

**CENTRO DE INVESTIGACIÓN Y DE
ESTUDIOS AVANZADOS DEL INSTITUTO
POLITÉCNICO NACIONAL**

UNIDAD ZACATENCO
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA
SECCIÓN MECATRÓNICA

**“Estimación de la Orientación de un
Cuerpo Rígido: Estudio Experimental
Comparativo”**

Tesis que presenta:
Edwin Rosario Gabriel

Para obtener el grado de:
Maestro en Ciencias

En la especialidad de:
Ingeniería Eléctrica

Directores de Tesis:
Dr. Hugo Rodríguez Cortés
Dr. Martín Velasco Villa

México, D.F.

Noviembre, 2013

[this page intentionally left blank]

Dedicatoria

A mis padres y hermanos.

[this page intentionally left blank]

Agradecimientos

- ♣ Al Departamento de Ingeniería Eléctrica, sección Mecatrónica del CINVESTAV unidad Zácatenco, por brindarme la oportunidad de crecer intelectualmente y compartir conocimientos y experiencias.
- ♣ Al Dr. Hugo Rodríguez Cortés y al Dr. Martín Velasco Villa por todas sus observaciones y recomendaciones para el desarrollo eficiente de esta Tesis.
- ♣ Al pueblo de México, que a través de CONACyT me brindo el soporte económico para realizar este trabajo.

[this page intentionally left blank]

Resumen

En este trabajo se presenta un estudio experimental comparativo de una muestra representativa de los métodos más utilizados en la literatura para estimar la orientación de un cuerpo rígido en el espacio utilizando sensores inerciales de tecnología MEMS, específicamente un acelerómetro, un magnetómetro y un giroscopio que conforman una unidad de medición inercial IMU; este tipo de sensores tienen el inconveniente de presentar ruido en las mediciones, pero por otra parte tienen ventajas como su tamaño reducido, un peso mínimo y un bajo costo que los hacen factibles para diferentes aplicaciones donde conocer la orientación es fundamental, el problema entonces consiste en estimar la orientación utilizando los vectores de referencia dados por los sensores y simultáneamente minimizar los efectos del ruido para que las estimaciones de los ángulos de orientación tengan un mínimo de error.

La primera parte de esta investigación consiste en revisar las diferentes parametrizaciones matemáticas para representar la orientación, así como las ecuaciones de la cinemática rotacional de un cuerpo rígido en el espacio y con esto abordar los esquemas de estimación que se dividen en tres grupos: Métodos Algebraicos, Métodos Estocásticos y los Observadores no Lineales. Cada método de estimación se compara bajo dos parámetros, el tiempo de calculo obtenido mediante su implementación en un sistema embebido basado en un procesador digital de señales DSP, así como la evaluación de un indice de desempeño del error de estimación basado en la norma \mathcal{L}_2 .

Para verificar y comparar de manera objetiva los métodos de estimación abordados, se utiliza como plataforma de experimentación un robot industrial antropomórfico ABB de la serie 2400/16 que tiene una alta resolución en cada articulación y además permite la repetibilidad del experimento. El DSP y la IMU se montan en el órgano terminal del robot que se programa para que realice rotaciones en los tres ejes, con el robot en movimiento las estimaciones calculadas en el DSP se transmiten mediante un canal inalámbrico a una computadora, donde los datos son guardados y comparados con los perfiles de rotación programados en el robot y de esta manera se evalúa la exactitud de cada método de estimación. Los resultados de cada método de estimación son presentados gráficamente para contrastar las ventajas y desventajas de cada uno.

[this page intentionally left blank]

Abstract

In this work we present a comparative experimental study of a representative group of methods used to estimate the attitude of a rigid body using inertial sensor with MEMS technology, specifically an accelerometer, a magnetometer and a gyroscope this is inertial measurement unit IMU. Such sensors have the disadvantage of delivering noisy measurements, but they have advantages such as small size, low weight and low cost. This makes them appealing for different applications where it is essential to know the attitude. The problem is then to estimate the attitude using the reference vectors given by the sensors while minimizing the effects of noise on the estimates.

The first part of this research reviews the different mathematical settings to represent the attitude and the equations of rotational kinematics of a rigid body. Then, the estimation schemes are reviewed, they are classified into three groups: Algebraic Methods, Stochastic Methods and Nonlinear Observers. Each estimation method is compared under two parameters, the computation time obtained through its implementation in an embedded system based on a digital signal processor DSP, and the evaluation of a performance index of error estimate based on the \mathcal{L}_2 norm.

To objectively compare the estimation methods discussed, we used as experimental platform an anthropomorphic industrial robot from ABB 2400/16 series has a high resolution in every joint and allows the repeatability of the experiment.

The DSP and the IMU are mounted at the end effector of the robot that is programmed to perform rotations in all three axes. The robot motion estimates calculated in the DSP are transmitted over a wireless channel to a computer, where the data are stored and compared with profiles programmed into the robot rotation. The results of each estimation method are presented graphically to contrast the advantages and disadvantages of each.

[this page intentionally left blank]

Índice general

Índice de figuras	x
Índice de tablas	xii
Lista de Símbolos	xv
Lista de Acrónimos	xvii
1. Introducción	1
1.1. Antecedentes y Estado del Arte	1
1.2. Sensores Inerciales	5
1.2.1. Acelerómetro MEMS	7
1.2.2. Giroscopio MEMS	8
1.3. Planteamiento del Problema	10
1.4. Objetivos	10
1.5. Organización del Documento	11
2. Cinemática de la Orientación	13
2.1. Propiedades Matemáticas	14
2.2. Representaciones de la Orientación	15
2.2.1. Ángulos de Euler	15
2.2.2. Matriz de Cosenos Directores	17
2.2.3. Cuaterniones	18
2.3. Transformaciones entre Representaciones	19
2.3.1. Transformación de Cuaterniones a DCM	19
2.3.2. Transformación de DCM a Cuaterniones	20
2.3.3. Transformación de Cuaterniones a Ángulos de Euler	20
2.3.4. Transformación de Ángulos de Euler a Cuaterniones	20
2.3.5. Transformación de Ángulos de Euler a DCM	21
2.3.6. Transformación de DCM a Ángulos de Euler	21
2.4. Ecuaciones Cinemáticas de la Orientación	21
2.4.1. EDC para Ángulos de Euler	22

ÍNDICE GENERAL

2.4.2. EDC para la Matriz de Cosenos Directores	23
2.4.3. EDC para Cuaterniones	25
3. Algoritmos de Estimación de Orientación	27
3.1. Mediciones	27
3.2. Algoritmos de Estimación Determinísticos	29
3.2.1. Método TRIAD	29
3.2.2. Método QUEST	31
3.3. Algoritmos de Estimación Estocásticos	35
3.3.1. Fusión Sensorial	36
3.3.2. Filtro de Kalman	40
3.4. Observadores no Lineales	43
3.4.1. Filtros Complementarios	43
3.4.2. Filtro basado en Inmersión e Invarianza	47
4. Resultados Experimentales	51
4.1. Sensores Inerciales	51
4.2. Procesador Digital de Señales	52
4.3. Plataforma de Experimentación	53
4.4. Descripción del Experimento	54
5. Conclusiones y Perspectivas	65
5.1. Conclusiones	65
5.2. Trabajo Futuro	66
A. Código en Matlab	69
B. Configuración del Bus-I²C del DSP	77
B.1. Introducción al módulo I ² C	77
B.2. Reloj del módulo I ² C	79
B.3. Registros del módulo I ² C	80
C. Unidad de Medición inercial Minimu9v2	83
D. Mediciones de la IMU	87

Índice de figuras

1.1.	Vehículos autónomos no tripulados que requieren estimar la orientación.	4
1.2.	Robots móviles que mantienen el equilibrio.	5
1.3.	Principio de funcionamiento y estructura básica de un acelerómetro.	7
1.4.	Principio de funcionamiento y estructura básica de un Giroscopio.	8
1.5.	Diagrama del problema.	10
2.1.	Marcos de referencia.	14
3.1.	Esquema de una IMU.	28
3.2.	Inclinación sobre el eje X - Alabeo.	37
3.3.	Inclinación sobre el eje Y - Cabeceo.	37
3.4.	Diagrama del método del filtro de Kalman.	41
3.5.	Marcos de referencia $\{i\}$, $\{b\}$, $\{e\}$.	44
3.6.	Filtro complementario directo.	45
3.7.	Filtro complementario directo	46
3.8.	Estimación del vector r_3 .	48
4.1.	Unidad de Medición Inercial - IMU.	52
4.2.	Procesador Digital de Señales - DSP.	52
4.3.	Robot ABB IRB 2400/16	53
4.4.	Plataforma de experimentación	54
4.5.	Esquema de comunicación sistema embebido - Computadora.	55
4.6.	Medición de tiempos en Método TRIAD	56
4.7.	Perfil de rotación programado en el robot en Alabeo	58
4.8.	Perfil de rotación programado en el robot en Cabeceo	58
4.9.	Perfil de rotación programado en el robot en Guiñada	58
4.10.	Método TRIAD.	59
4.11.	Método QUEST.	59
4.12.	Método de Fusión Sensorial.	60
4.13.	Filtro de Kalman.	60
4.14.	Filtro Complementario Directo.	60
4.15.	Filtro Complementario Pasivo.	61
4.16.	Filtro de Inmersión e Invarianza.	61

ÍNDICE DE FIGURAS

4.17. Índice de desempeño.	62
4.18. Desempeño vs Tiempo de cómputo.	63
B.1. Multiples dispositivos I ² C conectados.	78
B.2. Diagrama a bloques del módulo I ² C.	78
B.3. Diagrama de reloj del módulo I ² C.	79
D.1. Mediciones del Acelerómetro	87
D.2. Mediciones de Giroscopio	88
D.3. Mediciones del Magnetómetro	88
D.4. Acelerómetro eje X	89
D.5. Acelerómetro eje Y	89
D.6. Acelerómetro eje Z	89
D.7. Giroscopio eje X	90
D.8. Giroscopio eje Y	90
D.9. Giroscopio eje Z	90
D.10.Magnetómetro eje X	91
D.11.Magnetómetro eje Y	91
D.12.Magnetómetro eje Z	91

Índice de tablas

1.1.	Categorías de los dispositivos MEMS.	6
1.2.	Tipos de Giroscopios.	9
1.3.	Sensores magnéticos MEMS.	9
2.1.	Propiedades de las representaciones de la orientación.	13
4.1.	Tiempos de cómputo de cada método de estimación.	56
A.1.	Variables estándar en los códigos de simulación	69
B.1.	Registros del módulo I ² C del DSP	80

ÍNDICE DE TABLAS

Lista de Símbolos

$\{\mathbf{i}\} = \{\hat{\mathbf{i}}_x, \hat{\mathbf{i}}_y, \hat{\mathbf{i}}_z\}$	Marco inercial
$\{\mathbf{b}\} = \{\hat{\mathbf{b}}_x, \hat{\mathbf{b}}_y, \hat{\mathbf{b}}_z\}$	Marco cuerpo
$\{\mathbf{e}\} = \{\hat{\mathbf{e}}_x, \hat{\mathbf{e}}_y, \hat{\mathbf{e}}_z\}$	Marco estimado
ϕ	Alabeo - ángulo en X - Roll
θ	Cabeceo - ángulo en Y - Pitch
ψ	Guinada - ángulo en Z - Yaw
R_i^b	Matriz de rotación del marco $\{\mathbf{b}\}$ a $\{\mathbf{i}\}$
$r_{i,j}$	ij -ésimo elemento de la matriz R_i^b
$\boldsymbol{\Omega} = [p, q, r]$	Vector de velocidad angular
$\mathbf{a} = [a_x, a_y, a_z]$	Vector de aceleración
$\mathbf{m} = [m_x, m_y, m_z]$	Vector de campo magnético
\mathbf{v}_a	Vector de aceleración normalizado
\mathbf{v}_m	Vector de campo magnético normalizado
$\mathbf{q} = [\eta, \epsilon_1, \epsilon_2, \epsilon_3]$	Cuaternión
$\hat{x}_{k-1 k-1}$	Vector de estados inicial
$\hat{x}_{k k-1}$	Vector de estados a “priori”
$\hat{x}_{k k}$	Vector de estados a “posteriori”
ζ_i	Ruido en el Proceso
ξ_i	Ruido en la medición
$\mathbf{S}(\cdot)$	Matriz antisimétrica
Δt	Paso de integración
\mathcal{L}_2	Norma L_2
s_x	Función $\sin(x)$
c_x	Función $\cos(x)$
t_x	Función $\tan(x)$
(e, θ)	Eje y ángulo de Euler
v_k	Vector de observación
$d\theta_i$	Diferencial de ángulo
b_i	Sesgo
μ_i	Ruido aditivo de medición
ρ_i	Pesos de ponderación
\mathbf{P}	Parámetros de Rodrigues

LISTA DE SÍMBOLOS

Lista de Acrónimos

IMU	Unidad de Medición Inercial - (<i>Inertial Measurement Unit</i>)
MEMS	Sistemas Micro Electro Mecánicos - (<i>Micro Electro Mechanical Systems</i>)
NASA	Administración Nacional de Aeronáutica y del Espacio - (<i>National Aeronautics and Space Administration</i>)
TRIAD	Determinación de la Orientación en Tres-ejes - (<i>Tri-Axis Attitude Determination</i>)
HEAO	Observatorio Astronómico de Alta Energía - (<i>High Energy Astronomy Observatory</i>)
QUEST	Estimador de Cuaternión - (<i>QUaternion ESTimator</i>)
REQUEST	Estimador de Cuaternión Recursivo (<i>REcursive QUaternion ESTimator</i>)
ESOQ	Estimador de Cuaternión Óptimo - (<i>ESTimators of the Optimal Quaternion</i>)
SVD	Descomposición en Valores Singulares - (<i>Singular Value Decomposition</i>)
FOAM	Matriz Rápida de Orientación Óptima - (<i>Fast Optimal Attitude Matrix</i>)
EKF	Filtro de Kalman Extendido - (<i>Extended Kalman Filter</i>)
KF	Filtro de Kalman - (<i>Kalman Filter</i>)
MEFK	Filtro de Kalman Multiplicativo - (<i>Multiplicative Kalman Filtering</i>)
AEFK	Filtro de Kalman Aditivo - (<i>Additive Kalman Filtering</i>)
UKF	Filtro de Kalman “ <i>Unscented</i> ” - (<i>Unscented Kalman Filtering</i>)
PF	Filtro de Partículas - (<i>Particle Filter</i>)
VTOL	Despegue y aterrizaje vertical - (<i>Vertical Take-Off and Landing</i>)
MARGS	Sensores Magnéticos, Velocidad Angular y Gravedad - (<i>Magnetic, Angular Rate, and Gravity Sensor</i>)
MAPS	Sensor Multiple de Presión Absoluta - (<i>Manifold Absolute Pressure Sensor</i>)
DLP	Procesamiento Digital de Luz - (<i>Digital Light Processing</i>)
GLV	Rejillas de Válvulas de Luz - (<i>Grating Light Valve</i>)
SAW	Superficie de onda acústica - (<i>Surface Acoustic Waves</i>)
DCM	Matriz de Cosenos Directores - (<i>Director Cosine Matrix</i>)
GPS	Sistema de Posicionamiento Global - (<i>Global Positioning System</i>)
I&I	Inmersión e Invarianza - (<i>Immersion and Invariance</i>)
I²C	Circuitos Integrados Interconectados - (<i>Interconnect Integrated Circuit</i>)
SPI	Interfaz Serial de Periféricos - (<i>Serial Peripheral Interface</i>)

LISTA DE ACRÓNIMOS

DSP	Procesador Digital de Señales - (<i>Digital Signal Processor</i>)
IDE	Entorno de Desarrollo Integrado - (<i>Integrated Development Environment</i>)
FPU	Unidad de Punto Flotante - (<i>Floating Point Unit</i>)
PWM	Modulación por ancho de pulsos - (<i>Pulse width modulation</i>)
ADC	Convertidor Analógico a Digital - (<i>Analog to Digital Converter</i>)
GPIO	Entrada Salida de Propósito General - (<i>General Purpose Input Output</i>)
DMA	Acceso Directo a Memoria - (<i>Direct Memory Access</i>)
RAM	Memoria de Acceso Aleatorio - (<i>Random Access Memory</i>)
ISO	Organización Internacional para la Estandarización - (<i>International Organization for Standardization</i>)
FPGA	Arreglos de Compuertas Programables en Campo - (<i>Field Programmable Gate Array</i>)
SDA	Dato Serial - (<i>Serial Data</i>)
SCL	Reloj Serial - (<i>Serial Clock</i>)

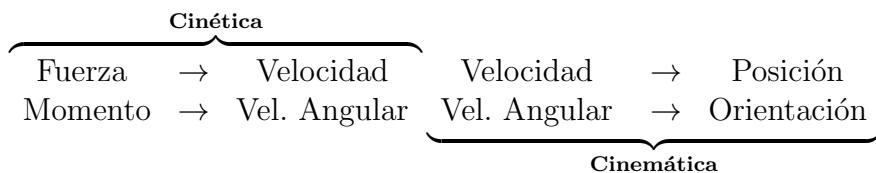
Capítulo 1

Introducción

1.1. Antecedentes y Estado del Arte

El problema de estimar la orientación de un cuerpo rígido en el espacio basado en sensores iniciales ha sido estudiado extensamente en la literatura de control, ingeniería aeroespacial, robótica y en diferentes áreas donde es importante conocer la orientación de un sistema en el espacio con respecto a un marco de referencia conocido, por ejemplo en los vehículos aéreos no tripulados la determinación de la orientación es fundamental para poder establecer sistemas de control que estabilicen y permitan el seguimiento de trayectorias, de aquí que el objetivo de estimar la orientación sea una tarea clave en muchas aplicaciones.

Este problema tiene sus fundamentos matemáticos en áreas de la mecánica clásica como la Cinemática, que estudia el movimiento sin tener en cuenta la causa que lo produce y la Cinética que estudia las fuerzas internas y externas que producen el movimiento en los cuerpos rígidos. La Cinemática a su vez se puede dividir en traslacional, que involucra el movimiento de un punto en el espacio tridimensional y la cinemática rotacional que implica la orientación de un marco de referencia en el espacio tridimensional, el cual abordaremos en este trabajo.



Esencialmente todos los sistemas a ser controlados requieren dos tipos de componentes de *hardware*, los sensores y los actuadores. Los sensores son usados para medir o muestrear el estado de los sistemas y los actuadores para ajustar el estado. Actualmente

CAPÍTULO 1. INTRODUCCIÓN

no existe un sensor con el cual, de forma directa, se pueda medir la orientación de un cuerpo rígido; para ello es necesario plantear diversas técnicas de estimación que fusionen las mediciones de una Unidad de Medición Inercial (IMU - *Inertial Measurement Unit*) para estimar la orientación. Una IMU está constituida por sensores iniciales que son capaces de captar el movimiento y de convertir los efectos de una fuerza inercial en una señal eléctrica, como los Giroscopios, los Acelerómetros y los Magnetómetros, que han sido desarrollados con mayores prestaciones gracias a las tecnologías emergentes de los Sistemas Micro-Electro-Mecánicos (MEMS - *Micro Electro Mechanical Systems*).

Sin embargo, las mediciones de los sensores iniciales son caracterizadas por un alto nivel de ruido, así como un término de desviación o sesgo variante en el tiempo conocido como *bias* [1],[2]. Conocer la orientación entonces involucra dos problemas principales, por un lado se necesita estimar la orientación a través de las mediciones iniciales y observaciones de referencias conocidas, y por otra parte es necesario considerar el filtrado de las mediciones ruidosas, convirtiéndose en un problema de naturaleza no lineal que necesita ser resuelto por enfoques apropiados. En general el problema de estimar la orientación utilizando mediciones con ruido se ha traducido en el desarrollo de diversas técnicas de estimación que se pueden clasificar en tres principales grupos: los métodos Determinísticos que son completamente predictivos si se conocen las entradas, los métodos Estocásticos que consideran variables aleatorias con una matriz de covarianza y los Observadores no lineales que contemplan términos de corrección en función del error de estimación [3].

El interés en el desarrollo de esta área comenzó a mediados del siglo XX cuando la agencia para la Administración Nacional de Aeronáutica y del Espacio (NASA - *National Aeronautics and Space Administration*) comenzó a desarrollar misiones espaciales. En 1964 fue presentado por Harold D. Black [4] el primer método que proporcionaba una solución determinista para estimar la orientación de un cuerpo rígido a partir de dos vectores de observación conocido como el algoritmo TRIAD (*Tri-Axis Attitude Determination*). Debido a su sencillez, se convirtió rápidamente en uno de los algoritmos más populares para determinar la orientación de vehículos espaciales y fue utilizado al menos por una década en muchas misiones espaciales de la NASA, por ejemplo en pequeños satélites astronómicos.

Sin embargo, aunque la carga computacional que demanda el algoritmo TRIAD es simple y sencilla, cuando más de dos vectores de observación están disponibles no es posible usar toda la información para estimar la orientación. Este problema fue abordado por primera vez en 1965 por G. Wahba [5], donde plantea encontrar una matriz ortogonal que minimiza una función de costo cuadrática que contempla los n vectores de observación, usando métodos de optimización.

El siguiente desarrollo significativo fue hecho una década más tarde por Paul Davenport quien mostró que la función de costo cuadrática que se planteaba en el problema de

Wahba para estimar la matriz de orientación podía ser transformada a una función de costo con la parametrización en cuaterniones; esto representó una gran simplificación al problema propuesto por Wahba, ya que los cuaterniones están sujetos a menores restricciones que los nueve elementos de la matriz de orientación, desarrollándose así el llamado Método-Q que tenía mayor carga de cómputo debido a que involucraba resolver los valores y vectores propios de una matriz de 4×4 . El Método-Q no fue publicado por Davenport por razones de confidencialidad, pero más tarde fue reportado en [6], [7] y fue usado para soportar la misión del Observatorio Astronómico de Alta Energía (HEAO - *High Energy Astronomy Observatory*) de la NASA [8].

Con el tiempo el método-Q se convirtió en el inmediato antecesor del algoritmo QUEST (QUaternion ESTimator) presentado en 1978 por M. Shuster [9], [10], el cual determina la orientación basándose en la mejor superposición ponderada de un número arbitrario de vectores de observación. Está basado en un esquema de aproximación que permite calcular el cuaternion óptimo sin tener que resolver explícitamente el problema de valores propios que involucra el Método-Q, logrando de esta forma una reducción significativa en los cálculos. El método QUEST desplazó al método TRIAD, ya que mantiene todas las ventajas computacionales de un rápido algoritmo determinístico mientras produce un resultado óptimo. QUEST dio soporte a las misiones de orientación de la NASA y la historia de su desarrollo puede encontrarse en [11]. Después se desarrollaron algunas variaciones como los métodos ESOQ y ESOQ-2 (EStimators of the Optimal Quaternion) comparados en [7] y los métodos REQUEST (REcursive QUEST) [12] y QUEST Extendido [13]. Por otro lado, los algoritmos SVD (Singular Value Decomposition) [14] y FOAM (Fast Optimal Attitude Matrix) [15] también son descendientes directos del problema de Wahba.

La necesidad de métodos que consideren la dinámica del sistema en la estrategia de estimación recibió gran atención a partir de 1960 cuando R. Kalman presentó una nueva estrategia de filtrado y estimación para sistemas lineales [16]. Entonces surgen los métodos de estimación de orientación estocásticos basados en el Filtro de Kalman Extendido (EKF) que consideran a los vectores de observación como variables aleatorias con ruido Gaussiano. Aunque el Filtro de Kalman (KF) funciona relativamente bien para estimar la orientación, se han desarrollado otras variantes, algunas manteniendo la estructura básica del KF, pero empleando diversas parametrizaciones de la orientación con el fin de proporcionar una mejor convergencia o mejorar otras características de rendimiento, como el Filtro de Kalman Multiplicativo (MEFK) y el Filtro de Kalman Aditivo (AEKF) que son comparados en [17]. Otros enfoques que no requieren la linealización del sistema original y el supuesto del ruido Gaussiano en las mediciones han sido desarrollados también, como ejemplos están el Filtro de Kalman “*Unscented*” (UKF) [18] y el Filtro de Partículas (PF) [19]. Aunque todas estas nuevas técnicas se han desarrollado para resolver el problema de estimación de la orientación, el EKF sigue siendo el más utilizado en aplicaciones prácticas.

Por otro lado, estimar la orientación basándose en los estados de un modelo dinámico

usando la información de entrada y salida del sistema ha sido un enfoque que ha cobrado gran relevancia en las últimas dos décadas, disponiendo actualmente de una importante cantidad de trabajos relacionados con Observadores no Lineales para estimar la orientación, ejemplos propuestos pueden encontrarse en [20], [21], [22]. Recientemente se han desarrollado observadores que recapturan la estructura clásica de los Filtros Complementarios [23], [24] y que además pueden estimar en línea el término de desviación de los sensores, *bias*, lo cual se traduce en una estimación más robusta. Existen también otros enfoques basados en el esquema de Inmersión e Invarianza [25] que han mostrado tener un buen rendimiento.

A lo largo de medio siglo de investigación y desarrollo en esta área, se han propuestos varios enfoques que han mostrado buenos resultados y han sido implementados exitosamente en diferentes aplicaciones y productos que usamos actualmente, sin embargo, estimar la orientación de un cuerpo rígido en el espacio sigue siendo un problema abierto y aún existe interés en aplicar nuevas técnicas de estimación no lineal que permitan aprovechar de mejor forma las propiedades de la cinemática rotacional.

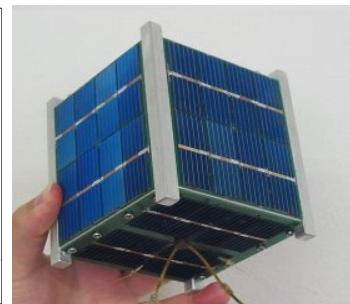
Por otro lado, gracias a los atributos favorables y el potencial de manufactura de los MEMS se ha creado una amplia gama de aplicaciones donde se usan los métodos de estimación de orientación, por ejemplo en los vehículos autónomos no tripulados (UAV) de bajo costo, ya sean aéreos como los helicópteros, aviones y vehículos de despegue y aterrizaje vertical (VTOL), donde conocer la orientación es un requisito clave para el control de estabilización, o vehículos acuáticos y terrestres como el conocido transportador personal SegwayPT®. Otras áreas que también se han desarrollado con el uso de estimadores de orientación están relacionadas con los robots móviles del tipo bípedos, cuadrúpedos y equilibristas donde uno de los objetivos principales es mantener el balance del robot. Los sensores iniciales se han utilizado también en aplicaciones biomédicas para reconstruir el movimiento humano y en sistemas de realidad virtual y videojuegos como complemento a los sensores de visión para lograr interfaces más realistas e interactivas.



(a) Helicóptero. [26]



(b) Cuadricóptero. [27]



(c) Nanosatélite. [28]

Figura 1.1: Vehículos autónomos no tripulados que requieren estimar la orientación.

Paralelo al trabajo desarrollado en robótica y control existe una significativa literatura referente a Sistemas de Referencia de Orientación y Rumbo (AHRS - *Attitude and Heading Reference System*) para aplicaciones aeroespaciales, tal como el control de orientación de nanosatélites, misiles y telescopios espaciales. También se han incorporado este tipo de sensores en aplicaciones de seguridad para automóviles, como es el caso de la activación de las bolsas de aire y en dispositivos de interfaz humana para computadoras. Como se puede ver, las aplicaciones son vastas y conocer la orientación es un requerimiento fundamental para el control de diferentes aplicaciones.



Figura 1.2: Robots móviles que mantienen el equilibrio.

En este trabajo se realiza una revisión limitada de métodos representativos para estimar la orientación basados en mediciones inerciales, con el objetivo de comparar, por una parte, los tiempos de cálculo que requiere cada método para su ejecución implementado en un sistema embebido, y por otra parte, evaluar la eficiencia de los estimadores implementados mediante su comparación a través de un índice de desempeño dado por la norma \mathcal{L}_2 del error de estimación. A lo largo de este trabajo, los términos Filtro y Estimador son usados como sinónimos, porque el ruido está involucrado en las mediciones. Cuando las mediciones son perfectas, es decir, no se tiene ruido en ellas entonces el término Observador es usado.

1.2. Sensores Inerciales

El reciente desarrollo y comercialización de sensores inerciales basados en tecnología MEMS, conocidos también como MARGs (Magnetic, Angular Rate, and Gravity sensor), ha permitido el desarrollo de unidades de medición inercial de bajo costo, de peso reducido y con un mínimo de consumo energético que pueden ser integrados fácilmente en distintas aplicaciones. MEMS es un término que fue acuñado alrededor de 1989 por el prof. R. Howe [32] para describir un campo de investigación emergente, donde los componentes mecánicos como los *cantilevers* o membranas, son manufacturados a

una escala por debajo de los $100 \mu m$. Mientras que los circuitos microelectrónicos son estructuras sólidas y compactas, los MEMS tienen agujeros, cavidades, canales, *cantilevers*, membranas, etc. que imitan partes mecánicas que son integradas directamente con la electrónica, incluyendo la adquisición y el almacenamiento de datos, el filtrado de las señales y las interfaces de comunicación, etc. El énfasis de los MEMS basados en silicio es un claro resultado del vasto conocimiento sobre este material, pero pueden ser fabricados también en polímeros, en vidrio, en cuarzo o incluso en metales.

El crecimiento del número de productos comercializados que hacen uso de los sensores MEMS ha sido exponencial y actualmente pueden ser divididos en seis categorías, ver Tabla 1.1.

Tabla 1.1: Categorías de los dispositivos MEMS.

Categoría de Producto	Ejemplos
Sensores de Presión	Sensores de presión en motores de combustión interna (MAPS), Presión en neumáticos, Sensores de presión arterial, etc.
Sensores Inerciales	Acelerómetros, Giroscopios, Magnetómetros, Sensores de choque, etc.
Microfluidos/bioMEMS	Boquillas de impresoras de inyección de tinta, Sistemas de micro-bio-análisis, Chips para ácido desoxirribonucleico (ADN), etc.
MEMS ópticos/MOEMS	Arreglo de micro-espejos para proyección (DLP), Arreglo de micro-rejillas para proyección (GLV), Interruptores de fibra óptica, Óptica adaptativa,etc.
RF MEMS	Inductores de alto Q, Antenas, Filtros, etc.
Otros	Relevadores, Micrófonos, Almacenamiento de datos, Juguetes, etc.

Generalmente los sensores inerciales presentan distorsiones o errores en las mediciones asociadas principalmente a una desviación conocida como *bias*, también se deben a efectos de temperatura y errores de calibración que se modelan de manera conjunta como una secuencia de ruido blanco asociado a una variable aleatoria. El término *bias* se interpreta como una salida promedio del sensor cuando éste no está bajo el efecto de ninguna fuerza, para mayor detalle consultar el reporte técnico de Woodman [33].

Para el desarrollo de este trabajo utilizamos un acelerómetro, un giroscopio y un magnetómetro, cada uno de tres ejes, de los cuales a continuación explicaremos brevemente el principio físico en el que basan su funcionamiento. Para mayores especificaciones sobre los sensores inerciales MEMS consultar [34] y sobre las tecnologías MEMS los libros de Beeby [35], Gardner [36] y Kaajakari [37] por citar algunos.

1.2.1. Acelerómetro MEMS

Los acelerómetros convierten la fuerza de aceleración en una señal eléctrica, esta fuerza puede ser estática como la fuerza de gravedad, o puede ser dinámica causada por el movimiento y las vibraciones del acelerómetro. El primer acelerómetro MEMS fue diseñado el 1979 en la universidad de Stanford, pero tomó más de 15 años para que estos dispositivos fueran producidos en grandes volúmenes [38]. En 1990 los acelerómetros MEMS revolucionaron una gama de aplicaciones industriales y comerciales, incluyendo el análisis de vibraciones, la protección de los discos duros de las computadoras, la detección de choque para el despliegue de las bolsas de aire (*air-bag*) en los automóviles, etc. Los acelerómetros son los dispositivos MEMS más extensamente usados después de los sensores de presión gracias a su reducido volumen, peso, costo y a su rendimiento mejorado que está tomando rápidamente ventajas en nuevas aplicaciones.

Los acelerómetros están basados en la segunda ley de Newton, en el cual una masa de prueba en voladizo m se desplaza bajo una fuerza inercial, utilizan un sensor capacitivo para medir el desplazamiento o también puede ser un sensor piezoelectrónico que contiene una estructura cristalina que se deforma por la fuerza de aceleración, y un circuito apropiado para el acondicionamiento de la señal, ver Figura 1.3. La masa es una barra de silicio independiente y el sistema de resortes es implementado por cuatro cuerdas, las cuales unen cada esquina de la masa.

La medición del desplazamiento de la barra independiente está basado en un capacitor diferencial con un plato paralelo móvil; esta configuración permite conocer la dirección del movimiento a través de los platos fijos, en un lado incrementa la capacitancia mientras que en el otro decrece.

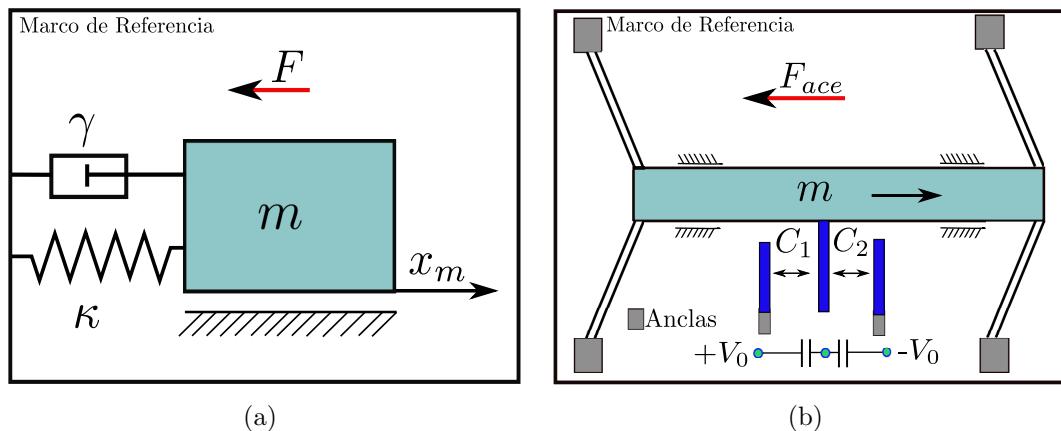


Figura 1.3: Principio de funcionamiento y estructura básica de un acelerómetro.

1.2.2. Giroscopio MEMS

Los giroscopios se basan en el fenómeno de la aceleración de Coriolis para poder medir la velocidad angular. Debido a este hecho, un giroscopio tiene una estructura mecánica que soporta una masa de prueba m que es excitada a una frecuencia y amplitud constante mediante un oscilador, el cual es necesario para generar la fuerza de Coriolis.

En la Figura 1.4 se muestra un modelo simplificado de un giroscopio, la masa de prueba se excita para oscilar a lo largo del eje x , cuando se presenta una rotación alrededor del eje z se suman ambas energías y se presenta una oscilación a lo largo del eje y cuya amplitud es proporcional a la velocidad de rotación.

Se pueden usar sensores capacitivos o piezoresistivos para detectar el movimiento en el eje de sensado, pero también es posible incorporar un lazo de realimentación de fuerza, cualquier movimiento a lo largo del eje y es detectado y una fuerza es aplicada para contrarrestar este movimiento, la magnitud de la fuerza requerida es entonces la señal de medición de la velocidad angular.

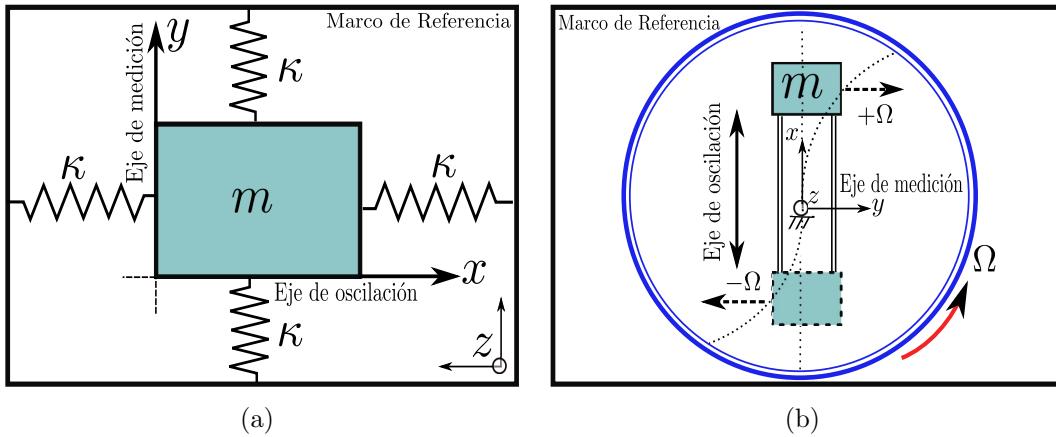


Figura 1.4: Principio de funcionamiento y estructura básica de un Giroscopio.

Una característica que tienen los giroscopios es que presentan una ligera desviación conocida como *bias*, que corresponde a una salida promedio del giroscopio cuando no está experimentando ninguna rotación [33]. Es un desplazamiento del valor verdadero de la salida medida en $[\text{°}/\text{hrs}]$. Un error constante de bias ς , puede causar un error angular en la integración de las velocidades angulares que crece linealmente con el tiempo $\theta(t) = \varsigma t$. La constante de error bias de un giroscopio puede ser estimada tomando un promedio a largo plazo de la salida del giroscopio sin movimiento. Una vez que se conoce el bias se compensa simplemente restandolo de la salida. Además de los giroscopios MEMS, existen giroscopios basados en otros principios físicos que tienen sus ventajas y desventajas dependiendo de su aplicación, la Tabla 1.2 muestra brevemente las características de cada uno.

Tabla 1.2: Tipos de Giroscopios.

Tipo	Comentarios
Mecánicos	Se basan en el principio de conservación de momento angular, contienen partes móviles que pueden presentar fricción, son sensibles al cambio de temperatura por la dilatación que puedan presentar sus componentes.
Ópticos	Usan la interferencia de la luz para medir la velocidad angular y están basados en el efecto Sagnac. La exactitud del giroscopio óptico depende de la longitud de la trayectoria de la luz, lo cual es una restricción al tamaño del dispositivo.
MEMS	Hacen uso del efecto de Coriolis, aún no pueden igualar la precisión de los giroscopios ópticos pero tienen otras ventajas en cuanto a peso, tamaño, fiabilidad, etc. Son sensibles al cambio de temperatura.

Magnetómetro MEMS

Los magnetómetros son sensores ampliamente usados en la industria de la aviación, la oceanografía y otras áreas donde medir la fuerza y la dirección del campo magnético es necesario, permitiendo encontrar la dirección del polo norte. Los magnetómetros no son muy exactos debido a que son afectados por las perturbaciones locales causadas por objetos magnéticos cercanos, sus datos sin embargo se pueden fusionar con los datos del giroscopio para mejorar la precisión de la orientación calculada. Existen varios tipos de magnetómetros MEMS que se pueden clasificar de acuerdo a su principio físico [39], en la Tabla 1.3 se describen brevemente.

Tabla 1.3: Sensores magnéticos MEMS.

Tipo	Comentarios
Galvánicos	Sensores de Efecto Hall, presentan buena linealidad pero baja sensitividad.
Conductivos	Sensores Magnetoresistivos, presentan alta sensitividad y son apropiados para medir campos magnéticos intensos.
Voltaicos	Magnetodiodos , Magnetotransistores, están basados en el efecto <i>fluxgate</i> .
Acústicos	Dispositivos de superficie de onda acústica (SAW), Necesitan un material magnetoelástico.
Resonantes	Magnetómetros basados en el principio de la fuerza de Lorentz y no requieren materiales magnéticos.

1.3. Planteamiento del Problema

Cuando se desea implementar algún método para estimar la orientación, la pregunta inmediata que surge es ¿Cuál método es el mejor?, de igual forma para aplicaciones donde se requiere tener una frecuencia de muestreo mayor, ¿Qué método es más rápido? o ¿Cuál es más fácil de programar?, estas son algunas de las preguntas que se proponen responder en este trabajo de investigación.

Basándonos en los antecedentes y el marco teórico anterior, primero necesitamos definir dos marcos de referencia, el marco inercial o marco fijo $\{i\}$, donde el origen será nuestro punto de referencia para todas las mediciones, y el marco de referencia cuerpo o marco móvil $\{b\}$, que estará adosado al cuerpo rígido en movimiento.

El problema consiste en implementar y evaluar distintos métodos para estimar la orientación de un cuerpo rígido en el espacio, utilizando la combinación de sensores iniciales y modelos matemáticos que recogen componentes vectoriales en ambos marcos de referencia, y los fusionan en algoritmos para determinar la orientación de un cuerpo rígido en el espacio en alguna de sus parametrizaciones.



Figura 1.5: Diagrama del problema.

Esto implica conocer al menos dos vectores $v_i, v_b \in \mathbb{R}^3$ de referencia para calcular una matriz de rotación $R_i^b \in \mathbb{R}^{3 \times 3}$ tal que, $v_i = R_i^b v_b$ y además considerar el filtrado del ruido de las mediciones para lograr una estimación adecuada, todo esto implementado en un sistema embebido.

1.4. Objetivos

El objetivo principal en esta tesis es realizar un estudio experimental comparativo de diferentes estrategias y algoritmos que fusionen las mediciones de una IMU para estimar la orientación de un cuerpo rígido en el espacio, y así poder evaluar el tiempo de cómputo y el desempeño de cada método de manera objetiva.

Los objetivos particulares de desarrollo son:

- Estudiar las distintas parametrizaciones de la orientación y las ecuaciones cinemáticas que relacionan la orientación con las velocidades angulares, así como

los diferentes métodos para estimar la orientación de un cuerpo rígido en el espacio.

- Integrar el *hardware* necesario para establecer la comunicación entre los sensores y el procesador digital de señales, así como el envío y almacenamiento de los datos estimados, y de esta forma disponer de una plataforma experimental de evaluación.
- Programar en el sistema embebido los algoritmos de estimación.
- Realizar los experimentos correspondientes de comparación entre cada método de estimación.

1.5. Organización del Documento

El presente trabajo está dividido en cinco Capítulos que describen cada etapa de desarrollo. En el Capítulo 1 se presenta un panorama general del desarrollo del problema de estimación de la orientación con una reseña de los principales avances y los tópicos que actualmente se desarrollan, haciendo énfasis en las aplicaciones que han tenido estos métodos y el desarrollo que han alcanzado los sensores inerciales MEMS. En el Capítulo 2 se aborda brevemente las herramientas matemáticas disponibles para representar la orientación así como las ecuaciones cinemáticas que describen el movimiento rotacional de un cuerpo rígido, enfatizando en las singularidades de cada parametrización y sus respectivas ventajas. En el Capítulo 3 se explican los métodos de estimación que se han estudiado e implementado, revisando los fundamentos matemáticos de cada uno. En el Capítulo 4 se presenta la parte experimental, donde se detalla la plataforma de evaluación y se muestran los resultados obtenidos de la implementación de cada método en el sistema embebido. Finalmente en el Capítulo 5 se presentan las conclusiones y el trabajo futuro de esta investigación.

Capítulo 2

Cinemática de la Orientación

El objetivo de este Capítulo es describir de forma breve las parametrizaciones matemáticas de la orientación de un cuerpo rígido en el espacio, analizar sus singularidades y las relaciones necesarias para convertir de una representación a otra. El desarrollo de las diferentes parametrizaciones de la orientación puede ser encontrada en libros de mecánica clásica y de control de orientación, en particular en Goldstein [40], Hughes [41], Wertz [6] y el libro de Junkins y Turner [42], por citar los más usados.

Existen varias representaciones que describen la orientación pero comúnmente las más utilizadas son tres, los Ángulos de Euler, la Matriz de Cosenos Directores y los Cuaterniones. En la tabla 2.1 se presentan las propiedades de unicidad y globalidad de las diferentes parametrizaciones de la orientación, donde se puede observar que aunque las representaciones (d), (e) y (f) pueden representar todo el espacio de orientaciones, es decir cualquier punto de una esfera tridimensional, solo (f) puede representar de manera única todas las orientaciones posibles de un cuerpo rígido [43].

Tabla 2.1: Propiedades de las representaciones de la orientación.

Representaciones de orientación	Global	Única
a) Ángulos de Euler	No	No
b) Parámetros de Rodrigues	No	No
c) Parámetros de Rodrigues Modificados	No	No
d) Cuaterniones	Si	No
e) Eje - Ángulo	Si	No
f) Matriz de Cosenos Directores (DCM)	Si	Si

2.1. Propiedades Matemáticas

Un marco de referencia o sistema coordenado se define generalmente como un conjunto de tres vectores unitarios que son mutuamente perpendiculares, es decir son ortogonales. Conocer la orientación de un marco de referencia es completamente equivalente a conocer la orientación de un cuerpo rígido. A lo largo de este documento vamos a establecer dos marcos de referencia, un marco inercial o fijo que denotaremos como $\{\mathbf{i}\} = \{\hat{\mathbf{i}}_x, \hat{\mathbf{i}}_y, \hat{\mathbf{i}}_z\}$ y un marco cuerpo o móvil que estará adosado al cuerpo en movimiento $\{\mathbf{b}\} = \{\hat{\mathbf{b}}_x, \hat{\mathbf{b}}_y, \hat{\mathbf{b}}_z\}$. Manteniendo fijo el origen del marco $\{\mathbf{i}\}$, los ejes pueden rotar libremente formando el marco $\{\mathbf{b}\}$, a este cambio precisamente se le llama una rotación.

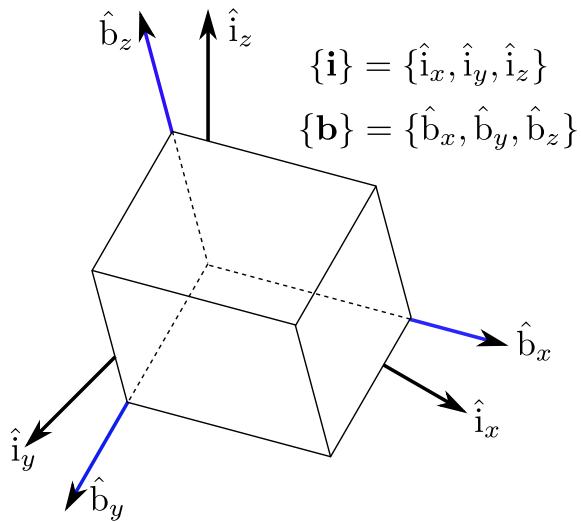


Figura 2.1: Marcos de referencia.

La propiedad de ortogonalidad de un marco de referencia está definida por el producto punto entre cada uno de los vectores, específicamente para cada conjunto de vectores base ortogonales, el producto punto debe satisfacer,

$$\hat{\mathbf{i}}_i \cdot \hat{\mathbf{i}}_j = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{si } i \neq j \end{cases}. \quad (2.1)$$

Una rotación puede verse como una transformación entre dos marcos de referencia, teniendo un punto P_b con sus respectivas coordenadas $[p_x, p_y, p_z]^T \in \mathbb{R}^3$ en el marco $\{\mathbf{b}\}$ y sean $X_i^b, Y_i^b, Z_i^b \in \mathbb{R}^3$ las coordenadas de los ejes principales del marco $\{\mathbf{b}\}$ relativo a $\{\mathbf{i}\}$, el punto P_b puede ser transformado a las coordenadas $P_i = [p_x, p_y, p_z]$ en $\{\mathbf{i}\}$ mediante:

$$P_i = R_i^b P_b \quad (2.2)$$

donde la matriz de rotación $R_i^b \in \mathbb{R}^{3 \times 3}$ está definida de la siguiente forma:

$$R_i^b = \begin{bmatrix} X_i^b & Y_i^b & Z_i^b \end{bmatrix} \quad (2.3)$$

y cumple con las siguientes propiedades:

$$RR^T = R^T R = I \quad (2.4)$$

$$\det(R) = \pm 1 \quad (2.5)$$

para determinar el signo del determinante de R recordemos del álgebra lineal que $\det(R) = X_i^{bT} \cdot (Y_i^b \times Z_i^b)$ y considerando sistemas de referencia dextrógiros tenemos que $Y_i^b \times Z_i^b = X_i^b$, entonces:

$$\det(R) = X_i^{bT} \cdot X_i^b = 1. \quad (2.6)$$

El conjunto de todas las matrices de 3×3 que satisfacen (2.4) y (2.6) es llamado grupo especial ortogonal y es denotado como $SO(3)$, especial se refiere al hecho de que $\det(R) = +1$ en lugar de ± 1 . De manera general se define el espacio de matrices de rotación en $\mathbb{R}^{n \times n}$ con $n \in \mathbb{Z}^+$ como:

$$SO(n) = \{R \in \mathbb{R}^{n \times n} : RR^T = I_n, \det(R) = 1\} \quad (2.7)$$

que corresponde a un grupo bajo la operación de la multiplicación de matrices y satisface los siguientes axiomas:

1. *Cerradura*: Si $R_1, R_2 \in SO(3)$, entonces $R_1 R_2 \in SO(3)$.
2. *Identidad*: Existe un elemento identidad I , tal que $RI = IR = R$ para cada $R \in SO(3)$, donde la matriz $I_{(i,j)} = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{si } i \neq j \end{cases}$
3. *Inversa*: Por (2.4) se tiene que para cada $R \in SO(3)$ existe una inversa única $R^{-1} = R^T \in SO(3)$, tal que $RR^{-1} = R^{-1}R = I$.
4. *Asociatividad*: Está dada por la multiplicación de matrices. Si $R_1, R_2, R_3 \in SO(3)$, entonces $(R_1 R_2) R_3 = R_1 (R_2 R_3)$.

2.2. Representaciones de la Orientación

2.2.1. Ángulos de Euler

Los ángulos de Euler son una serie de tres rotaciones que inician en el marco $\{\mathbf{i}\}$ y terminan en $\{\mathbf{b}\}$ y solo utilizan tres parámetros, (ϕ, θ, ψ) , para determinar la orientación de un cuerpo rígido en el espacio. Cada rotación de la serie corresponde al

movimiento sobre un eje arbitrario del marco móvil $\{\mathbf{b}\}$, por ello debemos tener cuidado al momento de definir la secuencia de rotaciones debido a que no son conmutativas. En este documento consideramos la convención de ángulos de Euler - ZYX, donde los marcos $\{\mathbf{i}\}$ y $\{\mathbf{b}\}$ son coincidentes inicialmente para después llevar el marco $\{\mathbf{b}\}$ hacia cualquier orientación considerando los siguientes pasos:

1. Girar el marco $\{\mathbf{b}\}$ un ángulo ψ con respecto al eje $\hat{\mathbf{b}}_z$.
2. Girar el marco $\{\mathbf{b}\}$ un ángulo θ con respecto al eje $\hat{\mathbf{b}}_y$.
3. Girar el marco $\{\mathbf{b}\}$ un ángulo ϕ con respecto al eje $\hat{\mathbf{b}}_x$.

Observemos que cada rotación se lleva a cabo sobre un eje del marco móvil $\{\mathbf{b}\}$ en lugar de hacerlo sobre los ejes del marco fijo $\{\mathbf{i}\}$, como sucede en la convención de ángulos fijos que también es conocida como ángulos de Alabeo, Cabeceo y Guiñada (*Roll, Pitch, Yaw*) [44]. Usaremos la siguiente notación para representar de forma compacta los tres pasos anteriores, renombrando el marco de referencia después de cada rotación,

$$\{\mathbf{i}\} \xrightarrow[\hat{\mathbf{b}}_z]{\psi} \{\mathbf{b}'\} \xrightarrow[\hat{\mathbf{b}}_y]{\theta} \{\mathbf{b}''\} \xrightarrow[\hat{\mathbf{b}}_x]{\phi} \{\mathbf{b}\}$$

donde se entiende que iniciamos en $\{\mathbf{i}\}$ y giramos un ángulo ψ sobre el eje $\hat{\mathbf{b}}_z$, con esta rotación llegamos a un nuevo marco secundario o provisional que llamaremos $\{\mathbf{b}'\}$, aplicamos la rotación sobre el eje correspondiente como en el caso anterior para llegar a $\{\mathbf{b}''\}$, y finalmente terminamos en $\{\mathbf{b}\}$. Respectivamente las matrices de rotación en cada eje son

$$R_i^{b'} = \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad R_{b'}^{b''} = \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix}, \quad R_{b''}^b = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & s_\phi \\ 0 & -s_\phi & c_\phi \end{bmatrix} \quad (2.8)$$

donde se ha considerado la notación $s_x = \sin(x)$ y $c_x = \cos(x)$, de manera que la matriz de rotación que describe la orientación del marco de referencia $\{\mathbf{b}\}$ relativo al marco de referencia $\{\mathbf{i}\}$ usando los ángulos de Euler está dada por:

$$R_i^b = R_i^{b'} R_{b'}^{b''} R_{b''}^b = \begin{bmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi - s_\psi c_\phi & c_\psi s_\theta s_\phi + s_\psi s_\phi \\ s_\psi c_\theta & s_\psi s_\theta s_\phi + c_\psi c_\phi & s_\psi s_\theta s_\phi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix}. \quad (2.9)$$

Observemos que $(180^\circ + \psi, 180^\circ - \theta, 180^\circ + \phi)$ representa la misma orientación que (ψ, θ, ϕ) , por ello se suelen limitar los ángulos, típicamente a $\theta \in [-90^\circ, 90^\circ]$.

2.2.2. Matriz de Cosenos Directores

En la matriz de cosenos directores (DCM - Director Cosine Matrix), en comparación a los ángulos de Euler, no es necesario definir explícitamente una secuencia de rotaciones; sin embargo, utiliza nueve parámetros para representar la orientación en lugar de tres. Dado un marco de referencia fijo $\{\mathbf{i}\}$ determinado por una base de vectores unitarios ortonormales $\{\hat{\mathbf{i}}_x, \hat{\mathbf{i}}_y, \hat{\mathbf{i}}_z\}$ y otro marco $\{\mathbf{b}\}$ determinado por $\{\hat{\mathbf{b}}_x, \hat{\mathbf{b}}_y, \hat{\mathbf{b}}_z\}$,

$$\begin{aligned}\hat{\mathbf{i}}_x &= [1, 0, 0]^T & \hat{\mathbf{b}}_x &= [1, 0, 0]^T \\ \hat{\mathbf{i}}_y &= [0, 1, 0]^T & \hat{\mathbf{b}}_y &= [0, 1, 0]^T \\ \hat{\mathbf{i}}_z &= [0, 0, 1]^T & \hat{\mathbf{b}}_z &= [0, 0, 1]^T\end{aligned}\quad (2.10)$$

entonces la orientación de $\{\mathbf{b}\}$ respecto de $\{\mathbf{i}\}$ está totalmente determinada por la matriz R_i^b definida como:

$$R_i^b = \begin{bmatrix} r_{1,1} & r_{1,2} & r_{1,3} \\ r_{2,1} & r_{2,2} & r_{2,3} \\ r_{3,1} & r_{3,2} & r_{3,3} \end{bmatrix}. \quad (2.11)$$

Ahora, si realizamos una proyección de cada uno de los ejes del marco $\{\mathbf{b}\}$ sobre los ejes del marco $\{\mathbf{i}\}$, como se muestra en las operaciones

$$V = R_i^b \hat{\mathbf{i}}_x = R_i^b [1, 0, 0]^T = [r_{1,1}, r_{2,1}, r_{3,1}]^T \quad (2.12)$$

$$\hat{\mathbf{b}}_x \cdot \hat{\mathbf{b}}_x = \hat{\mathbf{b}}_x^T \cdot V = [1, 0, 0] [r_{1,1}, r_{2,1}, r_{3,1}]^T = r_{1,1}, \quad (2.13)$$

observamos que cada elemento $r_{i,j}$ de R_i^b está determinado por el producto punto entre dos vectores unitarios. Entonces si generalizamos este resultado tenemos que:

$$\begin{aligned}r_{1,1} &= \hat{\mathbf{b}}_x \cdot \hat{\mathbf{i}}_x & r_{1,2} &= \hat{\mathbf{b}}_x \cdot \hat{\mathbf{i}}_y & r_{1,3} &= \hat{\mathbf{b}}_x \cdot \hat{\mathbf{i}}_z \\ r_{2,1} &= \hat{\mathbf{b}}_y \cdot \hat{\mathbf{i}}_x & r_{2,2} &= \hat{\mathbf{b}}_y \cdot \hat{\mathbf{i}}_y & r_{2,3} &= \hat{\mathbf{b}}_y \cdot \hat{\mathbf{i}}_z \\ r_{3,1} &= \hat{\mathbf{b}}_z \cdot \hat{\mathbf{i}}_x & r_{3,2} &= \hat{\mathbf{b}}_z \cdot \hat{\mathbf{i}}_y & r_{3,3} &= \hat{\mathbf{b}}_z \cdot \hat{\mathbf{i}}_z\end{aligned}\quad (2.14)$$

Por lo tanto la matriz de cosenos directores toma la siguiente forma:

$$DCM = R_i^b = \begin{bmatrix} \hat{\mathbf{b}}_x \cdot \hat{\mathbf{i}}_x & \hat{\mathbf{b}}_x \cdot \hat{\mathbf{i}}_y & \hat{\mathbf{b}}_x \cdot \hat{\mathbf{i}}_z \\ \hat{\mathbf{b}}_y \cdot \hat{\mathbf{i}}_x & \hat{\mathbf{b}}_y \cdot \hat{\mathbf{i}}_y & \hat{\mathbf{b}}_y \cdot \hat{\mathbf{i}}_z \\ \hat{\mathbf{b}}_z \cdot \hat{\mathbf{i}}_x & \hat{\mathbf{b}}_z \cdot \hat{\mathbf{i}}_y & \hat{\mathbf{b}}_z \cdot \hat{\mathbf{i}}_z \end{bmatrix}. \quad (2.15)$$

El producto punto entre dos vectores está definido como $x \cdot y = \|x\| \|y\| \cos(\theta)$, donde el operador $\|\cdot\|$ representa la norma Euclidiana, conociendo que los vectores

son unitarios tenemos que $x \cdot y = \cos(\theta_{x,y})$, donde $\theta_{x,y}$ es el ángulo entre los vectores x y y . Considerando lo anterior la matriz de cosenos directores adquiere esta forma,

$$DCM = R_i^b = \begin{bmatrix} \cos(\theta_{b_x,i_x}) & \cos(\theta_{b_x,i_y}) & \cos(\theta_{b_x,i_z}) \\ \cos(\theta_{b_y,i_x}) & \cos(\theta_{b_y,i_y}) & \cos(\theta_{b_y,i_z}) \\ \cos(\theta_{b_z,i_x}) & \cos(\theta_{b_z,i_y}) & \cos(\theta_{b_z,i_z}) \end{bmatrix}. \quad (2.16)$$

La DCM es la representación matemática que cubre de manera global y única el espacio de orientaciones posibles, a expensas de usar nueve parámetros.

2.2.3. Cuaterniones

Una tercera alternativa son los cuaterniones que capturan la esencia del teorema de Euler, que menciona que cualquier composición de rotaciones puede ser lograda a partir de un solo ángulo de rotación a través de un solo eje, en otras palabras, el conocimiento de la magnitud de rotación y el eje de Euler es información suficiente para caracterizar la transformación. Definimos un cuaternion como $\mathbf{q} = [\eta, \epsilon^T]^T \in \mathbb{S}^3$, con η como la parte escalar y $\epsilon = [\epsilon_1, \epsilon_2, \epsilon_3]^T$ la parte vectorial, donde $\mathbb{S}^3 = \{(x_1, x_2, x_3, x_4) \in \mathbb{R}^4 : \| (x_1, x_2, x_3, x_4) \| = 1\}$ es la 3-esfera o también llamada hiperesfera.

Para entender el concepto de las rotaciones con cuaterniones es importante conocer su álgebra y además la representación física que tienen. En este apartado sólo mostraremos la multiplicación, una operación fundamental para representar con cuaterniones la orientación de un cuerpo. Definamos dos cuaterniones tales que,

$$\mathbf{q}_1 = [a_0, \mathbf{a}^T]^T, \quad \mathbf{a} = [a_1, a_2, a_3]^T$$

$$\mathbf{q}_2 = [b_0, \mathbf{b}^T]^T, \quad \mathbf{b} = [b_1, b_2, b_3]^T$$

entonces podemos definir el producto de cuaterniones $\mathbf{q}_3 = \mathbf{q}_1 \otimes \mathbf{q}_2$, de tres formas:

- Forma estándar del producto:

$$\begin{aligned} \mathbf{q}_3 = \mathbf{q}_1 \otimes \mathbf{q}_2 = & [(a_0 b_0 - a_1 b_1 - a_2 b_2 - a_3 b_3) \\ & , (a_0 b_1 + a_1 b_0 + a_2 b_3 - a_3 b_2) \\ & , (a_0 b_2 - a_1 b_3 + a_2 b_0 - a_3 b_1) \\ & , (a_0 b_3 + a_1 b_2 - a_2 b_1 + a_3 b_0)]^T \end{aligned} \quad (2.17)$$

- Forma matricial del producto:

$$\mathbf{q}_3 = \mathbf{q}_1 \otimes \mathbf{q}_2 = \begin{bmatrix} a_0 & -a_1 & -a_2 & -a_3 \\ a_1 & a_0 & -a_3 & a_2 \\ a_2 & a_3 & a_0 & -a_1 \\ a_3 & -a_2 & a_1 & a_0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad (2.18)$$

- Forma vectorial del producto:

$$\mathbf{q}_3 = \mathbf{q}_1 \otimes \mathbf{q}_2 = [a_0 b_0 - \mathbf{a}^T \mathbf{b}, b_0 \mathbf{a} + a_0 \mathbf{b} + \mathbf{a} \times \mathbf{b}]. \quad (2.19)$$

Dada una orientación representada mediante un vector unitario $e = [e_x, e_y, e_z]^T$ y un ángulo θ , se puede escribir tal orientación en forma de cuaternion mediante:

$$\mathbf{q} = [\eta, \boldsymbol{\epsilon}^T]^T = [\cos(\frac{\theta}{2}), \sin(\frac{\theta}{2}) \boldsymbol{\epsilon}^T]^T. \quad (2.20)$$

Observemos que un cuaternion \mathbf{q} representa la orientación con cuatro parámetros, con la restricción $\|\mathbf{q}\| = \mathbf{q} \otimes \mathbf{q}^T = 1$. Existe un problema de ambigüedad ya que \mathbf{q} y $-\mathbf{q}$ representan la misma orientación, debido a que si \mathbf{q} corresponde a (e, θ) , entonces $-\mathbf{q}$ corresponde a $(-e, 360 - \theta)$. Esto se debe a que la transformación del espacio \mathbb{S}^3 de los cuaterniones unitarios al espacio $SO(3)$ de las rotaciones no es único, generando que cada orientación física $R \in SO(3)$ sea representado por un par de cuaterniones unitarios $\pm \mathbf{q} \in \mathbb{S}^3$.

2.3. Transformaciones entre Representaciones

Entre las tres representaciones que hemos descrito anteriormente podemos convertir de una representación a otra, por ejemplo, en los ángulos de Euler son más evidentes o intuitivos las rotaciones, sin embargo para los cuaterniones o para la matriz de cosenos directores las rotaciones no son fáciles de visualizar o puede ser posible que el cálculo numérico para cierta representación sea menor, a partir de estas cualidades es importante convertir de una representación a otra.

2.3.1. Transformación de Cuaterniones a DCM

Para convertir de cuaterniones $\mathbf{q} = [\eta, \boldsymbol{\epsilon}^T]^T = [\eta, \epsilon_1, \epsilon_2, \epsilon_3]^T$ a la matriz de cosenos directores, se utiliza la formula de Euler-Rodrigues

$$R = (\eta^2 - \boldsymbol{\epsilon}^T \boldsymbol{\epsilon}) I_3 + 2\boldsymbol{\epsilon} \boldsymbol{\epsilon}^T - 2\eta \mathbf{S}(\boldsymbol{\epsilon})$$

donde I_3 corresponde a la matriz identidad y el operador $\mathbf{S}(\cdot) : \mathbb{R}^3 \rightarrow \mathfrak{so}(3)$ está definido como:

$$\mathbf{S}(\boldsymbol{\epsilon}) = \begin{bmatrix} 0 & -\epsilon_3 & \epsilon_2 \\ \epsilon_3 & 0 & -\epsilon_1 \\ -\epsilon_2 & \epsilon_1 & 0 \end{bmatrix} \in \mathfrak{so}(3), \quad (2.21)$$

con $\mathfrak{so}(3) = \{U \in \mathbb{R}^{3 \times 3} : U^T = -U\}$ el espacio de matrices antisimétricas de 3×3 . La DCM en función de los parámetros del cuaternion toma la siguiente forma presentada

en [45].

$$R(\mathbf{q}) = \begin{bmatrix} \eta^2 + \epsilon_1^2 - \epsilon_2^2 - \epsilon_3^2 & 2(\epsilon_1\epsilon_2 + \eta\epsilon_3) & 2(\epsilon_1\epsilon_3 - \eta\epsilon_2) \\ 2(\epsilon_1\epsilon_2 - \eta\epsilon_3) & \eta^2 - \epsilon_1^2 + \epsilon_2^2 - \epsilon_3^2 & 2(\epsilon_2\epsilon_3 + \eta\epsilon_1) \\ 2(\epsilon_1\epsilon_3 + \eta\epsilon_2) & 2(\epsilon_2\epsilon_3 - \eta\epsilon_1) & \eta^2 - \epsilon_1^2 - \epsilon_2^2 + \epsilon_3^2 \end{bmatrix}. \quad (2.22)$$

2.3.2. Transformación de DCM a Cuaterniones

Conociendo la matriz de cosenos directores R podemos obtener la representación en cuaterniones mediante las siguientes expresiones presentadas en [45].

$$\eta = \frac{\sqrt{1 + \text{Tr}(R)}}{2} \quad (2.23)$$

$$\mathbf{S}(\epsilon) = \frac{1}{4\eta}(R^T - R) \quad (2.24)$$

donde el eje unitario y el ángulo de rotación están dados por

$$\begin{aligned} e &= \frac{1}{\|\epsilon\|}\epsilon, \quad \|\epsilon\| \neq 0 \\ \theta &= 2 \arccos(\eta). \end{aligned} \quad (2.25)$$

2.3.3. Transformación de Cuaterniones a Ángulos de Euler

Dada una orientación representada por el cuaternion $\mathbf{q} = [\eta, \epsilon_1, \epsilon_2, \epsilon_3]^T$, los ángulos de Euler se calculan con las siguientes expresiones presentadas en [45]:

$$\begin{aligned} \phi(\mathbf{q}) &= \arctan(2\epsilon_2\epsilon_3 + 2\eta\epsilon_1, \eta^2 - \epsilon_1^2 - \epsilon_2^2 + \epsilon_3^2) \\ \theta(\mathbf{q}) &= \arcsin(-2(2\epsilon_1\epsilon_3 - 2\eta\epsilon_2)) \\ \psi(\mathbf{q}) &= \arctan(2\epsilon_1\epsilon_2 + 2\eta\epsilon_3, \eta^2 + \epsilon_1^2 - \epsilon_2^2 - \epsilon_3^2). \end{aligned}$$

2.3.4. Transformación de Ángulos de Euler a Cuaterniones

Dados los ángulos de Euler (ϕ, θ, ψ) observemos que a cada ángulo de rotación le corresponde un cuaternion, es decir,

- Al ángulo de Euler $(\phi, 0, 0)$ le corresponde el cuaternion $\mathbf{q}_\phi = \cos(\frac{\phi}{2}) + \sin(\frac{\phi}{2})e_x$
- Al ángulo de Euler $(0, \theta, 0)$ le corresponde el cuaternion $\mathbf{q}_\theta = \cos(\frac{\theta}{2}) + \sin(\frac{\theta}{2})e_y$
- Al ángulo de Euler $(0, 0, \psi)$ le corresponde el cuaternion $\mathbf{q}_\psi = \cos(\frac{\psi}{2}) + \sin(\frac{\psi}{2})e_z$

Por lo tanto, el cuaternion que representa dicha orientación está dado por $\mathbf{q} = \mathbf{q}_\psi \otimes \mathbf{q}_\theta \otimes \mathbf{q}_\phi$, realizando las operaciones correspondientes, se tiene la siguiente expresión presentada en [45]:

$$\begin{aligned}\mathbf{q}(\psi, \theta, \phi) = & [\cos(\frac{\psi}{2}) \cos(\frac{\theta}{2}) \cos(\frac{\phi}{2}) + \sin(\frac{\psi}{2}) \sin(\frac{\theta}{2}) \sin(\frac{\phi}{2}) \\ & , \cos(\frac{\psi}{2}) \cos(\frac{\theta}{2}) \sin(\frac{\phi}{2}) - \sin(\frac{\psi}{2}) \sin(\frac{\theta}{2}) \cos(\frac{\phi}{2}) \\ & , \cos(\frac{\psi}{2}) \sin(\frac{\theta}{2}) \cos(\frac{\phi}{2}) + \sin(\frac{\psi}{2}) \cos(\frac{\theta}{2}) \sin(\frac{\phi}{2}) \\ & , \sin(\frac{\psi}{2}) \cos(\frac{\theta}{2}) \cos(\frac{\phi}{2}) - \cos(\frac{\psi}{2}) \sin(\frac{\theta}{2}) \sin(\frac{\phi}{2})]^T.\end{aligned}\quad (2.26)$$

2.3.5. Transformación de Ángulos de Euler a DCM

Dados los ángulos de Euler (ψ, θ, ϕ) , la matriz que representa tal rotación está dada por

$$R(\psi, \theta, \phi) = \begin{bmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi - s_\psi c_\phi & c_\psi s_\theta s_\phi + s_\psi s_\phi \\ s_\psi c_\theta & s_\psi s_\theta s_\phi + c_\psi c_\phi & s_\psi s_\theta s_\phi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix}. \quad (2.27)$$

como se explica en la Sección 2.2.1.

2.3.6. Transformación de DCM a Ángulos de Euler

Dada una matriz de rotación R , los ángulos de Euler son calculados de la siguiente forma [45]:

$$\begin{aligned}\phi &= \arctan(r_{2,3}, r_{3,3}) \\ \theta &= \arcsin(-r_{1,3}) \\ \psi &= \arctan(r_{1,2}, r_{1,1}).\end{aligned}$$

donde $r_{i,j}$ corresponden a los elementos de la matriz de rotación.

2.4. Ecuaciones Cinemáticas de la Orientación

Es importante notar que las representaciones anteriores describen la orientación relativa entre dos marcos de referencia, pero no capturan ningún detalle del movimiento entre ellos. Si nosotros deseáramos un detalle preciso de como $\{\mathbf{b}\}$ se mueve en relación a $\{\mathbf{i}\}$, entonces es necesario conocer las ecuaciones cinemáticas que rigen el comportamiento de un cuerpo rígido cuando éste se mueve con una velocidad angular $\Omega_i^b = [p, q, r]^T \in \mathbb{R}^3$.

Para el caso de movimiento rotacional, las ecuaciones diferenciales cinemáticas (EDC) relacionan los parámetros de la orientación (DCM, ángulos de Euler, cuaterniones) con el vector de velocidad angular Ω_i^b . En los sistema de navegación inercial el giroscopio nos proporciona las mediciones de Ω_i^b y habrá que integrarlas para calcular la orientación.

2.4.1. EDC para Ángulos de Euler

Partimos de la notación que definimos para los ángulos de Euler, iniciamos en el marco $\{\mathbf{i}\}$ y por cada rotación vamos cambiando de marco hasta llegar a $\{\mathbf{b}\}$.

$$\{\mathbf{i}\} \xrightarrow[\hat{\mathbf{b}}_z]{\psi} \{\mathbf{b}'\} \xrightarrow[\hat{\mathbf{b}}_y]{\theta} \{\mathbf{b}''\} \xrightarrow[\hat{\mathbf{b}}_x]{\phi} \{\mathbf{b}\}$$

La velocidad angular total es igual a la suma de las velocidades relativas entre cada marco de referencia. Observemos que Ω_i^b se define como la velocidad del marco móvil con respecto del marco fijo

$$\Omega_i^b = \Omega_{b''}^b + \Omega_{b'}^{b''} + \Omega_i^{b'}$$
 (2.28)

por otro lado es claro que las velocidades relativas entre marcos de referencia son

$$\Omega_{b''}^b = [\dot{\phi}, 0, 0]^T, \quad \Omega_{b'}^{b''} = [0, \dot{\theta}, 0]^T, \quad \Omega_i^{b'} = [0, 0, \dot{\psi}]^T$$
 (2.29)

Entre cada rotación conocemos $R_{b''}^b$, $R_{b'}^{b''}$ y $R_i^{b'}$, luego entonces tenemos que

$$\Omega_i^b = \Omega_{b''}^b + R_{b''}^b \Omega_{b'}^{b''} + R_{b'}^{b''} \Omega_i^{b'}$$
 (2.30)

y puesto que $R_{b'}^{b''} = R_{b''}^b R_{b'}^{b''}$, podemos escribir entonces:

$$\Omega_i^b = \Omega_{b''}^b + R_{b''}^b \Omega_{b'}^{b''} + R_{b''}^b R_{b'}^{b''} \Omega_i^{b'}.$$
 (2.31)

Desarrollando la ecuación (2.31)

$$\begin{aligned} \Omega_i^b &= \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & s_\phi \\ 0 & -s_\phi & c_\phi \end{bmatrix} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & s_\phi \\ 0 & -s_\phi & c_\phi \end{bmatrix} \begin{bmatrix} c_\theta & 0 & -s_\theta \\ 0 & 1 & 0 \\ s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \\ \Omega_i^b &= \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & s_\phi c_\theta \\ 0 & -s_\phi & c_\phi c_\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}. \end{aligned}$$
 (2.32)

Estamos interesados en una expresión para las derivadas de los ángulos de Euler en función de las velocidades angulares $[p, q, r]^T$ medidas en el marco móvil $\{\mathbf{b}\}$, por lo tanto resolviendo para los ángulos de Euler llegamos a

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \frac{1}{c_\theta} \begin{bmatrix} c_\theta & s_\theta s_\phi & s_\theta c_\phi \\ 0 & c_\theta c_\phi & -c_\theta s_\phi \\ 0 & s_\phi & c_\phi \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & t_\theta s_\phi & t_\theta c_\phi \\ 0 & c_\phi & -s_\phi \\ 0 & \frac{s_\phi}{c_\theta} & \frac{c_\phi}{c_\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \quad (2.33)$$

La expresión (2.33) representa el modelo cinemático rotacional de un cuerpo rígido usando los ángulos de Euler, observemos que hemos llegado a tres ecuaciones diferenciales no lineales con funciones trigonométricas que poseen una singularidad para $\theta = \pm 90^\circ$, es decir, $c_\theta \neq 0$. En realidad, los ángulos de Euler no están definidos para esta situación, de aquí que esta representación no sea global, ver Tabla 2.1.

2.4.2. EDC para la Matriz de Cosenos Directores

Para las ecuaciones diferenciales cinemáticas de la orientación a través de su representación con la matriz de cosenos directores, como se ha hecho anteriormente, consideramos dos sistemas de referencia $\{\mathbf{i}\}$ y $\{\mathbf{b}\}$ relacionados de la siguiente forma como se explica en la Sección 2.2.1,

$$\{\mathbf{i}\} \xrightarrow[\hat{\mathbf{b}}_z]{d\theta_1} \{\mathbf{b}'\} \xrightarrow[\hat{\mathbf{b}}_y]{d\theta_2} \{\mathbf{b}''\} \xrightarrow[\hat{\mathbf{b}}_x]{d\theta_3} \{\mathbf{b}\}$$

donde suponemos que $d\theta_i$ representa un ángulo pequeño $d\theta_i \approx 0$ de tal forma que podemos realizar la aproximación $\cos(d\theta_i) \approx 1$ y $\sin(d\theta_i) \approx d\theta_i$. Si escribimos las matrices de rotación teniendo en cuenta la aproximación anterior, obtenemos

$$R_i^b' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & d\theta_1 \\ 0 & -d\theta_1 & 1 \end{bmatrix}, R_{b'}^b'' = \begin{bmatrix} 1 & 0 & -d\theta_2 \\ 0 & 1 & 0 \\ d\theta_2 & 0 & 1 \end{bmatrix}, R_{b''}^b = \begin{bmatrix} 1 & d\theta_3 & 0 \\ -d\theta_3 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.34)$$

Realizando el producto $R_i^b = R_{b''}^b R_{b'}^b R_i^b'$ y despreciando todos los productos dobles de ángulos, es decir, $d\theta_i \cdot d\theta_j \approx 0$, obtenemos

$$R_i^b = \begin{bmatrix} 1 & d\theta_3 & -d\theta_2 \\ -d\theta_3 & 1 & d\theta_1 \\ d\theta_2 & -d\theta_1 & 1 \end{bmatrix} = I_3 - \begin{bmatrix} 0 & d\theta_3 & -d\theta_2 \\ -d\theta_3 & 0 & d\theta_1 \\ d\theta_2 & -d\theta_1 & 0 \end{bmatrix} = I_3 - \mathbf{S}(d\theta) \quad (2.35)$$

donde se ha considerado,

$$d\theta = [d\theta_1, d\theta_2, d\theta_3]^T \quad \text{y} \quad \mathbf{S}(d\theta) = \begin{bmatrix} 0 & -d\theta_3 & d\theta_2 \\ d\theta_3 & 0 & -d\theta_1 \\ -d\theta_2 & d\theta_1 & 0 \end{bmatrix}$$

con $\mathbf{S}(d\theta)$ como la matriz antisimétrica que se define de forma similar en (2.21).

Queremos calcular la orientación de $\{\mathbf{b}\}$ respecto de $\{\mathbf{i}\}$ mediante $R_i^b(t)$ que depende el tiempo. Sabiendo que $\{\mathbf{b}\}$ gira con respecto a $\{\mathbf{i}\}$, con una velocidad angular Ω_i^b , por definición tenemos

$$\frac{d}{dt}R_i^b(t) \approx \frac{R_i^b(t + \Delta t) - R_i^b(t)}{\Delta t}. \quad (2.36)$$

$\{\mathbf{i}\}$ es fijo y $\{\mathbf{b}\}$ varía en el tiempo, entonces es posible escribir $\{\mathbf{b}\} = \{\mathbf{b}(t)\}$ y por tanto $R_i^b(t) = R_i^{b(t)}$, usando este razonamiento,

$$R_i^b(t + \Delta t) = R_i^{b(t+\Delta t)} = R_{b(t)}^{b(t+\Delta t)} R_i^{b(t)}. \quad (2.37)$$

En el tiempo Δt , el marco $\{\mathbf{b}\}$ habrá girado respecto a sí mismo un ángulo muy pequeño en cada eje, por lo tanto usando la aproximación de ángulos pequeños (2.35) tenemos que

$$R_{b(t)}^{b(t+\Delta t)} = I_3 - \mathbf{S}(d\theta^b). \quad (2.38)$$

Siguiendo el mismo razonamiento llegamos a,

$$\begin{aligned} \frac{d}{dt}R_i^b(t) &\approx \frac{R_i^b(t + \Delta t) - R_i^b(t)}{\Delta t} \approx \frac{R_{b(t)}^{b(t+\Delta t)} R_i^b(t) - R_i^b(t)}{\Delta t} \\ \frac{d}{dt}R_i^b(t) &\approx \frac{(I_3 - \mathbf{S}(d\theta^b)) R_i^b(t) - R_i^b(t)}{\Delta t} \approx -\frac{\mathbf{S}(d\theta^b)}{\Delta t} R_i^b(t) \end{aligned} \quad (2.39)$$

donde Δt se puede considerar como un diferencial de tiempo dt y la matriz $\frac{\mathbf{S}(d\theta^b)}{\Delta t}$ está representada en la forma

$$\frac{\mathbf{S}(d\theta^b)}{dt} = \begin{bmatrix} 0 & -\frac{d\theta_3}{dt} & \frac{d\theta_2}{dt} \\ \frac{d\theta_3}{dt} & 0 & -\frac{d\theta_1}{dt} \\ -\frac{d\theta_2}{dt} & \frac{d\theta_1}{dt} & 0 \end{bmatrix} = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \quad (2.40)$$

con $\Omega_i^b = [p, q, r]^T$ como las velocidades angulares en cada eje. Debido a que $d\theta^b$ representaba el ángulo girado por $\{\mathbf{b}\}$ en un tiempo pequeño dt , entonces por definición se refiere a la velocidad angular, se tiene entonces

$$\begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} = \mathbf{S}(\Omega_i^b). \quad (2.41)$$

Finalmente llegamos al modelo de la cinemática rotacional usando la DCM

$$\frac{d}{dt}R_i^b = \dot{R}_i^b = -\mathbf{S}(\Omega_i^b) R_i^b. \quad (2.42)$$

Algunos autores presentan otra variación a la que se puede llegar transponiendo ambos miembros de la ecuación anterior, de tal forma que se obtiene

$$\dot{R}_b^i = R_b^i \mathbf{S}(\Omega_i^b). \quad (2.43)$$

En general, el modelo diferencial cinemático de la matriz de cosenos directores (2.42) representa nueve ecuaciones diferenciales acopladas que habrá que resolver componente a componente. El principal problema de resolver numéricamente esta ecuación es garantizar que la matriz resultante sea ortogonal, es decir, $R_i^b R_i^{bT} = I_3$.

2.4.3. EDC para Cuaterniones

Partimos de la definición de cuaterniones en función de un ángulo de rotación θ y el eje de Euler e dados por,

$$\eta = \cos\left(\frac{\theta}{2}\right), \quad \epsilon = \sin\left(\frac{\theta}{2}\right) e \quad (2.44)$$

se consideran también las ecuaciones diferenciales cinemáticas de la representación en forma de eje y ángulo de Euler (e, θ) descritas por

$$\dot{\theta} = e^T \Omega_i^b, \quad \dot{e} = \frac{1}{2} \left[\mathbf{S}(e) + \frac{1}{\tan(\theta/2)} (I_3 - e e^T) \right] \Omega_i^b, \quad \tan(\theta/2) \neq 0 \quad (2.45)$$

donde $\Omega_i^b = [p, q, r]^T$ representan las velocidades angulares en los ejes $\hat{b}_x, \hat{b}_y, \hat{b}_z$ respectivamente.

Ahora, si derivamos las ecuaciones (2.44) se obtiene,

$$\dot{\eta} = -\frac{1}{2} \sin\left(\frac{\theta}{2}\right) \dot{\theta} \quad (2.46)$$

$$\dot{\epsilon} = \frac{1}{2} \cos\left(\frac{\theta}{2}\right) e \dot{\theta} + \sin\left(\frac{\theta}{2}\right) \dot{e}. \quad (2.47)$$

Sustituyendo (2.45) en (2.46) y (2.47) tenemos,

$$\begin{aligned} \dot{\eta} &= -\frac{1}{2} \sin\left(\frac{\theta}{2}\right) e^T \Omega_i^b \\ \dot{\eta} &= -\frac{1}{2} \epsilon^T \Omega_i^b \end{aligned} \quad (2.48)$$

$$\begin{aligned} \dot{\epsilon} &= \frac{1}{2} \cos\left(\frac{\theta}{2}\right) e e^T \Omega_i^b + \frac{1}{2} \sin\left(\frac{\theta}{2}\right) \left[\mathbf{S}(e) + \frac{1}{\tan(\theta/2)} (I_3 - e e^T) \right] \Omega_i^b \\ \dot{\epsilon} &= \frac{1}{2} [\mathbf{S}(\epsilon) + \eta I_3] \Omega_i^b. \end{aligned} \quad (2.49)$$

Finalmente podemos escribir (2.48) y (2.49) en forma matricial para llegar a la siguiente ecuación cinemática usando la representación de cuaterniones

$$\frac{d}{dt} \begin{bmatrix} \eta \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -\epsilon_1 & -\epsilon_2 & -\epsilon_3 \\ \eta & -\epsilon_3 & \epsilon_2 \\ \epsilon_3 & \eta & -\epsilon_1 \\ -\epsilon_2 & \epsilon_1 & \eta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \quad (2.50)$$

Observemos que (2.50) representa cuatro ecuaciones diferenciales con ausencia de singularidades y de funciones trigonométricas, esta propiedad es la razón más importante por la cual su uso es generalizado para representar la orientación de vehículos espaciales y muchas otras aplicaciones, donde todos los cálculos se realizan mediante cuaterniones y en caso necesario, se transforman a ángulos de Euler.

Recordando la definición matricial del producto de cuaterniones (2.18) y definiendo un cuaternion puro

$$\mathbf{q}_\omega = \begin{bmatrix} 0 \\ p \\ q \\ r \end{bmatrix}, \quad (2.51)$$

cuya parte escalar es cero y la parte vectorial es igual a los componentes de la velocidad angular, podemos expresar (2.50) como

$$\dot{\mathbf{q}} = \frac{1}{2} \mathbf{q} \otimes \mathbf{q}_\omega. \quad (2.52)$$

Realizando la multiplicación de cuaterniones definida en (2.18) llegamos a la forma siguiente,

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{\eta} \\ \dot{\epsilon}_1 \\ \dot{\epsilon}_2 \\ \dot{\epsilon}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \eta & -\epsilon_1 & -\epsilon_2 & -\epsilon_3 \\ \epsilon_1 & \eta & -\epsilon_3 & \epsilon_2 \\ \epsilon_2 & \epsilon_3 & \eta & -\epsilon_1 \\ \epsilon_3 & -\epsilon_2 & \epsilon_1 & \eta \end{bmatrix} \begin{bmatrix} 0 \\ p \\ q \\ r \end{bmatrix} \quad (2.53)$$

que también se puede escribir como

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{\eta} \\ \dot{\epsilon}_1 \\ \dot{\epsilon}_2 \\ \dot{\epsilon}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \begin{bmatrix} \eta \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix}. \quad (2.54)$$

De manera breve hemos presentado las parametrizaciones más utilizadas para representar la orientación de un cuerpo en el espacio, junto con sus ecuaciones cinemáticas de rotación, haciendo énfasis en las ventajas y desventajas de cada una de ellas. En general, las ecuaciones de la cinemática de rotación son utilizadas en el diseño de los estimadores de orientación, de ahí la importancia de su estudio para poder abordar los métodos implementados en este trabajo y que son presentados en el siguiente capítulo.

Capítulo 3

Algoritmos de Estimación de Orientación

En este capítulo explicaremos la formulación matemática de los métodos para estimar la orientación que fueron implementados y comparados en esta investigación. En primer término están los métodos Determinísticos TRIAD y QUEST, los métodos de Fusion Sensorial y Filtro de Kalman que corresponden a los algoritmos Estocásticos y finalmente los filtros complementarios y el filtro basado en Inmersión e Invarianza que corresponden a los Observadores no lineales.

Independientemente de qué tipo de algoritmo se use para estimar la orientación de un cuerpo rígido, todos los métodos existentes usan de alguna forma el principio de comparación entre un conjunto de vectores conocidos v_k^i , $k = 1, \dots, m$ medidos en un marco fijo o inercial $\{\mathbf{i}\}$ y su valor correspondiente v_k^b , $k = 1, \dots, m$ en el marco móvil o cuerpo $\{\mathbf{b}\}$, tal que el vector $b_k^b \in \mathbb{R}^3$ pueda ser transformado al marco $\{\mathbf{i}\}$ mediante,

$$v_k^i = R_b^i v_k^b \quad (3.1)$$

donde R_b^i representa la matriz de rotación. Primero abordaremos el modelo de cada una de las mediciones y a partir de la información disponible analizaremos los enfoques probables para estimar la orientación.

3.1. Mediciones

Las señales disponibles en la unidad de medición inercial son las velocidades angulares $\Omega_b = [p, q, r]^T$, las aceleraciones $\mathbf{a}_b = [a_x, a_y, a_z]^T$ y el campo magnético $\mathbf{m}_b = [m_x, m_y, m_z]^T$. Los sensores son de tres ejes cada uno y se encuentran dispuestos de manera ortogonal como se muestra en la Figura 3.1, todas las medidas

son con respecto al marco $\{\mathbf{b}\}$ debido a se considera un sistema *Strapdown*, es decir, la IMU va montada sobre el cuerpo rígido en movimiento.

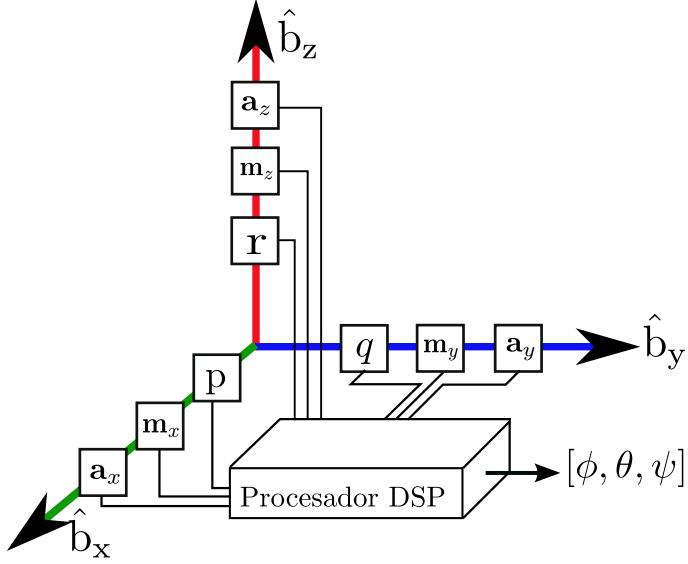


Figura 3.1: Esquema de una IMU.

Giroscopio. Mide las velocidades angulares Ω_b expresadas en el marco $\{\mathbf{b}\}$, usando el siguiente modelo,

$$\Omega_b = \Omega + b_\Omega + \mu_\Omega, \quad \Omega_b \in \mathbb{R}^3, \quad (3.2)$$

donde Ω representa el valor verdadero de la velocidad angular, μ_Ω corresponde al ruido aditivo que se atribuye a las mediciones y b_Ω representa la constante de desviación *bias*.

Acelerómetro. Mide la aceleración lineal instantánea de $\{\mathbf{b}\}$ relativo a $\{\mathbf{i}\}$ denotada como \dot{v} , menos la aceleración gravitacional g expresadas en el marco inercial $\{\mathbf{i}\}$, esto es,

$$\mathbf{a}_b = R_i^{b^T}(\dot{v} - g) + b_a + \mu_a, \quad \mathbf{a}_b \in \mathbb{R}^3 \quad (3.3)$$

donde μ_a denota el ruido aditivo de medición, b_a representa la constante de desviación *bias* y $R_i^{b^T}$ transforma las mediciones al marco $\{\mathbf{b}\}$. En condiciones estáticas normalmente el campo gravitacional $g = \|g_0\| \hat{\mathbf{i}}_z$, con $g_0 = 9.81$ domina el valor de \mathbf{a}_b . Durante condiciones dinámicas el acelerómetro no sólo es sensible a la gravedad, sino también a aceleraciones lineales que se pueden originar por vibraciones o movimientos bruscos. Considerando perfiles bajos de aceleración lineal, es decir $\dot{v} \approx 0$, es común usar la siguiente aproximación [24],

$$\mathbf{v}_a = \frac{\mathbf{a}_b}{|\mathbf{a}_b|} \approx -R_i^{b^T} \hat{\mathbf{i}}_z \quad (3.4)$$

como un vector estimado del eje inercial $\hat{\mathbf{i}}_z$ expresado en el marco $\{\mathbf{b}\}$. Sin embargo, es importante mencionar que cuando la aceleración externa $\dot{\mathbf{v}}$ es considerable, la aproximación (3.4) no es eficiente, teniendo errores considerables en la estimación.

Magnetómetro. Proporciona las mediciones del campo magnético \mathbf{m}_b , con el siguiente modelo

$$\mathbf{m}_b = R_i^{bT} \mathbf{m}_i + \mathbf{B}_m + \boldsymbol{\mu}_m, \quad \mathbf{m}_b \in \mathbb{R}^3 \quad (3.5)$$

donde \mathbf{m}_i representa el campo magnético real de la tierra expresado en $\{\mathbf{i}\}$, \mathbf{B}_m corresponde a la perturbación magnética local en $\{\mathbf{b}\}$ y $\boldsymbol{\mu}_m$ al ruido de medición. Sólo la dirección del campo magnético es relevante para la estimación de la orientación, por lo que es común realizar la normalización en la forma [24],

$$\mathbf{v}_m = \frac{\mathbf{m}_b}{\|\mathbf{m}_b\|}, \quad \mathbf{m}_b \neq 0. \quad (3.6)$$

3.2. Algoritmos de Estimación Determinísticos

3.2.1. Método TRIAD

El método TRIAD (1964, Harold D. Black [4]) utiliza dos vectores de referencia para estimar la matriz de orientación R_i^b , recordemos que tres parámetros de la matriz de rotación son independientes y suficientes para determinar completamente R_i^b , en tanto que cada vector de referencia tiene dos elementos independientes, esto hace que se necesiten al menos dos vectores diferentes para determinar la orientación.

Comenzamos definiendo dos vectores proporcionados por el magnetómetro y el acelerómetro que representaremos como \mathbf{m}_b y \mathbf{a}_b y los vectores medidos con respecto al marco $\{\mathbf{i}\}$ como \mathbf{m}_i y \mathbf{a}_i . La matriz de rotación R_i^b debe satisfacer las siguientes ecuaciones

$$\begin{aligned} \mathbf{a}_b &= R_i^b \mathbf{a}_i \\ \mathbf{m}_b &= R_i^b \mathbf{m}_i. \end{aligned} \quad (3.7)$$

Este algoritmo se basa en la construcción de dos triadas de vectores unitarios ortogonales $[t_{1b}, t_{2b}, t_{3b}]$ y $[t_{1i}, t_{2i}, t_{3i}]$, usando las mediciones del acelerómetro y el magnetómetro \mathbf{a}_b y \mathbf{m}_b y los vectores \mathbf{a}_i y \mathbf{m}_i que representan también la aceleración y el campo magnético pero en el marco $\{\mathbf{i}\}$ y son calculados por modelos matemáticos del campo gravitatorio y el campo magnético de la tierra.

Cada triada de vectores construidos son componentes de un marco de referencia auxiliar que denotaremos como $\{\mathbf{t}\}$, este nuevo marco de referencia es construido suponiendo que uno de los pares de vectores $[\mathbf{b}, \mathbf{i}]$ es exacto. Por ejemplo seleccionamos

las mediciones de los vectores de aceleración \mathbf{a}_b y \mathbf{a}_i como las más correctas y usamos estas direcciones como los primeros vectores base de $\{\mathbf{t}\}$,

$$\begin{aligned} t_{1b} &= \mathbf{a}_b \\ t_{1i} &= \mathbf{a}_i. \end{aligned} \quad (3.8)$$

Construimos el segundo vector base de $\{\mathbf{t}\}$ como un vector unitario en la dirección perpendicular al plano de las dos observaciones iniciales, esto es,

$$\begin{aligned} t_{2b} &= \frac{\mathbf{a}_b \times \mathbf{m}_b}{|\mathbf{a}_b \times \mathbf{m}_b|} \\ t_{2i} &= \frac{\mathbf{a}_i \times \mathbf{m}_i}{|\mathbf{a}_i \times \mathbf{m}_i|}. \end{aligned} \quad (3.9)$$

El tercer vector base está dado de la siguiente forma,

$$\begin{aligned} t_{3b} &= t_{1b} \times t_{2b} \\ t_{3i} &= t_{1i} \times t_{2i}. \end{aligned} \quad (3.10)$$

Ahora construimos dos matrices de rotación colocando los componentes de los vectores $[t_{1b}, t_{2b}, t_{3b}]$ y $[t_{1i}, t_{2i}, t_{3i}]$ en columnas, de tal forma que tenemos

$$\begin{bmatrix} t_{1b} & t_{2b} & t_{3b} \end{bmatrix} \quad \text{y} \quad \begin{bmatrix} t_{1i} & t_{2i} & t_{3i} \end{bmatrix}, \quad (3.11)$$

es evidente que estas matrices son R_t^b y R_i^t respectivamente. Para obtener la matriz deseada R_i^b que planteamos en (3.7) simplemente realizamos la multiplicación,

$$R_i^b = R_t^b R_i^t = \begin{bmatrix} t_{1b} & t_{2b} & t_{3b} \end{bmatrix} \begin{bmatrix} t_{1i} & t_{2i} & t_{3i} \end{bmatrix}^T, \quad (3.12)$$

es así como la ecuación (3.12) completa el algoritmo TRIAD.

Es importante resaltar que para conocer los vectores inerciales \mathbf{a}_i y \mathbf{m}_i se debe tener un modelo que describa el campo magnético y gravitatorio de la tierra. De esta forma, conociendo las coordenadas geográficas (Latitud, Longitud, Altitud) del marco de referencia $\{\mathbf{i}\}$ podemos calcular \mathbf{a}_i y \mathbf{m}_i . Sin embargo, pueden existir interferencias de campos generados por otras fuentes que el modelo no contemple; por ejemplo, en el caso del campo magnético, un motor eléctrico en funcionamiento o material ferromagnético cercano al marco fijo $\{\mathbf{i}\}$ pueden alterar los valores de \mathbf{m}_i . Una forma práctica para medir \mathbf{a}_i y \mathbf{m}_i es obteniendo el promedio de una serie de muestras tomadas con el magnetómetro y el acelerómetro de la IMU orientada con el marco $\{\mathbf{i}\}$ y cuidando que las velocidades angulares de los sensores sean cero.

De manera resumida presentamos el algoritmo del método TRIAD, donde observamos que los valores de \mathbf{a}_i y \mathbf{m}_i corresponden en este caso al campo gravitacional y magnético del marco inercial, pero pueden variar dependiendo de la posición y orientación de $\{\mathbf{i}\}$.

Algoritmo 1 Método TRIAD.

Entrada: $\mathbf{a}_b = [a_x; a_y; a_z]^T$, $\mathbf{m}_b = [m_x; m_y; m_z]^T$
 $\mathbf{a}_i = [0; 0; -9,81]^T$, $\mathbf{m}_i = [0,377; -0,499; -1,124]^T$

Salida: R_i^b Matriz de orientación.

```

1: for  $k = 1, 2, \dots, N$  do
2:    $a_B = \frac{\mathbf{a}_b}{|\mathbf{a}_b|}$  ,  $m_B = \frac{\mathbf{m}_b}{|\mathbf{m}_b|}$ 
3:    $a_A = \frac{\mathbf{a}_i}{|\mathbf{a}_i|}$  ,  $m_A = \frac{\mathbf{m}_i}{|\mathbf{m}_i|}$ 
4:    $t_{1b} = a_B$  ,  $t_{1i} = a_A$ 
5:    $t_{2b} = \frac{a_B \times m_B}{|a_B \times m_B|}$  ,  $t_{2i} = \frac{a_A \times m_A}{|a_A \times m_A|}$ 
6:    $t_{3b} = t_{1b} \times t_{2b}$  ,  $t_{3i} = t_{1i} \times t_{2i}$ 
7:    $R_i^b(k) = [t_{1b}, t_{2b}, t_{3b}] [t_{1i}, t_{2i}, t_{3i}]^T$ 
8: end for

```

3.2.2. Método QUEST

El método QUEST (1978, M. Shuster [9]) es una aproximación eficiente del método Q, y se utiliza cuando dos o más vectores de observación están disponibles. En este método iniciamos considerando n vectores linealmente independientes v_k con $k = 1, 2, \dots, n$, y por cada uno de ellos tenemos un sensor que mide su valor correspondiente v_{kb} en el marco móvil $\{\mathbf{b}\}$, y un modelo matemático para los componentes de v_{ki} en el marco fijo $\{\mathbf{i}\}$. El objetivo es encontrar la matriz R_i^b , tal que se cumpla

$$v_{kb} = R_i^b v_{ki} \quad (3.13)$$

y además que minimice la función de costo del problema de Wahba [5],

$$J(R_i^b) = \frac{1}{2} \sum_{k=1}^n \rho_k |v_{kb} - R_i^b v_{ki}|^2 \quad (3.14)$$

donde $J(R_i^b)$ es la función a ser minimizada y representa la suma del cuadrado de los errores para cada vector muestrado, v_{kb} y v_{ki} son los vectores de observación, k es el contador para los n vectores y ρ_k es una constante de peso para ponderar cada vector. Si las mediciones v_{kb} y v_{ki} son exactas, entonces (3.13) se satisface para los n vectores y $J(R_i^b) = 0$; de lo contrario si existen errores o ruido en las mediciones entonces $J(R_i^b) > 0$. La mejor aproximación de R_i^b se presenta cuando $J(R_i^b)$ sea lo

más pequeña posible. Comenzamos por expandir la función de costo de la siguiente forma,

$$J(R_i^b) = \frac{1}{2} \sum_{k=1}^n \rho_k (v_{kb} - R_i^b v_{ki})^T (v_{kb} - R_i^b v_{ki}) \quad (3.15)$$

$$J(R_i^b) = \frac{1}{2} \sum_{k=1}^n \rho_k (v_{kb}^T v_{kb} + v_{ki}^T v_{ki} - 2v_{kb}^T R_i^b v_{ki}). \quad (3.16)$$

Suponemos que los vectores de observación están normalizados, por lo tanto los primeros dos términos satisfacen $v_{kb}^T v_{kb} = v_{ki}^T v_{ki} = 1$ y la función de costo se transforma en,

$$J(R_i^b) = \sum_{k=1}^n \rho_k (1 - v_{kb}^T R_i^b v_{ki}). \quad (3.17)$$

Minimizar $J(R_i^b)$ en (3.17) es igual a maximizar la función de ganancia,

$$g(R_i^b) = \sum_{k=1}^n \rho_k v_{kb}^T R_i^b v_{ki}. \quad (3.18)$$

La clave para resolver este problema de optimización es plantearlo en términos de cuaterniones $\mathbf{q} = [\eta, \epsilon^T]^T$, para lo cual usamos la transformación,

$$R_i^b = (\eta^2 - \epsilon^T \epsilon) I_3 + 2\epsilon \epsilon^T - 2\eta \mathbf{S}(\epsilon). \quad (3.19)$$

Entonces, la función de ganancia $g(R_i^b)$, puede ser reescrita en términos de cuaterniones en lugar de la matriz de rotación. Realizando las sustituciones pertinentes llegamos a

$$g(\mathbf{q}) = \mathbf{q}^T K \mathbf{q} \quad (3.20)$$

donde K es una matriz de 4×4 dada por

$$K = \begin{bmatrix} S - \sigma I & Z \\ Z^T & \sigma \end{bmatrix} \quad (3.21)$$

con

$$\begin{aligned} B &= \sum_{k=1}^n \rho_k (v_{kb} v_{ki}^T) \\ S &= B + B^T \\ Z &= [B_{23} - B_{32}, B_{31} - B_{13}, B_{12} - B_{21}]^T \\ \sigma &= \text{tr}[B]. \end{aligned} \quad (3.22)$$

Para maximizar la función de ganancia (3.20) necesitamos derivarla con respecto a \mathbf{q} , pero como los elementos del cuaternión no son independientes, la restricción $\mathbf{q}^T \mathbf{q} = 1$, también debe ser satisfecha. Añadiendo entonces la restricción a la función $g(\mathbf{q})$ con un multiplicador de Lagrange generamos una nueva función,

$$g'(\mathbf{q}) = \mathbf{q}^T K \mathbf{q} - \lambda \mathbf{q}^T \mathbf{q} \quad (3.23)$$

derivando (3.23) nos damos cuenta que $g'(q)$ tiene un valor extremo cuando

$$K \mathbf{q} = \lambda \mathbf{q}. \quad (3.24)$$

La ecuación ecuación (3.24) es fácilmente reconocida como un problema de valores propios. La orientación óptima estará determinada entonces por un vector propio de la matriz K . Sin embargo, hay cuatro valores propios y a cada uno le corresponde un vector propio diferente. Para saber qué valor propio corresponde al vector propio óptimo el cual maximiza la función de ganancia, sustituimos (3.24) en (3.20),

$$\begin{aligned} g(\mathbf{q}) &= \mathbf{q}^T K \mathbf{q} \\ &= \mathbf{q}^T \lambda \mathbf{q} \\ &= \lambda \mathbf{q}^T \mathbf{q} \\ &= \lambda. \end{aligned} \quad (3.25)$$

Esto demuestra que el valor propio mas grande de K maximiza la función de ganancia, de esta forma, el vector propio correspondiente a λ_{max} es el mínimo cuadrado óptimo estimado de la orientación.

El método Q involucra resolver el problema de valor-vector propio directamente, que se soluciona fácilmente usando Matlab u otras herramientas computacionales; sin embargo, es una desventaja cuando se desea implementar en un sistema embebido. El método QUEST proporciona una forma fácil de estimar el valor propio más grande y resuelve para el correspondiente vector propio.

Recordemos que la orientación óptima se obtiene minimizando la función de costo,

$$J(R_i^b) = \sum_{k=1}^N \rho_k (1 - V_{kB}^T R_B^A V_{kA}) \quad (3.26)$$

o maximizando la función de ganancia,

$$g = \sum_{k=1}^N \rho_k v_{kb}^T R_i^b v_{ki} = \mathbf{q}^T K \mathbf{q} = \lambda_{opt}. \quad (3.27)$$

Reordenando las expresiones (3.26) y (3.27) obtenemos un resultado útil para el valor propio óptimo,

$$\lambda_{opt} = \sum_{k=1}^n \rho_k - J. \quad (3.28)$$

Como la función de costo debe ser mínima, $J \approx 0$, entonces una buena aproximación para el valor propio óptimo es,

$$\lambda_{opt} \approx \sum_{k=1}^n \rho_k. \quad (3.29)$$

Esta es la aproximación que se hace en el método QUEST para evitar resolver el problema de valor-vector propio del método Q. Una vez que el valor propio óptimo ha sido estimado, se calcula el correspondiente vector propio que corresponde al cuaternion de orientación. Una forma de hacer ésto es parametrizar el cuaternion obtenido con los parámetros de Rodrigues, definidos de la siguiente forma,

$$\mathbf{p} = [p_1, p_2, p_3]^T = \frac{\epsilon}{\eta} \begin{bmatrix} \theta \\ 2e \end{bmatrix}. \quad (3.30)$$

Sustituyendo (3.30) y (3.21) en (3.24) y haciendo las operaciones necesarias tenemos

$$\begin{bmatrix} S - \sigma I_3 & Z \\ Z^T & \sigma \end{bmatrix} \begin{bmatrix} 1 \\ \mathbf{p} \end{bmatrix} = \lambda_{opt} \begin{bmatrix} 1 \\ \mathbf{p} \end{bmatrix} \quad (3.31)$$

$$S - \sigma I_3 + Z \mathbf{p}^T = \lambda_{opt} I_3 \quad (3.32)$$

$$Z + \sigma \mathbf{p} = \lambda_{opt} \mathbf{p}. \quad (3.33)$$

A partir de (3.32) llegamos a la siguiente ecuación para calcular los parámetros de Rodrigues,

$$\mathbf{p} = [(\lambda_{opt} + \sigma)I_3 - S]^{-1}Z. \quad (3.34)$$

Obtener la inversa de esta expresión es también una operación que requiere muchas operaciones, por lo tanto podemos considerar enfoques como la eliminación de Gauss-Jordan u otros métodos de sistemas lineales para resolver la matriz inversa y calcular \mathbf{p} ,

$$[(\lambda_{opt} + \sigma)I_3 - S]\mathbf{p} = Z. \quad (3.35)$$

Una vez que los parámetros de Rodrigues se han calculado, realizamos la transformación a cuaterniones utilizando la siguiente expresión

$$\mathbf{q} = \frac{1}{\sqrt{1 + \mathbf{p}^T \mathbf{p}}} \begin{bmatrix} 1 \\ \mathbf{p} \end{bmatrix}, \quad (3.36)$$

obteniéndose una orientación estimada representada en cuaterniones a través del método QUEST. Un problema con este enfoque es que los parámetros de Rodrigues presentan una singularidad en la ecuación (3.30) cuando la rotación es de π radianes,

ya que $\frac{\theta}{2} \rightarrow \infty$ cuando $\theta \rightarrow \pi$. A continuación mostramos de manera resumida la forma en como se programaría el método QUEST.

Algoritmo 2 Método QUEST.

Entrada: $\mathbf{a}_b = [a_x; a_y; a_z]^T$, $\mathbf{m}_b = [m_x; m_y; m_z]^T$
 $\mathbf{a}_i = [0; 0; -9,81]^T$, $\mathbf{m}_i = [0,377; -0,499; -1,124]^T$

Salida: $\mathbf{q} = [\eta, \epsilon^T]^T = [\eta, \epsilon_1, \epsilon_2, \epsilon_3]^T$ Cuaternión de orientación.

- 1: **for** $k = 1, 2, \dots, N$ **do**
 - 2: $v_{1b} = \frac{\mathbf{m}_i}{|\mathbf{m}_i|}$, $v_{2b} = \frac{\mathbf{m}_b}{|\mathbf{m}_b|}$
 - 3: $v_{1i} = \frac{\mathbf{a}_i}{|\mathbf{a}_i|}$, $v_{2i} = \frac{\mathbf{m}_i}{|\mathbf{m}_i|}$
 - 4: Define: $\rho_k = [\rho_1, \rho_2]^T$, $\lambda_{opt} \approx \sum_{k=1}^N \rho_k = \rho_1 + \rho_2$
 - 5: $B = \rho_1(v_{1b} v_{1i}^T) + \rho_2(v_{2b} v_{2i}^T)$
 - 6: $S = B + B^T$
 - 7: $Z = [B_{23} - B_{32}, B_{31} - B_{13}, B_{12} - B_{21}]^T$
 - 8: $\sigma = \text{tr}[B]$
 - 9: $\mathbf{p} = [(\lambda_{opt} + \sigma)I_3 - S]^{-1}Z$
 - 10: $\mathbf{q}(k) = \frac{1}{\sqrt{1+\mathbf{p}^T\mathbf{p}}} \begin{bmatrix} 1 \\ \mathbf{p} \end{bmatrix} = [\eta, \epsilon_1, \epsilon_2, \epsilon_3]^T$
 - 11: **end for**
-

3.3. Algoritmos de Estimación Estocásticos

En los métodos determinísticos supusimos que las mediciones de los sensores están libres de ruido, pero ésto regularmente no es así. En esta sección describiremos dos métodos para estimar la orientación basados en el filtro de Kalman donde se consideran el ruido en los sensores y los errores del modelo. Empezaremos introduciendo la formulación matemática de cada método de estimación y describiremos brevemente el filtro de Kalman.

3.3.1. Fusión Sensorial

La fusión de datos o fusión sensorial (*sensor fusion*) se refiere a la combinación de datos de distintos sensores de tal forma que el resultado sea en algún sentido mejor que cuando los sensores se usan de manera individual. Por ejemplo, el resultado puede ser más exacto o se puede lograr un nuevo resultado, como sucede en la visión estereoscópica donde se puede crear una ilusión 3D a partir de dos imágenes 2D, otros ejemplos prácticos de la fusión de datos son el GPS (*Global Positioning System*), el radar, los sensores sísmicos, etc.

En el método reportado en [46] veremos como estimar los ángulos de alabeo y cabeceo usando sólo el giroscopio por un lado y por el otro usando el acelerómetro. Después fusionaremos ambas mediciones a través del filtro de Kalman para obtener una estimación más exacta eliminando la deriva del giroscopio.

Estimación a partir del Giroscopio

En comparación con el acelerómetro, el giroscopio presenta las mediciones menos sensibles a ruidos externos; por lo tanto las velocidades angulares $\Omega = [p, q, r]^T = [\dot{\phi}, \dot{\psi}, \dot{\theta}]^T$ medidas ya sea en ($^{\circ}/s$) o (rad/s), se pueden integrar para calcular los ángulos de rotación en cada eje simplificando las ecuaciones cinemáticas (2.33) y considerando la aproximación de ángulos pequeños $\theta_i \approx 0$, $\cos(\theta_i) \approx 1$ y $\sin(\theta_i) = \theta_i$.

$$\phi_g = \int_{t_0}^t p(t) dt \quad (3.37)$$

$$\theta_g = \int_{t_0}^t q(t) dt \quad (3.38)$$

Aproximando las integrales a su forma discreta, podemos calcular el ángulo de rotación a través del producto entre la velocidad angular instantánea y el tiempo de muestreo Δt , esto es,

$$\phi_{g_{k+1}} = \phi_{g_k} + \Delta t p_k \quad (3.39)$$

$$\theta_{g_{k+1}} = \theta_{g_k} + \Delta t q_k. \quad (3.40)$$

Existen dos inconvenientes principales al tratar de encontrar la orientación usando solo el giroscopio. El primero se conoce como deriva o *drift* que se produce a lo largo del tiempo debido a que las ecuaciones (3.40) son acumulativas, es decir, el ruido en las mediciones del giroscopio y los errores de integración provocarán que para lapsos pequeños de tiempo tengamos un valor cercano al ángulo de rotación real; pero para periodos largos los valores calculados estarán lejos de la medida real. El segundo inconveniente al usar sólo el giroscopio es que no se puede saber el estado inicial o

ángulo de inicio del cuerpo rígido, a menos que el sistema tenga siempre un estado inicial con el que se inicie el algoritmo acumulativo. Con los argumentos anteriores, podemos notar que es complicado obtener una estimación confiable usando solo el giroscopio.

Estimación a partir del acelerómetro

Por otro lado el acelerómetro proporciona el vector de aceleración $\mathbf{a}_b = [a_x, a_y, a_z]^T$ de cada uno de los ejes del marco $\{\mathbf{b}\}$, cuando los ejes \hat{b}_x y \hat{b}_y del sensor forman un plano paralelo a la superficie de la tierra, el eje \hat{b}_z del acelerómetro permanece perpendicular y mide la aceleración gravitacional $\mathbf{a}_b = [0, 0, g]^T$, con $g = 9,81 \text{ m/s}^2$. Cuando el cuerpo rígido toma una inclinación con respecto al eje \hat{b}_x provoca en este caso que el eje \hat{b}_y , que corresponde al alabeo, entregue una señal debido a la componente de gravedad presente como se puede observar en la Figura 3.2.

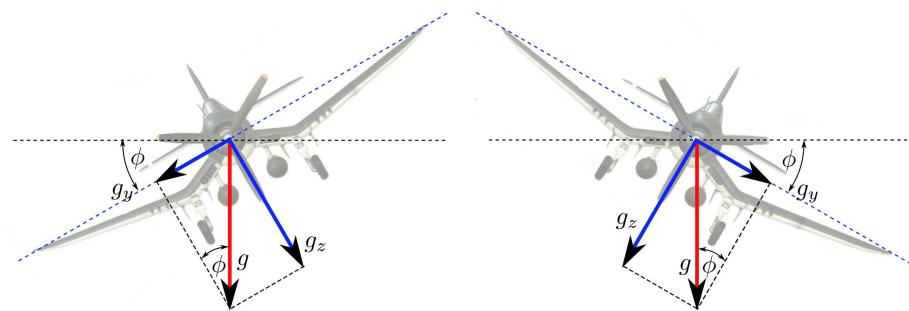


Figura 3.2: Inclinación sobre el eje X - Alabeo.

Esto sucede de igual forma en el caso del cabeceo, donde la inclinación es con respecto al eje \hat{b}_y , ver Figura 3.3.

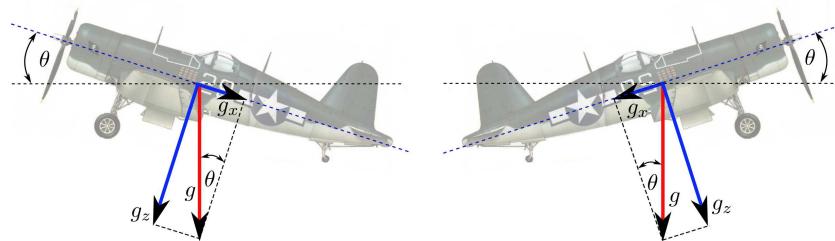


Figura 3.3: Inclinación sobre el eje Y - Cabeceo.

Para obtener los ángulos de inclinación, ϕ y θ , se utiliza las siguientes fórmulas que se derivan geométricamente a partir del vector de aceleración, véase [47].

$$\begin{aligned}\phi_{ak} &= \arctan\left(\frac{a_x}{\sqrt{a_y^2 + a_z^2}}\right) \\ \theta_{ak} &= \arctan\left(\frac{a_y}{\sqrt{a_x^2 + a_z^2}}\right)\end{aligned}\quad (3.41)$$

La sensibilidad del acelerómetro al ruido y a perturbaciones externas provoca que los ángulos ϕ_{ak} y θ_{ak} calculados mediante (3.41) presenten discontinuidades y variaciones pequeñas, pero tienen la ventaja de no presentar deriva o *drift*. Debemos tener en cuenta que el acelerómetro estimará un ángulo correcto sólo en las condiciones donde no se presenta aceleración lineal en alguno de sus ejes, de lo contrario el ángulo calculado no será correcto. esto se debe a que el acelerómetro mide la aceleración de la gravedad más la aceleración lineal.

Fusion sensorial

Tomando como referencia los dos esquemas anteriores, apreciamos claramente que al usar tanto el acelerómetro como el giroscopio de manera individual se tienen varios inconvenientes que producen que el resultado deseado tienda a presentar errores. Surge entonces la necesidad de fusionar ambos sensores con el fin de tomar las ventajas de cada uno a través del filtro de Kalman que se aplica a sistemas de la siguiente forma,

$$x_{k+1} = A x_k + B u_k + H_{k-1} \zeta_k \quad (3.42)$$

$$y_{k+1} = C x_k + \xi_k \quad (3.43)$$

donde se tiene un modelo del sistema (3.42) y un modelo de medición (3.43), ζ_k y ξ_k son variables aleatorias que representan el ruido del modelo y ruido de medición respectivamente, tienen una distribución Gaussiana $\zeta_k \sim N(\mu, \sigma^2)$, $\xi_k \sim N(\mu, \sigma^2)$ con media μ y varianza σ^2 , véase [48].

La idea principal del algoritmo de fusión es estimar el ángulo de inclinación de cada eje a través de las mediciones del giroscopio en el modelo del sistema y corregir tales estimaciones a través del modelo de medición que utiliza las mediciones del acelerómetro. De esta forma se tiene un estimado de la orientación donde se atenúa el ruido en la medición y el ruido del sistema. Usando los modelos acumulativos (3.40) y (3.39) para calcular los ángulos de inclinación con el giroscopio tenemos,

$$\phi_{gk+1} = \phi_{gk} + \Delta t(p_k - \zeta_{1k}) \quad (3.44)$$

$$\theta_{gk+1} = \theta_{gk} + \Delta t(q_k - \zeta_{2k}) \quad (3.45)$$

donde ϕ_{gk+1} y θ_{gk+1} son los ángulos a estimar y los términos ζ_{1k} y ζ_{2k} que se restan a las velocidades angulares representan el ruido en el sistema y se pueden ver como la

deriva o la desviación de los ángulos estimados. Tomando a ζ_{1k} y ζ_{2k} como estados del sistema y considerando la aproximación $\zeta_{k+1} = \zeta_k$, se pueden reescribir las ecuaciones (3.45) y (3.44) en su representación de espacio de estados de la siguiente forma

$$x_k = \begin{bmatrix} \phi_{gk+1} \\ \zeta_{1k+1} \\ \theta_{gk+1} \\ \zeta_{2k+1} \end{bmatrix} = \begin{bmatrix} 1 & -\Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -\Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \phi_{gk} \\ \zeta_{1k} \\ \theta_{gk} \\ \zeta_{2k} \end{bmatrix} + \begin{bmatrix} \Delta t p_k \\ 0 \\ \Delta t q_k \\ 0 \end{bmatrix}. \quad (3.46)$$

Las mediciones del acelerómetro son parte del filtro de Kalman a través del modelo de medición; de esta manera los ángulos ϕ_{ak} y θ_{ak} dados por (3.41) sirven para determinar el error entre el ángulo estimado por el giroscopio y la medición del acelerómetro en ese instante, atenuando así el llamado *drift* en la etapa de innovación del filtro de Kalman mediante $\hat{x}_{k|k} = \hat{x}_{k|k-1} + G_k(y_k - C\hat{x}_{k|k-1})$. El modelo de medición toma la siguiente forma

$$y_{k+1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \phi_{ak} \\ \zeta_{1k} \\ \theta_{ak} \\ \zeta_{2k} \end{bmatrix} + \begin{bmatrix} \xi_{1k} \\ 0 \\ \xi_{2k} \\ 0 \end{bmatrix}, \quad (3.47)$$

donde los términos ξ_{1k} y ξ_{2k} representan el ruido en las mediciones del acelerómetro.

Con las ecuaciones (3.46) y (3.47) se puede implementar el filtro de Kalman con sus dos etapas de actualización y corrección para estimar ϕ_k y θ_k como se presenta de forma resumida en el Algoritmo 3, donde $P_{0,0} = var(x_0)$ representa la matriz de error de covarianza y $Q = \mathbf{I}_4 var(\zeta)$, $R = \mathbf{I}_4 var(\xi)$ las matrices de la varianza del ruido de proceso y de medición respectivamente. Para mayores detalles sobre el filtro de Kalman referirse a [49], [48].

Algoritmo 3 Fusión Sensorial usando el Filtro de Kalman

Entrada: $\mathbf{a}_b = [a_x, a_y, a_z]^T$, $\Omega_b = [p, q, r]^T$, $P_{0,0} = var(x_0)$, $\hat{x}_0 = E(x_0)$

Salida: $\hat{x}_{k|k} = [\hat{\phi}, \hat{\zeta}_1, \hat{\theta}, \hat{\zeta}_2]$ Vector de estados estimados.

- 1: **for** $k = 1, 2, \dots$ **do**
 - 2: $Q = \mathbf{I}_4 var(\zeta)$, $R = \mathbf{I}_4 var(\xi)$
 - 3: $P_{k|k-1} = A P_{k-1,k-1} A^T + H_{k-1} Q_{k-1} H_{k-1}^T$
 - 4: $\hat{x}_{k|k-1} = A \hat{x}_{k-1|k-1} + B_{k-1} u_{k-1}$
 - 5: $G_k = P_{k|k-1} C^T (C P_{k|k-1} C^T + R_k)^{-1}$
 - 6: $P_{k|k} = [I - G_k C] P_{k,k-1}$
 - 7: $\hat{x}_{k|k} = \hat{x}_{k|k-1} + G_k (y_k - C \hat{x}_{k|k-1})$
 - 8: **end for**
-

3.3.2. Filtro de Kalman

En esta sección abordaremos otro método para estimar la orientación utilizando un esquema diferente al método de Fusión sensorial, pero que también hace uso del filtro de Kalman. En la literatura se encuentran principalmente dos enfoques para diseñar un estimador de orientación, reportados en [50] y [51], que utilizan los cuaterniones para representar la orientación. Explicaremos brevemente cada uno de ellos y las modificaciones del método que se implementó en este trabajo.

Para diseñar el filtro de Kalman el primer paso es desarrollar el modelo del proceso, en este caso el modelo de un cuerpo rígido bajo movimiento rotacional como se explica en el Capítulo 2; el vector de estados del modelo de proceso consistirá entonces de las velocidades angulares $\Omega_b = [p, q, r]$ y de los parámetros que caractericen la orientación, que pueden ser los cuaterniones $\mathbf{q} = [\eta, \epsilon_1, \epsilon_2, \epsilon_3]$, los ángulos de Euler, etc.

El primer enfoque es el estándar, el cual contempla siete estados $x = [p, q, r, \eta, \epsilon_1, \epsilon_2, \epsilon_3]^T$ y nueve mediciones $y = [p, q, r, a_x, a_y, a_z, m_x, m_y, m_z]^T$ tomadas directamente de los sensores iniciales, lo que resulta en ecuaciones de medición no lineales con lo que el filtro de Kalman resultante se vuelve muy extenso, haciendo difícil la implementación en un sistema embebido.

El segundo enfoque hace uso de un algoritmo determinístico para aproximar la orientación de manera algebraica, que puede ser el método de Gauss-Newton, TRIAD o QUEST para calcular un cuaternion aproximado en cada medición del acelerómetro y del magnetómetro. El cuaternion calculado es tomado como parte de las mediciones del filtro de Kalman además de las velocidades angulares. Como resultado las salidas del filtro se reducen de nueve a siete, $y_k = [p, q, r, \eta, \epsilon_1, \epsilon_2, \epsilon_3]^T$ y más aún, las ecuaciones de salida se transforman en ecuaciones lineales, lo cual simplifica en gran medida el diseño del filtro.

Aunque las ecuaciones de salida son lineales se requiere un filtro de Kalman extendido ya que parte de las ecuaciones de estado del proceso son no lineales. Sin embargo, la linealidad en las ecuaciones de salida simplifica significativamente el diseño del filtro y reduce los requisitos computacionales para su implementación.

Lo que se planteó en este trabajo fue utilizar el método TRIAD para calcular el cuaternion instantáneo de la orientación en cada medición del acelerómetro y del magnetómetro y considerar a $\Omega = [p, q, r]^T$ como parámetros variantes en el tiempo de la matriz A en (3.42) del modelo del proceso, generando así un sistema lineal variante en el tiempo con vector de estados $x_k = [\eta, \epsilon_1, \epsilon_2, \epsilon_3]^T$. El cuaternion calculado por el método TRIAD formará parte de la ecuación de medición del filtro de Kalman junto con las mediciones de las velocidades angulares, manteniendo así cuatro estados en el modelo del sistema y siete en las ecuaciones de medición que son lineales, lo cual

nos permite usar el filtro de Kalman en su versión lineal. La Fig.(3.4) ejemplifica lo anterior.

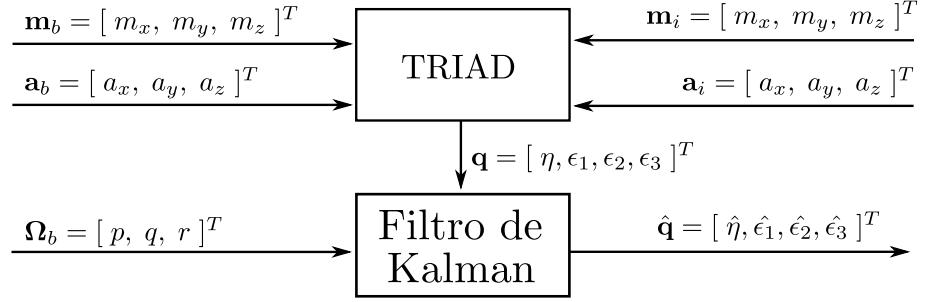


Figura 3.4: Diagrama del método del filtro de Kalman.

Utilizando las ecuaciones diferenciales (2.4.3) que describen la cinemática de un cuerpo rígido usando cuaterniones, que se derivan en la Sección 2.4.3, tenemos el siguiente modelo de proceso:

$$\begin{bmatrix} \dot{\eta} \\ \dot{\epsilon}_1 \\ \dot{\epsilon}_2 \\ \dot{\epsilon}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \begin{bmatrix} \eta \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix} + \begin{bmatrix} \zeta_\eta \\ \zeta_{\epsilon_1} \\ \zeta_{\epsilon_2} \\ \zeta_{\epsilon_3} \end{bmatrix} \quad (3.48)$$

donde los términos ζ_i representan las secuencias de ruido Gaussiano de cada elemento del cuaternion. Recordemos que tenemos en total nueve mediciones disponibles de la unidad de medición inercial, pero usando el algoritmo TRIAD se reducen a siete. El modelo de medición toma la forma,

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \\ \eta \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix} + \begin{bmatrix} \xi_p \\ \xi_q \\ \xi_r \\ \xi_{\epsilon_0} \\ \xi_{\epsilon_1} \\ \xi_{\epsilon_2} \\ \xi_{\epsilon_3} \end{bmatrix} \quad (3.49)$$

Los términos ξ_i representan el ruido en la medición de los sensores. Teniendo ahora el modelo continuo del sistema y de medición los podemos discretizar usando el método de Euler,

$$x = x_k \quad (3.50)$$

$$\dot{x} = \frac{x_{k+1} - x_k}{\Delta t}. \quad (3.51)$$

Realizando las operaciones necesarias, el modelo discretizado del sistema queda de la siguiente forma,

$$\begin{bmatrix} \eta_{k+1} \\ \epsilon_{1\ k+1} \\ \epsilon_{2\ k+1} \\ \epsilon_{3\ k+1} \end{bmatrix} = \frac{h}{2} \begin{bmatrix} \frac{2}{\Delta t} & -p_k & -q_k & -r_k \\ p_k & \frac{2}{\Delta t} & r_k & -q_k \\ q_k & -r_k & \frac{2}{\Delta t} & p_k \\ r_k & q_k & -p_k & \frac{2}{\Delta t} \end{bmatrix} \begin{bmatrix} \eta_k \\ \epsilon_{1\ k} \\ \epsilon_{2\ k} \\ \epsilon_{3\ k} \end{bmatrix} + \begin{bmatrix} \Delta t \xi_{\eta\ k} \\ \Delta t \xi_{\epsilon_1\ k} \\ \Delta t \xi_{\epsilon_2\ k} \\ \Delta t \xi_{\epsilon_3\ k} \end{bmatrix}. \quad (3.52)$$

Para el modelo de medición tenemos,

$$\begin{bmatrix} y_{1\ k} \\ y_{2\ k} \\ y_{3\ k} \\ y_{4\ k} \\ y_{5\ k} \\ y_{6\ k} \\ y_{7\ k} \end{bmatrix} = \begin{bmatrix} p_k + \xi_{p\ k} \\ q_k + \xi_{q\ k} \\ r_k + \xi_{r\ k} \\ \eta_k + \xi_{\eta\ k} \\ \epsilon_{1\ k} + \xi_{\epsilon_1\ k} \\ \epsilon_{2\ k} + \xi_{\epsilon_2\ k} \\ \epsilon_{3\ k} + \xi_{\epsilon_3\ k} \end{bmatrix}. \quad (3.53)$$

Teniendo los modelos (3.53) y (3.52) se implementa el filtro de Kalman en su versión lineal como se presenta en el Algoritmo 4, donde de igual forma $P_{0,0} = var(x_0)$ corresponde a la matriz de covarianza y $Q = \mathbf{I}_4 var(\zeta)$, $R = \mathbf{I}_4 var(\xi)$ representan las matrices de varianza de las variables aleatorias del ruido de proceso y medición respectivamente.

Algoritmo 4 Filtro de Kalman.

Entrada: $\mathbf{a}_b = [a_x, a_y, a_z]^T$, $\mathbf{m}_b = [m_x, m_y, m_z]^T$, $\Omega_b = [p, q, r]^T$, $\mathbf{a}_i = [0; 0; 9,81]^T$, $\mathbf{m}_i = [0,377; -0,499; -1,124]^T$, $P_{0,0} = var(x_0)$, $\hat{x}_0 = E(x_0)$, $\mathbf{q}(0) = [0, 0, 0, 0]$

Salida: $\hat{x}_{k|k} = [\hat{\eta}, \hat{\epsilon}_1, \hat{\epsilon}_2, \hat{\epsilon}_3]$ Vector de estados estimados.

- 1: **for** $k = 1, 2, \dots$ **do**
 - 2: $\mathbf{q}(k) = [\eta, \epsilon_1, \epsilon_2, \epsilon_3] = TRIAD(\mathbf{a}_b, \mathbf{m}_b, \mathbf{a}_i, \mathbf{m}_i)$
 - 3: $Q = \mathbf{I}_4 var(\zeta)$, $R = \mathbf{I}_4 var(\xi)$, $C = \mathbf{I}_4$, $H = \mathbf{I}_4 \Delta t$, $B = 0$
 - 4: $P_{k|k-1} = A P_{k-1,k-1} A^T + H_{k-1} Q_{k-1} H_{k-1}^T$
 - 5: $\hat{x}_{k|k-1} = A \hat{x}_{k-1|k-1} + B_{k-1} u_{k-1}$
 - 6: $G_k = P_{k|k-1} C^T (C P_{k|k-1} C^T + R_k)^{-1}$
 - 7: $P_{k|k} = [I - G_k C] P_{k,k-1}$
 - 8: $\hat{x}_{k|k} = \hat{x}_{k|k-1} + G_k (y_k - C \hat{x}_{k|k-1})$
 - 9: **end for**
-

3.4. Observadores no Lineales

La observación del estado de un sistema dinámico se refiere a tratar de estimar tal estado utilizando la información de entrada y salida del sistema. En los sistemas lineales se ha hecho una amplia investigación que se ha traducido en aplicaciones muy útiles como el diseño de leyes de control basadas en observadores; sin embargo, para los sistemas no lineales la teoría de observadores es un área de investigación abierta que presenta mayores retos en su aplicación.

La ventaja de los métodos de estimación basados en observadores no lineales es que están diseñados sobre la verdadera naturaleza del sistema, es decir, éstos no se linealizan para tomar ventaja de las técnicas de estimación lineal y su desarrollo está basado en pruebas de estabilidad, permitiendo establecer propiedades sobre la estabilidad del estimador más precisas tales como la determinación de su región de atracción.

En este trabajo hacemos una revisión de los filtros complementarios y de un método basado en la teoría de Inmersión e Invarianza propuestos en la última década.

3.4.1. Filtros Complementarios

Este enfoque está basado en los desarrollos presentados en [24] donde se formula el problema de estimación como un observador cinemático planteado directamente en el grupo especial ortogonal $SO(3)$, que es llamado filtro complementario por la analogía que tiene con la arquitectura de los filtros complementarios lineales, aunque en el caso no lineal no se tenga una interpretación en el dominio de la frecuencia.

Para abordar este método de estimación, además de los marcos cuerpo $\{\mathbf{b}\}$ e inercial $\{\mathbf{i}\}$, se define $\{\mathbf{e}\}$ como el marco del estimador, también se consideran los vectores \mathbf{v}_a (3.4) y \mathbf{v}_m (3.6) que corresponden a las mediciones normalizadas del acelerómetro y el magnetómetro definidos en la Sección 3.1 que son usados para reconstruir una medición algebraica instantánea R_y de la matriz de rotación R_i^b , donde R_y está definida como

$$R_y = \min_{R \in SO(3)} (\rho_1 \| e_3 - R\mathbf{v}_a \|^2 + \rho_2 \| v_m - R\mathbf{v}_b \|^2) \approx R_i^b \quad (3.54)$$

El término e_3 representa el vector de aceleración y v_m representa la dirección del campo magnético, ambos en el marco inercial $\{\mathbf{i}\}$; ρ_1 y ρ_2 son pesos de ponderación que dependen de la exactitud relativa entre las mediciones de los sensores. La ecuación (3.54) implica resolver un problema de minimización que puede llevarse a cabo a través del método de Gauss-Newton o algún otro método de minimización; sin embargo, también se pueden usar los métodos TRIAD o QUEST para calcular R_y .

Ahora, si \hat{R} denota un estimado de la matriz de rotación R_i^b , entonces \hat{R} puede considerarse con coordenadas en el marco de referencia $\{\mathbf{e}\}$, que asociado al marco

inercial puede verse como,

$$\hat{R} = \hat{R}_i^e : \{\mathbf{i}\} \rightarrow \{\mathbf{e}\}. \quad (3.55)$$

Entonces, el objetivo de estimar la orientación es hacer que $\hat{R}_i^e \rightarrow R_i^b$, definiendo así el error de estimación \tilde{R} como la rotación relativa entre el marco $\{\mathbf{b}\}$ y el marco $\{\mathbf{e}\}$, definida de la siguiente forma,

$$\tilde{R} = \hat{R}^T R_i^b = \hat{R}_e^i R_i^b, \quad \tilde{R} = \tilde{R}_e^b : \{\mathbf{e}\} \rightarrow \{\mathbf{b}\} \quad (3.56)$$

como se observa en la Figura 3.5. El propósito de este método de estimación es determinar la dinámica de $\hat{R}(t) \in SO(3)$ tal que el error de rotación $\tilde{R}(t) \rightarrow I_3$, es decir, que los marcos $\{\mathbf{e}\}$ y $\{\mathbf{b}\}$ tengan la misma orientación.

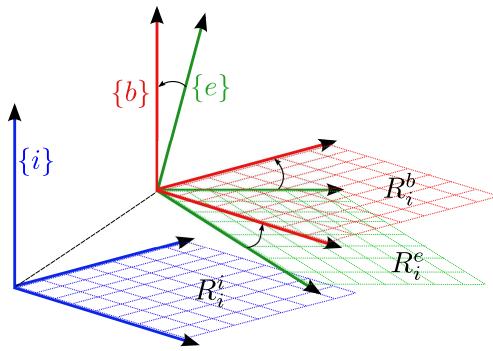


Figura 3.5: Marcos de referencia $\{\mathbf{i}\}$, $\{\mathbf{b}\}$, $\{\mathbf{e}\}$.

Usando las ecuaciones (2.42) de la cinemática de rotación basada en la matriz de cosenos directores derivada en la Sección 2.4.2, que tienen la siguiente estructura,

$$\dot{R}_b^i = R_b^i \mathbf{S}(\Omega_i^b) = \mathbf{S}(R_b^i \Omega_i^b) R_b^i \quad (3.57)$$

el observador se propone con la siguiente forma,

$$\dot{\hat{R}} = \mathbf{S}(R\Omega_i^b + k_p \hat{R}\omega) \hat{R}, \quad \hat{R}(0) = \hat{R}_0 \quad (3.58)$$

que incluye un término de predicción $R\Omega_i^b$ basado en las mediciones de las velocidades angulares Ω_i^b , un término de corrección o innovación $\omega = \omega(\tilde{R})$ que depende del error \tilde{R} y una ganancia positiva k_p . El término $(R\Omega_i^b + k_p \hat{R}\omega)$ está expresado en el marco inercial $\{\mathbf{i}\}$, las velocidades angulares medidas en el marco $\{\mathbf{b}\}$ son transformadas al marco inercial mediante $\Omega_i = R_b^i \Omega_i^b$ y $\omega(\tilde{R})$ está definida como,

$$\omega = \text{vex}(\mathbb{P}_a(\tilde{R})) = \text{vex}(\mathbb{P}_a(\hat{R}^T R_y)) \quad (3.59)$$

donde el operador $\text{vex}(\cdot) : \mathfrak{so}(3) \rightarrow \mathbb{R}^3$ denota la inversa del operador $\mathbf{S}(\cdot)$ definido en (2.21) y el operador $\mathbb{P}_a(\cdot)$ devuelve la parte antisimétrica de una matriz cuadrada

$$\mathbb{P}_a(\tilde{R}) = \frac{1}{2}(\tilde{R} - \tilde{R}^T), \quad \tilde{R} \in \mathbb{R}^{n \times n}. \quad (3.60)$$

El término de corrección $\omega = \omega(\tilde{R})$ tiene como referencia el marco del estimador $\{\mathbf{e}\}$ y se considera como una aproximación no lineal del error entre R_i^b y \hat{R} medido desde el marco de referencia asociado con \hat{R} . En la práctica se implementa como un error entre una medición algebraica instantánea R_y de R_i^b y el estimado \hat{R} . De igual forma, en la implementación se considera que $\Omega_y \approx \Omega + b$, donde Ω corresponde a la velocidad angular real y b a una constante de desviación, recordando que la velocidad angular medida por el giroscopio corresponde a $\Omega_i^b = \Omega + b_\Omega + \mu_\Omega$.

Observemos de la ecuación (3.58) que es necesario conocer R_i^b para mapear las velocidades angulares de $\{\mathbf{b}\}$ al marco inercial $\{\mathbf{i}\}$ para poder calcular $\dot{\hat{R}}$; sin embargo, la rotación verdadera R_i^b no está disponible, de hecho es la rotación que deseamos estimar. Por lo tanto, es necesario utilizar una aproximación eficiente de R_i^b , ésto abre dos posibilidades para implementar el filtro complementario.

Filtro Complementario Directo

En este caso la matriz R_y es usada para transformar las velocidades angulares hacia el marco inercial, de manera que el observador toma la forma,

$$\dot{\hat{R}} = \mathbf{S}(R_y \Omega_y + k_p \hat{R} \omega) \cdot \hat{R} \quad (3.61)$$

Un diagrama de bloques del diseño de este filtro se muestra en la Figura 3.6. Esta versión tiene la ventaja de que no introduce un lazo de realimentación adicional en la dinámica del filtro; sin embargo, el ruido en la rotación reconstruida R_y entrará en el término de prealimentación directa del filtro.

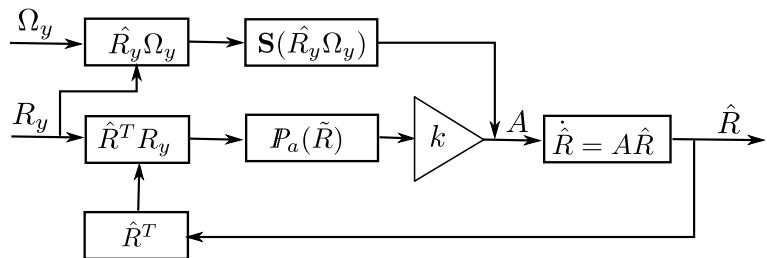


Figura 3.6: Filtro complementario directo.

Se puede extender (3.61) para estimar en línea el término de desviación *bias* de las mediciones del giroscopio. El enfoque adoptado es agregar un integrador para el término compensador en la ecuación de realimentación del filtro. Con $k_p, k_i > 0$, ganancias positivas, se define el filtro complementario directo con corrección de *bias*

de la siguiente forma:

$$\begin{aligned}\dot{\hat{R}} &= \mathbf{S}(R_y(\Omega_y - \hat{b}) + k_p \hat{R} \omega) \hat{R}, & \hat{R}(0) &= \hat{R}_0 \\ \dot{\hat{b}} &= -K_i \omega, & \hat{b}(0) &= \hat{b}_0 \\ \omega &= \text{vex}(\mathbb{P}_a(\tilde{R})), & \tilde{R} &= \hat{R}^T R_y.\end{aligned}\tag{3.62}$$

Filtro Complementario Pasivo

En este caso la orientación filtrada \hat{R} es usada en el término de predicción $\hat{R}\Omega$ para transformar las velocidades angulares al marco inercial $\{1\}$,

$$\dot{\hat{R}} = \mathbf{S}(\hat{R}\Omega_y + k_p \hat{R}\omega) \cdot \hat{R}\tag{3.63}$$

Un diagrama de bloques de esta arquitectura se muestra en al Figura 3.7, la ventaja radica en evitar corromper el término de velocidad angular con el ruido en la rotación reconstruida R_y , sin embargo, el enfoque introduce un lazo de realimentación secundario en el filtro.

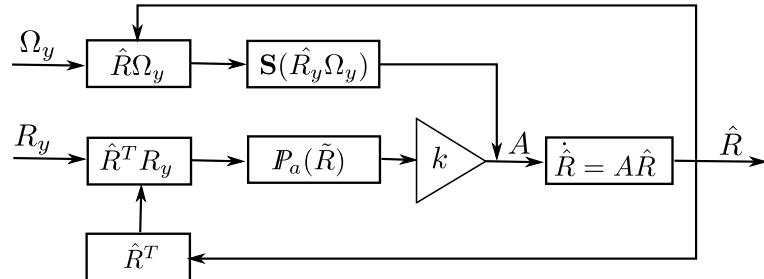


Figura 3.7: Filtro complementario directo

De igual forma, se puede extender (3.63) para obtener un filtro complementario pasivo con corrección de *bias*:

$$\begin{aligned}\dot{\hat{R}} &= \hat{R}\mathbf{S}(\Omega_y - \hat{b} + k_p \omega) \hat{R}, & \hat{R}(0) &= \hat{R}_0 \\ \dot{\hat{b}} &= -K_i \omega, & \hat{b}(0) &= \hat{b}_0 \\ \omega &= \text{vex}(\mathbb{P}_a(\tilde{R})), & \tilde{R} &= \hat{R}^T R_y.\end{aligned}\tag{3.64}$$

A continuación se presenta de manera resumida el filtro complementario directo, que tiene una estructura similar al filtro complementario pasivo, excepto por la matriz R_y que cambia a \hat{R} . En la implementación de este esquema de estimación se utilizó de igual forma el método TRIAD para calcular R_y .

Algoritmo 5 Filtro complementario directo

Entrada: $\mathbf{a}_b = [a_x, a_y, a_z]^T$, $\mathbf{m}_b = [m_x, m_y, m_z]^T$, $\Omega_b = [p, q, r]^T$, $\mathbf{a}_i = [0; 0; 9,81]^T$,
 $\mathbf{m}_i = [0,377; -0,499; -1,124]^T$, k_p, k_i , $\hat{R}(0) = I_3$

Salida: \hat{R}_i^b Matriz de rotación.

```

1: for  $k = 1, 2, \dots$  do
2:    $R_y = \text{TRIAD}(\mathbf{a}_b, \mathbf{m}_b, \mathbf{a}_i, \mathbf{m}_i)$ 
3:    $\tilde{R} = \hat{R}R_y$ 
4:    $\mathbb{P}_a(\tilde{R}) = \frac{1}{2}(\tilde{R} - \tilde{R}^T)$ 
5:    $\omega = [-\mathbb{P}_a(2, 3), \mathbb{P}_a(1, 3), -\mathbb{P}_a(1, 2)]^T$ 
6:    $\dot{\hat{R}} = \mathbf{S}(R_y \Omega_b + k_p \hat{R} \omega) \hat{R}$ 
7:    $\hat{R} = \hat{R} + \Delta t \dot{\hat{R}}$ 
8: end for

```

3.4.2. Filtro basado en Inmersión e Invarianza

Este método para estimar la orientación está basado en los desarrollos presentados en [25] y se basa en las nociones de Inmersión de sistemas e Invarianza de campos (I&I), las cuales son herramientas clásicas de la teoría de regulación no lineal y control geométrico no lineal presentada en [52].

En este método se estiman los ángulos de alabeo y cabeceo usando las mediciones del acelerómetro y del giroscopio que están modeladas respectivamente en la forma,

$$\Omega_i^b = \Omega + b_\Omega + \mu_\Omega, \quad \Omega_i^b \in \mathbb{R}^3 \quad (3.65)$$

$$\mathbf{a}_b = R_i^{bT}(\dot{v} - g) + b_a + \mu_a, \quad \mathbf{a}_b \in \mathbb{R}^3. \quad (3.66)$$

como se explica en la Sección 3.1 y en la observación de que la tercera columna de la matriz de rotación R de la ecuación (2.27) es independiente del ángulo ψ , estimando esta columna haría posible extraer los ángulos ϕ y θ .

Usando las ecuaciones (2.42) de la cinemática de rotación basada en la matriz de cosenos directores derivada en la Sección 2.4.2,

$$\dot{R}_i^b = -\mathbf{S}(\Omega_i^b)R_i^b \quad (3.67)$$

podemos ver que separando las columnas de R_i^b es posible obtener,

$$\dot{r}_j = -\mathbf{S}(\Omega_i^b)r_j, \quad j = 1, 2, 3. \quad (3.68)$$

donde r_j es la j -ésima columna de R_i^b y además $r_3 = R_i^{b^T} g$, con $g = |g_0| \hat{i}_z$ que representa la dirección del campo gravitacional medido desde el marco $\{i\}$. Se plantea entonces el problema de estimación considerando el siguiente modelo,

$$\begin{aligned}\dot{r}_3 &= -\mathbf{S}(\Omega_i^b)r_3 + \xi \\ \mathbf{a}_b &= -r_3 + R_i^{b^T} \dot{\nu} + \zeta_a\end{aligned}\quad (3.69)$$

donde ξ representa una variación de la dinámica real de r_3 causada en parte por los términos $b_\Omega + \mu_\Omega$ del modelo de medición (3.65) de las velocidades angulares, y ζ_a que corresponde a la suma de los términos $b_a + \mu_a$ del modelo de medición (3.66) del acelerómetro. Hay que notar que las desviaciones ξ y ζ_a pueden verse como los ruidos de proceso y medición respectivamente como en el filtro de Kalman. Se deben considerar perfiles bajos de aceleración lineal que no cambien bruscamente, es decir, $\dot{\nu} \approx 0$ para que el vector de referencia que mida el acelerómetro sea perpendicular al plano \hat{i}_x, \hat{i}_y del marco $\{i\}$, como se observa en la Figura 3.8.

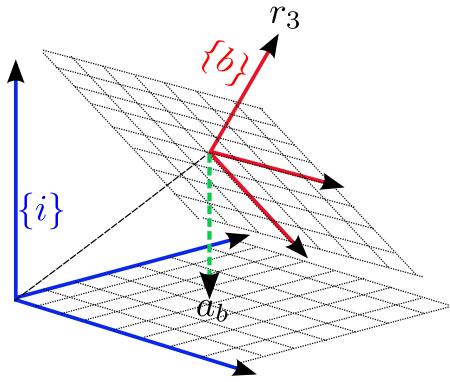


Figura 3.8: Estimación del vector r_3 .

Considerando la extensión dinámica,

$$\dot{\nu} = -r_3 + R_i^{b^T} \dot{\nu} + \zeta_a, \quad (3.70)$$

el sistema dinámico en (3.69) puede ser escrito en la forma,

$$\begin{aligned}\dot{r}_3 &= -\mathbf{S}(\Omega_b)r_3 + \xi \\ \dot{\nu} &= -r_3 + R_i^{b^T} \dot{\nu} + \zeta_a \\ y &= \nu\end{aligned}\quad (3.71)$$

con $y = \nu$ como la nueva salida medida, r_3 corresponde a la tercera columna de la matriz de rotación que representa la dirección del eje \hat{i}_z y ν representa la integral de la aceleración, $\dot{\nu} = [a_x, a_y, a_z]^T$.

Se definen los errores de estimación z_1, z_2, z_3 como:

$$\begin{aligned} z_1 &= r_3 - \hat{r}_3 + \beta_1(\nu) \\ z_2 &= \bar{\xi} \tanh \left[\frac{\hat{\xi} + \beta_2(\nu)}{\bar{\xi}} \right] - \xi \\ z_3 &= \bar{\zeta}_a \tanh \left[\frac{\hat{\zeta}_a + \beta_3(\nu)}{\bar{\zeta}_a} \right] - \zeta_a \end{aligned} \quad (3.72)$$

donde $\beta_i(\nu)$, $i = 1, 2, 3$ son funciones a ser definidas, $\bar{\xi}, \bar{\zeta}_a$ son cotas superiores aproximadas de ξ y ζ_a respectivamente. Observemos que si los errores de estimación en (3.72) convergen a cero, se tiene entonces,

$$\begin{aligned} \lim_{t \rightarrow \infty} (\hat{r}_3 - \beta_1(\nu)) &= r_3, \\ \lim_{t \rightarrow \infty} \left(\bar{\xi} \tanh \left[\frac{\hat{\xi} + \beta_2(\nu)}{\bar{\xi}} \right] \right) &= \xi, \\ \lim_{t \rightarrow \infty} \left(\bar{\zeta}_a \tanh \left[\frac{\hat{\zeta}_a + \beta_3(\nu)}{\bar{\zeta}_a} \right] \right) &= \zeta_a. \end{aligned} \quad (3.73)$$

Tomando la derivada de los errores de estimación a lo largo del sistema dinámico (3.71) y aplicando el enfoque de Inmersión en Invarianza desarrollado en [25], es posible obtener el siguiente sistema que estima el vector \hat{r}_3 ,

$$\begin{aligned} \dot{\hat{r}}_3 &= \mathbf{S}(\Omega_b)(\hat{r}_3 - \beta_1) + \bar{\xi} \tanh \left[\frac{\hat{\xi} + \beta_2}{\bar{\xi}} \right] + \frac{\partial \beta_1}{\partial \nu} \left(-\hat{r}_3 + \beta_1 + \bar{\zeta}_a \tanh \left[\frac{\hat{\zeta}_a + \beta_3}{\bar{\zeta}_a} \right] \right) \\ \dot{\hat{\xi}} &= -\frac{\partial \beta_2}{\partial \nu} \left(-\hat{r}_3 + \beta_1 + \bar{\zeta}_a \tanh \left[\frac{\hat{\zeta}_a + \beta_3}{\bar{\zeta}_a} \right] \right) \\ \dot{\hat{\zeta}}_a &= -\frac{\partial \beta_3}{\partial \nu} \left(-\hat{r}_3 + \beta_1 + \bar{\zeta}_a \tanh \left[\frac{\hat{\zeta}_a + \beta_3}{\bar{\zeta}_a} \right] \right) \end{aligned} \quad (3.74)$$

donde $\bar{\zeta}_a$ y $\bar{\xi}$ son constantes, $\beta_1, \beta_2, \beta_3$ están en función de las mediciones tomadas por el acelerómetro y por las constantes γ_1, γ_2 y γ_3 definidas como $\beta_i = \gamma_i \nu$, $\gamma_i \in \mathbb{R}^+$, donde ν se obtiene mediante la integración de las mediciones del acelerómetro, es decir,

$$\nu = \int \dot{\eta} dt, \quad \dot{\nu} = [a_x, a_y, a_z]^T. \quad (3.75)$$

Finalmente, las derivadas parciales $\frac{\partial \beta_i}{\partial \nu}$ se proponen con la forma,

$$\frac{\partial \beta_i}{\partial \nu} = \text{diag}\{\gamma_i\}, \quad \gamma_i > 0, \quad i = 1, 2, 3. \quad (3.76)$$

Con las ecuaciones (3.72) y (3.74) se calcula el vector estimado y se obtienen los ángulos en la forma:

$$r_3 = \hat{r}_3 - \beta_1 \quad (3.77)$$

$$\phi = (180/\pi) \arctan(r_3(2), r_3(3)) \quad (3.78)$$

$$\theta = -(180/\pi) \arcsin(r_3(1)). \quad (3.79)$$

La implementación en el procesador digital de señales (DSP - *Digital Signal Processor*) de este esquema de estimación se presenta en el Algoritmo 6 propuesto para calcular los ángulos alabeo y cabeceo. De igual forma se utiliza el método de Euler para la implementación de las rutinas de integración.

Algoritmo 6 Filtro basado en Inmersión e Invarianza.

Entrada: $\mathbf{a}_b = [a_x, a_y, a_z]^T, \Omega_b = [p, q, r]^T, \gamma_1, \gamma_2, \gamma_3, \bar{\mu}_G, \bar{\mu}_A$

Salida: $[\phi, \theta]$ Ángulos de alabeo y cabeceo.

- 1: **for** $k = 1, 2, \dots$ **do**
 - 2: $\dot{\nu} = [a_x, a_y, a_z]^T$
 - 3: $\nu = \nu + \Delta t \dot{\nu}$
 - 4: $\beta_i = [\nu(1), \nu(2), \nu(3)]^T, i = 1, 2, 3$
 - 5: $\frac{\partial \beta_i}{\partial \nu} = \gamma_i I_3, i = 1, 2, 3$
 - 6: $\dot{\hat{r}}_3 = \mathbf{S}(\Omega_b)(\hat{r}_3 - \beta_1) + \bar{\xi} \tanh(\hat{\xi} + \beta_2) + \frac{\partial \beta_1}{\partial \nu}(-\hat{r}_3 + \beta_1 + \bar{\mu}_a \tanh(\hat{\mu}_a + \beta_3))$
 - 7: $\dot{\hat{\xi}} = -\frac{\partial \beta_2}{\partial \nu}(-\hat{r}_3 + \beta_1 + \bar{\mu}_a \tanh(\hat{\mu}_a + \beta_3))$
 - 8: $\dot{\hat{\mu}}_a = -\frac{\partial \beta_3}{\partial \nu}(-\hat{r}_3 + \beta_1 + \bar{\mu}_a \tanh(\hat{\mu}_a + \beta_3))$
 - 9: $\hat{r}_3 = \hat{r}_3 + \Delta t \dot{\hat{r}}_3$
 - 10: $\dot{\hat{\xi}} = \dot{\hat{\xi}} + \Delta t \dot{\hat{\xi}}$
 - 11: $\hat{\mu}_a = \hat{\mu}_a + h \dot{\hat{\mu}}_a$
 - 12: $r_3 = \text{norm}(\hat{r}_3 - \beta_1)$
 - 13: $\phi(k) = -(180/\pi) \arcsin(r_3(1))$
 - 14: $\theta(k) = (180/\pi) \arctan 2(r_3(2), r_3(3))$
 - 15: **end for**
-

Se ha explicado de forma breve las bases teóricas de cada método de estimación de orientación analizado e implementado en este trabajo. En el siguiente Capítulo se abordarán los resultados experimentales obtenidos.

Capítulo 4

Resultados Experimentales

En este Capítulo se describe de forma general el sistema embebido y la plataforma experimental en la que se programaron y evaluaron los métodos de estimación revisados en el Capítulo 3; de igual forma se presentan los resultados de los experimentos realizados para comparar el rendimiento de cada estimador.

4.1. Sensores Inerciales

En este trabajo usamos la tarjeta MinIMU-9v2 de la marca *Pololu* que está compuesta por un Giroscopio L3GD20, un Acelerómetro y un Magnetómetro encapsulados en el mismo *chip* LSM303DLHC fabricados con tecnología MEMS, todos de tres ejes. La tarjeta tiene una interfaz I^2C (*Interconnect Integrated Circuit*) para acceder a los registros de los sensores y poder establecer la transmisión de datos. También se pueden configurar opciones tal como la sensibilidad, la velocidad de transmisión de cada sensor, etc. todo esto a partir de la arquitectura de registros que se describe de forma detallada en [53].

Las nueve lecturas de los sensores, la velocidad angular $\Omega = [p, q, r]^T$, la aceleración $\mathbf{a} = [a_x, a_y, a_z]^T$, y el magnetómetro $\mathbf{m} = [m_x, m_y, m_z]^T$ proporcionan la información necesaria para estimar la orientación con un algoritmo apropiado implementado en un procesador digital de señales, donde el giroscopio puede usarse para calcular la rotación, mientras que el acelerómetro y el magnetómetro pueden ayudar a compensar la deriva del giroscopio con el tiempo.

El DSP está configurado como el dispositivo maestro y los tres sensores actúan como dispositivos esclavos en el bus I^2C , cada uno con una dirección diferente configurado de fábrica, para el giroscopio corresponde a: $1101011b = 0x6B$, el acelerómetro: $0011001b = 0x19$ y el magnetómetro: $0011110b = 0x1E$. Una explicación detallada acerca del protocolo de comunicación I^2C puede ser encontrado en [54].

Características generales de la IMU:

- Dimensiones: $20 \times 13 \times 3 \text{ mm}$
- Peso sin conectores (pines): $0,7 \text{ g}$
- Voltaje de operación: $2,5 \text{ a } 5,5 \text{ v}$
- Rango de sensibilidad:
 - Giroscopio:
 $\pm 250, \pm 500, \pm 2000 \text{ }^{\circ}/\text{s}$
 - Acelerómetro:
 $\pm 2, \pm 4, \pm 8, \pm 19 \text{ g}$
 - Magnetómetro:
 $\pm 1,3, \pm 2,5, \pm 4,7, \pm 8,1 \text{ gauss}$

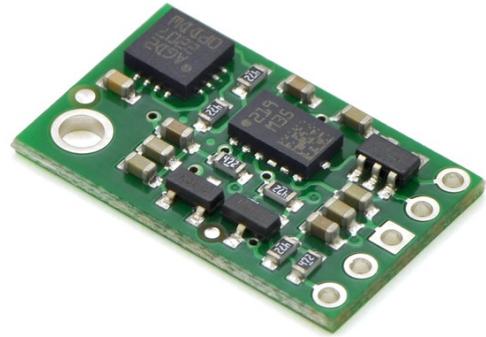


Figura 4.1: Unidad de Medición Inercial - IMU.

4.2. Procesador Digital de Señales

El DSP que se utilizó es el TMS320F28335 de la marca *Texas Instruments* que tiene un alto rendimiento para aplicaciones que demanden un procesamiento numérico extenso y un entorno de desarrollo integrado (IDE - *Integrated Development Environment*) para programar los estimadores en lenguaje C. Cuenta con distintos periféricos y protocolos de interconexión para comunicar distintos dispositivos, en nuestro caso la IMU. Todos los detalles sobre la arquitectura interna del dispositivo y el *software* de desarrollo pueden ser encontrados en [55].

Características generales del DSP:

- Frecuencia: 150 MHz (6.67-ns Cycle Time).
- Unidad de Punto Flotante *IEEE* – 754 de precisión simple (32 bits).
- Voltaje de operación: $1,9 \text{ v}$ Core, $3,3 \text{ v}$ I/O.
- 18 canales *PWM*, 16 *ADC* de 12 bits, 88 *GPIO*, 6 canales de *DMA*.
- Protocolos de comunicación: *I²C*, *SCI*, *SPI*.
- Memoria *Flash* y *RAM* de 512 y 68 KB respectivamente.



Figura 4.2: Procesador Digital de Señales - DSP.

4.3. Plataforma de Experimentación

Uno de los retos importantes de esta tesis fue encontrar una plataforma de experimentación que garantizará una comparación justa e imparcial de los métodos de estimación estudiados. Las principales opciones fueron: adquirir una plataforma robótica expresamente diseñada para validar sistemas de orientación o construir una plataforma similar. La primera opción se puede realizar con una plataforma de posicionamiento de tres grados de libertad de gran precisión modelo 2103HT de la marca *Ideal AeroSmith* [56] que es utilizada en la validación experimental en [57], cuyo costo rebasa el presupuesto de esta tesis. Debido a las limitaciones en las técnicas de manufactura que se han desarrollado en la Sección de Mecatrónica la segunda opción tampoco es viable. Por lo tanto, se decidió por una tercera opción que fue utilizar el brazo robot de la marca ABB de la serie IRB 2400/16, el cual si bien tiene una resolución angular menor que la de la plataforma 2103HT, dicha resolución y repetibilidad es considerablemente mejor que la que se podría lograr con un prototipo construido en la Sección de Mecatrónica.

Características generales del robot ABB IRB 2400/16:

- Capacidad de carga: 16 kg
- Peso: 380 kg
- Alcance máximo: 1,55 m
- Resolución en cada articulación: 0,01°
- Repetibilidad de posición media (ISO 9283): 0,03 mm
- Exactitud de posición (ISO 9283): 0,03 mm
- Repetibilidad de trayectoria media (ISO 9283): 0,15 mm
- Exactitud de trayectoria (ISO 9283): 0,41 mm

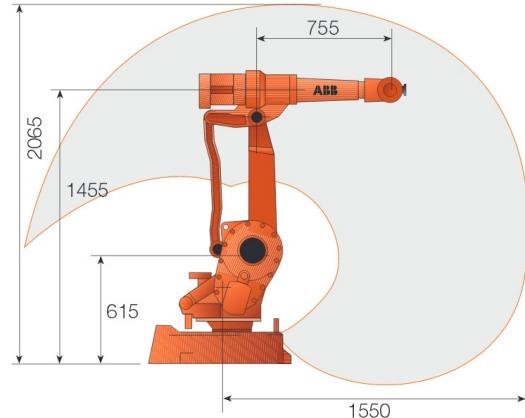


Figura 4.3: Robot ABB IRB 2400/16

El robot IRB 2400/16 tiene un *hardware* de control IRC5 que permite programarlo en un lenguaje de alto nivel llamado RAPID, ver [58]; todos los detalles técnicos del robot pueden encontrarse en [59]. La ventaja de este robot como nuestra plataforma de experimentación es que nos permite generar rotaciones con una exactitud y una repetibilidad aceptables, debido a que están calibrados bajo la norma internacional de robots manipuladores industriales *ISO 9283* (*ISO - International Organization for Standardization*) que define los criterios y métodos para pruebas de desempeño y rendimiento.

La repetibilidad de las rotaciones programadas en el robot fue una condición imprescindible para comparar los métodos de estimación de manera justa e imparcial, sin embargo, el inconveniente con estos robots es que no se puede tener acceso directo a la mediciones de los ángulos de rotación, las velocidades y aceleraciones angulares de cada articulación del robot; en su defecto lo que se hace es programar al robot para que recorra un perfil de rotaciones deseado; el controlador del robot se encarga de interpolar la trayectoria de tal forma que inicie en una ángulo inicial y termine en un ángulo final a una velocidad constante, con las mismas características de posición, velocidad y aceleración en cada experimento.

4.4. Descripción del Experimento

Con los sensores conectados a través del protocolo I^2C al DSP y los estimadores programados debidamente, los experimentos para evaluar y comparar cada método de estimación fueron realizados en el robot IRB 2400/16 con el DSP y la IMU montados en el órgano terminal como se muestra en la Figura 4.4.

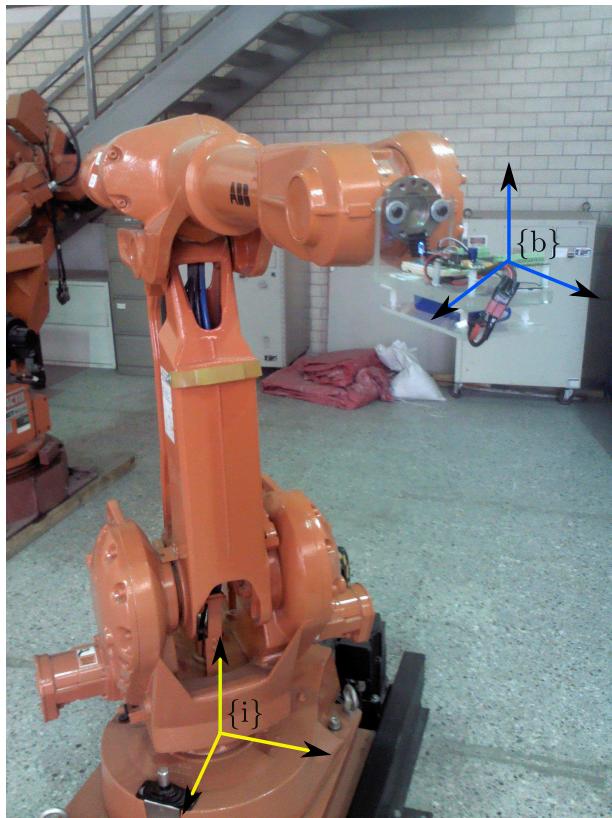


Figura 4.4: Plataforma de experimentación con el DSP y la IMU montados en el órgano final. Video disponible en: <http://www.youtube.com/watch?v=LTR-hAu0JqY>

Para iniciar el experimento es importante calibrar la IMU con la orientación del robot, en nuestro caso posicionamos el órgano terminal para que el plano XY de la IMU se encuentre de manera perpendicular con el vector de aceleración gravitacional, en otras palabras, fijamos como nuestro marco inercial $\{\mathbf{i}\}$ la orientación del robot en la cual el acelerómetro midiera $\mathbf{a}_b = [0, 0, -9,81]^T \text{ m/s}^2$, si embargo, esto no siempre debe ser así, se puede orientar la IMU en cualquier dirección y a partir de ese marco fijo empezar a estimar la orientación del marco móvil $\{\mathbf{b}\}$. Es importante mencionar que en el caso de los métodos de estimación TRIAD y QUEST los vectores $\mathbf{a}_i = [0; 0; -9,81]^T$, $\mathbf{m}_i = [0,377; -0,499; -1,124]^T$ son tomados justo en esta posición inicial.

Cada algoritmo de estimación se programó en el DSP con tres rutinas o tareas principales. La primera rutina corresponde a la adquisición de datos, donde se accede a los registros de salida de la IMU a una frecuencia máxima de 565 Hz para leer los vectores $\boldsymbol{\Omega}_b$, \mathbf{a}_b y \mathbf{m}_b . Enseguida una segunda rutina realiza los cálculos correspondientes para estimar los ángulos de inclinación [Alabeo, Cabeceo, Guiñada] = $[\phi, \theta, \psi]$; cuando los ángulos se han calculado la tercera rutina los envía a través de un canal inalámbrico mediante un dispositivo XbeePro bajo el protocolo RS232 a una velocidad de transmisión de 115200 bit/s respetando el estándar para aritmética en coma flotante IEEE 754. El dispositivo XbeePro receptor está conectado a una computadora para que se puedan guardar y graficar los datos mediante Matlab. Esto se ejemplifica mejor con la Figura 4.5.

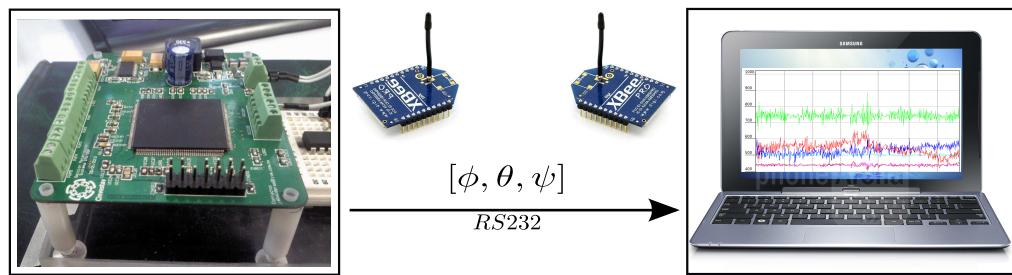


Figura 4.5: Esquema de comunicación sistema embebido - Computadora.

Como se describe, en la lógica de programación tenemos tres rutinas o tareas principales, la rutina de adquisición de datos, la rutina de estimación y la rutina de envío de datos a la computadora, las cuales son controladas a través de interrupciones generadas por contadores del DSP, lográndose un control estricto del inicio de ejecución de cada tarea. Es importante mencionar que el tiempo de ejecución de la rutina 2 varía entre cada método de estimación como se observa en la Tabla 4.1, para comparar de manera equitativa y tener la misma frecuencia de estimación, todos los algoritmos incluyendo las rutinas de adquisición y transmisión de datos se realizaron a una frecuencia de 200 Hz.

Los tiempos de procesamiento de cada algoritmo se midieron usando un osciloscopio de dos canales y dos salidas digitales del DSP; en el primer canal se mide el tiempo de adquisición de las mediciones y en el segundo canal se mide el tiempo de procesamiento de cada algoritmo como se observa en la Figura 4.6. El tiempo de adquisición que corresponde a la señal S_1 es igual en todos los métodos de estimación así como el tiempo de transmisión da datos, la señal S_2 representa el tiempo de cómputo de cada método y difiere entre cada uno de ellos. La medición de estos tiempos también se corroboraron con los contadores internos del DSP.

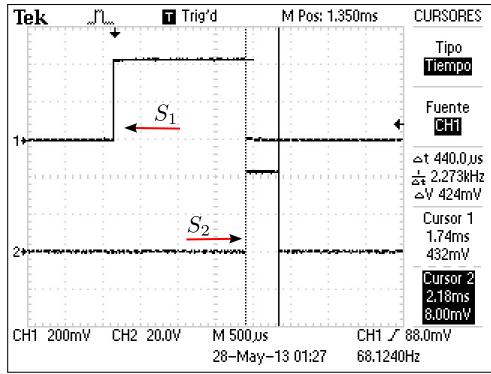


Figura 4.6: Medición de tiempos en Método TRIAD

Los tiempos de cómputo obtenidos en cada algoritmo se presentan en la Tabla 4.1, donde se observa que el menor tiempo corresponde al método (3) y el método que requiere más operaciones de cálculo corresponde a (4), debido a que se necesita calcular la inversa de una matriz de 4×4 y se requieren más multiplicaciones de matrices respecto a los demás métodos.

Tabla 4.1: Tiempos de cómputo de cada método de estimación.

No.	Algoritmo de Estimación	Tiempo (μs)	Frec. (KHz)
1	TRIAD	440	2.273
2	QUEST	360	2.778
3	Fusion Sensorial	140	7.143
4	Filtro de Kalman	2300	0.434
5	F. Complementario Directo	560	1.786
6	F. Complementario Pasivo	560	1.786
7	F. de Inmersión e Invarianza	200	5.000

Las ganancias y las condiciones iniciales con los que se realizaron los experimentos se fijaron para el método QUEST en $\rho_1 = 2$, $\rho_2 = 2$; para el filtro de Kalman $\hat{\mathbf{q}} = [0, 0, 0, 0]^T$, $P_{0,0} = I_4$, $\sigma_\zeta = 0.1$ y $\sigma_\xi = 0.1$; para el método de Fusion Sensorial

$\sigma_\zeta = 5$ y $\sigma_\xi = 3$. Para los filtros complementario en sus dos versiones, pasivo y directo se propusieron $k_i = 10$ y $k_p = 25$. Para el filtro basado en el esquema de Inmersión e Invarianza se consideró $\bar{\mu}_a = 10$, $\bar{\xi} = -10$, $\gamma_1 = 0.5$, $\gamma_2 = 0.7$ y $\gamma_3 = 0.9$. Es importante mencionar que dichas ganancias se lograron de forma heurística con una serie de pruebas de ensayo y error, excepto para las varianzas del filtro de Kalman que se aproximan tomando una secuencia de datos de los sensores iniciales con la IMU en forma estática, para poder calcular la media $\mu_x = E(X)$ que corresponde al valor esperado de la variable aleatoria X y las varianzas $\sigma^2 = \text{Var}(X) = E[(X - \mu)^2]$ que son una medida de dispersión respecto a la media como se muestra en el Apéndice D, sin embargo, de forma real no existe ninguna señal de ruido que tenga exactamente una distribución Gaussiana. Por lo tanto, las mediciones de media y varianza únicamente sirvieron como referencia para la sintonización del filtro de Kalman. La sintonización fina también fue heurística.

El robot IRB 2400/16 se programó para ejecutar dos series de rotaciones de $[-80^\circ, 80^\circ]$ grados en alabeo, $[-60^\circ, 60^\circ]$ grados en cabeceo y $[-40^\circ, 40^\circ]$ en guiñada que están especificados con las variables P_i del tipo *JointTarget* como se muestra en el siguiente código.

```
! _____
! Progra para las rotaciones del robot
! _____
MODULE Test01
    PROC main()
        FOR i:1:2
            MOVEJ P0,v20,z10,tool1
            MOVEJ P1,v20,z10,tool1
            MOVEJ P2,v20,z10,tool1
            MOVEJ P0,v20,z10,tool1
            MOVEJ P3,v20,z10,tool1
            MOVEJ P4,v20,z10,tool1
            MOVEJ P0,v20,z10,tool1
            MOVEJ P5,v20,z10,tool1
            MOVEJ P6,v20,z10,tool1
            WAITTIME 3
        ENDFOR
    ENDPROC
END MODULE
!
```

El perfil de estas rotaciones se pueden observar por separado en las Figuras 4.7 - 4.9, pero se ejecutan simultáneamente en el robot.

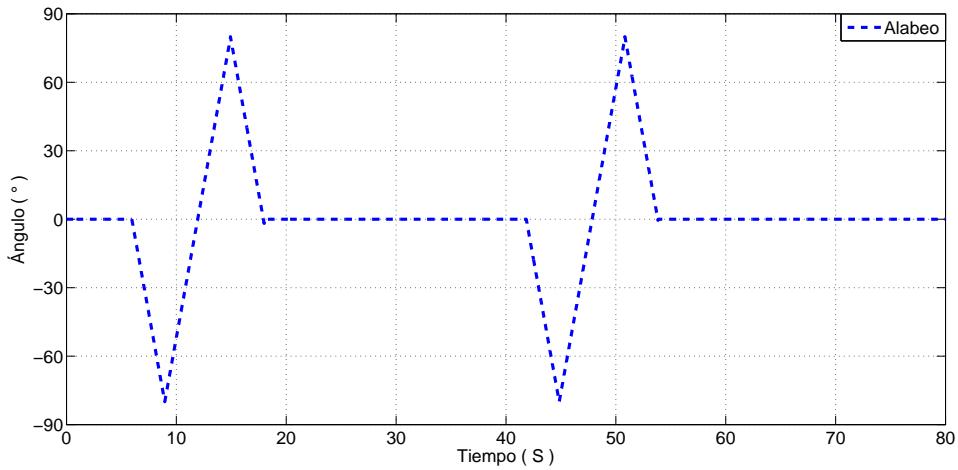


Figura 4.7: Perfil de rotación programado en el robot en Alabeo

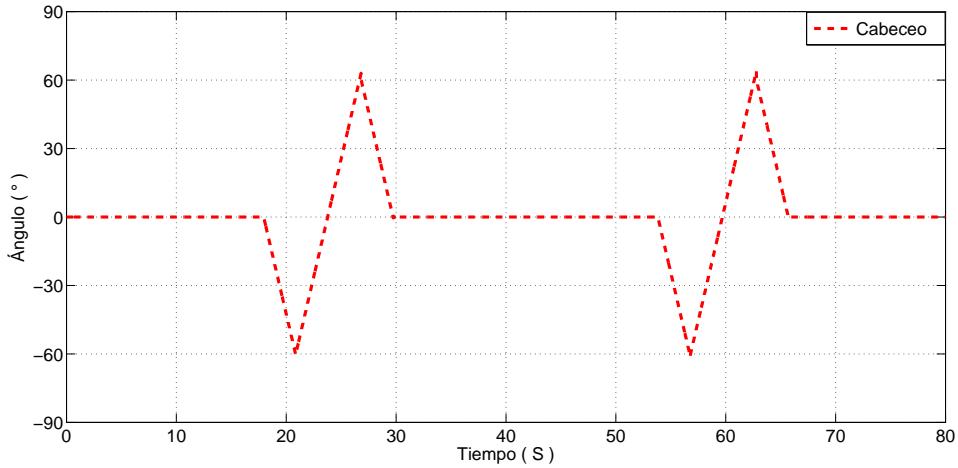


Figura 4.8: Perfil de rotación programado en el robot en Cabeceo

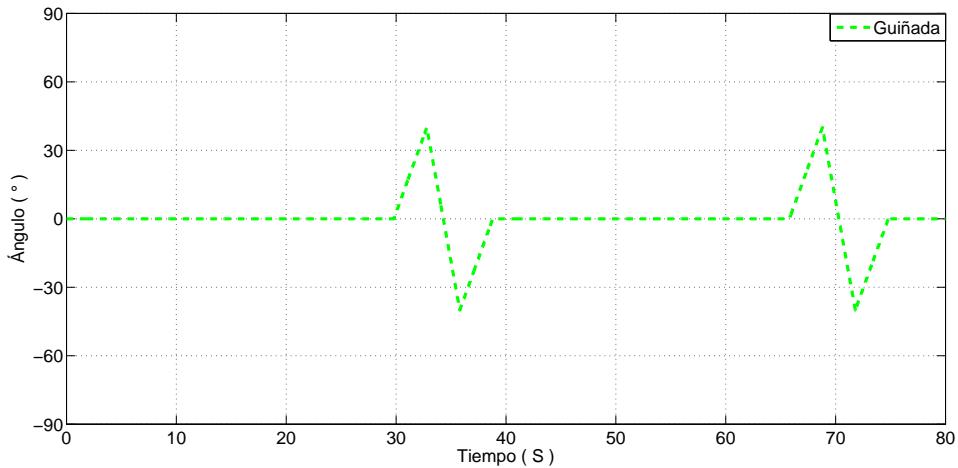


Figura 4.9: Perfil de rotación programado en el robot en Guiñada

Con el robot siguiendo los perfiles de rotación programados, los resultados obtenidos usando el método TRIAD y QUEST se presentan en la Figura 4.10 y 4.11 respectivamente, observándose un rendimiento menor debido al ruido presente en las mediciones, en contraste con los otros métodos donde se obtienen mejores resultados. Esto se debe a que estos métodos de estimación son algebraicos y no consideran ningún término que modele el ruido de los vectores de observación.

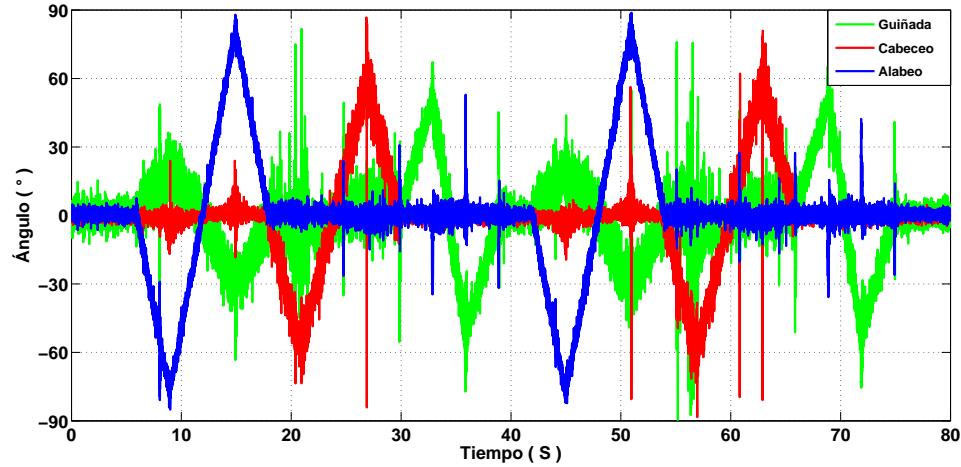


Figura 4.10: Método TRIAD.

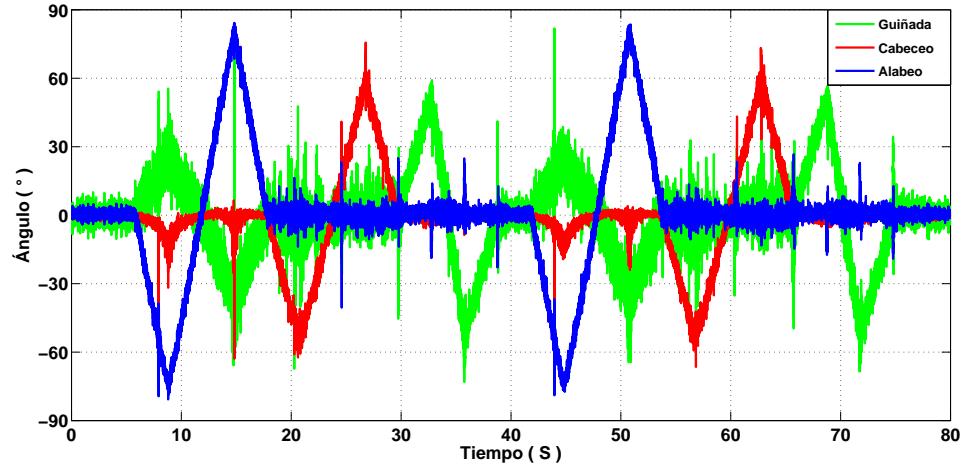


Figura 4.11: Método QUEST.

Observemos que los método de Fusión Sensorial, Figura 4.12 y el método de Inmersión en Invarianza, Figura 4.16 solo estiman los ángulos $[\phi, \theta]$ y los métodos restantes estiman los tres ángulos $[\phi, \theta, \psi]$. En las gráficas siguientes notemos que el ruido de las mediciones se reducen considerablemente, logrando mejores estimaciones respecto a las rotaciones programadas en el robot.

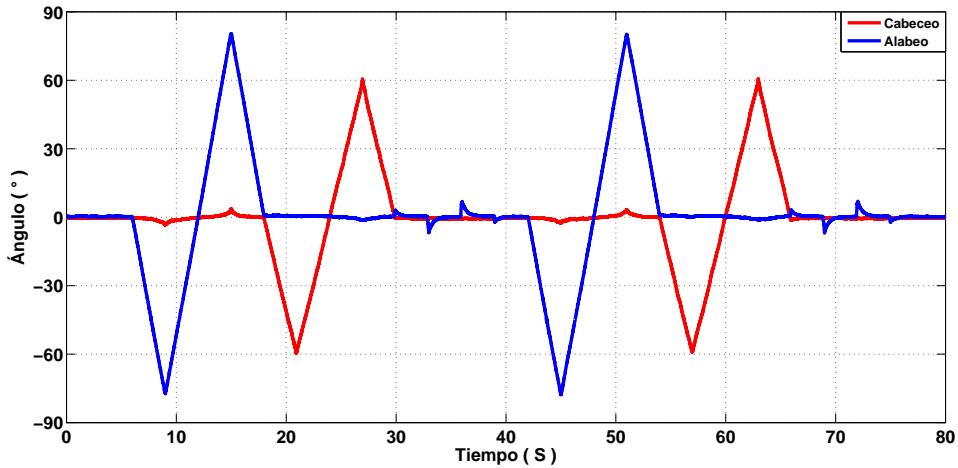


Figura 4.12: Método de Fusión Sensorial.

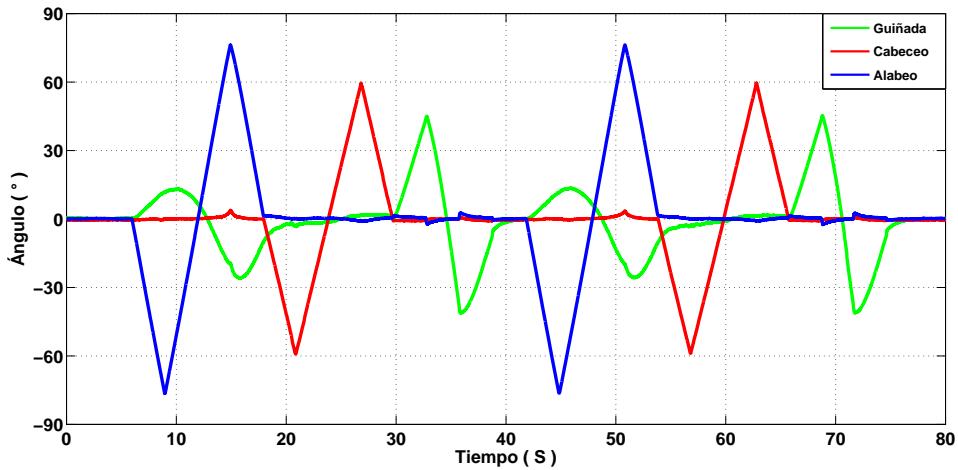


Figura 4.13: Filtro de Kalman.

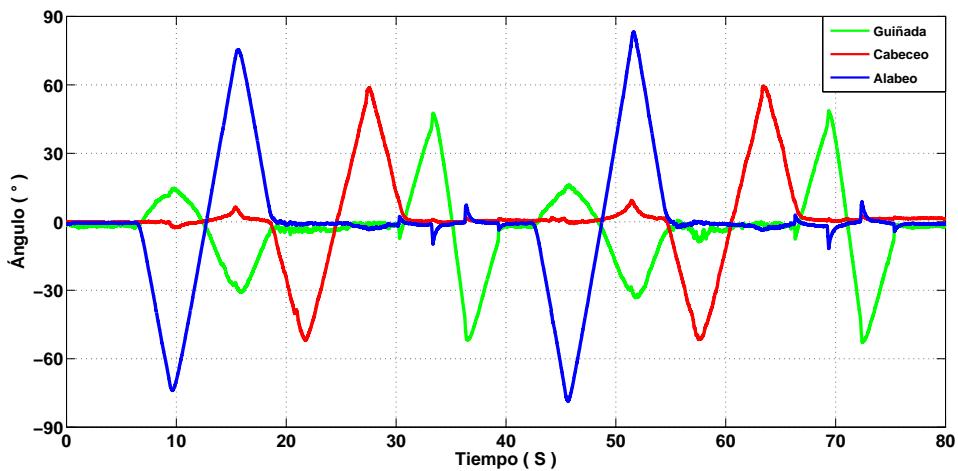


Figura 4.14: Filtro Complementario Directo.

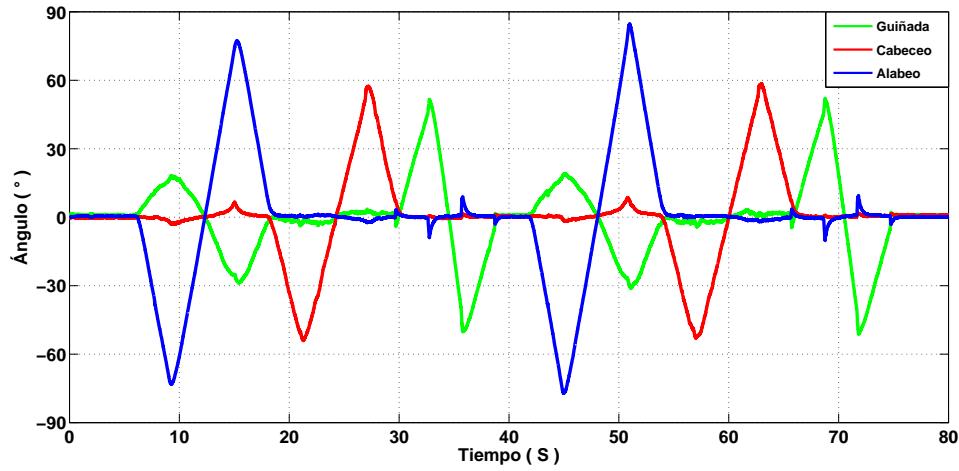


Figura 4.15: Filtro Complementario Pasivo.

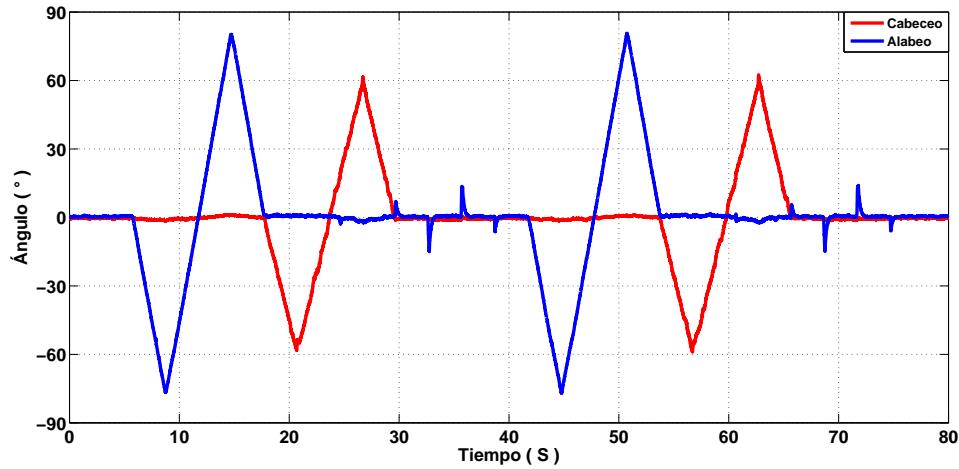


Figura 4.16: Filtro de Inmersión e Invarianza.

En los métodos que estiman los tres ángulos de orientación se observa claramente que se presenta un error de estimación considerable en el ángulo de guinada [ψ]. Esto se atribuye al campo magnético generado por los motores de las articulaciones del robot. Este es un problema común en aplicaciones donde existen perturbaciones magnéticas locales que afectan directamente al magnetómetro, este problema también se reporta en los experimentos de [57].

Otro problema que también afecta a las estimaciones, además del ruido en los sensores, es la aceleración lineal que cobra relevancia cuando se presenta un cambio en el sentido de rotación, generando pequeñas desviaciones como se observa en las Figuras 4.12 y 4.16 donde no se hace uso del magnetómetro y las desviaciones se atribuyen únicamente al acelerómetro. Estos picos del ángulo de Alabeo que se notan bien en la Figura 4.16

se deben precisamente al cambio del sentido de giro que realiza el robot en el perfil programado para el ángulo de guiñada $[\psi]$.

Para evaluar objetivamente los métodos de estimación implementados se consideró un índice de desempeño basado en la norma \mathcal{L}_2 de los errores de estimación \tilde{e}_χ definido como $\tilde{e}_\chi = \chi_r - \chi_e$ como un parámetro para medir la exactitud, con χ_r el ángulo del perfil programado en el robot, χ_e el ángulo estimado y $\chi = \phi, \theta, \psi$. El índice mencionado tiene la siguiente forma,

$$\mathcal{L}_2(\tilde{e}_\chi) = \left(\frac{1}{t - t_0} \sum_{n=t_0}^t \| \tilde{e}_\chi \|^2 \right)^{1/2}, \quad \tilde{e}_\chi = \chi_r - \chi_e, \quad t - t_0 \neq 0 \quad (4.1)$$

donde $t, t_0 \in \mathbb{R}^+$ son los tiempos inicial y final respectivamente. Es claro que un índice pequeño de $\mathcal{L}_2(\tilde{e}_\chi)$ representa un mejor desempeño.

Utilizando (4.1) y siguiendo la numeración de la Tabla 4.1 los índices de desempeño de cada método están dados en la Figura 4.17 donde es posible observar que los métodos TRIAD y QUEST tienen menor desempeño por el ruido en las estimaciones que se puede observar claramente en las Figuras 4.10 y 4.11, sin embargo se tienen mejores resultados con el filtro de fusión sensorial, seguido por el filtro de Inmersión e Invarianza que atenúan de mejor forma los errores de los vectores de observación.

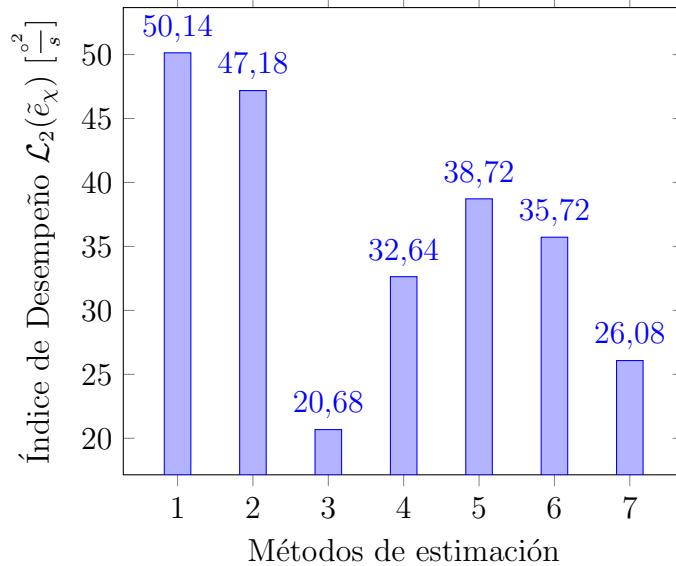


Figura 4.17: Índice de desempeño.

A partir del compromiso entre un bajo tiempo de cómputo y un adecuado índice de desempeño en la Figura 4.18 se muestra la relación normalizada existente entre ellos, donde podemos ver que el método (3) y (7) representan las mejores alternativas.

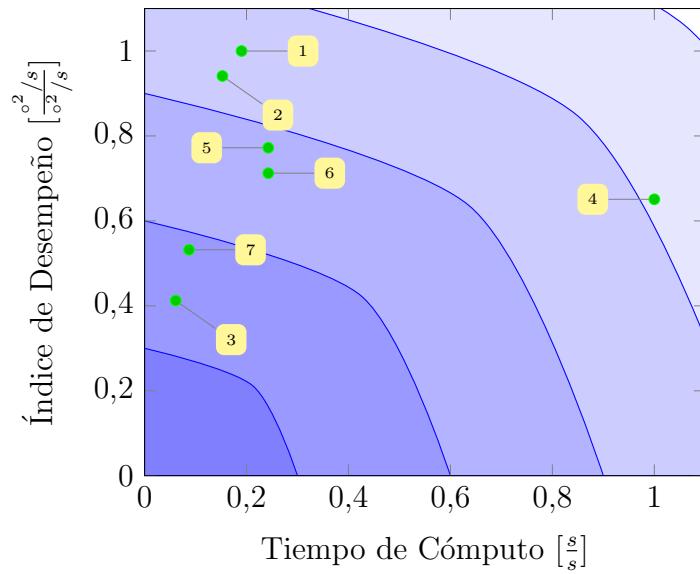


Figura 4.18: Desempeño vs Tiempo de cómputo.

Se ha presentado un estudio experimental comparativo de diversos métodos utilizados en la literatura para estimar la orientación de un cuerpo rígido, considerando como parámetros a evaluar sus tiempos de cálculo obtenidos mediante su implementación en un sistema embebido basado en un procesador digital de señales, así como la evaluación de un índice de desempeño del error de estimación basado en la norma \mathcal{L}_2 . Es notable como la mejor estimación dada por el filtro de Kalman presenta el problema de largos tiempos de cómputo, teniendo entonces un compromiso entre velocidad de cálculo y exactitud de estimación, generándose alternativas dadas por los Filtros Complementarios para estimar la orientación con respecto a los tres ejes o el método de Fusión Sensorial si únicamente se desea obtener la orientación con respecto al vector de gravedad.

Capítulo 5

Conclusiones y Perspectivas

5.1. Conclusiones

En este trabajo se ha presentado un estudio experimental comparativo de una muestra representativa de los métodos más utilizados reportados en la literatura para estimar la orientación de un cuerpo rígido en el espacio, evaluando sus tiempos de computo, así como la evaluación de un índice de desempeño del error de estimación basado en la norma \mathcal{L}_2 . En respuesta a las preguntas hechas en la sección de planteamiento del problema, podemos concluir que los métodos que presentan mejores estimaciones comparados con el perfil de rotaciones programados en el robot son, el método de Fusión Sensorial y el método del Filtro de Kalman, sin embargo los métodos mas rápidos en cuanto al tiempo de ejecución son los métodos de Fusión Sensorial y el método de Inmersión e Invarianza.

En la bibliografía consultada referente a la estimación de la orientación se pueden encontrar diferentes esquemas con algunas diferencias en su estructura; por ejemplo, utilizan diferentes parametrizaciones de la orientación o en la forma de abordar los efectos del ruido de los sensores, sin embargo todos preservan de manera general una idea principal, que básicamente consiste en comparar los ángulos de orientación calculados usando el giroscopio a través del modelo cinemático rotacional, con los vectores de referencia que proporcionan el acelerómetro y el magnetómetro y de esta forma eliminar la deriva o los errores de estimación.

En general, constatamos que usando sensores iniciales tipo MEMS se puede estimar la orientación con una precisión aceptable, aunque con este tipo de sensores persisten dos problemas, por un lado el ruido en las mediciones en los tres sensores que se debe principalmente a la miniaturización de los mismos y por otro lado estamos restringidos a que los perfiles de aceleración lineal sean bajos para que el vector de aceleración gravitacional que mide el acelerómetro no sea afectado considerablemente. Estos son

problemas propios de los sensores, sin embargo también ofrecen otras ventajas como un peso mínimo y un tamaño reducido que los hacen factibles en diferentes aplicaciones.

5.2. Trabajo Futuro

El área referente a la estimación de la orientación utilizando sensores inerciales lleva poco más de medio siglo de investigación, pero aún sigue siendo un tema abierto y de gran interés por la gama de aplicaciones prácticas que tiene, sobre todo por la ventajas que tienen los dispositivos basados en tecnologías MEMS para ser integrados en diferentes sistemas.

Si bien es cierto que ya se tienen diferentes marcas de IMU's comerciales que integran en una sola tarjeta tanto los sensores inerciales como un procesador programado con un algoritmo de estimación, es importante conocer a fondo cuales son los mecanismos y las ecuaciones que logran dicha estimación y así poder replantear o proponer nuevos esquemas de estimación.

Últimamente han surgido otras tecnologías, como los métodos de identificación óptica de posición y orientación usando un grupo de cámaras infrarrojas, que también ofrecen una buena medición de la orientación de un cuerpo rígido, pero el sistema de estimación no va montada sobre el cuerpo en movimiento, es decir, se tiene un sistema de plataforma estable. Otro ejemplo son los dispositivos *Kinects* que a través de algoritmos adecuados pueden también estimar la orientación. Como puede verse existen otras alternativas, pero con los sensores inerciales se tienen resultados aceptables, aún así todavía es necesario seguir mejorando desde los esquemas de estimación hasta lo referente al desarrollo de nuevos sensores que se puedan integrar para resolver el problema de las aceleraciones lineales en los acelerómetros, y reducir aun más el bias en los giroscopios.

En este trabajo consideramos que los objetivos a futuro son:

- Construir un *cluster* de sensores del mismo tipo para promediar las mediciones y así reducir el ruido.
- Implementar estos métodos de estimación en un Arreglos de Compuertas Programables en Campo (FPGA - *Field Programmable Gate Array*) para trabajar con procesos paralelos, reduciendo el tiempo de cómputo.
- Proponer esquemas de estimación descentralizados, es decir, utilizar dos modelos cinemáticos que utilicen una parametrización diferente de la orientación pero que usen las mismas mediciones, teniendo de esta forma dos estimaciones que después pueden fusionarse para minimizar el error.
- Utilizar un inclinómetro como un sensor auxiliar para el acelerómetro, de tal

forma que se pueda medir el vector de aceleración gravitacional cuando no sea posible cumplir la restricción de mantener perfiles de aceleración lineal bajos.

También estos métodos de estimación se podrían aplicar en nuevas investigaciones, por ejemplo en el área de robótica puede ser factible estimar la orientación del órgano terminal de los robots antropomórficos sin depender totalmente de los ángulos de las articulaciones, de igual forma en robots de cadena cerrada, específicamente en el robot paralelo 3-RPS es realizable unir el modelo de la orientación de la plataforma superior usando sensores iniciales con el modelado de la cinemática inversa del robot y tratar de explotar las ventajas que esto pueda ofrecer. En otras áreas, también puede ser posible usar estos sensores en automóviles para controlar la estabilidad a través de suspensiones activas. En general, se tienen un sin fin de aplicaciones donde estimar la orientación es vital para el control y donde los sensores iniciales basados en tecnología MEMS junto con los métodos de estimación pueden ser aplicados.

Apéndice A

Código en Matlab

En este apartado se presentan los códigos de simulación de cada método de estimación, donde previamente se estableció la comunicación via RS232 entre la IMU y Matlab para adquirir los nueve datos iniciales del acelerómetro, del magnetómetro y del giroscopio. Todos los códigos están programados siguiendo una estandarización en las siguientes variables.

Tabla A.1: Variables estándar en los códigos de simulación

Variable	Descripción
Ax, Ay, Az	Aceleración en el eje X, Y y Z
Mx, My, Mz	Campo magnético en el eje X, Y y Z
Gx, Gy, Gz	Velocidad angular en el eje X, Y y Z
Ace_i, Ace_b	Vector de aceleración en el marco $\{i\}$ y $\{b\}$
Mag_i, Mag_b	Vector de campo magnético en el marco $\{i\}$ y $\{b\}$
q_0, q_1, q_2, q_3	Cuaternion estimado
R_i	Matriz de rotación
r, p, y	Roll, Pitch, Yaw

Algoritmo A.1: Código del método TRIAD

```
1 -----
2 Algoritmo: T R I A D
3 Author: Edwin Rosario Gabriel
4 Date: 17/04/2013 , 08:37 pm
5 -----
6 function [q0, q1, q2, q3] = TRIAD(Ax,Ay,Az,Mx,My,Mz)
7 % Datos del acelerometro
8 Ace_b = [Ax ; Ay ; Az];
9 Ace_b = Ace_b/norm(Ace_b);
10 Ace_i = [0 ; 0 ; 9.8];
```

```

11 Ace_i = Ace_i/norm(Ace_i);
12
13 % Datos del magnetometro
14 Mag_b = [Mx ; My ; Mz];
15 Mag_b = Mag_b/norm(Mag_b);
16 Mag_i = [0.2584; 0.0510; -0.8610];
17 Mag_i = Mag_i/norm(Mag_i);
18
19 % Operaciones del metodo
20 t1b = Ace_b;
21 t1i = Ace_i;
22 t2b = cross(Ace_b, Mag_b)/norm(cross(Ace_b, Mag_b));
23 t2i = cross(Ace_i, Mag_i)/norm(cross(Ace_i, Mag_i));
24 t3b = cross(t1b, t2b);
25 t3i = cross(t1i, t2i);
26
27 % Matriz de rotacion tb = R*t
28 R=[t1b t2b t3b]*[t1i t2i t3i]';
29
30 % Convierte de DCM a Euler
31 r= (180/pi)*atan(R(2,3)/R(3,3));
32 p= (180/pi)*asin(-R(1,3));
33 y= (180/pi)*atan(R(1,2)/R(1,1));
34
35 % Convierte de DCM a Cuaterniones
36 q0 = sqrt((trace(R)+1))/2;
37 qx = (1/(4*q0))*(R'-R);
38 q1 = qx(3,2);
39 q2 = qx(1,3);
40 q3 = qx(2,1);
41 end

```

Algoritmo A.2: Código del método QUEST

```

1 -----
2 Algoritmo: Q U E S T
3 Author: Edwin Rosario Gabriel
4 Date: 17/04/2013 , 08:37 pm
5 -----
6 function [q0,q1,q2,q3] = QUEST(Ax,Ay,Az,Mx,My,Mz)
7 % Datos del acelerometro
8 Ace_b = [Ax ; Ay ; Az];
9 Ace_b = Ace_b/norm(Ace_b);
10 Ace_i = [0 ; 0 ; 9.8];
11 Ace_i = Ace_i/norm(Ace_i);
12
13 % Datos del magnetometro
14 Mag_b = [Mx ; My ; Mz];
15 Mag_b = Mag_b/norm(Mag_b);
16 Mag_i = [0.2584; 0.0510; -0.8610];
17 Mag_i = Mag_i/norm(Mag_i);

```

```

18
19      % Operaciones del metodo
20      rho=2;
21      lamda_op=rho+rho;
22      B = rho*(Ace_b*Ace_i') + rho*(Mag_b*Mag_i')
23      S = B + B';
24      Z = [B(2,3)-B(3,2) , B(3,1)-B(1,3) , B(1,2)-B(2,1)]';
25      sigma = trace(B);
26      p= inv((lamda_op + sigma)*eye(3) - S)*Z;
27      quat = (1/sqrt(1 + p'*p)) * [p;1];
28
29      % Quaternion calculado
30      q0 = quat(4,1);
31      q1 = quat(1,1);
32      q2 = quat(2,1);
33      q3 = quat(3,1);
34 end

```

Algoritmo A.3: Código del método Q-METHOD

```

1 -----
2 Algoritmo: Q - M E T H O D
3 Author: Edwin Rosario Gabriel
4 Date: 17/04/2013 , 08:17 pm.
5 -----
6 function [q0,q1,q2,q3] = QMETHOD(Ax,Ay,Az,Mx,My,Mz)
7     % Datos del acelerometro
8     Ace_b = [Ax ; Ay ; Az];
9     Ace_b = Ace_b/norm(Ace_b);
10    Ace_i = [0.5102 ; 1.1978 ; 9.7186];
11    Ace_i = Ace_i/norm(Ace_i);
12
13    % Datos del magnetometro
14    Mag_b = [Mx ; My ; Mz];
15    Mag_b = Mag_b/norm(Mag_b);
16    Mag_i = [0.4482 ; -0.2163 ; -0.8996];
17    Mag_i = Mag_i/norm(Mag_i);
18
19    % Operaciones del metodo
20    rho=2;
21    B = rho*(Ace_b*Ace_i') + rho*(Mag_b*Mag_i');
22    S = B + B';
23    Z = [B(2,3)-B(3,2) , B(3,1)-B(1,3) , B(1,2)-B(2,1)]';
24    sigma = trace(B);
25    K=[S-sigma*eye(3) , Z; Z' , sigma ];
26    [A,B] = eig(K);
27
28    % Quaternion calculado
29    q1= A(1,4);
30    q2= A(2,4);
31    q3= A(3,4);

```

```

32 q0= A(4,4);
33
34 if q0<0
35 q1= (-1)*A(1,4);
36 q2= (-1)*A(2,4);
37 q3= (-1)*A(3,4);
38 q0= (-1)*A(4,4);
39 end
40 end

```

Algoritmo A.4: Código del filtro de Kalman

```

1 -----
2 Algoritmo: F - K A L M A N
3 Author: Edwin Rosario Gabriel
4 Date: 17/04/2013 , 08:17 pm.
5 -----
6 function [q0,q1,q2,q3,Pr] = FKALMAN(q0e,q1e,q2e,q3e,wx,wy,wz,q0T,
7 q1T,q2T,q3T,h,P)
8 % Condiciones iniciales
9 varQ = 0.1;
10 Q=varQ*eye(4);
11 varR = 0.1;
12 R=varR*eye(4);
13 C=eye(4);
14
15 % Se evalua el sistema Linealizado con los valores estimados
16 A=[ 1 -(h/2)*wx -(h/2)*wy -(h/2)*wz ;
17 (h/2)*wx 1 (h/2)*wz -(h/2)*wy ;
18 (h/2)*wy -(h/2)*wz 1 (h/2)*wx ;
19 (h/2)*wz (h/2)*wy -(h/2)*wx 1 ];
20
21 % Consideraremos error en el sistema
22 H=h*eye(4)
23
24 % Se calcula P de prediccion
25 Pp = A*P*A' + H*Q*H';
26
27 %Se evalua el modelo del sistema no Lineal discreto
28 q0ep = (h/2)*(-wx*q1e - wy*q2e - wz*q3e) + q0e;
29 q1ep = (h/2)*(wx*q0e + wz*q2e - wy*q3e) + q1e;
30 q2ep = (h/2)*(wy*q0e - wz*q1e + wx*q3e) + q2e;
31 q3ep = (h/2)*(wz*q0e + wy*q1e - wx*q2e) + q3e;
32
33 % Calculamos la Ganancia
34 G=Pp*C'*inv(C*Pp*C' + R);
35
36 % Se actualiza P
37 P = (eye(4) - G*C)*Pp;
38
39 % Se actualizan los estados estimados

```

```

39 xe= [q0ep;q1ep;q2ep;q3ep] + G*([q0T;q1T;q2T;q3T] - C*[q0ep;
40 q1ep;q2ep;q3ep]);
41 norma = sqrt(xe(4,1)^2 + xe(5,1)^2 + xe(6,1)^2 + xe(7,1)^2);
42 q0 = xe(1,1)/norma;
43 q1 = xe(2,1)/norma;
44 q2 = xe(3,1)/norma;
45 q3 = xe(4,1)/norma;
46 Pr = P;
47 end

```

Algoritmo A.5: Código del método de Fusión sensorial

```

1 -----
2 Algoritmo: F - F U S I O N X
3 Author: Edwin Rosario Gabriel
4 Date: 17/04/2013 , 10:00 pm
5 -----
6 function [Angxe,Biasxe,Pr] = FKALMANX(Angx,Biasx,Ax,Ay,Az,Wx,P,h)
7 % Condiciones iniciales
8 Qang = 5 ;
9 Qbias = 5 ;
10 R = 3 ;
11 Wx=(180/pi)*Wx;

12 % Se calculan los estados del sistema
13 Angx = Angx + h*(Wx - Biasx);
14 Biasx = Biasx;

15 % Se calcula P de predicción
16 P(1,1)= P(1,1) + h*(h*P(2,2) - P(1,2) - P(2,1) + Qang);
17 P(1,2)= P(1,2) - h*P(2,2);
18 P(2,1)= P(2,1) - h*P(2,2);
19 P(2,2)= P(2,2) + Qbias*h;

20 % Calculamos la Ganancia
21 S = P(1,1) + R;
22 K0 = P(1,1)/S;
23 K1 = P(2,1)/S;

24 % Mediciones de Correcion
25 Z = atan(Ay/sqrt(Az^2 + Ax^2));
26 AngAce = (180/pi)*Z;

27 % Actualizamos P
28 P(1,1)= P(1,1) - K0*P(1,1);
29 P(1,2)= P(1,2) - K0*P(1,2);
30 P(2,1)= P(2,1) - K1*P(1,1);
31 P(2,2)= P(2,2) - K1*P(1,2);
32 Pr=P;
33
34
35
36
37
38

```

```
39 % Calculamos los estados estimados
40 Y = AngAce - Angx;
41 Angxe = Angx + KO*Y;
42 Biasxe = Biasx + K1*Y;
43 end
```

Algoritmo A.6: Código del filtro de Inmersión e Invarianza

```
1 -----
2 Algoritmo: F - INMERSION E INVARIANZA
3 Author: Edwin Rosario Gabriel
4 Date: 23/04/2013 , 05:01 am
5 -----
6 function [p,r] = FIINVARIANZA(Ax,Ay,Az,Wx,Wy,Wz)
7 % Condiciones iniciales
8 gamma1 = 0.5; gamma2 = 0.7; gamma3 = 0.9;
9 bmuG = -30; bmuA = 10 ; g = 9.81;
10 eta = zeros(3,1);
11 Hr3 = zeros(3,1);
12 HmuG = zeros(3,1);
13 HmuA = zeros(3,1);

14
15 % Se agrupan las Aceleraciones
16 deta=[Ax;Ay;Az];
17
18 % Integraremos el vector de aceleracion
19 eta = eta + h*deta;

20
21 % Se actualizan parametros de Beta
22 B1 = gamma1*[eta(1); eta(2); eta(3)];
23 dB1eta = gamma1*[1 0 0; 0 1 0; 0 0 1];
24 B2 = gamma2*[eta(1); eta(2); eta(3)];
25 dB2eta = gamma2*[1 0 0; 0 1 0; 0 0 1];
26 B3 = gamma3*[eta(1); eta(2); eta(3)];
27 dB3eta = gamma3*[1 0 0; 0 1 0; 0 0 1];

28
29 % Se actualizan la matriz antisimetrica
30 SOmega = [ 0 , -Wz , Wy;
31 Wz , 0 , -Wx;
32 -Wy , Wx , 0];
33
34 % Se evaluan las ecuaciones
35 Hr3_dot = -SOmega*(Hr3 - B1) + bmuG*tanh(HmuG + B2)+ dB1eta
36 *(-g*(Hr3 - B1) + bmuA*tanh(HmuA + B3));
37 HmuG_dot = -dB2eta*(-g*(Hr3 - B1) + bmuA*tanh(HmuA + B3));
38 HmuA_dot = -dB3eta*(-g*(Hr3 - B1) + bmuA*tanh(HmuA + B3));

39 % Se integran las ecuaciones
40 Hr3 = Hr3 + h*Hr3_dot;
41 HmuG = HmuG + h*HmuG_dot;
42 HmuA = HmuA + h*HmuA_dot;
```

```

43
44 % Se calculan los angulos
45 r3e = Hr3 - B1;
46 r3e = r3e/norm(r3e);
47 r = -(180/pi)*asin(r3e(1));
48 p = (180/pi)*atan2(r3e(2),r3e(3));
49 end

```

Algoritmo A.7: Código del filtro Complementario

```

1 -----
2 Algoritmo: F - COMPLEMENTARIO DIRECTO
3 Author: Edwin Rosario Gabriel
4 Date: 23/04/2013 , 05:01 am
5 -----
6 function [R1,P1,Y1,R2,P2,Y2] = FCOMPLEMENTARIOD(Ax,Ay,Az,Wx,Wy,Wz)
7 % Condiciones iniciales
8 Re=[1 0 0 ; 0 1 0 ; 0 0 1];
9 be=[0; 0; 0];
10
11 % Se calcula Ry instantanea
12 [Ry]=TRIADR(Ax,Ay,Az,Mx,My,Mz);
13 Omega=[Wx;Wy;Wz];
14 Kp = 25;
15
16 % Operaciones del metodo
17 E = Re'*Ry;
18 Pa = (1/2)*(E - E');
19 w=[-Pa(2,3);Pa(1,3);-Pa(1,2)];
20 A = Ry*Omega + Kp*Re*w;
21 Ax=A(1,1); Ay=A(2,1); Az=A(3,1);
22 AA=[ 0 -Az Ay;
23 Az 0 -Ax;
24 -Ay Ax 0];
25
26 % Se integra la matriz de rotacion Re
27 dRe = AA*Re
28 Re = Re + h*dRe;
29
30 % Convertir de DCM 2 Euler
31 [r1,p1,y1] = DCM2EULER(Re); % Estimado
32 [r2,p2,y2] = DCM2EULER(Ry); % Triad
33 end

```


Apéndice B

Configuración del Bus-I²C del DSP

En este apéndice se describe brevemente la configuración y operación del módulo I²C disponible en el DSP TMS320F28335 para establecer la comunicación con la IMU MINIMU9v2. Para detalles mas precisos existe una amplia literatura referente al bus-I²C que pueden ser consultadas en [60],[54]. Para el modulo I²C del DSP deberá referirse a [61].

B.1. Introducción al módulo I²C

El bus-I²C fue diseñado por la marca Philips y proporciona una interfaz entre dispositivos compatibles a un velocidad de 100 kbit/s en el modo estándar. Es muy usado en la industria, principalmente para comunicar microcontroladores y sus periféricos en sistemas integrados que normalmente residen en un mismo circuito impreso.

El bus-I²C utiliza dos líneas bidireccionales para transmitir la información, una para los datos SDA y otra para la señal de reloj SCL. Los dispositivos conectados al bus tienen una dirección única que los identifica, pueden ser configurados como Maestros o Esclavos y funcionan como transmisor o receptor en función de las transacciones de los datos. El dispositivo maestro inicia la transferencia de datos y además genera la señal de reloj, pero no es necesario que el maestro sea siempre el mismo, esta característica hace que al bus-I²C sea también un bus multi-maestro. La figura B.1 muestra un ejemplo de multiples dispositivos I²C conectados, donde cada línea debe estar conectada a una tensión positiva mediante una resistencia pull-up.

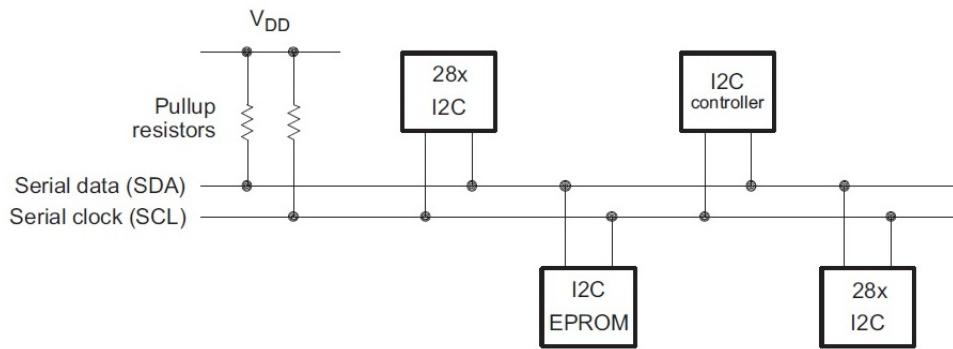


Figura B.1: Multiples dispositivos I²C conectados.

El módulo I²C consta de los siguientes bloques principales, ver Figura B.2, donde el CPU escribe los datos para su transmisión en el registro I2CDXR y lee los datos recibidos de I2CDRR. Cuando el módulo de I²C está configurado como un transmisor, los datos escritos en I2CDXR se copian en I2CXSR y se transmiten en el pin SDA un bit a la vez. Cuando el módulo de I²C está configurado como un receptor, los datos recibidos se desplazan al I2CRCSR y luego se copian a I2CDRR.

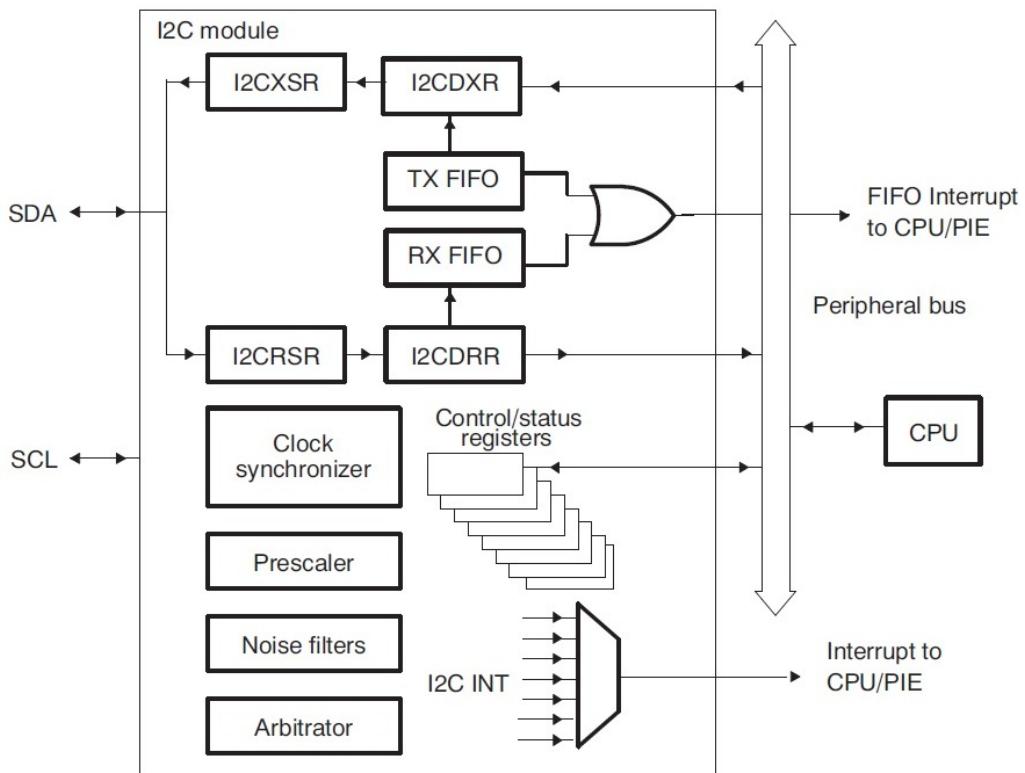


Figura B.2: Diagrama a bloques del módulo I²C.

B.2. Reloj del módulo I²C

Es muy importante configurar adecuadamente el dispositivo generador de reloj del bus-I²C, ver Figura B.3, para sincronizar la comunicación entre el dispositivo maestro y esclavo, este módulo recibe una señal de reloj externa InputCLOCKI2C equivalente al reloj del CPU (150 MHz) y genera la señal de reloj del módulo I²C CLOCKI2C y la señal de comunicación SCL con una cierta frecuencia a través de un preescalador programable.

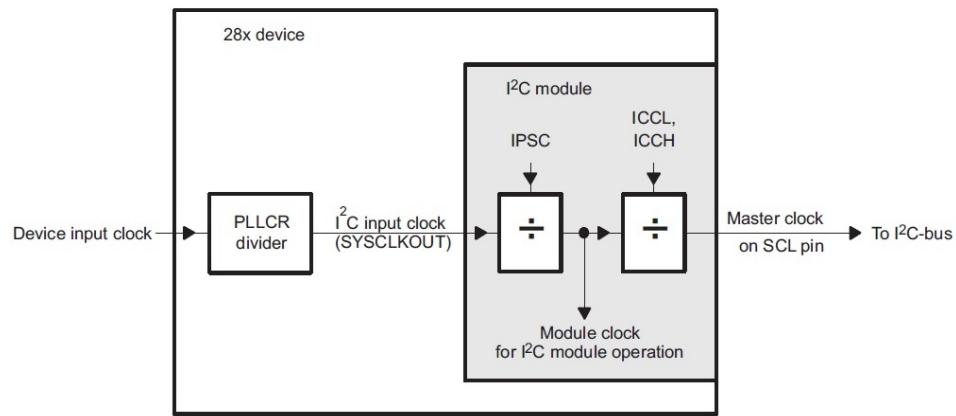


Figura B.3: Diagrama de reloj del módulo I2C.

Para especificar la frecuencia del módulo I²C se debe inicializar el campo IPSC del registro I2CPSC con la siguiente expresión.

$$CLOCKI2C = \frac{InputCLOCKI2C}{IPSC + 1} \quad (B.1)$$

Para satisfacer todas las especificaciones de tiempo del protocolo I²C, el reloj del módulo CLOCKI2C debe configurarse entre 7-12 MHz. El preescalador debe ser inicializado solo mientras el módulo I²C esta en modo Reset, IRS=0 in I2CMDR. La frecuencia del preescalador toma efecto sólo cuando el campo IRS es cambiado a 1. Cambiar el valor de IPSC mientras IRS=1 no tiene efecto. El reloj del dispositivo maestro aparece en el pin SCL cuando el módulo I²C esta configurado para ser un maestro en el bus-I²C. Este reloj controla la sincronización de la comunicación entre el módulo de I²C y un esclavo.

B.3. Registros del módulo I²C

El módulo de I²C del DSP tiene quince registros B.1, donde cada registro contienen campos para configurar la comunicación entre los dispositivos, para mayor detalle ver [61]. A continuación se detallan las funciones para inicializar el módulo I²C del DSP y para leer y escribir datos en la IMU.

Tabla B.1: Registros del módulo I²C del DSP

Nombre	Dirección	Descripción
I2CSTR	0x7902	Registro de estado
I2CCLKL	0x7903	Registro para el tiempo en bajo del reloj
I2CCLKH	0x7904	Registro para el tiempo en alto del reloj
I2CCNT	0x7905	Registro contador de datos
I2CDRR	0x7906	Registro de datos recibidos
I2CSAR	0x7907	Registro de la dirección del esclavo
I2CDXR	0x7908	Registro de datos transmitidos
I2CMDR	0x7909	Registro para configurar el modo
I2CISRC	0x790A	Registro de fuentes de interrupción
I2CEMDR	0x790B	Registro del modo extendido
I2CPSC	0x790C	Registro del preescalador
I2CFFTX	0x7920	Registro de transmisión del FIFO
I2CFFRX	0x7921	Registro de recepción del FIFO
I2CRSR	-	Registro de recepción desplazado
I2CXSR	-	Registro de transmisión desplazado

Con la función B.1 primero se reinicia el módulo I²C, se configura el DSP como maestro y la IMU como esclavo, después se fija a 10MHz el reloj del módulo I²C y la frecuencia de transmisión SCL en 400KHz que corresponde al modo rápido de la IMU, se configuran los tiempos en alto y en bajo de SCL y finalmente se habilita el módulo I²C para la transmisión y recepción de datos.

Algoritmo B.1: Configuración del módulo I²C

```
1 void I2C_Init(void)
2 {
3     /* Reiniciar el modulo. */
4     I2caRegs.I2CMDR.bit.IRS = 0;
5
6     /* Conf. el registro de modo. DSP-Master , IMU-Slave */
7     I2caRegs.I2CMDR.all = 0x0000;
8
9     /* Deshabilitar las interrupciones. */
10    I2caRegs.I2CIER.all = 0x0000;
```

```

12  /* Configurar el preescalador del modulo. 14 -> 150 Mhz/15 = 10
   MHz Clok Modulo I2C */
13  I2caRegs.I2CPSC.all = 14;
14
15  /* Duracion de la parte baja del reloj I2C. 7 -> 400 KHz */
16  I2caRegs.I2CCLKL = 7;
17
18  /* Duracion de la parte alta del reloj I2C. 8 -> 400 KHz */
19  I2caRegs.I2CCLKH = 8;
20
21  /* Habilitar el modulo. */
22  I2caRegs.I2CMDR.bit.IRS = 1;
23
24  /* Habilitar la interrupcion del I2C .*/
25  PieCtrlRegs.PIEIER8.bit.INTx1 = 1;
26  IER |= 0x0080;
27 }
```

Con la función B.2 se leen los datos del giroscopio, del magnetómetro y del aerómetro, primero se configura el DSP como maestro-transmisor y se envía la dirección del dispositivo esclavo y el registro del cual se quiere hacer una lectura, después se configura el DSP como maestro-receptor para recibir el dato.

Algoritmo B.2: Leer un dato en el bus-I²C

```

1 unsigned char I2C_Read(unsigned char ADDRESS, unsigned char
REGISTER)
2 {
3     unsigned char DATO=0x00;
4
5     /* Asignamos la direccion del esclavo. */
6     I2caRegs.I2CSAR = ADDRESS;
7
8     /* Transmitir. */
9     I2caRegs.I2CCNT = 1;
10    I2caRegs.I2CMDR.all = 0x6E20;
11    I2caRegs.I2CDXR = REGISTER;
12    while(I2caRegs.I2CSTR.bit.XRDY != 1){ }
13
14    /* Recibir */
15    I2caRegs.I2CCNT = 1;
16    I2caRegs.I2CMDR.all = 0x6C20;
17    while(I2caRegs.I2CSTR.bit.RRDY != 1){ }
18    DATO = I2caRegs.I2CDRR;
19
20    I2caRegs.I2CMDR.bit.IRS=0;
21    I2caRegs.I2CMDR.bit.IRS=1;
22    return DATO;
23 }
```

Con la función B.3 se pueden escribir datos en la IMU para configurarla adecuadamente,

primero se fija la dirección del dispositivo esclavo al cual se desea enviar un dato y después se configura el DSP como maestro-transmisor, se envía entonces la dirección del registro a sobreescribir y después el dato.

Algoritmo B.3: Escribir un dato en el bus-I²C

```
1 int I2C_Write(unsigned char ADDRESS, unsigned char REGISTER,
2                 unsigned char DATA)
3 {
4
5     /* Asignar la direccion de el esclavo */
6     I2caRegs.I2CSAR = ADDRESS;
7
8     /* Numero de datos a transmitir */
9     I2caRegs.I2CCNT = 2;
10
11    /* Fijar modo maestro transmisor */
12    I2caRegs.I2CMDR.all = 0x6E20;
13    i = 0;
14    while(1)
15    {
16        if (I2caRegs.I2CSTR.bit.XRDY == 1)
17        {
18            if(i == 0)
19            {
20                I2caRegs.I2CDXR = REGISTER;
21            }
22            if(i == 1)
23            {
24                I2caRegs.I2CDXR = DATA;
25            }
26            i = i + 1;
27            if(i == 2)
28            {
29                break;
30            }
31        }
32    }
33    /* Esperar condicion de paro */
34    while(I2caRegs.I2CMDR.bit.STP == 1){}
35    I2caRegs.I2CMDR.bit.IRS=0;
36    I2caRegs.I2CMDR.bit.IRS=1;
37    return (0);
38 }
```

Apéndice C

Unidad de Medición inercial Minimu9v2

La MinIMU-9v2 de la marca Pololu® es una unidad de medición inercial que tiene integrado un giroscopio L3GD20, un acelerómetro y magnetómetro LSM303DLHC en una pequeña tarjeta de 0.8 x 0.5 in. Usa una interfaz I²C para acceder a los nueve datos independientes de cada sensor que tienen varias opciones configurables incluyendo la sensibilidad y la velocidad de la salida de datos de cada sensor. Para la configuración de los sensores se recomienda leer cuidadosamente [62] y [63]. Algunas especificaciones importantes se presentan a continuación.

- Dimensiones: 0,8 × 0,5 × 0,1 in.
- Peso sin conectores (pines): 0,7 g.
- Interfaz de comunicación: I²C.
- Voltaje de operación: 10 ma.
- Corriente de operación: 2,5 a 5,5 v.
- Ejes: *pitch* (x), *roll* (x) y *yaw* (x).
- Rango de sensibilidad:
 - Giroscopio (16 b/*axis*): ±250, ±500, ±2000 ° / s.
 - Acelerómetro (12 b/*axis*): ±2, ±4, ±8, ±16 g.
 - Magnetómetro (12 b/*axis*): ±1,3, ±1,9, ±2,5, ±4,0, ±4,7, ±5,6, ±8,1 gauss.

Los tres sensores actúan como dispositivos esclavos en el mismo bus-I²C con diferentes direcciones, la dirección para el giroscopio con el pin *SA0* en alto es 1101011b, la dirección del acelerómetro es 0011001b y para el magnetómetro 0011110b. De igual

forma cada sensor tiene un conjunto de registros, a los cuales se accede especificando la dirección del esclavo y del registro. Presentamos las funciones utilizadas para configurar la IMU.

Algoritmo C.1: Inicializar la IMU

```

1 void InitSensors(void)
2 {
3     /* Configuracion del Giroscopio */
4     I2C_Write(ADDRESS_GYR, L3G_CTRL_REG1, 0xFF);
5     I2C_Write(ADDRESS_GYR, L3G_CTRL_REG4, 0x00);
6
7     /* Configuracion del Acelerometro */
8     I2C_Write(ADDRESS_ACE, LSM303_CTRL_REG1_A, 0x97);
9     I2C_Write(ADDRESS_ACE, LSM303_CTRL_REG4_A, 0x10);
10
11    /* Configuracion del Magnetometro */
12    I2C_Write(ADDRESS_MAG, LSM303_CRA_REG_M, 0x1C);
13    I2C_Write(ADDRESS_MAG, LSM303_CRB_REG_M, 0x80);
14    I2C_Write(ADDRESS_MAG, LSM303_MR_REG_M, 0x00);
15 }
```

Una vez que la IMU ha sido inicializada y configurada correctamente podemos adquirir los datos de los sensores con la función [C.2](#), donde primero se accede a la parte alta y baja de cada registro y después se compone la muestra con un corrimiento de 8 bits de la parte alta, finalmente se calibran los datos en base a la configuración de la sensibilidad de cada sensor.

Algoritmo C.2: Adquirir datos de la IMU

```

1 void AdDatos(void)
2 {
3     /* Obteccion de los datos del sensor */
4     GYR_Data[0] = I2C_Read(ADDRESS_GYR, L3G_OUT_X_L);
5     GYR_Data[1] = I2C_Read(ADDRESS_GYR, L3G_OUT_X_H);
6     GYR_Data[2] = I2C_Read(ADDRESS_GYR, L3G_OUT_Y_L);
7     GYR_Data[3] = I2C_Read(ADDRESS_GYR, L3G_OUT_Y_H);
8     GYR_Data[4] = I2C_Read(ADDRESS_GYR, L3G_OUT_Z_L);
9     GYR_Data[5] = I2C_Read(ADDRESS_GYR, L3G_OUT_Z_H);
10    Gx = (int)((GYR_Data[1] << 8) + GYR_Data[0]);
11    Gy = (int)((GYR_Data[3] << 8) + GYR_Data[2]);
12    Gz = (int)((GYR_Data[5] << 8) + GYR_Data[4]);
13
14    ACC_Data[0] = I2C_Read(ADDRESS_ACE, LSM303_OUT_X_L_A);
15    ACC_Data[1] = I2C_Read(ADDRESS_ACE, LSM303_OUT_X_H_A);
16    ACC_Data[2] = I2C_Read(ADDRESS_ACE, LSM303_OUT_Y_L_A);
17    ACC_Data[3] = I2C_Read(ADDRESS_ACE, LSM303_OUT_Y_H_A);
18    ACC_Data[4] = I2C_Read(ADDRESS_ACE, LSM303_OUT_Z_L_A);
19    ACC_Data[5] = I2C_Read(ADDRESS_ACE, LSM303_OUT_Z_H_A);
20    Ax = (int)((ACC_Data[1] << 8) + ACC_Data[0]);
21    Ay = (int)((ACC_Data[3] << 8) + ACC_Data[2]);
```

```

22     Az = (int) ((ACC_Data[5] << 8) + ACC_Data[4]);
23
24     MAG_Data[0] = I2C_Read(ADDRESS_MAG, LSM303_OUT_X_H_M);
25     MAG_Data[1] = I2C_Read(ADDRESS_MAG, LSM303_OUT_X_L_M);
26     MAG_Data[2] = I2C_Read(ADDRESS_MAG, LSM303_OUT_Y_H_M);
27     MAG_Data[3] = I2C_Read(ADDRESS_MAG, LSM303_OUT_Y_L_M);
28     MAG_Data[4] = I2C_Read(ADDRESS_MAG, LSM303_OUT_Z_H_M);
29     MAG_Data[5] = I2C_Read(ADDRESS_MAG, LSM303_OUT_Z_L_M);
30     Mx = (int) ((MAG_Data[0] << 8) + MAG_Data[1]);
31     My = (int) ((MAG_Data[2] << 8) + MAG_Data[3]);
32     Mz = (int) ((MAG_Data[4] << 8) + MAG_Data[5]);
33
34     /* Calibracion de los datos */
35     Gxc = (float)((float)Gx/114);
36     Gyc = (float)((float)Gy/114);
37     Gzc = (float)((float)Gz/114);
38
39     Axc = (float)((float)Ax/8000)*g;
40     Ayc = (float)((float)Ay/8000)*g;
41     Azc = (float)((float)Az/8000)*g;
42
43     Mxc = (float)Mx;
44     Myc = (float)My;
45     Mzc = (float)Mz;
46 }
```


Apéndice D

Mediciones de la IMU

En este apéndice se presentan las mediciones que proporciona la Unidad de Medición Inercial con la que se realizaron los experimentos presentados en el Capítulo 5. Se miden las velocidades angulares $\Omega_b = \Omega + b_\Omega + \mu_\Omega$, la aceleración $\mathbf{a}_b = R_i^{bT}(\dot{v} - g) + b_a + \mu_a$ y el campo magnético $\mathbf{m}_b = R_i^{bT}m_i + B_m + \mu_m$ mostradas en las Figuras D.1, D.2 y D.3 respectivamente. Estas mediciones son tomadas con la IMU en reposo con una orientación arbitraria y se puede apreciar el ruido modelado a través de los términos μ_Ω , μ_a y μ_m que presentan estos sensores MEMS. El ruido en las mediciones y los sesgos o *bias* de acelerómetro y del giroscopio son los principales inconvenientes que se deben atenuar para estimar la orientación.

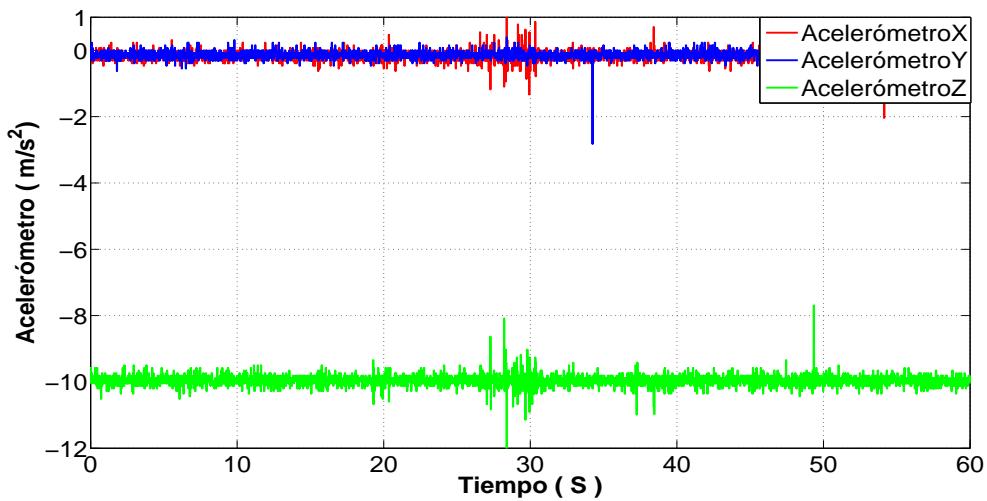


Figura D.1: Mediciones del Acelerómetro

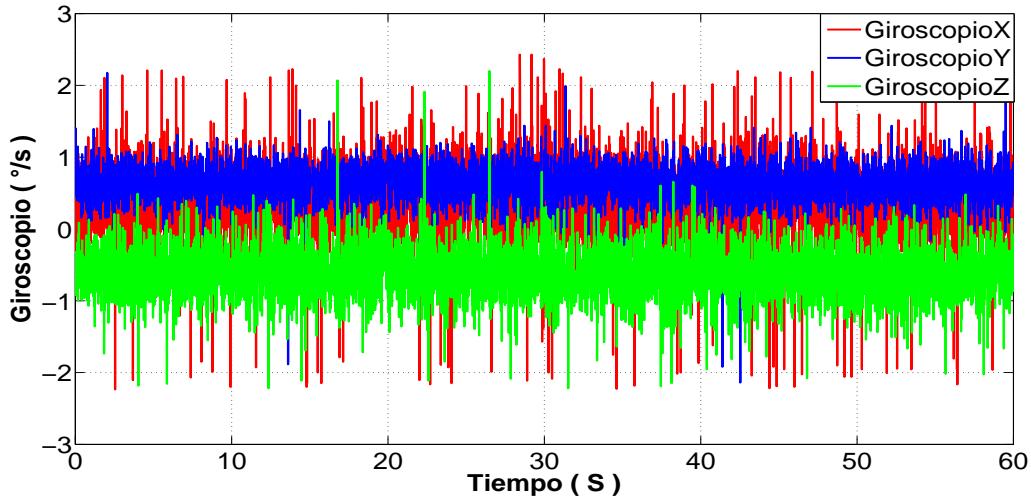


Figura D.2: Mediciones de Giroscopio

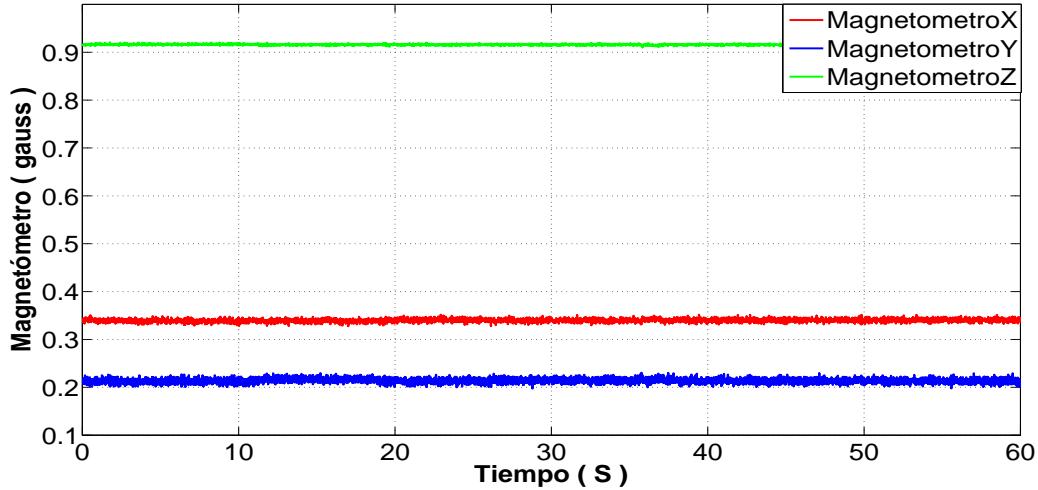


Figura D.3: Mediciones del Magnetómetro

Cada señal de la IMU se puede modelar como una variable aleatoria X con una distribución de probabilidad Normal $N \sim (\mu_x, \sigma^2)$ de manera aproximada como se describe en [64]. En las Figuras siguientes se muestran por separado un análisis de la distribución de probabilidad de cada señal usando histogramas y se puede apreciar que los ruidos del giroscopio mostrados en las Figuras D.7, D.8 y D.9 se aproximan de mejor forma a esta distribución. Estas gráficas fueron obtenidas tomando un conjunto de muestras de las señales de los sensores y después se analizaron en Matlab utilizando comandos de estadística. La distribución de probabilidad nos permite aproximar la media $\mu_x = E(X)$ y las varianzas $\sigma^2 = Var(X) = E[(X - \mu)^2]$ de cada señal.

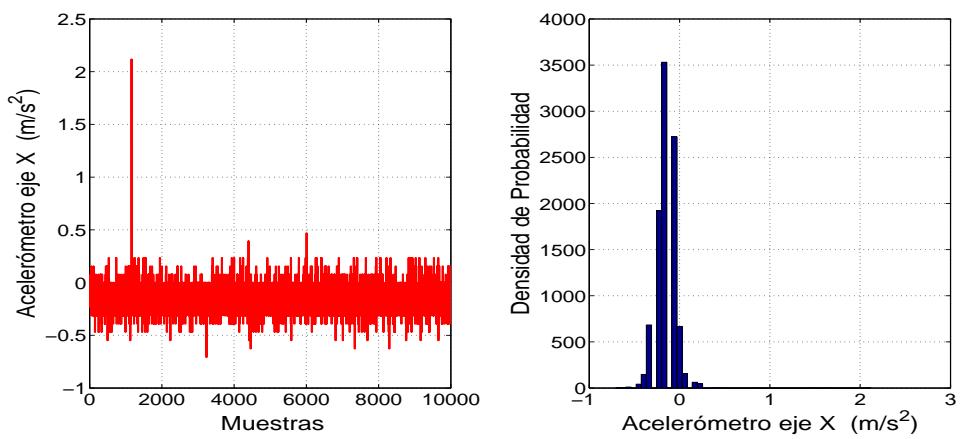


Figura D.4: Acelerómetro eje X

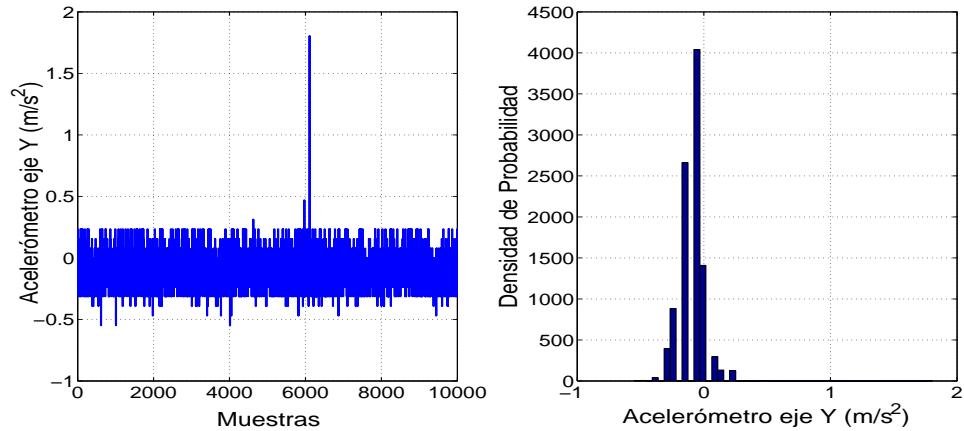


Figura D.5: Acelerómetro eje Y

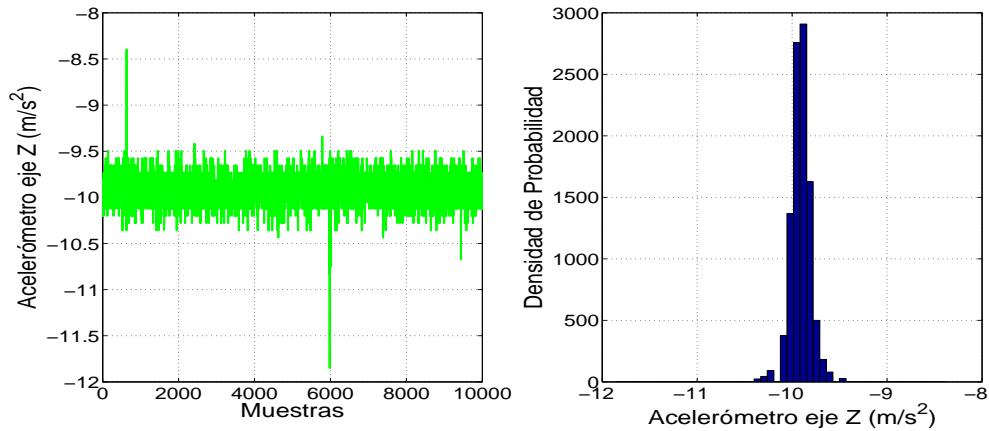


Figura D.6: Acelerómetro eje Z

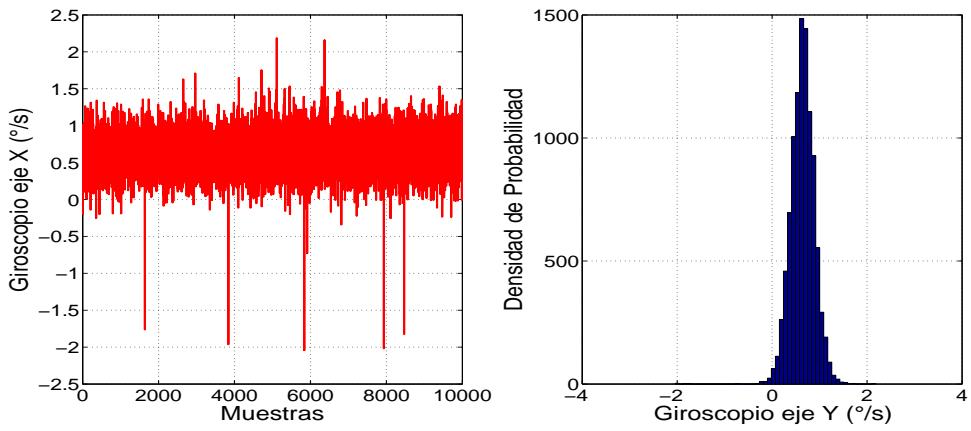


Figura D.7: Giroscopio eje X

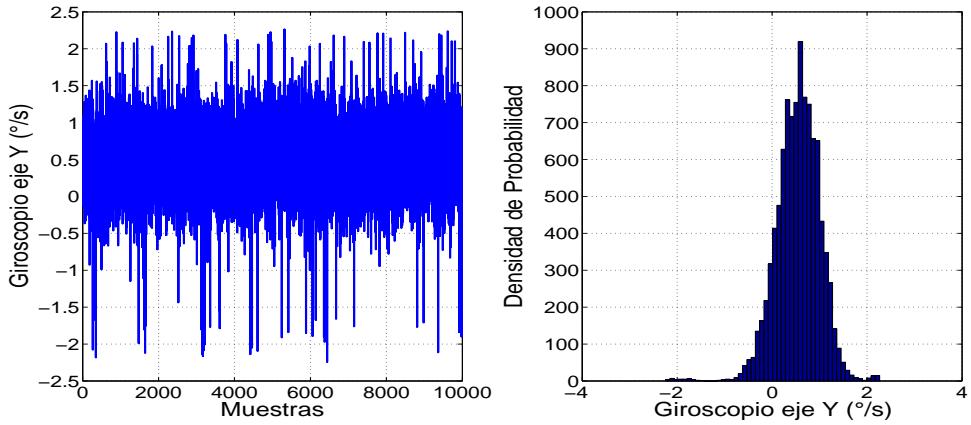


Figura D.8: Giroscopio eje Y

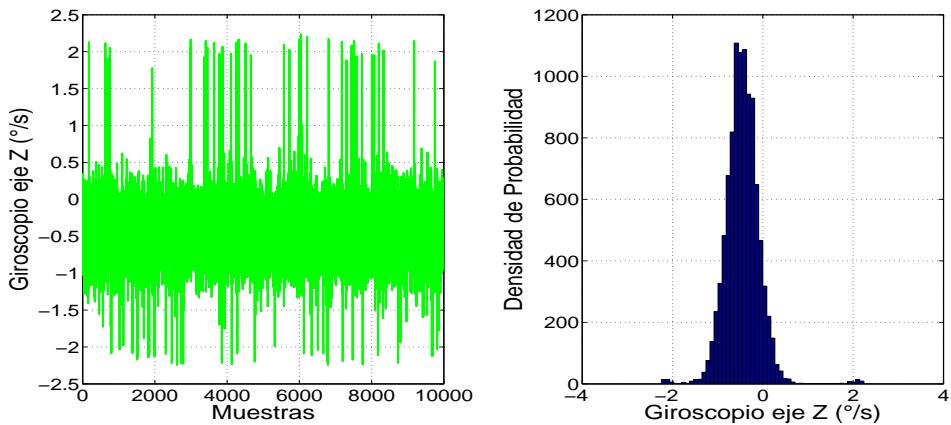


Figura D.9: Giroscopio eje Z

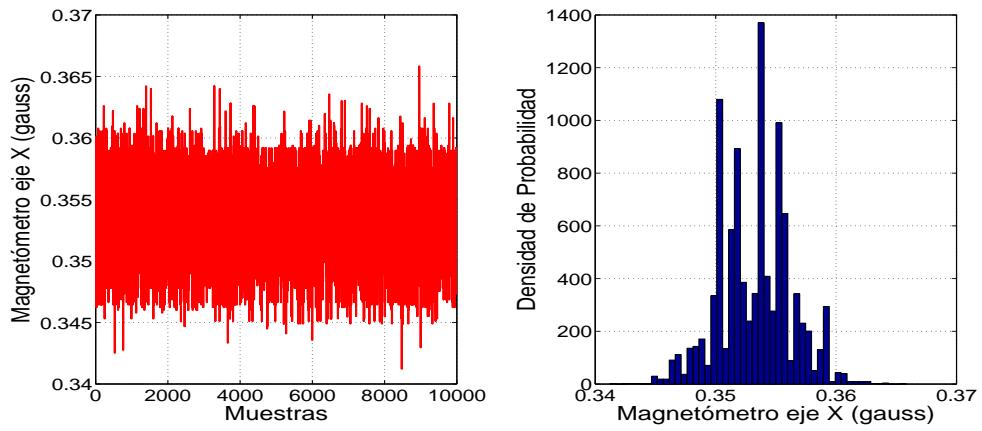


Figura D.10: Magnetómetro eje X

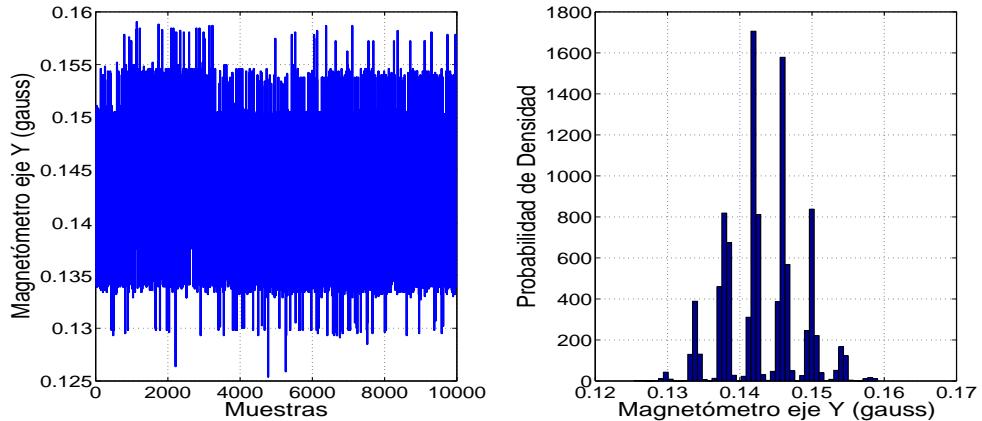


Figura D.11: Magnetómetro eje Y

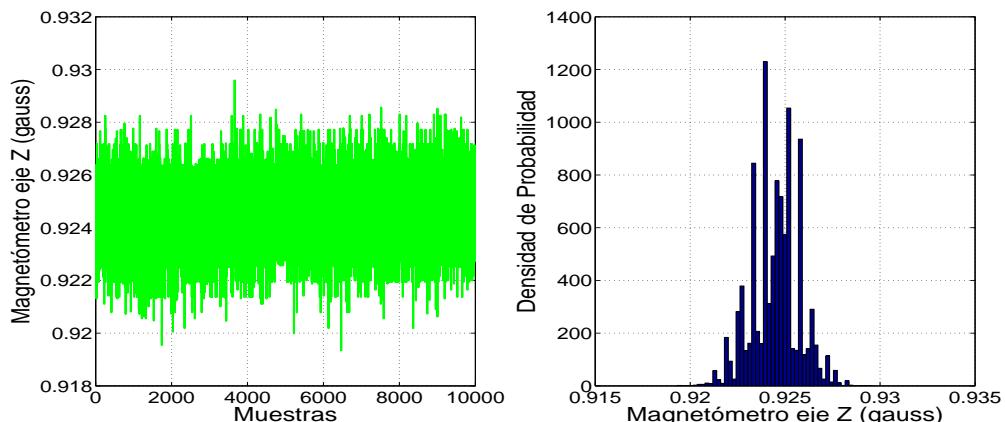


Figura D.12: Magnetómetro eje Z

Bibliografía

- [1] A El Hadri and A Benallegue. Attitude estimation with gyros-bias compensation using low-cost sensors. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 8077–8082. IEEE, 2009.
- [2] Kevin J Walchko and Paul AC Mason. Inertial navigation. In *Proceedings of Florida Conference on Recent Advances in Robotics*, 2002.
- [3] Nojan Madinehi. Rigid body attitude estimation: An overview and comparative study. Master’s thesis, The Western University London, Ontario, Canada, 2013.
- [4] Hyoung-Jin Kim, Chongam Kim, Oh-Hyun Rho, and HD BLACK. A passive system for determining the attitude of a satellite. *AIAA Journal*, 2(7):1350–1351, 1964.
- [5] Grace Wahba. A least squares estimate of satellite attitude. *Siam Review*, 7(3):409–409, 1965.
- [6] James R Wertz. *Spacecraft attitude determination and control*, volume 73. Kluwer Academic Pub, 1978.
- [7] F Landis Markley and Daniele Mortari. Quaternion attitude estimation using vector observations. *Journal of the Astronautical Sciences*, 48(2):359–380, 2000.
- [8] L Fallon III, IH Harrop, and CR Sturch. Ground attitude determination and gyro calibration procedures for the heao missions. In *AIAA, Aerospace Sciences Meeting*, volume 1, 1979.
- [9] Malcolm D Shuster. Approximate algorithms for fast optimal attitude computation. In *Guidance and Control Conference*, volume 1, pages 88–95, 1978.
- [10] Malcolm D Shuster and S D Oh. Three-axis attitude determination from vector observations. *Journal of Guidance, Control, and Dynamics*, 4(1):70–77, 1981.
- [11] Malcolm D Shuster. In quest of better attitudes. In *Proceeding of the 11 th Annual AAS/AIAA Space Flight Mechanics Meeting, Santa Barbara, CA*, pages 2089–2117, 2001.

- [12] Malcolm D Shuster. A simple kalman filter and smoother for spacecraft attitude. *Journal of the Astronautical Sciences*, 37(1):89–106, 1989.
- [13] Mark L Psiaki. Attitude-determination filtering via extended quaternion estimation. *Journal of Guidance, Control, and Dynamics*, 23(2):206–214, 2000.
- [14] F Landis Markley. Attitude determination using vector observations and the singular value decomposition. *The Journal of the Astronautical Sciences*, 36(3):245–258, 1988.
- [15] F Landis Markley. Attitude determination using vector observations: A fast optimal matrix algorithm. *Journal of the Astronautical Sciences*, 41(2):261–280, 1993.
- [16] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.
- [17] F Landis Markley. Multiplicative vs. additive filtering for spacecraft attitude determination. *Dynamics and Control of Systems and Structures in Space*, 2004.
- [18] Eric A Wan and Rudolph Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pages 153–158. IEEE, 2000.
- [19] Neil J Gordon, David J Salmond, and Adrian FM Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, pages 107–113. IET, 1993.
- [20] Andrew Roberts and Abdelhamid Tayebi. On the attitude estimation of accelerating rigid-bodies using gps and imu measurements. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 8088–8093. IEEE, 2011.
- [21] Philippe Martin and Erwan Salaün. Design and implementation of a low-cost observer-based attitude and heading reference system. *Control Engineering Practice*, 18(7):712–722, 2010.
- [22] José Fernandes Vasconcelos, Carlos Silvestre, and Paulo Jorge Oliveira. A nonlinear observer for rigid body attitude estimation using vector observations. In *Proceedings of the 17th IFAC World Congress*, volume 17, pages 8599–8604, 2008.
- [23] Robert Mahony, Tarek Hamel, and J-M Pflimlin. Complementary filter design on the special orthogonal group $so(3)$. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pages 1477–1484. IEEE, 2005.

- [24] Robert Mahony, Tarek Hamel, and J-M Pflimlin. Nonlinear complementary filters on the special orthogonal group. *Automatic Control, IEEE Transactions on*, 53(5):1203–1218, 2008.
- [25] I Estrada-Sanchez, H Rodriguez-Cortes, and M Velasco-Villa. Immersion and invariance based experimental attitude estimation for mobile robots with low acceleration profiles. In *Electrical Engineering, Computing Science and Automatic Control, 2008. CCE 2008. 5th International Conference on*, pages 374–379. IEEE, 2008.
- [26] Miniplanes. RC helicopter U13A. <http://www.miniplanes.fr/helicoptere-24ghz-avec-camera-udi-rc>. Accessed: 2013-11-23.
- [27] Heliplane. Cuadricoptero HUBSAN X4. <http://www.heliplane.es/es/parrot-ardrone-20-hd/897-cuadricoptero-hubsan-x4.html>. Accessed: 2013-11-23.
- [28] Montana State University. Cubesats. <http://kandaka.com/tag/cubesat/>. Accessed: 2013-11-23.
- [29] A.I. & Robot. Robot bípedo en bicicleta. <http://ai2001 ifdef .jp/>. Accessed: 2013-11-23.
- [30] Segway LLC of Manchester. Segway human transporter. <http://www.segway.com/>. Accessed: 2013-11-23.
- [31] Boston Dynamics. Bigdog. http://www.bostondynamics.com/robot_bigdog.html. Accessed: 2013-11-23.
- [32] Nadim Maluf and Kirt Williams. *Introduction to microelectromechanical systems engineering*. Artech house publishers, 2004.
- [33] Oliver J Woodman. An introduction to inertial navigation. *University of Cambridge, Computer Laboratory, Tech. Rep. UCAMCL-TR-696*, 2007.
- [34] Anthony Lawrence. *MODERN INERTIAL TECHNOLOGY.: Navigation, Guidance, and Control*. Springer, 1998.
- [35] Stephen Beeby. *MEMS mechanical sensors*. Artech House, 2004.
- [36] Julian W Gardner and Vijay K Varadan. *Microsensors, MEMS and smart devices*. John Wiley & Sons, Inc., 2001.
- [37] Ville Kaajakari. Practical mems: Design of microsystems, accelerometers, gyroscopes, rf mems, optical mems, and microfluidic systems. *Small Gear Publishing*, 2009.

- [38] Innam Lee, Gil Ho Yoon, Jungyul Park, Seonho Seok, Kukjin Chun, and Kyo-Il Lee. Development and analysis of the vertical capacitive accelerometer. *Sensors and Actuators A: Physical*, 119(1):8–18, 2005.
- [39] Dennis K Wickenden, Thomas J Kistenmacher, Robert Osiander, Scott A Ecelberger, R Ben Givens, and John C Murphy. Development of miniature magnetometers. *Johns Hopkins APL Technical Digest*, 18(2):271–278, 1997.
- [40] Herbert Goldstein. *Classical mechanics*, volume 4. Pearson Education India, 1962.
- [41] Peter Carlisle Hughes. *Spacecraft attitude dynamics*. DoverPublications.com, 2012.
- [42] John L Junkins and James D Turner. *Optimal spacecraft rotational maneuvers*. Access Online via Elsevier, 1986.
- [43] Nalin A Chaturvedi, Amit K Sanyal, and N Harris McClamroch. Rigid-body attitude control. *IEEE Control Systems Magazine*, 31(3):30–51, 2011.
- [44] John J Craig. *Introduction to robotics: mechanics and control*. Prentice Hall, 2004.
- [45] Esmat Bekir. *Introduction to modern navigation systems*. World Scientific, 2007.
- [46] Young Soo Suh. Attitude estimation using low cost accelerometer and gyroscope. In *Science and Technology, 2003. Proceedings KORUS 2003. The 7th Korea-Russia International Symposium on*, volume 2, pages 423–427. IEEE, 2003.
- [47] Christopher J Fisher. Using an accelerometer for inclination sensing. *AN-1057, Application note, Analog Devices*, 2010.
- [48] Charles K Chui and Guanrong Chen. *Kalman filtering: with real-time applications*. Springer, 2009.
- [49] Mohinder S Grewal and Angus P Andrews. *Kalman filtering: theory and practice using MATLAB*. Wiley. com, 2011.
- [50] João Luís Marins, Xiaoping Yun, Eric R Bachmann, Robert B McGhee, and Michael J Zyda. An extended kalman filter for quaternion-based orientation estimation using marg sensors. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 4, pages 2003–2011. IEEE, 2001.
- [51] Xiaoping Yun and Eric R Bachmann. Design, implementation, and experimental results of a quaternion-based kalman filter for human body motion tracking. *Robotics, IEEE Transactions on*, 22(6):1216–1227, 2006.

- [52] Alessandro Astolfi, Dimitrios Karagiannis, and Romeo Ortega. *Nonlinear and adaptive control with applications*. Springer, 2008.
- [53] Pololu Robotics & Electronics. Minimu-9 v2 gyro, accelerometer, and compass. <http://www.pololu.com/catalog/product/1268>. Accessed: 2013-10-08.
- [54] NXP Semiconductors. *UM10204 I²C-bus specification and user manual*, 2012.
- [55] Texas Instruments. Tms320f28335 - features. <http://www.ti.com/product/tms320f28335>. Accessed: 2013-09-07.
- [56] Ideal Aerosmith. Model 2103ht three-axis positioning and rate table system. <http://www.ideal-aerosmith.com/pdf/2103HT.pdf>. Accessed: 2013-08-09.
- [57] P. Batista, C. Silvestre, and P. Oliveira. Sensor-based globally asymptotically stable filters for attitude estimation: Analysis, design, and performance evaluation. *Automatic Control, IEEE Transactions on*, 57(8):2095–2100, 2012.
- [58] ABB Robotics Products. Introducción a rapid IRB 2400/16. <http://personal.biada.org/~jhorillo/RAPIDManualoperador.pdf>. Num. de Documento: 3HAC029364-005, Accessed: 2013-08-09.
- [59] ABB Robotics Products. Especificaciones del producto IRB 2400/16. <http://www.abb.es/product/seitp327/f597ea096948f833c125717900224b21.aspx>. Num. de Documento: 3HAC10759-1, Accessed: 2013-08-07.
- [60] Dominique Paret and Carl Fenger. *The I²C bus: from theory to practice*. John Wiley & Sons, Inc., 1997.
- [61] Texas Instruments. *TMS320x2833x, 2823x Inter-Integrated Circuit(I²C) Module*, 2011.
- [62] STMicroelectronics. *MEMS motion sensor: three-axis digital output gyroscope*, 2011.
- [63] STMicroelectronics. *Ultra compact high performance e-compass 3D accelerometer and 3D magnetometer module*, 2011.
- [64] Bin Fang, Wusheng Chou, and Li Ding. Attitude estimation of rigid bodies using mems inertial sensors. In *Intelligent Computation Technology and Automation (ICICTA), 2011 International Conference on*, volume 1, pages 592–595, 2011.