k_y \\ k_{yzx} \end{bmatrix} \quad (4.6)$$

$$\begin{bmatrix} v_{x1} \\ v_{x2} \\ v_{x3} \\ v_{x4} \end{bmatrix} = \begin{bmatrix} 1 & g_{x1} & g_{y1} & -g_{z1} \\ 1 & g_{x2} & g_{y2} & -g_{z2} \\ 1 & g_{x3} & g_{y3} & -g_{z3} \\ 1 & g_{x4} & g_{y4} & -g_{z4} \end{bmatrix} \begin{bmatrix} b_x \\ k_x \\ k_{xyz} \\ k_{xzy} \end{bmatrix} \quad (4.7)$$

donde

$$a_{zx} = \frac{k_{yzx}}{k_y} \quad (4.8)$$

$$a_{yz} = \frac{k_{xyz}}{k_x} \quad (4.9)$$

$$a_{zy} = \frac{k_{xzy}}{k_x} - a_{yz}a_{zx} \quad (4.10)$$

Para determinar los coeficientes de calibración del acelerómetro (4.3) es necesario realizar  $n$  mediciones del campo gravitatorio en  $n$  orientaciones diferentes para las que se conozca el valor teórico de la dirección del vector de gravedad  $\mathbf{g}$ . Siendo  $k$  el número de incógnitas por eje y  $n$  el número de orientaciones evaluadas, existen un total de,

$$\mathcal{C} = \frac{n!}{k!(n-k)!} \quad (4.11)$$

combinaciones de medidas por cada eje para calcular los coeficientes de calibración. Aumentando el número de direcciones evaluadas  $n$  es posible reducir el error en la estimación de (4.3). La tabla 4.2 muestra los parámetros de calibración obtenidos tras haber evaluado cuatro orientaciones del sensor. Nótese que los valores de los factores de escala son adimensionales y pueden utilizarse para reescalar el módulo total del vector gravedad.

La figura 4.4 representa las lecturas del sensor en tres orientaciones distintas antes y después de aplicar la calibración. En las orientaciones evaluadas, el vector de la gravedad apunta según la dirección de uno de los ejes del acelerómetro. Por ello, las lecturas del acelerómetro en las otras dos direcciones deberían ser nulas. La calibración del sensor permite reducir de manera muy significativa los errores de bias y de ortogonalidad ya que el módulo de la aceleración se mantiene casi constante e igual a  $g$  en las orientaciones evaluadas.

### 4.3. Calibración del magnetómetro.

Localización	Latitud [N]	Longitud [W]	$h_x^I$ [nT]	$h_y^I$ [nT]	$h_z^I$ [nT]	$\delta h_x^I$ [nT]	$\delta h_y^I$ [nT]	$\delta h_z^I$ [nT]
Madrid	40° 25' 10"	3° 41' 31"	25749.4	97.1	36931.8	131	94	157
Toulouse	43° 36' 16"	1° 26' 39"	23869.8	637.5	39984.3	131	94	157

Tabla 4.3: Componentes del vector del campo magnético terrestre en ejes inerciales (NED) para dos localizaciones de referencia.

El magnetómetro actúa como una brújula, proporcionando las tres componentes del vector del campo magnético terrestre medidas en ejes cuerpo  $-b$ .

$$\mathbf{h}_b = [h_x, h_y, h_z]_b \quad (4.12)$$

El vector en ejes cuerpo debe ser transformado a ejes inerciales para poderlo comparar con la dirección del campo magnético terrestre en esa posición, que es un vector de referencia conocido. La dirección del vector del campo magnético terrestre varía a lo largo de la superficie del planeta pero puede considerarse constante para pequeños desplazamientos. A partir de las coordenadas de un punto de la superficie terrestre es posible obtener los valores de las componentes del vector del campo magnético utilizando herramientas tales como NCEI Geomagnetic Calculators. En la tabla 4.3 se recogen las componentes del campo magnético para dos localizaciones distintas junto con sus incertidumbres.

La comparación entre las lecturas del magnetómetro y el valor de referencia del campo magnético puede utilizarse para estimar la actitud del sistema de referencia ejes cuerpo  $-b$ . En contraste con el acelerómetro, el vector de referencia del campo magnético tiene componentes no nulas en el plano  $x_I y_I$  por lo que permite estimar el ángulo de guiñada  $\psi$ . Otra de las ventajas del magnetómetro es que sus medidas no se ven afectadas por las aceleraciones externas, que sí afectan a las lecturas del acelerómetro.

A día de hoy existen diferentes tipos de sensores del campo magnético basados en diferentes tecnologías. Por una parte podemos encontrar los sensores magnetorresistivos anisotrópicos (AMR por sus siglas en inglés). En este tipo de sensores existen elementos resistivos cuya resistencia efectiva cambia cuando son sometidos a un campo magnético. Su sensibilidad al campo magnético depende de la dirección del campo magnético incidente con respecto a la orientación del elemento magnetorresistivo, de ahí la denominación de anisotrópicos. Para la medición del campo magnético es necesario disponer cuatro elementos magnetorresistivos en configuración de puente de Wheatstone y medir el voltaje de salida. Es posible encontrar más detalles sobre esta tecnología en [35, 36].

Por otra parte se encuentran los sensores magnéticos de efecto Hall [37]. Los sensores magnéticos de efecto Hall son dispositivos electrónicos que detectan la presencia de un



campo magnético utilizando el efecto Hall, descubierto por el físico estadounidense Edwin Hall en 1879. El efecto Hall se refiere al fenómeno que consiste en la aparición de una diferencia de potencial en un conductor cuando se coloca en un campo magnético perpendicular y fluye una corriente a través de él. Los sensores de efecto Hall consisten en una delgada tira de material semiconductor (como arseniuro de galio o silicio) a través de la cuál se hace pasar una corriente eléctrica. Al aproximar el sensor a un campo magnético perpendicular a la dirección de la corriente, se generará un voltaje proporcional al producto de la fuerza del campo magnético y la corriente. Conociendo el valor de la corriente, es posible calcular la fuerza del campo magnético. De entre las diferentes tecnologías de detección del campo magnético, los sensores de efecto Hall son los más extendidos. Una de las principales razones es que es posible producir sensores de gran calidad utilizando las técnicas convencionales de construcción de circuitos integrados. Esto permite integrar el sensor dentro de un chip de manera sencilla y barata. Los sensores de efecto Hall se utilizan ampliamente en una variedad de aplicaciones, incluyendo sistemas de control automotriz e industrial, robótica y electrónica de consumo. La IMU MPU9250 utiliza un sensor de tipo efecto Hall para medir las componentes del campo magnético.

No obstante, el magnetómetro sufre de una serie de limitaciones que hacen que su calibración sea extremadamente importante para poderlo utilizar de forma efectiva en la estimación de la actitud del vehículo. Una descripción exhaustiva sobre las fuentes de error en los sensores de campo magnético magnetorresistivos puede encontrarse en [35]. De forma resumida, este tipo de sensores están sometidos a errores en los factores de escala asociados a la falta de proporcionalidad entre la salida del sensor y el campo magnético detectado, errores de offset debido a la presencia de campos parásitos, errores debido a la sensibilidad cruzada entre ejes o desalineamiento y, finalmente, a errores debido al ruido. De entre las fuentes de error que causan errores de offset y de desalineamiento podemos destacar los errores de hierro. Los errores de hierro están causados por las perturbaciones magnéticas originadas por la presencia de materiales ferromagnéticos y sistemas electromagnéticos en las proximidades del sensor. Estos errores están causados por los diferentes elementos de la planta donde se instala el sensor y dependen por tanto de su configuración. Esto hace que la calibración del sensor sea diferente en función de si este se instala en un placa de prototipado o en la planta final. Los errores de hierro se clasifican a su vez en errores de hierro duro y blando. Los errores de hierro duro están causados por fuentes de campo magnético cuyas componente del campo magnético son constantes en todas la direcciones. En otras palabras, son perturbaciones del campo magnético generadas por los diferentes subsistemas electrónicos presentes en las proximidades del sensor y que no dependen del campo magnético terrestre. Por otra parte, los errores de hierro blando surgen de la interacción entre el campo magnético terrestre y los elementos ferromagnéticos presentes en las proximidades del sensor. Estas perturbaciones tienen una naturaleza más compleja ya que dependen de la dirección del campo magnético terrestre incidente y, por tanto, de la orientación del sensor.

A continuación se presenta el modelo de error empleado para calibrar el magnetómetro y se describe el procedimiento de calibración. Sea  $\hat{h}$  el vector del campo magnético medido por el sensor y  $h$  el vector del campo magnético real en ejes cuerpo, el modelo de error de las medidas del magnetómetro que permite relacionar ambos términos viene dado por la ecuación (4.13).

$$\hat{h} = SM(A_{si}h + b_{hi}) + b_m + \epsilon \quad (4.13)$$

donde,

- $S$  es la matriz diagonal de factores de escala,

$$S = \text{diag}(s_x, s_y, s_z) \quad (4.14)$$

Los factores de escala permiten modelar los errores de proporcionalidad entre la salida del sensor y las componentes del campo magnético en las tres direcciones del espacio. Además, este término permite normalizar la salida del sensor.

- $M$  es la matriz de desalineamiento, modela los errores de ortogonalidad entre los ejes del sensor y viene dada por

$$M = N^{-1} = [\mathbf{m}_x \ \mathbf{m}_y \ \mathbf{m}_z]^{-1} \quad (4.15)$$

donde  $\mathbf{m}_x$ ,  $\mathbf{m}_y$  y  $\mathbf{m}_z$  son vectores columna de dimensión tres que definen la orientación de los ejes del sensor  $F(O_s, x_s, y_s, z_s)$  con respecto a los ejes cuerpo  $F(O_b, x_b, y_b, z_b)$ .

- $b_m$  representa el sesgo (bias) en la salida del sensor modelado como un *offset* (desplazamiento)

$$b_m = [b_{m,x} \ b_{m,y} \ b_{m,z}]^T \quad (4.16)$$

- $b_{hi}$  modela los errores de desviación magnética de tipo hierro duro. La presencia de campos magnéticos parásitos causados por fuentes de campo magnético presentes en las proximidades del sensor se representa mediante un desplazamiento o sesgo.

$$b_{hi} = [b_{hi,x} \ b_{hi,y} \ b_{hi,z}]^T \quad (4.17)$$

- $A$  es la matriz que modela los errores de hierro suave causados por fuentes externas de campo magnético que cambian dependiendo en función de la orientación del sensor. La matriz  $A$  viene dada por,

$$A_{si} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad (4.18)$$

La ecuación (4.13) puede reescribirse como,

$$\hat{h} = A \cdot h + b + \epsilon \quad (4.19)$$

donde

$$A = SMA_{si} \quad y \quad b = SMb_{hi} + b_m \quad (4.20)$$

La matriz  $A$  combina los errores de factor de escala, desalineamiento y perturbaciones de hierro suave mientras que el vector  $b$  agrupa los errores de sesgo (bias).

Calibracion del magnetometro

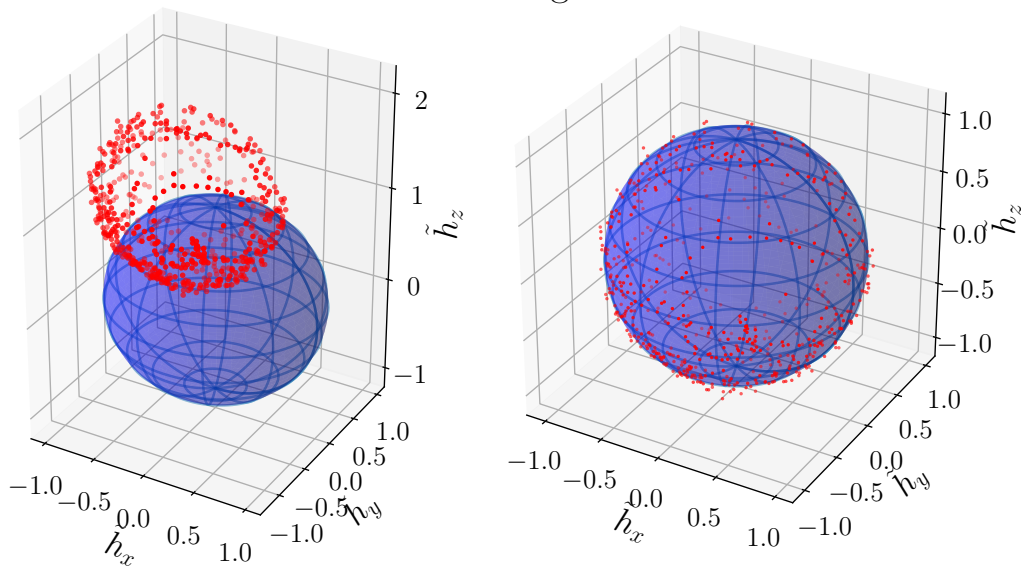


Figura 4.5: Lugar geométrico de los puntos del espacio descrito por el vector del campo magnético medido por el sensor cuando le se somete a rotaciones aleatorias en las tres direcciones del espacio. A la izquierda se representan las lecturas brutas previas a la calibración y una esfera de radio unitario centrada en el origen. A la derecha se representan esas mismas lecturas tras aplicar la calibración descrita. Las lecturas del magnetómetro se han normalizado con el módulo del vector del campo magnético en Toulouse (ver tabla 4.3).

A partir del modelo de error descrito podemos manipular la ecuación (4.20) para aislar el campo magnético  $h$  real en ejes cuerpo.

$$h = A^{-1}(h_m - b) \quad (4.21)$$

done  $h$  es el vector del campo magnético real en ejes cuerpo y  $h_m$  es el vector del campo magnético medido por el sensor y sometido a las diferentes fuentes de error descritas. La calibración consiste en estimar los términos de la matriz  $A$  y del vector  $b$  para poder corregir las lecturas del sensor empleando la ecuación (4.21). De ahora en adelante, las lecturas del

sensor se asumen normalizadas por el módulo del campo magnético de manera que el vector del campo magnético  $h$  satisface,

$$h^T h = 1 \quad (4.22)$$

Substituyendo la ecuación (4.21) en (4.22), se obtiene.

$$(h_m - b)^T (A^{-1})^T A^{-1} (h_m - b) = 1 \quad (4.23)$$

Se denota por  $Q$  el producto  $Q = (A^{-1})^T A^{-1}$ . Es fácil comprobar que  $Q$  es una matriz simétrica ya que es el producto de una matriz multiplicada por su transpuesta. Sustituyendo se obtiene,

$$(h_m - b)^T Q (h_m - b) = 1 \quad (4.24)$$

Expandiendo la ecuación anterior y simplificando se llega a,

$$h_m^T Q h_m - h_m^T Q b - b^T Q h_m + b^T Q b - 1 = 0 \quad (4.25)$$

dado que  $Q$  es una matriz simétrica,  $b^T Q h = h^T Q b$ . Por tanto, podemos reescribir la ecuación (4.25) como

$$h_m^T Q h_m - 2h_m^T Q b + b^T Q b - 1 = 0 \quad (4.26)$$

La ecuación (4.26) puede finalmente escribirse como la ecuación de una forma cuadrática

$$h_m^T Q h_m + h_m^T n + d = 0 \quad (4.27)$$

donde

$$Q = (A^{-1})^T A^{-1} \quad (4.28)$$

$$n = -2Qb \quad (4.29)$$

$$d = b^T Q b - 1 \quad (4.30)$$

El método de calibración del magnetómetro consiste en una calibración tridimensional que emplea el ajuste por mínimos cuadrados y se presenta con detalle en [38, 35, 39]. En ausencia de perturbaciones ni errores de desalineamiento, el módulo del vector del campo magnético medido por el sensor debe ser igual al módulo campo magnético terrestre independientemente de la orientación del sensor. En consecuencia, cuando el sensor se gira en los tres ejes del espacio, el vector del campo magnético medido por el sensor en ejes cuerpo debe describir una esfera de radio igual al módulo del vector del campo magnético. Debido a las diferentes fuentes de error descritas anteriormente, la traza del vector del campo magnético obtenida al girar el sensor formará un elipsoide.

La calibración consiste en encontrar los parámetros  $Q$ ,  $n$  y  $d$  que satisfacen la ecuación (4.27). Se trata de un problema de ajuste de elipsoides utilizando las lecturas del sensor recopiladas para diferentes orientaciones. De entre los diferentes métodos de ajuste de elipsoides existentes es posible citar los algoritmos algebraicos [40, 41], geométricos [42]

y adaptativos [43]. En la presente implementación se ha utilizado la metodología descrita en [40].

La calibración requiere medir el campo magnético en un gran número de orientaciones espaciales diferentes. A diferencia del acelerómetro, que requiere una calibración estática, el magnetómetro puede ser calibrado de forma dinámica, registrando los valores de las lecturas del magnetómetro durante un periodo de tiempo y girando el sensor en las tres direcciones del espacio. No obstante, debido a los ya mencionados errores de hierro duros y suaves, para ser efectiva la calibración debe realizarse directamente en la planta final. Esta operación puede resultar complicada en sistemas reales. Otros métodos de calibración 2D como el presentado por [44] permiten simplificar la fase de adquisición de datos para la calibración.

Tras ajustar los valores de  $Q$ ,  $n$  y  $d$  utilizando los datos experimentales es posible calcular la matriz  $A^{-1}$  y el vector de bias  $b$  aplicando,

$$b = -Q^{-1}n \quad (4.31)$$

$$A^{-1} = \frac{Q^{1/2}}{\sqrt{n^T Q^{-1} n - d}} \quad (4.32)$$

La calibración representada en la figura 4.5 arroja los siguientes valores para la matriz  $A^{-1}$  y el vector de sesgo  $b$ ,

$$A^{-1} = \begin{bmatrix} 1.2535 & -0.0207 & 0.0725 \\ -0.0207 & 1.2612 & -0.0207 \\ 0.0725 & -0.0207 & 1.2737 \end{bmatrix} \quad b = \begin{bmatrix} -0.0724 \\ -0.3814 \\ 1.2595 \end{bmatrix} \quad (4.33)$$

Nótese que los valores del sesgo  $b$  son adimensionales ya que se ha normalizado cada lectura del magnetómetro con el módulo del vector del campo magnético  $h$ .



# Bibliografía

- [1] Mohamed Nadir Boukoberine, Zhibin Zhou y Mohamed Benbouzid. “A critical review on unmanned aerial vehicles power supply and energy management: Solutions, strategies, and prospects”. En: *Applied Energy* 255 (2019), pág. 113823.
- [2] Georgi Valentinov HrISToV, Plamen Zlatkov ZAHArIEV e Ivan Hristov BELoEV. “A review of the characteristics of modern unmanned aerial vehicles”. En: *Acta technologica agriculturae* 19.2 (2016), págs. 33-38.
- [3] Michał Mazur, Adam Wisniewski, Jeffery McMillan et al. “Clarity from above: PwC global report on the commercial applications of drone technology”. En: *Warsaw: Drone Powered Solutions, PriceWater house Coopers* (2016).
- [4] Pifu Zhang et al. “Navigation with IMU/GPS/digital compass with unscented Kalman filter”. En: *IEEE International Conference Mechatronics and Automation, 2005*. Vol. 3. IEEE. 2005, págs. 1497-1502.
- [5] Martin Pettersson. *Extended kalman filter for robust uav attitude estimation*. 2015.
- [6] Nikolas Trawny y Stergios I Roumeliotis. “Indirect Kalman filter for 3D attitude estimation”. En: *University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep 2* (2005), pág. 2005.
- [7] Mohinder S Grewal, Angus P Andrews y Chris G Bartone. “Kalman filtering”. En: (2020).
- [8] Charles K Chui, Guanrong Chen et al. *Kalman filtering*. Springer, 2017.
- [9] Rudolph Emil Kalman. “A new approach to linear filtering and prediction problems”. En: (1960).
- [10] François Auger et al. “Industrial applications of the Kalman filter: A review”. En: *IEEE Transactions on Industrial Electronics* 60.12 (2013), págs. 5458-5471.
- [11] SY Chen. “Kalman filter for robot vision: a survey”. En: *IEEE Transactions on industrial electronics* 59.11 (2011), págs. 4409-4420.
- [12] Mohinder S Grewal y Angus P Andrews. *Kalman filtering: Theory and Practice with MATLAB*. John Wiley & Sons, 2014.

- [13] N Emer y N Özbek. “A survey on Kalman filtering for unmanned aerial vehicles: Recent trends, applications, and challenges”. En: *Proceedings of the International Conference on Engineering Technologies (ICENTE'20), Konya, Turkey*. 2020, págs. 19-21.
- [14] Walter T Higgins. “A comparison of complementary and Kalman filtering”. En: *IEEE Transactions on Aerospace and Electronic Systems* 3 (1975), págs. 321-325.
- [15] Mark Euston et al. “A complementary filter for attitude estimation of a fixed-wing UAV”. En: *2008 IEEE/RSJ international conference on intelligent robots and systems*. IEEE. 2008, págs. 340-345.
- [16] Pengfei Gui, Liqiong Tang y Subhas Mukhopadhyay. “MEMS based IMU for tilting measurement: Comparison of complementary and kalman filter based data fusion”. En: *2015 IEEE 10th conference on Industrial Electronics and Applications (ICIEA)*. IEEE. 2015, págs. 2004-2009.
- [17] Peter Marwedel. *Embedded system design: embedded systems foundations of cyber-physical systems, and the internet of things*. Springer Nature, 2021.
- [18] Dimosthenis E Bolanakis. “A survey of research in microcontroller education”. En: *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje* 14.2 (2019), págs. 50-57.
- [19] Hari Kishan Kondaveeti et al. “A systematic literature review on prototyping with Arduino: Applications, challenges, advantages, and limitations”. En: *Computer Science Review* 40 (2021), págs. 100364.
- [20] Eben Upton y Gareth Halfacree. *Raspberry Pi user guide*. John Wiley & Sons, 2016.
- [21] Wojciech Kunikowski et al. “An overview of ATmega AVR microcontrollers used in scientific research and industrial applications”. En: *Pomiary Automatyka Robotyka* 19.1 (2015), págs. 15-19.
- [22] Jon Mullen, Sean CC Bailey y Jesse B Hoagg. “Filtered dynamic inversion for altitude control of fixed-wing unmanned air vehicles”. En: *Aerospace Science and Technology* 54 (2016), págs. 241-252.
- [23] Raspberry. “Raspberry pi Pico datasheet”. En: *Copyright ©2020-2022 Raspberry Pi Ltd* (2022).
- [24] Raspberry. “Raspberry pi Pico Python SDK: A MicroPython environment for RP2040 microcontrollers”. En: *Copyright ©2020-2022 Raspberry Pi Ltd* (2022).
- [25] Raspberry. “Getting started with Raspberry pi Pico: C/C++ development with Raspberry pi Pico and other RP2040-based microcontroller boards”. En: *Copyright ©2020-2022 Raspberry Pi Ltd* (2022).
- [26] Raspberry. “Raspberry pi Pico C/C++ SDK: Libraries and tools for C++ development on RP2040 microcontrollers”. En: *Copyright ©2020-2022 Raspberry Pi Ltd* (2022).



- [27] InvenSense. “MPU 9250 Product Specification”. En: *PS-MPU-9250A-01* (2014).
- [28] Dominique Paret y Carl Fenger. *The I2C bus: from theory to practice*. John Wiley & Sons, Inc., 1997.
- [29] Robert Bogue. “MEMS sensors: past, present and future”. En: *Sensor Review* (2007).
- [30] Stephen Beeby. *MEMS mechanical sensors*. Artech House, 2004.
- [31] Yuanxin Wu y Ling Pei. “Gyroscope calibration via magnetometer”. En: *IEEE Sensors Journal* 17.16 (2017), págs. 5269-5275.
- [32] Haotian Yang et al. “A novel tri-axial MEMS gyroscope calibration method over a full temperature range”. En: *Sensors* 18.9 (2018), pág. 3004.
- [33] Kian Sek Tee et al. “Triaxial accelerometer static calibration”. En: *Proceedings of the World Congress on Engineering*. Vol. 3. 2011, págs. 25-27.
- [34] Isaac Skog y Peter Händel. “Calibration of a MEMS inertial measurement unit”. En: *XVII IMEKO world congress*. 2006, págs. 1-6.
- [35] Valérie Renaudin, Muhammad Haris Afzal y Gérard Lachapelle. “Complete triaxis magnetometer calibration in the magnetic domain”. En: *Journal of sensors* 2010 (2010).
- [36] PP Freitas et al. “Magnetoresistive sensors”. En: *Journal of Physics: Condensed Matter* 19.16 (2007), pág. 165221.
- [37] Edward Ramsden. *Hall-effect sensors: theory and application*. Elsevier, 2011.
- [38] Manon Kok et al. “Calibration of a magnetometer in combination with inertial sensors”. En: *2012 15th International Conference on Information Fusion*. IEEE. 2012, págs. 787-793.
- [39] Thom Magnusson. *State estimation of uav using extended kalman filter*. 2013.
- [40] Qingde Li y John G Griffiths. “Least squares ellipsoid specific fitting”. En: *Geometric modeling and processing, 2004. proceedings*. IEEE. 2004, págs. 335-340.
- [41] Andrew Fitzgibbon, Maurizio Pilu y Robert B Fisher. “Direct least square fitting of ellipses”. En: *IEEE Transactions on pattern analysis and machine intelligence* 21.5 (1999), págs. 476-480.
- [42] Chi-Lun Cheng y John W Van Ness. “Statistical regression with measurement error”. En: *(No Title)* (1999).
- [43] Ivan Markovsky, Alexander Kukush y S Van Huffel. “Consistent least squares fitting of ellipsoids”. En: *Numerische Mathematik* 98 (2004), págs. 177-194.
- [44] Michael J Caruso. “Applications of magnetoresistive sensors in navigation systems”. En: *Progress in technology* 72 (1998), págs. 159-168.