



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

Facultad de Ciencias de la Electrónica

Maestría en Ingeniería Electrónica,
opción Instrumentación Electrónica

Tesis para obtener el grado de
MAESTRO EN INGENIERÍA ELECTRÓNICA

**INSTRUMENTACIÓN DE UNA BOYA PARA EL
MONITOREO DE VARIABLES OCEANOGRÁFICAS**

Presenta:

Osberto Abirán Betanzos Ramírez*

Director:

Dr. José Fermi Guerrero Castellanos

Co-directoras:

Dra. Josefina Castañeda Camacho

Dra. Flor Lizeth Torres Ortiz (UNAM)

Resumen

En las últimas décadas, el uso de sensores inerciales y magneto-resistivos ha abierto un amplio campo de investigación científica para el desarrollo de diversos algoritmos aplicados principalmente a sistemas de navegación en vehículos autónomos. Específicamente, estos sensores han sido aplicados en el desarrollo de sistemas AHRS (del inglés, *Attitude and Heading Reference System*), para la estimación de la orientación de un cuerpo rígido, haciendo uso de algoritmos computacionales que permiten la fusión de todos los datos provenientes de los sensores. Generalmente estos algoritmos hacen uso de un único conjunto de sensores (acelerómetros, magnetómetros y girómetros), que en la literatura es comúnmente llamado IMU (del inglés, *Inertial Measurement Unit*), y que en este trabajo hemos llamado unidad de medición (MU, *Measurement Unit*).

En este trabajo se diseñó, simuló e implementó, un sistema AHRS robusto y tolerante a fallas. El sistema consiste en tres MU's espacialmente distribuidas, donde cada MU es visto como un nodo o agente dentro de una red de sensores y el sistema completo se modela usando la teoría de grafos. Cada agente tiene la capacidad de estimar un cuaternión unitario de orientación, haciendo uso de un algoritmo de estimación determinista y un observador no lineal. Entonces, para fusionar la información de los tres nodos, se utiliza un algoritmo de consenso dinámico, donde cada nodo se comunica con su nodo vecino. La información transmitida permite a cada nodo actualizar su información local para posteriormente ser transmitida a sus vecinos con el fin de alcanzar un consenso. Además, se presentan los resultados experimentales, realizados durante una estancia de tres meses en el laboratorio ICUBE y el INSA (*Institut National des Sciences Appliquées*) en la ciudad de Estrasburgo Francia. Las estimaciones de orientación del sistema propuesto en este trabajo son comparadas con una orientación de referencia obtenida de un sistema de alta precisión basada en cámaras Optitrack.

Resumen gráfico

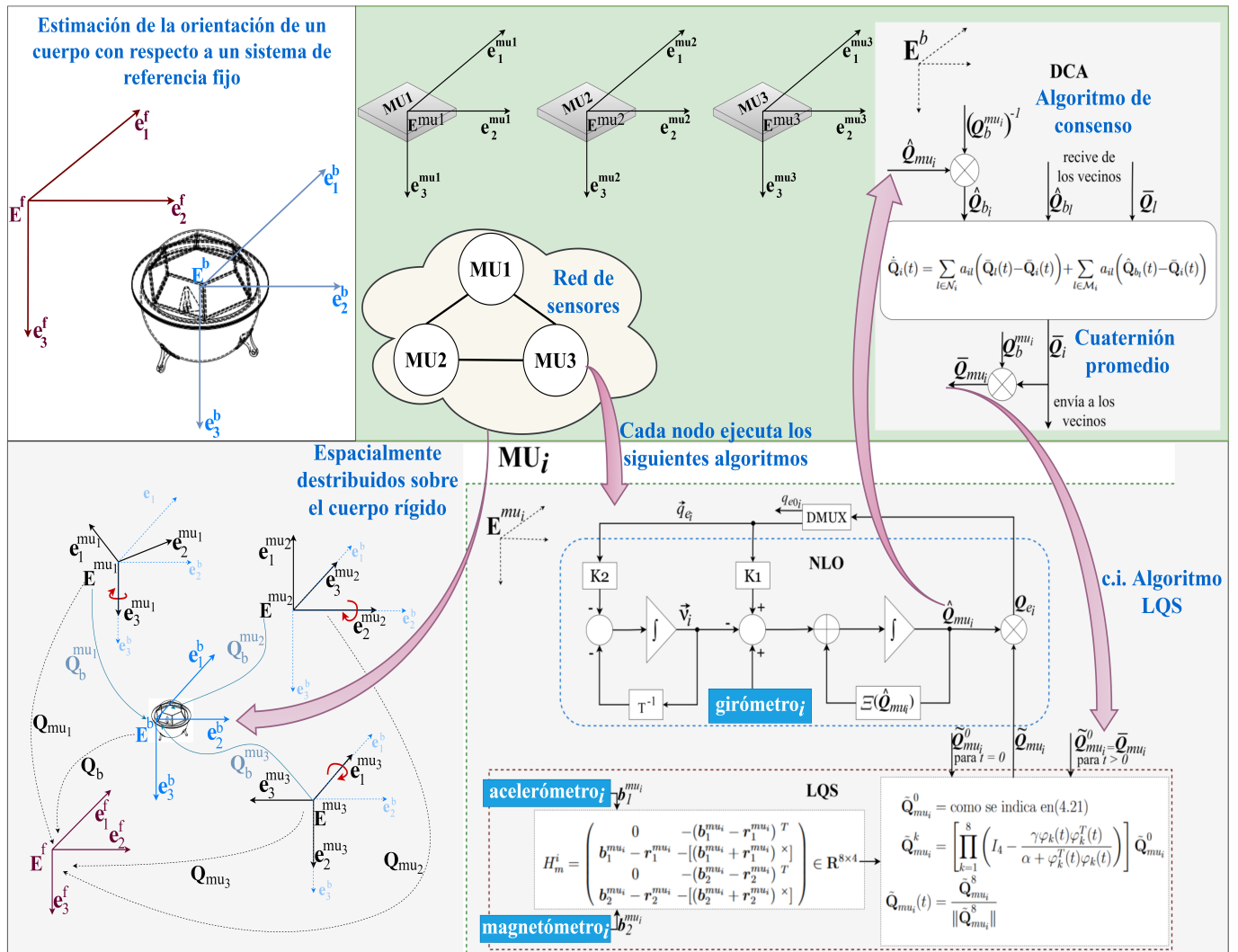


Figura 1: Resumen gráfico del trabajo de tesis desarrollado

Agradecimientos

Gracias a Dios por ser guía, modelo y ejemplo a seguir en todo cuanto se refiere a una persona de valores, vida con propósitos y amor. Porque siempre que mis pensamientos me ahogaban pedía su ayuda y la recibía.

Gracias a mis padres, Ismael Betanzos y Ociel Ramírez quienes siempre han apoyado a mis hermanos y a mí y nos han brindado valiosos consejos de vida además de motivarnos a seguir adelante.

Gracias a mis hermanos, Cinthia y Mauricio porque siempre me han apoyado y me han impulsado a tomar buenas decisiones que en ocasiones dudaba o no tenía el valor de tomar.

Gracias a mi familia quienes me mostraron aprecio y apoyo durante este caminar y me acompañaron de una u otra forma.

Gracias a mis amigos, que, a pesar de no conocerlos físicamente durante los dos años de maestría, siempre estuvieron apoyándome y sumando a mi formación para alcanzar mis objetivos.

Gracias a cada uno de mis asesores y profesores de la maestría quienes compartieron con mucho gusto su conocimiento y experiencia. En especial al Dr. Fermi quién me brindo la oportunidad y confianza de trabajar con él, además de apoyarme, guiarme y ser un amigo durante mis estudios de maestría. Agradezco también al Dr. Sylvain Durand quién me brindo la oportunidad de trabajar y realizar una estancia con su equipo de trabajo en Estrasburgo Francia, gracias por el apoyo y amistad brindados por él y por mis amigos Charlotte, Miguel, Florián, Tibault, Ethan y Devesh.

Gracias al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo económico otorgado durante el periodo de agosto 2020 a julio 2022, sin el cual no habría sido posible realizar mis estudios de maestría.

Gracias a la Facultad de Ciencias de la Electrónica (FCE) de la BUAP, por la invaluable oportunidad de permitirme continuar con mi preparación académica con la maestría en Ingeniería, por los medios y recursos proporcionados.

Finalmente me gustaría agradecer a todas aquellas personas que comparten sus conocimientos de diversas formas como por ejemplo en videos en la página de YouTube, porque también han sido parte de este logro.

Índice general

Resumen	2
Agradecimientos	I
Lista de Figuras	VII
Lista de Tablas	VIII
Lista de Pseudocódigos	IX
1. Introducción	1
1.1. Motivación y Antecedentes	1
1.2. Objetivos	5
1.2.1. Objetivo general	5
1.2.2. Objetivos específicos	5
1.3. Justificación	6
1.4. Descripción	7
2. Marco teórico	9
2.1. Métodos de navegación	10
2.1.1. Método de extrapolación	10
2.1.2. Método de referenciamiento absoluto	11
2.1.3. Navegación inercial	11
2.1.4. Navegación integrada	11
2.2. Sistemas de referencia	12
2.2.1. Sistema de referencia inercial	13
2.2.2. Sistema de referencia terrestre	13
2.2.3. Sistema de referencia geográfico	14
2.2.4. Sistema de referencia del cuerpo	14
2.3. Parametrizaciones de la orientación	15
2.3.1. Matriz de rotación	15
2.3.2. Ángulos de Euler	17
2.3.3. Cuaterniones	18
2.4. Sensores	21
2.4.1. Acelerómetros	21
2.4.2. Girómetros	22

2.4.3.	Magnetómetro	23
2.4.4.	Sistema de Posicionamiento Global (<i>GPS</i>)	24
2.5.	Fusión de Datos multi-sensorial	26
2.5.1.	Algoritmos deterministas para la estimación de la orientación	27
2.5.2.	Observadores	29
2.5.3.	Filtro de Kalman	31
2.5.4.	Observadores No Lineales	36
2.6.	Teoría de grafos	37
3.	Estimación de la orientación: el caso individual	39
3.1.	Planteamiento del problema	41
3.2.	Estimación de la orientación con algoritmo determinista	41
3.3.	Estimación de la orientación implementando un observador no lineal	42
3.4.	Resultados del capítulo	43
3.4.1.	Descripción de la plataforma experimental	44
3.4.2.	Calibración de los sensores	44
3.4.3.	Implementación de los algoritmos de estimación de la orientación	48
4.	Estimación distribuida de la orientación	54
4.1.	Rotación del sistema $\mathbf{E}^{m_{u_i}}$ con respecto al sistema \mathbf{E}^b	55
4.1.1.	Mapeo con matrices de rotación	55
4.1.2.	Mapeo con cuaterniones	58
4.2.	Planteamiento del problema	59
4.3.	Diseño del algoritmo de consenso dinámico	60
4.4.	Resultados basados en simulación	62
4.4.1.	Robustez tolerante a fallas	63
4.5.	Resultados experimentales	66
4.5.1.	Algoritmo de consenso dinámico caso 1	66
4.5.2.	Algoritmo de consenso dinámico caso 2	76
5.	Integración GPS/INS	82
5.1.	Coordenadas del GPS	83
5.2.	Cambio de coordenadas	84
5.3.	Resultados del capítulo	85
5.3.1.	Resultados de la conversión de coordenadas EC-EF a coordenadas cartesianas	85
5.3.2.	Plataforma experimental (Hardware In the Loop)	87
	Conclusiones	89
	Trabajo futuro	90
A.	Métodos numéricos	91
A.1.	Métodos numéricos en la solución de Ecuaciones diferenciales ordinarias	91
A.1.1.	Integración numérica del Observador no lineal	92
B.	Conversiones entre sistemas de referencias	95

C. Artículos publicados	98
C.1. Artículo 1	98
C.2. Artículo 2	105
D. Conexión entre los software MATLAB y Motive	112
D.1. Introducción	112
D.1.1. Instalación del Toolbox Optitrack	112
D.1.2. Configuración de Motive	113
D.1.3. Ejemplo de comunicación entre MATLAB y Motive	115
E. Cronograma de Actividades	117
F. Tabla de avances	118
Referencias	119

Índice de figuras

1.	Resumen gráfico del trabajo de tesis desarrollado	3
1.1.	Sistema de guiado, control y navegación de un vehículo autónomo marino (basado en [1])	3
1.2.	Diagrama a bloque de la implementación de un estimador de posición y orientación utilizando Observadores No Lineales. (Fuente:[13])	4
1.3.	Problemática a abordar: sistema de navegación de una boya	5
1.4.	Diagrama a bloques propuesto para el subsistema de navegación de la boya.	8
2.1.	Vehículos que hacen uso de sistemas de navegación (fuente: google imágenes).	9
2.2.	Obtención de los parámetros de navegación (Fuente:[20]).	10
2.3.	sistemas de referencia mínimos a utilizar en un sistema de navegación (Fuente: [21])	13
2.4.	Coordenadas inerciales y terrestres (Fuente: [19])	14
2.5.	Coordenadas del cuerpo y de navegación (Fuente: [19])	15
2.6.	Rotaciones de ángulos Euler (fuente: [27])	18
2.7.	Representación visual de los ángulos de Euler (fuente: https://www.luisllamas.es/medir-la-inclinacion-imu-arduino-filtro-complementario/).	19
2.8.	Sistema de coordenadas empleado para el acelerómetro	23
2.9.	Sistema de coordenadas empleado para el girómetro	24
2.10.	Constelación de satélites NAVSTAR (Fuente: https://www.gps.gov/spanish.php)	25
2.11.	Ejemplo de una sentencia NMEA-0183 (Fuente: https://brandidowns.com/?p=77)	26
2.12.	Estructura general del observador (Fuente: [35])	30
2.13.	Sistema con perturbación en la entrada y ruido en la medición(Fuente: http://mocha-java.uccs.edu/ECE5550/index.html)	31
2.14.	Algoritmo del filtro de Kalman (Basado en: http://mocha-java.uccs.edu/ECE5550/index.html)	33
2.15.	Arquitectura de fusión, dominio de la observación (Fuente: [38])	34
2.16.	Arquitectura de fusión, dominio de la estimación (Fuente: [38])	35
2.17.	Grafo \mathcal{G}	37
3.1.	Diagrama de bloques de un Attitude and Heading Reference System	39
3.2.	Diagrama de bloques del observador no lineal (3.5)(\otimes representa la multiplicación cuaternión y \oplus la multiplicación matricial).	43
3.3.	Mediciones del girómetro antes y después del proceso de calibración	45
3.4.	Mediciones del acelerómetro antes y después del proceso de calibración	47
3.5.	Mediciones del magnetómetro sin calibrar	48
3.6.	Mediciones del magnetómetro calibradas	48

3.7. Diagrama de bloques de la implementación del algoritmo LQS en Simulink.	49
3.8. Ángulos de Euler obtenidos de una secuencia de rotaciones bien definida.	50
3.9. Diagrama de bloques de la implementación de la ecuación cinemática en Simulink.	50
3.10. Deriva y ángulos de Euler a partir de la integración de la ecuación cinemática.	51
3.11. Diagrama de bloques de la implementación del observador no lineal en Simulink.	51
3.12. Ángulos de Euler obtenidos de una secuencia de rotaciones bien definida a partir del observador no lineal.	52
3.13. Estructura general del buffer de datos.	53
3.14. Sistema instrumentado para la estimación de la orientación.	53
4.1. Sistemas de referencia \mathbf{E}^f , \mathbf{E}^b y \mathbf{E}^{mu}	54
4.2. Sistemas de referencia \mathbf{E}^{mu_i} y \mathbf{E}^b alineados	55
4.3. Relación entre los sistemas de referencia \mathbf{E}^{mu_i} , \mathbf{E}^b y \mathbf{E}^f	56
4.4. Relación de los sistemas de referencia \mathbf{E}^f , \mathbf{E}^b y \mathbf{E}^{mu_i} mediante matrices de rotación	57
4.5. Grafo de comunicación entre cada MU	60
4.6. Diagrama de bloques de un agente en la red de sensores	62
4.7. Comparación entre los ángulos de Euler obtenidos de \mathbf{Q}_b y los ángulos de Euler obtenidos de $\bar{\mathbf{Q}}_i$	64
4.8. Criterio de desempeño ISE aplicado a β_e	65
4.9. Evolución en el tiempo de la parte escalar de \mathbf{Q}_b , $\hat{\mathbf{Q}}_{MU}$ y $\bar{\mathbf{Q}}_i$	66
4.10. Esquema del funcionamiento de las pruebas experimentales para el caso 1	67
4.11. Diagrama de bloque de la interacción de los elementos de la Figura 4.10	68
4.12. Turtlebot instrumentado con unidades de medición y marcadores de cuerpo rígido para estimación de la orientación	68
4.13. Sala de captura de movimiento con cuatro cámaras Optitrack	69
4.14. Turtlebot con una posición inicial de $+45^\circ$ y -45°	70
4.15. Ángulos de Euler obtenidos al mover aleatoriamente el Turtlebot	70
4.16. Ángulos de Euler obtenidos al mover al Turtlebot enviando comandos por ROS	71
4.17. Cuaternión de referencia (Obtenido del Optitrack) y cuaternión estimado por el consenso dinámico, con movimientos aleatorios aplicados al Turtlebot	71
4.18. Cuaternión de referencia (Obtenido del Optitrack) y cuaternión estimado por el consenso dinámico, con movimientos realizados por el Turtlebot mediante comandos de ROS	72
4.19. Ángulos de Euler obtenidos del consenso dinámico, asociado al escenario 1	73
4.20. Ángulos de Euler obtenidos del consenso dinámico, asociado al escenario 2	74
4.21. Ángulos de Euler obtenidos del consenso dinámico, asociado al escenario 3	74
4.22. Ángulos de Euler obtenidos del consenso dinámico, asociado al escenario 4	75
4.23. Ángulos de Euler obtenidos del consenso dinámico, asociado al escenario 5	75
4.24. Ángulos de Euler obtenidos del consenso dinámico, asociado al escenario 6	75
4.25. Ángulos de Euler obtenidos del consenso dinámico, para los escenarios 7 y 8	76
4.26. Características del microcontrolador utilizado para ejecutar los diferentes algoritmos de estimación (Fuente: https://www.st.com/en/microcontrollers-microprocessors/stm32f411re.html)	77
4.27. Implementación de la red de sensores, para la estimación de la orientación de manera distribuida	78
4.28. Diagrama de bloques del intercambio de información entre los tres nodos	79

4.29. Ángulos de Euler obtenido del consenso dinámico procesado a nivel del microcontrolador.	80
4.30. Cuaternión 1 obtenido del consenso dinámico implementado en la tarjeta de desarrollo NUCLEO-64.	81
4.31. Cuaternión 2 obtenido del consenso dinámico implementado en la tarjeta de desarrollo NUCLEO-64.	81
5.1. Trayectoria recorrida por la boya, las líneas discontinuas representan datos de posición obtenidos a partir de la integración de las medidas de un acelerómetro	82
5.2. Trayectoria de la boya registrada por el receptor GPS	82
5.3. Sistema de referencia centrado y fijo en la Tierra \mathbf{E}^{EC-EF} , sistema de referencia local \mathbf{E}^{NED}	83
5.4. Origen del sistema de referencia \mathbf{E}^f	84
5.5. Trayectoria del dron, con respecto al sistema de coordenadas curvilíneo \mathbf{E}^{EC-EF}	86
5.6. Trayectoria del dron, con respecto al sistema de coordenadas rectangular \mathbf{E}^{EC-EF}	86
5.7. Trayectoria del dron, con respecto a un nuevo sistema de referencia \mathbf{E}^f , el cual puede ser definido por el usuario	87
5.8. Diagrama de bloques de la plataforma experimental implementada	87
5.9. Plataforma experimental implementada	88
5.10. Rotación bien conocida con respecto al ángulo ψ y obtención de la posición expresado en el sistema cartesiano	88
A.1. RK4 correspondiente a la ecuación (A.5) implementado en Matlab.	93
A.2. RK4 correspondiente a la ecuación (A.6) implementado en Matlab.	94
C.1. Artículo enviado a congreso internacional ICMEAE.	98
C.2. Artículo enviado a congreso nacional AMCA.	105
D.1. Directorio de trabajo con el Toolbox optitrack para MATLAB	113
D.2. Software Motive y streaming pane option	114
D.3. Menú optiTrack Streaming Engine	114
D.4. Configuración del Up Axis	115
D.5. Info pane para visualizar la información referente a posición y orientación del cuerpo rígido	116
D.6. SCRIPT_DisplayData.m file ejecutándose para adquirir la posición y orientación del cuerpo rígido en tiempo real	116

Índice de tablas

2.1. Definición y dimensiones de variables	32
3.1. Características del girómetro.	45
3.2. Características del acelerómetro.	47
3.3. Características del magnetómetro.	49
4.1. Rotación aplicada a cada MU con respecto al sistema de referencia \mathbf{E}^b	64
4.2. Operación de los sensores del i -ésimo agente durante el proceso de simulación de la Figura 4.8	65
4.3. Cuaternión de rotación aplicada a cada MU en su correspondiente eje con grado de libertad	69
4.4. Operación de los sensores del i -ésimo agente durante el proceso experimental	72
E.1. Tabla de actividades	117
F.1. Tabla de porcentaje de avance de tesis	118

Índice de pseudocódigos

1.	LQS algorithm	42
2.	Non-linear observer (Simulink implementation)	44
3.	rotation matrix to axis-angle	95
4.	rotation matrix to Quaternion	96
5.	quaternion to Euler angles	96
6.	quaternion to axis-angle	97
7.	Conversion hexadecimal to float	97

Capítulo 1

Introducción

En este capítulo se da un panorama general del trabajo de investigación y desarrollo tecnológico llevado a cabo en el marco de la presente tesis. Primeramente se menciona la motivación y los antecedentes de nuestra investigación. En seguida se presentan el objetivo general y el objetivo específico, acompañados de una justificación. Finalmente se presenta una descripción del trabajo realizado y la organización del actual documento.

1.1. Motivación y Antecedentes

Los océanos del mundo cubren dos terceras partes de la superficie terrestre de la tierra y han sido fundamentales para el bienestar humano a lo largo de la historia [1], [2]. Actualmente los mares constituyen una gran red de comunicación marítima entre naciones, permitiendo el intercambio de mercancías y el transporte público o turístico entre ellas, lo que sostiene una vibrante economía a nivel mundial [3], es lugar también de operaciones militares y científicas, actividades de pesca comercial que garantizan la seguridad alimenticia a lo ancho del mundo [3] y muchas otras actividades recreativas y comerciales.

Sin embargo, el crecimiento de la población y con ello el crecimiento de las industrias a nivel mundial ha generado múltiples complicaciones [4]. Dos de las complicaciones más significativas son el agotamiento de los recursos no renovables del planeta, principalmente del gas y el petróleo, de los cuales no se tiene un conocimiento certero de cuando empezaran a notarse la falta de estos recursos, y el cambio climático el cual ha afectado tanto a la vida terrestre como a la vida marina.

Con la finalidad de hacer frente a estos retos, científicos de algunas naciones como la Unión Europea han concentrado sus esfuerzos en generar energía eléctrica a partir del potencial energético de los océanos [5] y en la predicción del cambio climático del mundo [4]. Las principales fuentes de energía marina que en los últimos años han experimentado un mayor auge son: energía del viento, energía undimotriz y energía de las corrientes marinas [5]. En consecuencia el análisis, medición y caracterización del oleaje es relevante para científicos e ingenieros dado que la energía almacenada en una ola depende principalmente de su altura y longitud; la medición de la velocidad del viento en el ambiente marino es otra variable de interés, ya que permite conocer el volumen del flujo de aire en zonas concretas de estudio, y con ello determinar la cantidad de energía que se podría producir con este recurso. También variables como temperatura de la superficie del mar (SST) y salinidad principalmente son de interés para los científicos en el estudio del cambio climático, ya que la variabilidad de estas es crucial para comprender el papel del océano en el sistema climático [6].

El monitoreo de la superficie marina se ha convertido en una tarea muy importante debido a

la necesidad de realizar un mapeo efectivo en diferentes zonas de la costa marina que contribuyan a un estudio de viabilidad económica y análisis de la cantidad de recursos energéticos existentes en cada una de estas zonas de interés [5]. Este monitoreo se ha logrado principalmente con boyas fijas, por ejemplo en [5] donde se propone el diseño de un sistema de boya con sensores para llevar a cabo un estudio de sitios de gran interés para la generación de energía proveniente del ambiente marino. En [7] los autores proponen el diseño de una boya con multi-sensores con el objetivo de realizar estudios de parámetros meteorológicos y marinos. Sin embargo, los entornos acuáticos son inherentemente dinámicos, lo que dificulta significativamente la instalación, mantenimiento y en consecuencia la efectividad de cualquier red de sensores fijos [8]. En los últimos años y con el gran avance de los sensores, actuadores, sistemas de almacenamiento, sistemas de cómputo embebido y sistemas de comunicaciones, se ha hecho posible la implementación de un sistema de monitorización dinámico, basado en boyas marinas que permite realizar un muestreo espacio-temporal. Por ejemplo, en [8] se reporta un sistema de sensado autónomo utilizando boyas marinas que permiten una mejor cobertura de medición espacio-temporal, lo cual contribuye a caracterizar de mejor manera la superficie marina, generando mapas de mediciones en diferentes puntos de la costa marina.

Debido a la importancia de un monitoreo dinámico, debe garantizarse la autonomía de la boya para que esta sea capaz de navegar en entornos desconocidos. En la Figura 1.1 se muestra un diagrama de bloques general y simplificado del sistema de navegación, control y guiado para un vehículo autónomo marino, se puede observar que son tres los principales bloques que garantizan la autonomía de la boya:

- Bloque del sistema de navegación
- Bloque del sistema de guiado
- Bloque del sistema de control

El sistema de guiado se encarga de generar la trayectoria para alcanzar la referencia, este sistema es alimentado por el sistema de navegación y por los puntos de paso o puntos de referencia, dados por el usuario. La información generada por el sistema de guiado es transferida al sistema de control, quien se encarga de tratar de alcanzar la referencia deseada, por medio de la aplicación de leyes de control clásico, control no lineal, control inteligente, etc. que calculan y regulan la potencia necesaria para los motores y es alimentado por el sistema de navegación y el sistema de guiado [9].

El sistema de navegación de una boya marina es clave en la autonomía de la misma, este sistema conlleva al uso de diferentes tipos de sensores que permiten determinar la posición y orientación, de la boya. La aplicación de sensores inerciales (acelerómetros y girómetros) en la estimación de posición y orientación de sistemas de navegación, conocidos usualmente como sistemas de navegación inercial (INS) se ha incrementado en los últimos años debido a su volumen reducido y bajo costo [10]. Dos son las principales formas de obtener la estima de posición y orientación a partir de sensores inerciales [11], la primera corresponde a la estimación directa donde la señal del girómetro es integrada para obtener la orientación, y la señal del acelerómetro es doblemente integrada, además de estar vinculada a la medición del girómetro, para obtener la posición. Sin embargo, este primer método no es muy utilizado por la baja precisión de las estimaciones, debido a ciertas características de error que estos sensores presentan. El segundo método corresponde al uso de algoritmos de fusión de datos, con el fin de fusionar y complementar las mediciones de los sensores inerciales con las medidas de otros sensores para compensar y disminuir el error que caracteriza a los sensores inerciales y así obtener una estimación más precisa de la orientación y posición [10], [12].

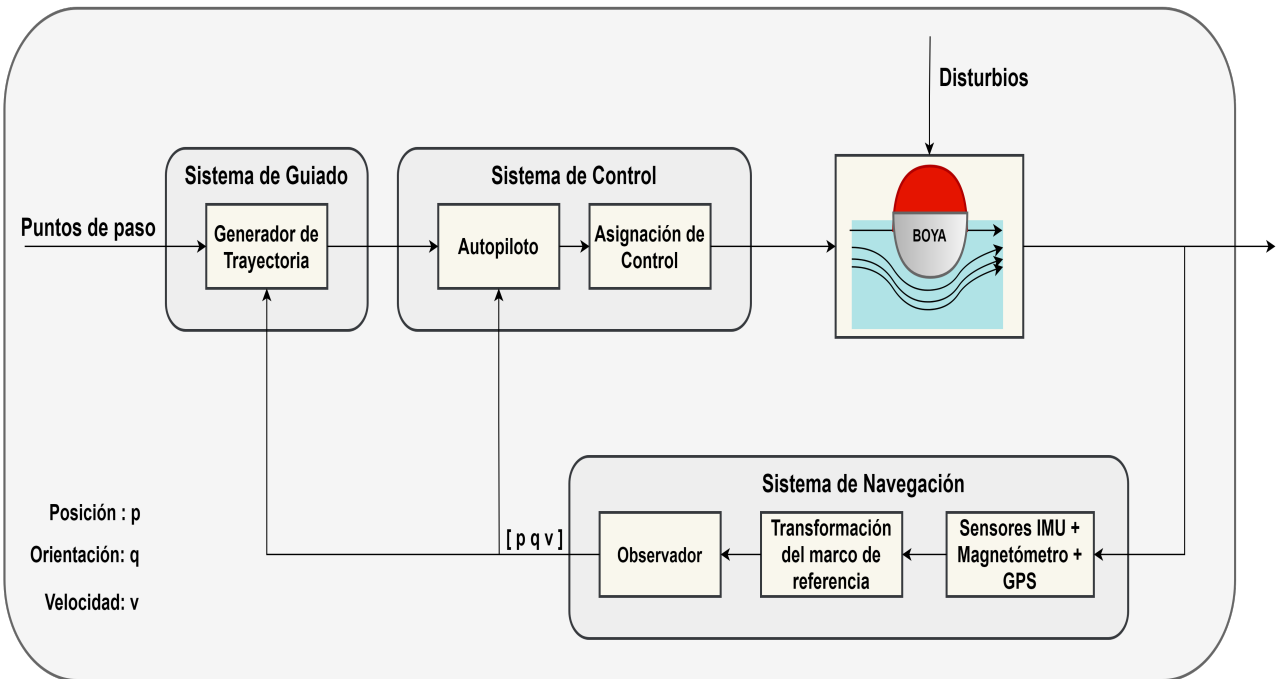


Figura 1.1: Sistema de guiado, control y navegación de un vehículo autónomo marino (basado en [1])

Entre los dispositivos más comunes para complementar a los sensores inerciales se encuentran los magnetómetros y el GPS (Global Positioning System). Los algoritmos de fusión son ampliamente utilizados en la actualidad en diversas aplicaciones de estimación de orientación y posición. Entre los algoritmos más utilizados están las diferentes variaciones del Filtro de Kalman, en especial el Filtro de Kalman Extendido (EKF) y los Observadores no lineales, estos últimos recientemente han tenido un mayor desarrollo e implementación en sistemas de navegación inercial, puesto que presentan buenas características de precisión y baja necesidad de cómputo embebido [13], [14]. En la Figura 1.2 se presenta un diagrama de bloques general de la implementación de un algoritmo basado en un observadores no lineal para la estimación de la orientación, utilizando sensores inerciales y magnetómetro.

Muchas otras metodologías de fusión de datos han sido desarrolladas en base a sensores inerciales. Por ejemplo en [8] los autores utilizan el algoritmo de Madwick para fusionar datos de sensores inerciales y GPS con el fin de estimar orientación y posición de la boya marina. En el trabajo reportado en [15] los autores diseñan un estimador basado en observadores no lineales utilizando sensores inerciales con el fin de estimar la orientación de un cuerpo rígido en tiempo real. Muy recientemente, se reporta en el trabajo [12], el desarrollo de estimadores (Observadores) no lineales tanto para la orientación como para la posición, de vehículos marinos usando una fusión de GPS, sensores inerciales y magnetómetros. Una de las mayores ventajas de utilizar Observadores no lineales, con respecto al EKF, (que es el algoritmo mayormente usado para la fusión de datos) es que es posible mostrar teóricamente algunas propiedades de convergencia, lo cual es imposible de hacer usando el EKF. Otro ejemplo de implementación de Observadores no lineales para fusión de datos de sensores inerciales, es el mostrado en [13], donde los autores diseñan un Attitude Heading Reference Systems (AHRS), basado en observadores no lineales y utilizando el método

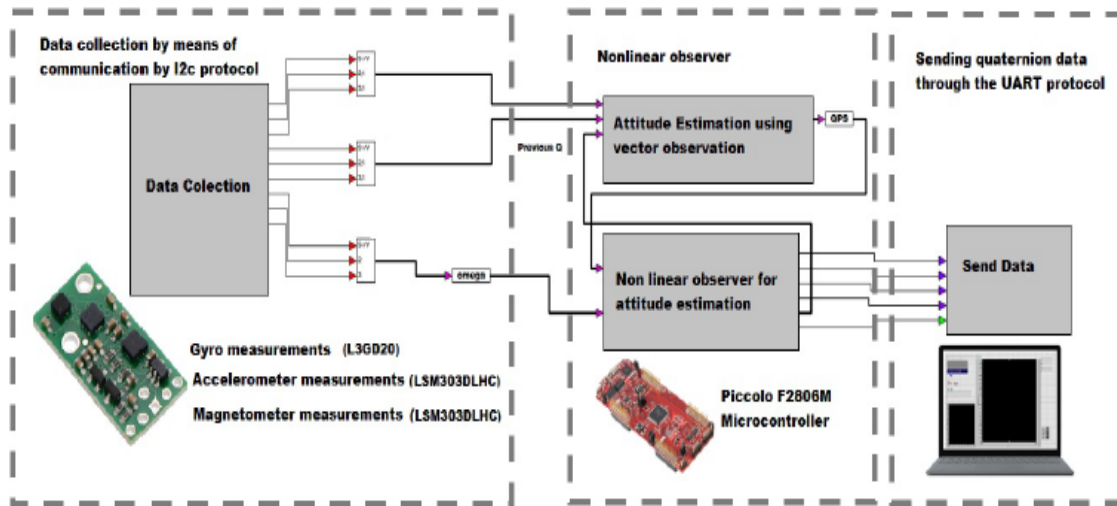


Figura 1.2: Diagrama a bloque de la implementación de un estimador de posición y orientación utilizando Observadores No Lineales. (Fuente:[13])

Kaczmarz, el cual tiene muy poca necesidad de cómputo embebido. Un estudio reportado recientemente en [16] indica que el uso de múltiples unidades de medición inercial y magnetómetros (MIMU), mejoran la precisión de la estima de posición y orientación, con el costo de requerir una mayor capacidad de cómputo embebido, lo que conlleva a un mayor tiempo de procesamiento. Sin embargo, en la investigación realizada en [16] los autores realizan pruebas paralelizando el algoritmo de fusión de los sensores, donde demuestran que el tiempo de procesamiento es reducido ampliamente.

La transmisión de toda la información generada por los sensores de la boya marina es crucial para su análisis. Actualmente la tecnología sobre redes de comunicaciones principalmente de telefonía celular, ha venido avanzando rápidamente, permitiendo una gran conexión entre usuarios casi en cualquier parte del mundo. El surgimiento del concepto internet de las cosas (IoT) ha habilitado nuevas tecnologías para la intercomunicación ya no solo entre usuarios de las diferentes compañías telefónicas, ahora también es posible la comunicación inalámbrica entre objetos, como por ejemplo entre redes de sensores inalámbricos (WSN, Wireless Sensor Networks), lo que presenta una gran ventaja a la hora de monitorear un sistema de medición ya que por medio de la utilización del IoT, el monitoreo puede realizarse de manera remota desde un servidor web o cualquier otra aplicación que permita visualizar los datos transmitidos por los sensores o por el propio procesador.

El enfoque principal del proyecto al cual pertenece este trabajo de tesis, consiste en desarrollar una boya marina autónoma para el monitoreo de variables en el ambiente marino, donde las variables medidas son transmitidas a una estación base en tierra, haciendo uso de algún sistema de comunicación inalámbrico. Sin embargo considerando los puntos antes mencionados, en este trabajo de tesis se realizó la instrumentación electrónica de un prototipo de boya marina, abordando principalmente la problemática del diseño del sistema de navegación y algoritmos de fusión de datos, como se observa en la Figura 1.3 que permitan a la boya navegar de manera autónoma cerca de la costa (10-15 metros de profundidad) y en aguas someras con el fin de medir, estimar y mapear variables oceanográficas.

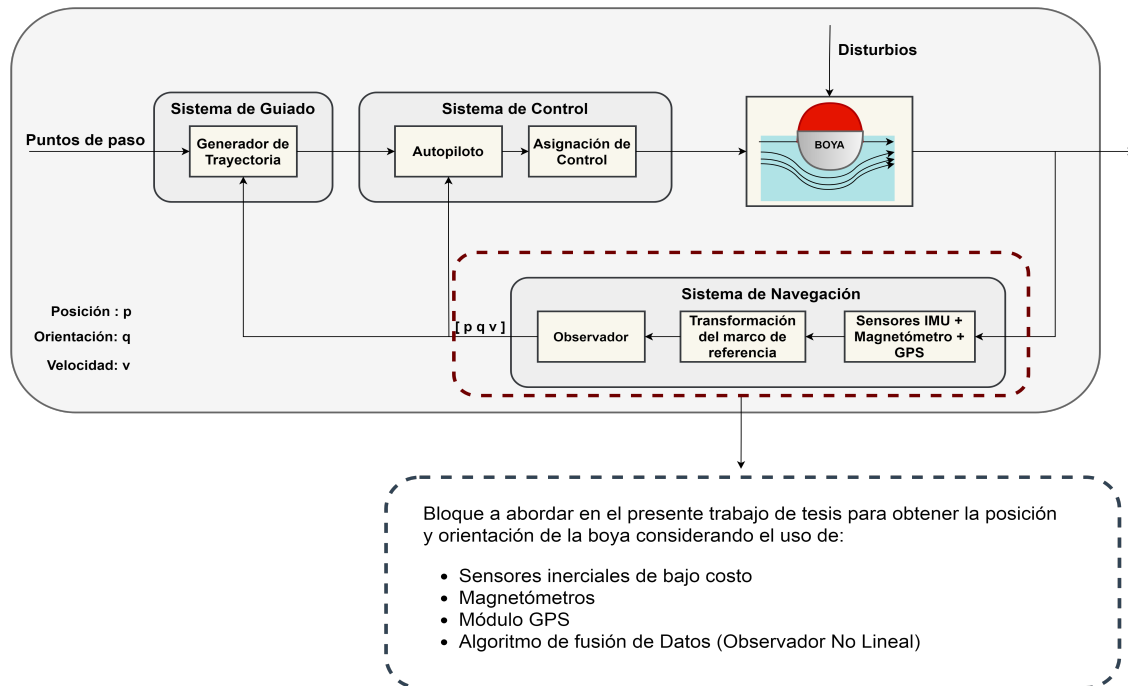


Figura 1.3: Problemática a abordar: sistema de navegación de una boya

1.2. Objetivos

1.2.1. Objetivo general

Implementar la instrumentación electrónica para el sistema de navegación de un prototipo de boya marina compacta, económica, autopropulsada y energéticamente autónoma que le permita navegar de manera automática cerca de la costa, con el fin de medir, estimar y transmitir variables oceanográficas en tiempo real a un centro de monitoreo.

1.2.2. Objetivos específicos

- Desarrollar un algoritmo de fusión de datos que permita estimar la orientación y posición de la boya, utilizando al menos 3 IMU (del inglés, Inertial Measurement Unit), 3 magnetómetros y receptor GPS.
- Implementar el algoritmo de fusión de datos desarrollado, utilizando un sistema de cómputo embebido y los sensores correspondientes.
- Validar el correcto funcionamiento de la boya, por medio de pruebas experimentales que permitan reconstruir la disposición espacial de la boya.

1.3. Justificación

Desde que el mar ha fungido como un puente entre naciones y ha permitido la navegación de grandes y pequeñas embarcaciones con diferentes fines como pueden ser: transporte público, embarcaciones de pesca, operaciones militares, investigaciones científicas, etc. y debido a que un porcentaje significativo de la población mundial vive en zonas costeras [6] el monitoreo de la superficie marina ha sido de gran relevancia. Además de permitir el estudio y análisis del cambio climático por medio de la medición de la temperatura de la superficie del mar (SST) y su salinidad principalmente, el monitoreo, registro y mapeo de diferentes variables oceanográficas permite realizar modelos matemáticos que ayudan a pronosticar el cambio climático o algún otro evento relacionado, como pueden ser huracanes o tsunamis y así evitar desastres que puedan perjudicar a la actividad humana e infraestructuras [6]. En especial para la detección de tsunamis el uso de boyas marinas se ha vuelto de gran relevancia en [17] se menciona la importancia que estas boyas fungen en la prevención de tsunamis a través de la medición de las características de las olas.

El monitoreo marino es esencial para científicos e ingenieros que buscan nuevas formas de generar energía eléctrica a partir de las corrientes marinas, oleaje marino y viento marino. Antes de realizarse un proyecto de este ámbito, estudios preliminares de viabilidad económica deben de ser efectuados con el fin de obtener un panorama general de la posible cantidad de energía que podría obtenerse en el área de estudio [5]. Esto permite conocer la viabilidad de implementar un sistema generador de energía en áreas específicas del entorno marino. La generación de energía a base del viento, o energía eólica comúnmente se lleva a cabo en zonas terrestres donde la cantidad de flujo de aire es suficientemente fuerte para mover los grandes rotores. Sin embargo, en las últimas décadas se ha planteado la conveniencia de trasladar los parques generadores de energía eólica al ámbito marino, donde el flujo de viento es un 30 % más alto que en tierra [5]. Trasladar este tipo de sistemas generadores de energía al ámbito marino ofrece las ventajas de una mayor generación de energía eléctrica, gracias a las pocas restricciones en cuanto a dimensiones del equipo generador de energía.

Diseñar un prototipo de boya marina compacta, de bajo costo, y que esta sea capaz de navegar de manera autónoma en ambientes desconocidos, presenta claras ventajas como por ejemplo el de conformar un sistema distribuido de boyas marinas que permitirán realizar estudios y caracterizar de manera más eficiente las costas mexicanas, con el fin de identificar las costas mexicanas con mayor potencial energético, estudiar los efectos del cambio climático en los mares mexicanos y establecer estrategias para la conservación del equilibrio ecológico de las zonas costeras. Conocer el comportamiento de la superficie marina a partir del análisis generado por medio de una red de sensores aporta grandes beneficios a muchos sectores de la población y la misma vida marina, ya que permite tomar mejores decisiones en ocasiones con anticipación lo que permitirá evitar alguna circunstancia o evento inesperado.

A partir de una serie de revisiones realizadas en [18], donde los autores mencionan que la fuerte inversión en sistemas de monitoreo ambiental acuático es una de las principales limitaciones en el desarrollo de estudios marinos, esto debido al alto costo de los dispositivos de monitoreo y al valor agregado que representan el mantenimiento de los mismos por parte de las empresas vendedoras. Aunado a que la mayoría de los sistemas de monitoreo marino son de tipo fijos y a que en México se han desarrollado muy pocos sistemas de monitoreo marino los cuales han sido de tipo fijos y de gran tamaño, es que en el presente proyecto se propuso realizar la instrumentación electrónica que forma parte del sistema de navegación de una boya marina autónoma de bajo costo, con

la finalidad de estimar su posición y orientación. Esperando también contribuir en el quehacer científico del país.

Debido a que se abordará como problemática principal la estimación de la posición y orientación de la boya, es deseable que este sistema de estimación sea lo más robusto y preciso con el fin de garantizar tolerancia ante fallos inesperados de alguno de los sistemas de sensado. Por consiguiente el uso de sensores inerciales y magnetómetros redundantes más las mediciones de un módulo GPS garantizarán un sistema de estimación de posición y orientación tolerante a fallas, además de mejorar la precisión de la estimación generada.

En el presente trabajo, se abordará el problema de la estimación de la posición y orientación de una boya marina, utilizando un enfoque de observación no lineal. Además, y a diferencia de los trabajos reportados, se busca utilizar el concepto de estimación colaborativa distribuida para la determinación de la orientación, mediante el uso de 3 unidades de medición (cada una con un acelerómetro, un magnetómetro y un girómetro) distribuidas espacialmente.

1.4. Descripción

El desarrollo de la tesis consistió en cuatro etapas principales que a su vez, dieron origen a los siguientes capítulos:

- **Capítulo 2** Describe los diferentes métodos de navegación. También se abordan los diferentes sistemas de referencia más utilizados en navegación, así como los sensores que permiten obtener la información de orientación, posición y velocidad de un móvil, con respecto a un sistema de referencia y haciendo uso de algoritmos computacionales. Finalmente, se introducen algunos trabajos relacionados para la estimación de la orientación.
- **Capítulo 3** Describe de manera general lo que es un sistema AHRS y como esta constituido. Se presentan el algoritmo LQS (*Learning Quaternion Sequence*) y el observador no lineal utilizados en este trabajo para la estimación de la orientación. Posteriormente, el procedimiento de calibración de acelerómetros, magnetómetros y girómetros es introducido. Finalmente se presentan una serie de resultados experimentales, haciendo uso de una MU y del software MATLAB/Simulink.
- **Capítulo 4** Se aborda la problemática de la estimación de la orientación de un cuerpo, de manera distribuida. Se presentan los sistemas de referencia involucrados en los algoritmos de estimación, así como la forma en que se relacionan haciendo uso de matrices de rotación y cuaterniones. Se describe también el planteamiento del problema para la estimación de la orientación distribuida, y se presenta el diseño del algoritmo de consenso dinámico desarrollado durante este trabajo de tesis, en este contexto se presenta un diagrama de bloques de los diferentes algoritmos que cada nodo de la red de sensores es capaz de ejecutar y la información que comparten con sus nodos vecinos. Además, a manera de simulación, se presenta una serie de escenarios de fallas en algunos sensores, con la finalidad de verificar que el sistema desarrollado es robusto ante la posible falla de alguno de los sensores. Finalmente, se presentan los resultados experimentales del sistema desarrollado para la estimación de la orientación distribuida. Las pruebas experimentales fueron desarrolladas durante una estancia de tres meses en el Laboratorio ICUBE y el INSA, en la ciudad de Estrasburgo, Francia. El sistema desarrollado fue colocado sobre un robot móvil, para estimar su orientación, y para verificar el buen funcionamiento del sistema, se obtuvo una orientación verdadera, de un sistema de alta precisión basado en cámaras Optitrack.
- **Capítulo 5** Describe las ventajas de integrar las medidas de un acelerómetro y un receptor GPS, para estimar la posición de un móvil, con respecto a un sistema de referencia fijo.

1. 1.4. DESCRIPCIÓN

También se describe la manera en que se pueden transformar de un sistema de coordenadas curvilíneo, que son las coordenadas que utiliza el GPS para dar sus mediciones, a un sistema de coordenadas cartesiano, que puede ser definido en un lugar conveniente a la aplicación. Finalmente se presentan resultados experimentales, donde las mediciones dadas por un receptor GPS son transformadas a un sistema de referencia fijo.

En la Figura 1.4 se describe la electrónica propuesta para el subsistema de Navegación, el cuál se encargara de estimar la orientación de la boya, haciendo uso de un algoritmo de fusión de datos y una unidad de medición (MU), formada por un acelerómetro, un magnetómetro y un girómetro y para obtener la posición se hace uso de un receptor GPS.

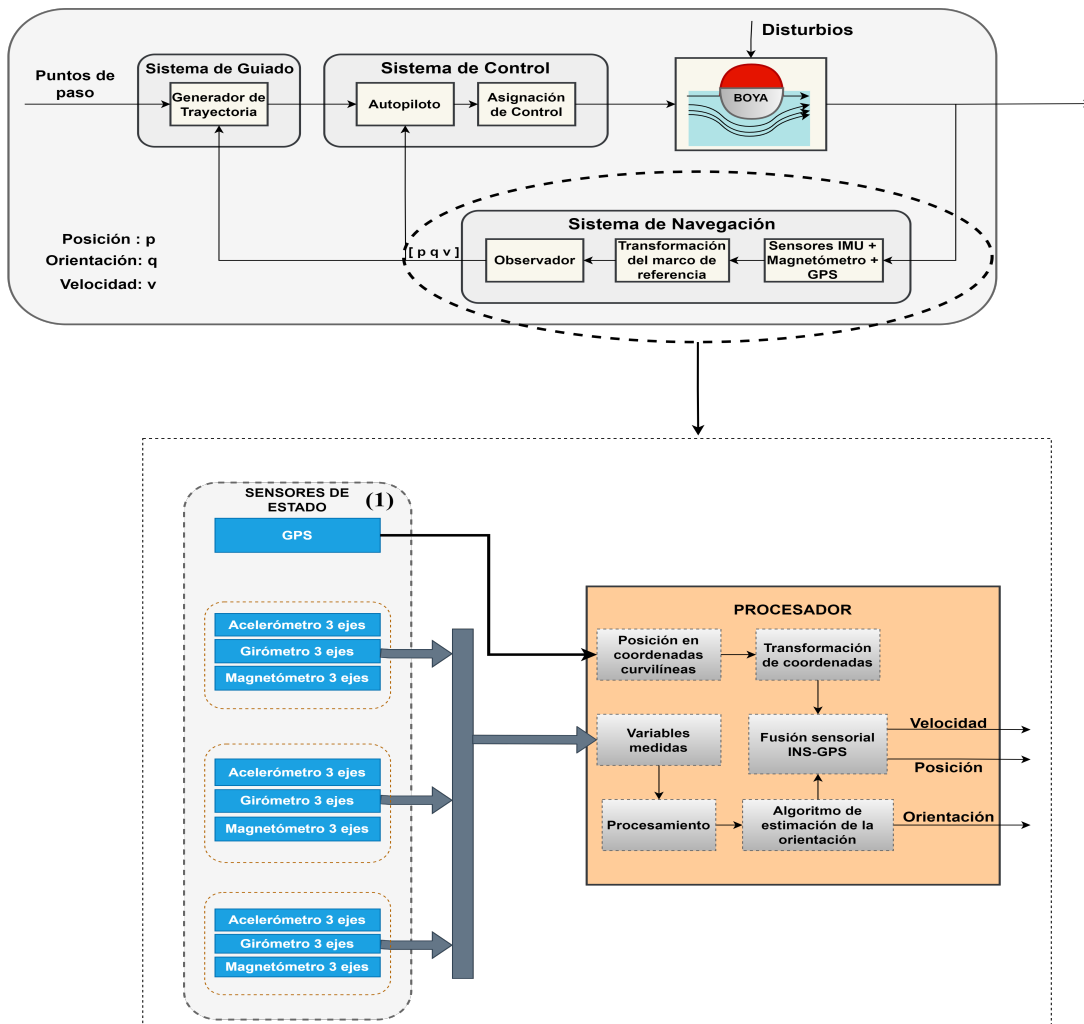


Figura 1.4: Diagrama a bloques propuesto para el subsistema de navegación de la boya.

Capítulo 2

Marco teórico

En la actualidad existe una amplia gama de aplicaciones que utilizan sistemas de navegación, que en su gran mayoría operan de manera autónoma como por ejemplo: aeronaves, autos, robots, vehículos subacuáticos, satélites, etc. (ver Figura 2.1). El sistema de navegación se encarga de obtener los llamados parámetros de navegación que consisten en la posición, orientación y velocidad del vehículo en cuestión, que lo definen en un espacio tridimensional con respecto a un sistema de referencia. El conocimiento de estos parámetros permite suministrar de información a otros sistemas que dotan de autonomía total al vehículo y le permiten navegar hacia un punto dado de referencia.

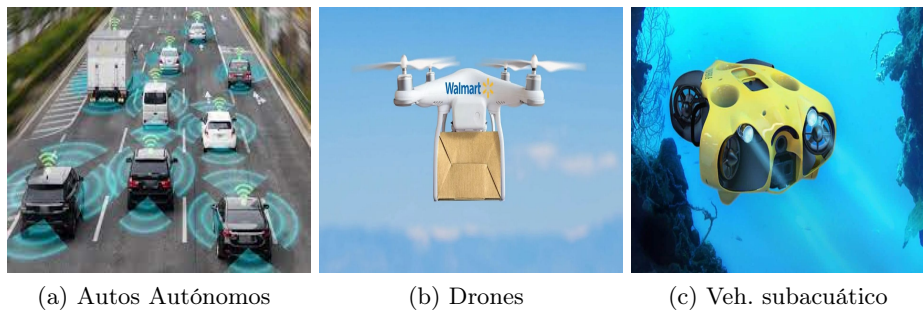


Figura 2.1: Vehículos que hacen uso de sistemas de navegación (fuente: google imágenes).

En la Figura 1.1 se presento el diagrama de bloques general de los sistemas que conforman a un vehículo autónomo marino, donde se resaltaba la importancia del sistema de navegación que conforma a todo vehículo autónomo.

Dado que en el presente proyecto se realizó el prototipo de la instrumentación electrónica del sistema de navegación de una boya marina (vehículo subacuático) en este capítulo se dará un panorama general de los diferentes tipos de sistemas de navegación que existen y son utilizados por los vehículos antes mencionados. Se abordará de manera general todo lo que conforma estos sistemas, como los diferentes sensores utilizados, sistemas de referencia, algoritmos de fusión de datos, modelos matemáticos de sensores, las diferentes representaciones de la orientación. Además, se presentará de manera sucinta, la teoría de grafos, la cual es utilizada para la propuesta de nuestro algoritmo de fusión de datos usando un conjunto de sensores, espacialmente distribuidos y que es lo que representa la innovación y contribución de nuestro trabajo al estado del arte y la técnica.

2.1. Métodos de navegación

Existen diversos métodos de navegación, los cuales van a depender de los principios físicos que permiten medir o estimar los parámetros de navegación, pueden destacarse dos grandes grupos: los métodos de extrapolación y los métodos de referenciamiento absoluto [19].

2.1.1. Método de extrapolación

En este grupo se especifica uno de los tipos de mediciones más utilizados a partir de las cuales se pueden obtener la posición, orientación y velocidad:

- Mediciones de tasa de cambio como velocidades y aceleraciones lineales o angulares.

La forma de obtener las diferentes variables de navegación es a través de la integración de las mediciones y para ello es necesario contar con una condición inicial. Los instrumentos característicos de este método para obtener las mediciones antes mencionadas son: sensores inerciales que serán abordados más adelante, odómetros, codificadores rotatorios, sensores de velocidad del aire, etc. Estos instrumentos tienen la peculiaridad de no depender de señales externas para efectuar sus mediciones por lo que son denominados introceptivos.

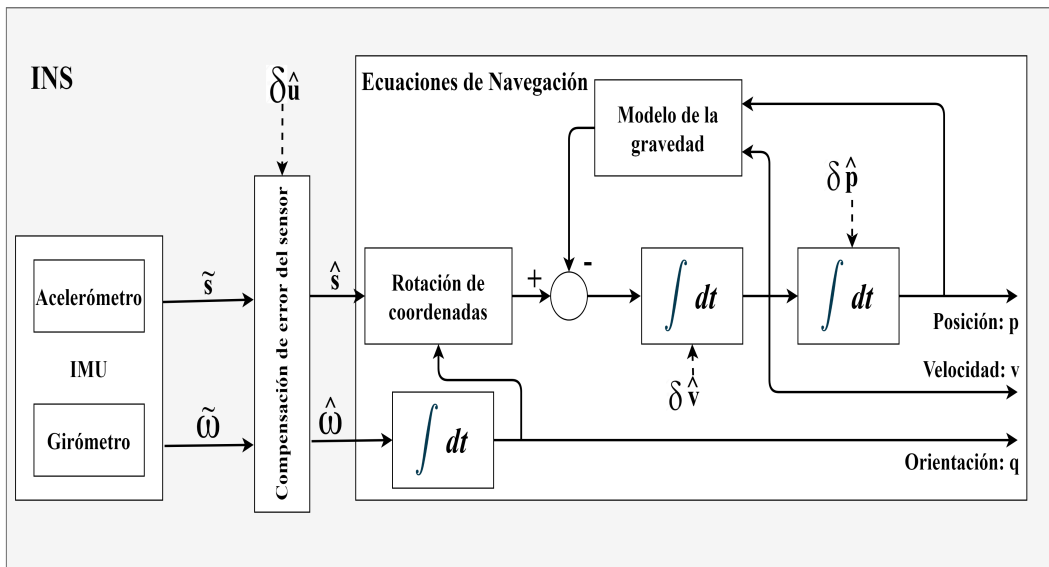


Figura 2.2: Obtención de los parámetros de navegación (Fuente:[20]).

Este método de navegación es susceptible a errores en la determinación de los parámetros de navegación, desde que estos se obtienen mediante la integración de las mediciones (ver Figura 2.2), es debido a este proceso de integración, que los errores ocasionados son lentamente variantes en el tiempo. Por lo que después de un tiempo suficientemente grande este método requiere de ser actualizado con mediciones absolutas de la posición o la orientación. Sin embargo como se mencionó anteriormente, este método no depende de señales externas, lo cual lo convierte en una herramienta muy útil en diversas aplicaciones, además de que pueden implementarse en diversos entornos y presentan una alta tasa de muestreo. Estas características han permitido que este método siga vigente, en especial, la navegación inercial, como se verá más adelante [19].

2.1.2. Método de referenciamiento absoluto

Este grupo se basa en diferentes principios en los que en cada uno se utilizan instrumentos para determinar de manera directa los parámetros de navegación (posición en base a coordenadas, orientación o velocidad) respecto a un sistema de referencia. Dado que los sensores utilizados en este método utilizan señales externas al vehículo, son denominados exoceptivos. Algunos de los principios empleados en este método son:

- Navegación satelital
- Navegación con mapa
- Rebote de señal (sonar, radar, lidar)
- Navegación por imágenes

2.1.3. Navegación inercial

Los vehículos con sistema de navegación inercial, forman parte del grupo que utilizan el método de navegación de extrapolación, es decir que no hacen uso de señales externas para la determinación de los parámetros de navegación. Se denomina navegación inercial ya que únicamente hace uso de sensores inerciales, los cuales se basan en principios físicos para obtener la posición, orientación y velocidad.

«La navegación inercial se sustenta en el siguiente principio básico de la cinemática: Conocidos en un instante inicial la velocidad, la orientación y la posición de un móvil así como los valores instantáneos presentes y futuros de su aceleración lineal y su velocidad angular relativas a un dado sistema de referencia, es posible calcular la posición, la velocidad y la orientación del vehículo en todo instante futuro» [19].

Los sensores por excelencia para estos sistemas de navegación, como se mencionó anteriormente son los inerciales, es decir, acelerómetros y girómetros que proporcionan información de aceleración lineal y velocidad angular respectivamente. A continuación se enumeran las ventajas y desventajas de este tipo de navegación.

Ventajas [19]:

1. Tienen una alta tasa de muestreo.
2. No requiere del uso de señales externas para generar sus mediciones.
3. Sus mediciones están siempre disponibles, independientemente del lugar donde el vehículo se encuentre operando.

Desventajas [19]:

1. Se necesita conocer las condiciones iniciales del vehículo.
2. Susceptible a errores que crecerán lentamente con el tiempo.
3. Para eliminar los errores inducidos en estos sistemas, se requiere la adición de otros sensores como GPS y magnetómetro.

2.1.4. Navegación integrada

Con el avance y mejora de las características en cuanto a tecnología de dispositivos micro-controladores, microprocesadores, sensores, en especial sensores inerciales de tecnología MEMS, sistemas de comunicación, etc. Los cuales han logrado mayor capacidad de procesamiento y cómputo embebido, bajo consumo de energía, volumen reducido y bajo costo, y debido a la creciente demanda de vehículos autónomos se dio paso al desarrollo de un nuevo enfoque de sistema de navegación. Este consiste en la navegación multi-sensor o navegación integrada, la cual busca usar las ventajas de los sensores inerciales y sensores de medición absoluta como GPS, magnetómetros, radar, etc. Para obtener los parámetros de navegación con la mayor exactitud posible a una alta

tasa de muestreo, lo que permitiría su implementación en el control de vehículos en tiempo real.

El uso complementario de sensores inerciales y sensores de medición absoluta como GPS y magnetómetro, para la obtención de posición, orientación y velocidad requiere de la fusión de las mediciones de cada uno de estos sensores. Esto es, se deben de combinar las diferentes mediciones obtenidas de cada sensor mediante el uso de algoritmos de fusión de datos (los cuales serán abordados más adelante). El uso de algoritmos de fusión de datos permite la estimación de los parámetros de navegación con una gran mejora en la exactitud y a una tasa de muestreo bastante aceptable, eliminando o reduciendo también los errores presentados en la navegación inercial pura, y mejorando la tasa de muestreo que utilizan los sistemas de navegación con GPS o navegación satelital (GNSS).

Dada las desventajas que presentan los sistemas de navegación inerciales, estos en la actualidad ya no se ocupan solos, la navegación integrada ha venido a mejorar al sistema de navegación inercial, y busca mejorar también a los sistemas de navegación de referenciamiento absoluto que dependen de señales externas para la obtención de los parámetros de navegación.

En el presente proyecto de tesis, una baja tasa de errores en los parámetros de navegación y una baja dependencia de señales externas son requeridas. Dadas las ventajas que presenta el sistema de navegación integrada, al utilizar características de la navegación inercial y navegación absoluta para mejorar la exactitud de los parámetros de navegación y eliminar errores producidos por los sensores, es que se ha elegido utilizar un sistema de navegación integrada para la boya marina. En la sección 2.2, se introducirá el concepto de sistemas de referencia, y los diferentes sistemas de referencia con los cuales se estarán trabajando durante la implementación del sistema de navegación.

2.2. Sistemas de referencia

En esta sección se presenta una herramienta muy importante para todo sistema de navegación, esta herramienta muy conocida son los sistemas de referencia, gracias a los sistemas de referencia es posible expresar la posición y orientación con respecto a una referencia. Por ejemplo supongamos que se requiere saber la posición de una persona dentro de una casa, en todo instante de tiempo, para obtener esta información, se define un sistema de referencia con origen en el centro de la casa, esto posibilita la descripción del desplazamiento realizado por toda la casa con respecto a la referencia seleccionada. Los sistemas de referencia facilitan la descripción del movimiento de un objeto, así como también definir su posición y orientación con respecto a un conjunto de ejes (normalmente tres: x,y,z) ortogonales entre sí. Los sistemas de navegación involucran al menos 3 sistemas de referencia [21], sin embargo dos de ellos pueden ser iguales, por lo que es usual encontrar únicamente dos sistemas de referencia involucrados en la descripción. A continuación se hace mención de los tres sistemas de referencia presentes en la mayoría de los sistemas de navegación [21],

- El sistema de referencia del cuerpo, que es donde el objeto esta expresado, conocido como el *object frame*.
- El sistema de referencia con respecto al cual el movimiento es realizado conocido como el *reference frame*.
- El sistema de referencia en el cual el movimiento es expresado, conocido como *resolving frame*.

Es necesario realizar la descripción de un cuerpo rígido al menos con respecto a dos sistemas de referencia (ver Figura 2.3). Del ejemplo anterior, donde interesa expresar el desplazamiento de una persona dentro de una casa, se escogieron el sistema de referencia del cuerpo (persona) y el

sistema de referencia fijo (centro de la casa, reference frame), que es con respecto al cual se realiza el desplazamiento, para expresar el movimiento, si ambos sistemas fueran iguales no se podría describir movimiento alguno ya que este no existiría.

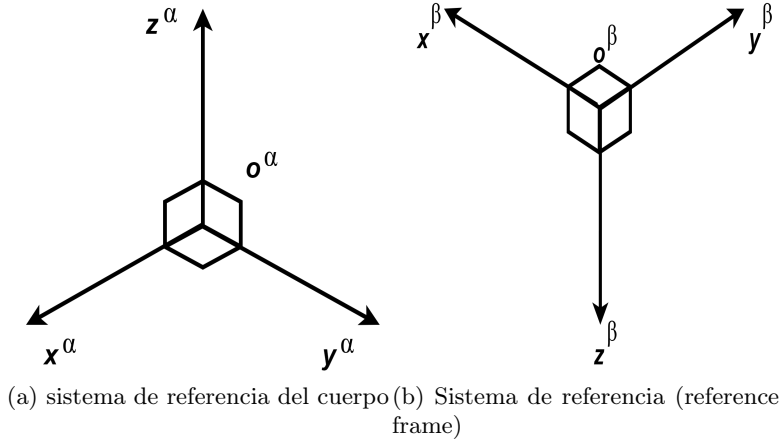


Figura 2.3: sistemas de referencia mínimos a utilizar en un sistema de navegación (Fuente: [21])

Para describir la posición, velocidad, aceleración y velocidad angular, es necesario indicar los tres sistemas de referencia involucrados en la descripción de cada una de estas cantidades, para este propósito se adopta la notación usada en [21].

$$X_{\beta\alpha}^{\gamma} \quad (2.1)$$

Esta notación describe la relación del sistema α , donde se encuentra el punto de interés, el sistema β , que es el sistema de referencia y el sistema γ , que es el sistema de solución. A continuación se describen de manera general los sistemas de referencia más utilizados en navegación.

2.2.1. Sistema de referencia inercial

Este sistema ampliamente conocido como *ECI* (del inglés: *Earth Centered Inertial Frame*). Este es un sistema de coordenadas inerciales, que carece de aceleración, y donde las leyes de movimiento de Newton pueden ser aplicados. Es denotado ampliamente por el símbolo $\{i\}$, su origen se encuentra ubicado en el centro de masas de la tierra, y los ejes x e y se encuentran contenidos en el plano ecuatorial y el eje z es coincidente con el eje de rotación de la tierra. Debido a que en el presente trabajo de tesis se pretende diseñar un sistema de navegación integrado, el sistema de referencia inercial es importante ya que este sistema de navegación involucra el uso de sensores inerciales, donde las aceleraciones y velocidades angulares son referenciadas respecto a un sistema inercial [21], [19].

2.2.2. Sistema de referencia terrestre

Este sistema de referencia conocido en la literatura inglesa como *Earth Centered-Earth Fixed* (*EC-EF*), para el presente trabajo será denotado como $\{\mathbf{E}^{EC-EF}\}$, tiene su origen en el centro de masas de la tierra, donde el eje z es coincidente con el eje de rotación de la tierra, el eje x pasa por el punto de latitud y longitud $(0, 0)$ [22] y el eje y completa el sistema de coordenadas

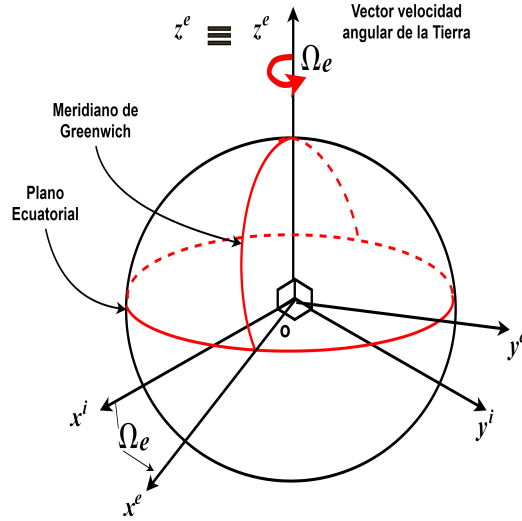


Figura 2.4: Coordenadas inerciales y terrestres (Fuente: [19])

cartesiana, aplicando la regla de la mano derecha. La diferencia entre la terna $EC-EF$ y la ECI , es que en la primera los ejes permanecen fijos en un punto de la tierra, es decir rotan junto con la tierra a una velocidad constante [21], [22].

Ya que este sistema de referencia rota junto con la tierra, existe un punto donde los ejes x e y del $EC-EF$ y ECI son coincidentes, por lo que se hace necesario especificar el tiempo en el que esto ocurre [19]. Una manera de realizar esto, es alineando los dos sistemas de referencia en el instante inicial de navegación [21], [22], en la Figura 2.4, se muestran los sistemas de referencia inercial y terrestres, coincidentes.

El $EC-EF$ es muy importante en todo sistema de navegación, ya que normalmente se requiere conocer la posición relativa del vehículo a la tierra, por lo que este sistema suele ser utilizado como sistema de solución [21].

2.2.3. Sistema de referencia geográfico

Es denotado por el símbolo $\{n\}$, tiene su origen en el punto donde se busca una solución de navegación, como por ejemplo el centro de masas del vehículo [21]. El eje z apunta hacia el centro de la tierra, es conocido como $DOWN$ (D), el eje x , es ortogonal al eje z , apuntando al norte verdadero [23] y es conocido como $NORTH$ (N) y completando la terna el eje y , apunta en dirección este, por lo que suele conocerse como $EAST$ (E). El sistema de navegación geográfico NED , es ampliamente utilizado para conocer la orientación del vehículo [21] y para posición y velocidad suele emplearse como sistema de solución [21]. En esta tesis se estará trabajando con este sistema de coordenadas, el cual se tomará como inercial debido a la navegación local de la aplicación y se denotará como \mathbf{E}^f , con componentes $\{e_1^f, e_2^f, e_3^f\}$.

2.2.4. Sistema de referencia del cuerpo

El sistema del cuerpo (objeto), denotado en la literatura por el símbolo $\{b\}$ suele tener su origen en el centro de masas del vehículo, suele también hacer coincidente el origen de esta terna con la del sistema de referencia geográfico, donde se busca tener información de posición y velocidad

[21],[19]. El eje x , suele asociarse con la dirección de desplazamiento del vehículo, el eje z se asocia a la dirección de la gravedad y el eje y completa la terna de la mano derecha (ver Figura 2.5). Muchos autores suelen denominar al eje x , como *roll*, el eje y como *pitch* y eje z como *yaw*, estos para representar movimiento rotacional [21].

Todo objeto que tenga asociado un sistema de navegación deberá tener asociado un sistema de referencia del cuerpo, para describir sus parámetros de navegación. En este trabajo de tesis, el sistema de referencia del cuerpo será denotado como \mathbf{E}^b , con componentes $\{e_1^b e_2^b e_3^b\}$.

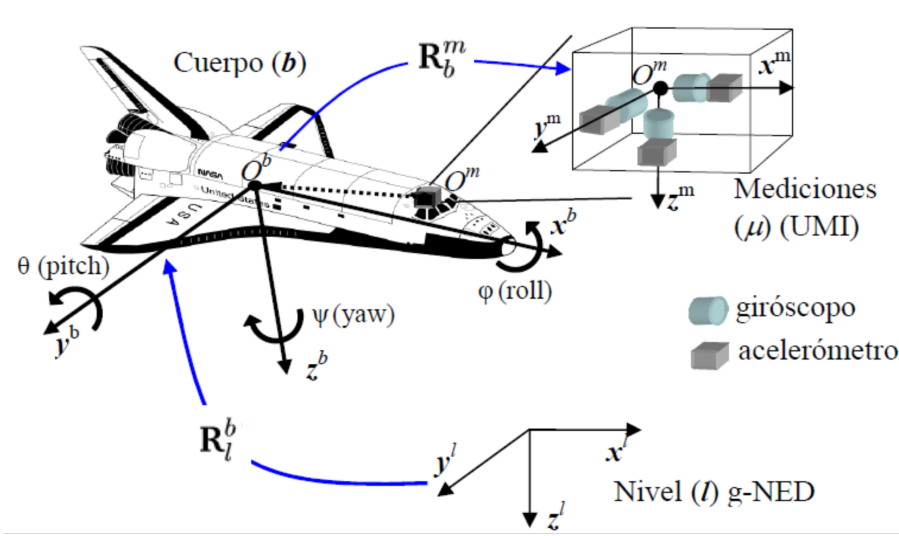


Figura 2.5: Coordenadas del cuerpo y de navegación (Fuente: [19])

Es importante conocer los diferentes tipos de sistemas de referencia, ya que para realizar la integración de mediciones de un GPS y sensores inerciales, se debe de situar ambas mediciones en un sistema de referencia en común con la finalidad de hacer posible tal integración de mediciones [24].

2.3. Parametrizaciones de la orientación

Entre los objetivos de todo sistema de navegación, se encuentra el proveer en todo instante información de la orientación del vehículo. Existen diversas formas de representar matemáticamente la orientación del vehículo en cuestión. En la literatura, la mayoría de los autores denominan a estas formas de representación de la orientación como "Parametrizaciones de la orientación", la cual consiste en un conjunto de valores numéricos que son actualizados permanentemente por el algoritmo de navegación [19].

Los diferentes métodos de parametrizar la orientación tienen un papel importante en navegación, ya que llevan a cabo dos funciones básicas e importantes: 1.-) Describen la orientación de un sistema de coordenada con respecto a otro, 2.-) proporcionan un medio para transformar un vector de un conjunto de ejes de resolución a otro [21]. A continuación se realiza un breve pero importante estudio de las principales formas de representar la orientación.

2.3.1. Matriz de rotación

La matriz de rotación, pertenece a $\mathbb{R}^{3 \times 3}$, es denotada como \mathbf{R}_a^b , que indica la rotación de un sistema de coordenadas $\{a\}$, a un sistema de coordenadas $\{b\}$, por lo que en el presente trabajo

de tesis se adopta la siguiente notación para representar la matriz de rotación [25], [26]:

$$\mathbf{R}_{from}^{to}$$

Dados los siguientes dos vectores,

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \text{ y } \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

donde a y b son vectores unitarios definidos en los sistemas $\{a\}$ y $\{b\}$ respectivamente, se tiene que:

$$\begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \mathbf{R}_a^b \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \quad (2.2)$$

Donde,

$$\mathbf{R}_a^b = \begin{pmatrix} (a_1 \cdot b_1) & (a_1 \cdot b_2) & (a_1 \cdot b_3) \\ (a_2 \cdot b_1) & (a_2 \cdot b_2) & (a_2 \cdot b_3) \\ (a_3 \cdot b_1) & (a_3 \cdot b_2) & (a_3 \cdot b_3) \end{pmatrix} \quad (2.3)$$

Sin embargo del cálculo vectorial, se tiene que el producto punto de dos vectores unitarios, define el coseno del ángulo entre estos dos vectores, de la siguiente manera,

$$\cos(\theta_{ab}) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}||\vec{b}|} \quad (2.4)$$

por lo tanto la matriz de rotación, puede ser definida como se muestra a continuación,

$$\mathbf{R}_a^b = \begin{pmatrix} \cos(\theta_{a1b1}) & \cos(\theta_{a1b2}) & \cos(\theta_{a1b3}) \\ \cos(\theta_{a2b1}) & \cos(\theta_{a2b2}) & \cos(\theta_{a2b3}) \\ \cos(\theta_{a3b1}) & \cos(\theta_{a3b2}) & \cos(\theta_{a3b3}) \end{pmatrix} \quad (2.5)$$

La representación dada en (2.5), es llamada matriz de cosenos directivos. Esta matriz es muy útil al momento de describir la rotación del sistema de referencia del cuerpo o móvil \mathbf{E}^b , con respecto a un sistema de referencia fijo o inercial \mathbf{E}^f . La matriz de rotación que lleva del sistema de coordenadas fijo al sistema de coordenadas del cuerpo, esta dada por:

$$\mathbf{R}_f^b = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & -\sin(\theta) & \cos(\theta) \end{pmatrix} \quad (2.6)$$

Las matrices de rotación pertenecen al grupo especial ortogonal $SO(3)$, donde

$$SO(3) = \{R | R \in \mathbb{R}^{3 \times 3}, RR^T = I, \det R = +1\} \quad (2.7)$$

y tienen las siguientes propiedades,

- $(\mathbf{R}_a^b)^{-1} = (\mathbf{R}_a^b)^T = (\mathbf{R}_b^a)$
- $\mathbf{R}_c^a = \mathbf{R}_b^a \mathbf{R}_c^b$ composición de rotaciones.
- $(\mathbf{R}_a^b)^T (\mathbf{R}_a^b) = (\mathbf{R}_a^b) (\mathbf{R}_a^b)^T = \mathbf{I}$
- $\det(\mathbf{R}_a^b) = 1$

- El conjunto de todas las rotaciones es un Grupo de Lie, $SO(3)$

Otra representación de la matriz de rotación poco abordada en la literatura, consiste en la representación exponencial de \mathbf{R} , en la cual se relacionan el eje sobre el cual se realiza la rotación y el ángulo de rotación, de esta manera la matriz de rotación \mathbf{R} queda en función de un eje $\vec{\lambda} \in \mathbb{R}^3$, el cual debe de ser un vector de norma 1, y en función de un ángulo $\beta \in \mathbb{R}$ el cual esta dado en radianes. La fórmula que relaciona la matriz de rotación \mathbf{R} con el eje de rotación $\vec{\lambda}$ y el ángulo de giro β esta dada por,

$$e^{[\vec{\lambda}^\times]\beta} = \mathbf{I} + [\vec{\lambda}^\times] \sin \beta + [\vec{\lambda}^\times]^2(1 - \cos \beta) \quad (2.8)$$

La ecuación (2.8) es conocida como la *Fórmula de Rodrigues*, donde $e^{[\vec{\lambda}^\times]\beta}$ es la matriz de rotación y $[\vec{\lambda}^\times]$ es la matriz antisimétrica asociada al vector $\vec{\lambda}$, en ingles llamada *skew matrix*, definida en (2.9). Esta forma de parametrizar la orientación es llamada representación ángulo-eje equivalente.

$$[\vec{\lambda}^\times] = \begin{pmatrix} 0 & -\lambda_3 & \lambda_2 \\ \lambda_3 & 0 & -\lambda_1 \\ -\lambda_2 & \lambda_1 & 0 \end{pmatrix} \quad (2.9)$$

En general para cualquier vector $\vec{\xi} \in \mathbb{R}^3$, esto es $\xi = [\xi_1 \ \xi_2 \ \xi_3]^T$, se puede definir su matriz antisimétrica asociada como se muestra a continuación,

$$[\vec{\xi}^\times] = \begin{pmatrix} 0 & -\xi_3 & \xi_2 \\ \xi_3 & 0 & -\xi_1 \\ -\xi_2 & \xi_1 & 0 \end{pmatrix} \quad (2.10)$$

La matriz antisimétrica ofrece una manera sencilla de realizar el producto cruz de dos vectores, sean los vectores \vec{a} y \vec{b} que pertenecen a \mathbb{R}^3 , su producto cruz esta dado por,

$$\vec{a} \times \vec{b} = \begin{pmatrix} a_2b_3 - a_3b_2 \\ a_3b_1 - a_1b_3 \\ a_1b_2 - a_2b_1 \end{pmatrix} = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = [\vec{a}^\times]\vec{b} \quad (2.11)$$

La ecuación (2.11) ofrece una manera eficiente de visualizar el producto cruz de dos vectores, a partir del producto de la matriz antisimétrica del primer vector por el segundo vector.

2.3.2. Ángulos de Euler

Los ángulos de Euler son una de las formas mayormente empleadas para parametrizar la orientación en navegación. Esta forma de representar la orientación, es utilizada también para de definir una transformación de coordenadas de un sistema de referencia con respecto a otro, donde uno de ellos es fijo, realizando una secuencia de tres rotaciones angulares, en una secuencia bien definida [21],[1].

Existen 12 formas de realizar la secuencia de rotaciones para definir la orientación, por medio de los ángulos de Euler [27],[1], sin embargo la más empleada es cuando se aplican rotaciones sobre los ejes de un sistema fijo. Si se tiene una terna $(x - y - z)$, se define la siguiente secuencia de rotación (ver Figura 2.6):

1. Se rota un ángulo ψ sobre el eje z
2. Se rota un ángulo θ sobre el eje y
3. Se rota un ángulo ϕ sobre el eje x

Donde ψ es denominado *yaw* o guiñada, θ es denomiando *pitch* y ϕ es denominado *roll*, como se observa en la Figura 2.7. A pesar de que esta forma de parametrización de la orientación, es una de las más intuitivas, presenta algunos inconvenientes que hacen que su uso en la parametrización de la orientación, se vea limitada y se prefiera utilizar alguna de las otras parametrizaciones [10],[21],[26]. La principal singularidad que caracteriza a esta parametrización, es cuando el primer y último ejes son coincidentes, esto se presenta cuando se da una rotación de $\pm 90^\circ$ en el ángulo θ [19]. Este fenómeno provoca la pérdida de un grado de libertad en las rotaciones, en la literatura inglesa, es conocido como *gimbal lock*. A continuación se presentan las matrices de transformación de coordenadas para la rotación de ángulos presentada anteriormente.

$$\begin{pmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{pmatrix} \quad (2.12)$$

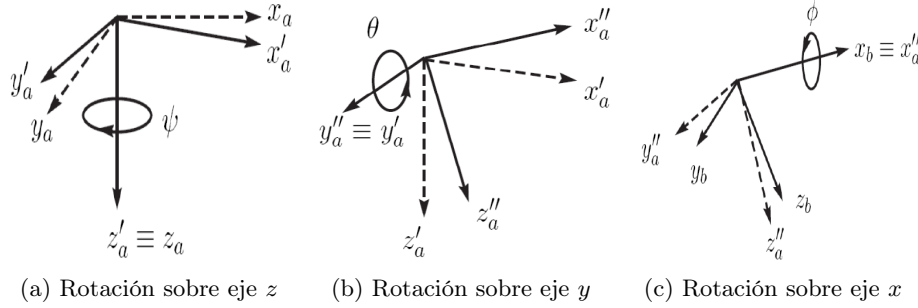


Figura 2.6: Rotaciones de ángulos Euler (fuente: [27])

• Razón de cambio de los ángulos Euler en función de la velocidad angular

Al igual que con las matrices de rotación, contar con una representación que relacione al vector de velocidad angular $\vec{\omega} = [\omega_1 \ \omega_2 \ \omega_3]^T$ con los ángulos de Euler, es de gran importancia en la práctica, ya que a partir de las mediciones de girómetros se podría obtener una aproximación de los ángulos de Euler en función del vector de velocidad angular medido. Esta relación viene dada por,

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \tan(\theta) \sin(\phi) & \tan(\theta) \cos(\phi) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \frac{\sin(\phi)}{\cos(\theta)} & \frac{\cos(\phi)}{\cos(\theta)} \end{pmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} \quad (2.13)$$

2.3.3. Cuaterniones

En temas relacionados a robótica, navegación, computación gráfica, es bien conocido el término cuaternión, el cual consiste en número hipercomplejo de rango 4, descubierto por el físico y matemático Sir William Rowan Hamilton (1805 – 1865) [26].

Un cuaternión se define de la siguiente manera:

$$\mathbf{Q} = q_0 + \vec{q} = q_0 + \mathbf{i}q_1 + \mathbf{j}q_2 + \mathbf{k}q_3 \quad (2.14)$$

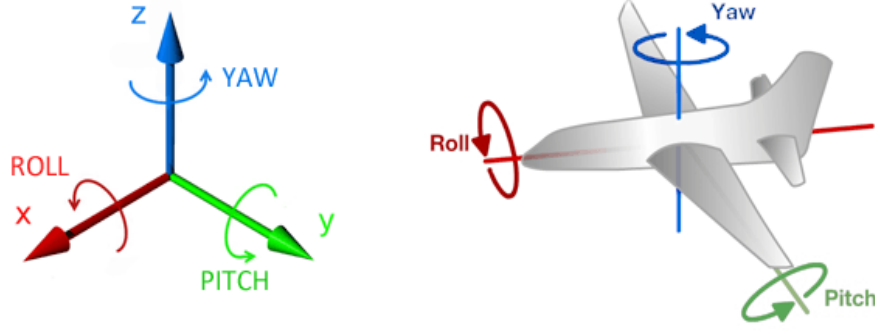


Figura 2.7: Representación visual de los ángulos de Euler (fuente: <https://www.luisllamas.es/medir-la-inclinacion-imu-arduino-filtro-complementario/>).

Donde q_0 es la parte escalar del cuaternión y pertenece a \mathbb{R} , \vec{q} es la parte vectorial del cuaternión, y q_1, q_2, q_3 también pertenecen a \mathbb{R} . De manera vectorial se tiene, $\mathbf{Q} = [q_0 \ q_1 \ q_2 \ q_3]^T$. En el campo de la ingeniería, como se menciono al inicio de esta subsección, los cuaternios son ampliamente utilizados para representar la orientación de un cuerpo rígido [28], además de que esta representación en \mathbb{R}^4 evita el *gimbal lock* de los ángulos de Euler [29]. A continuación se definen las operaciones matemáticas básicas con cuaternios.

Se definen el siguiente par de cuaternios,

$$\mathbf{Q}_1 = q_{10} + \mathbf{i}q_{11} + \mathbf{j}q_{12} + \mathbf{k}q_{13}$$

y

$$\mathbf{Q}_2 = q_{20} + \mathbf{i}q_{21} + \mathbf{j}q_{22} + \mathbf{k}q_{23}$$

• Suma

La suma de dos cuaternios, se realiza al igual que en el álgebra compleja, elemento a elemento de cada cuaternio, como se indica a continuación,

$$\begin{aligned} \mathbf{Q}_1 + \mathbf{Q}_2 &= (q_{10} + q_{20}) + \mathbf{i}(q_{11} + q_{21}) + \mathbf{j}(q_{12} + q_{22}) + \mathbf{k}(q_{13} + q_{23}) \\ \mathbf{Q}_1 + \mathbf{Q}_2 &= [q_{10} + q_{20} \quad q_{11} + q_{21} \quad q_{12} + q_{22} \quad q_{13} + q_{23}]^T \end{aligned} \quad (2.15)$$

• Multiplicación

El producto de dos cuaternios satisface los siguientes productos elementales,

$$\begin{aligned} \mathbf{i}^2 &= \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1 \\ \mathbf{ij} &= \mathbf{k} = -\mathbf{ji} \\ \mathbf{jk} &= \mathbf{i} = -\mathbf{kj} \\ \mathbf{ki} &= \mathbf{j} = -\mathbf{ik} \end{aligned} \quad (2.16)$$

De manera general, el producto de dos cuaternios se expresa como,

$$\mathbf{Q}_1 \otimes \mathbf{Q}_2 = \begin{pmatrix} q_{10} & -q_{11} & -q_{12} & -q_{13} \\ q_{11} & q_{10} & -q_{13} & q_{12} \\ q_{12} & q_{13} & q_{10} & -q_{11} \\ q_{13} & -q_{12} & q_{11} & q_{10} \end{pmatrix} \begin{pmatrix} q_{20} \\ q_{21} \\ q_{22} \\ q_{23} \end{pmatrix} \quad (2.17)$$

Denotando como q_{1_0} , \vec{q}_1 y q_{2_0} , \vec{q}_2 , la parte escalar y vectorial de cada correspondiente cuaternión, se tiene que el producto cuaternión puede ser expresado de la siguiente manera [13], [15]:

$$\mathbf{Q}_1 \otimes \mathbf{Q}_2 = \begin{pmatrix} q_{1_0} & -\vec{q}_1^T \\ \vec{q}_1 & I_3 q_{1_0} + [\vec{q}_1^\times] \end{pmatrix} \begin{pmatrix} q_{2_0} \\ \vec{q}_2 \end{pmatrix} \quad (2.18)$$

Donde el resultado es otro cuaternión y el producto de cuaterniones no es conmutativo.

- **Conjugado complejo**

El conjugado complejo de un cuaternión se define de la siguiente manera,

$$\mathbf{Q}^{-1} = [q_0 \quad -q_1 \quad -q_2 \quad -q_3]^T \quad (2.19)$$

- **Representación de rotaciones**

La representación de una secuencia de rotaciones, empleando cuaterniones, es efectuada mediante rotaciones simples, es decir, utilizando un vector unitario que exprese el eje y ángulo de rotación [28]. Es sabido que los cuaterniones pertenecen a \mathbb{R}^4 , en consecuencia para representar un vector de \mathbb{R}^3 , como un cuaternión, se emplean los llamados cuaterniones puros, que se definen de la siguiente manera,

$$\mathbf{Q}(\mathbf{v}) = [0, \mathbf{v}^T]^T \quad (2.20)$$

Donde el escalar es 0 y solamente queda la representación del cuaternión puro en tres dimensiones.

De esta manera una rotación de \mathbf{w}_a a \mathbf{v}_b , se representa de la siguiente manera [28],[30]

$$\mathbf{Q}(\mathbf{v}_b) = \mathbf{Q}_a^b \otimes \mathbf{Q}(\mathbf{w}_a) \otimes (\mathbf{Q}_a^b)^* \quad (2.21)$$

Donde $\mathbf{Q}(\mathbf{v})$ y $\mathbf{Q}(\mathbf{w})$, son cuaterniones puros. Similarmente se define la rotación inversa de \mathbf{v}_b a \mathbf{w}_a como sigue [28],[30]

$$\mathbf{Q}(\mathbf{w}_a) = (\mathbf{Q}_a^b)^* \otimes \mathbf{Q}(\mathbf{v}_b) \otimes \mathbf{Q}_a^b \quad (2.22)$$

La derivada del producto de dos cuaterniones, respeta la regla de derivación de productos, sin embargo al ser una de las funciones de los cuaterniones unitarios la de describir la orientación de un cuerpo rígido con respecto a una referencia fija, es mejor expresar esta derivada en términos de la velocidad angular $\vec{\omega}(t)$ del cuerpo móvil, respecto a la referencia fija. Obteniéndose la igualdad mostrada en la ecuación (2.23), en [30] y [28], puede encontrarse el desarrollo completo para llegar a esta expresión.

$$\dot{\mathbf{Q}}(t) = \frac{1}{2} \mathbf{Q} \otimes \mathbf{Q}(\vec{\omega})(t) \quad (2.23)$$

Donde $\mathbf{Q}(\omega)$ es un cuaternión puro. Expandiendo (2.23), se tiene,

$$\dot{\mathbf{Q}} = \frac{1}{2} \begin{pmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{pmatrix} \begin{pmatrix} 0 \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} q_0 & -\vec{q}^T \\ \vec{q} & I_3 q_0 + [\vec{q}^\times] \end{pmatrix} \begin{pmatrix} 0 \\ \vec{\omega} \end{pmatrix} \quad (2.24)$$

por lo tanto de (2.24), la ecuación cinemática que representa la orientación de un cuerpo rígido en función del vector de velocidad angular puede expresarse como [13], [15],

$$\dot{\mathbf{Q}} = \frac{1}{2} \begin{pmatrix} -\vec{q}^T \\ I_3 q_0 + [\vec{q}^\times] \end{pmatrix} \vec{\omega} = \frac{1}{2} \Xi(\mathbf{Q}) \vec{\omega} \quad (2.25)$$

Una vez introducidos los conceptos de sistemas de referencia y parametrizaciones, en la siguiente sección, se introducen los sensores empleados en la navegación integrada, los cuales hacen uso de los conceptos anteriores, para representar sus mediciones.

2.4. Sensores

Los sensores están presentes en una gran cantidad de dispositivos y cosas de las que hacemos uso diariamente, ya que permiten obtener información externa o interna del dispositivo al que se encuentran integrados, esto posibilita una interacción entre el dispositivo y el medio que lo rodea o una interacción con el estado interno del propio dispositivo. Los sensores cubren un amplio campo de estudio, sin embargo lo que se pretende en la presente sección es realizar un estudio general de los sensores que brindan información acerca del estado interno de un vehículo autónomo, en especial los utilizados en el sistema de navegación del mismo.

El uso de sensores en un vehículo autónomo es esencial ya que permiten que este interactúe con el medio que lo rodea con la finalidad de evadir obstáculos, reconocer donde se encuentra en cada instante de tiempo y que sepa cómo moverse hacia un punto dado de referencia. Toda esta información le permitirá al vehículo generar mejores trayectorias y alcanzar el punto de referencia con el mínimo error posible [31].

Como se ha mencionado en secciones anteriores, los sensores más usados en la navegación integrada son los inerciales (acelerómetros y girómetros), que ofrecen mediciones de aceleración lineal y velocidad angular respectivamente, con las cuales es posible obtener información de posición, orientación y velocidad del vehículo, se tiene también el uso de magnetómetros y GPS que complementan las mediciones de los sensores inerciales disminuyendo en gran medida los errores presentados en la navegación inercial.

Es conocido de la física que tanto la aceleración como la velocidad angular de un cuerpo son magnitudes vectoriales, por lo que es importante conocer la proyección de estas mediciones sobre al menos 3 ejes no coplanares. Es por ello que a continuación se hace referencia a sensores inerciales de 3 ejes o de 3 grados de libertad cada uno, y que son ampliamente comerciales, los circuitos integrados que contienen estos sensores son conocidos en el mercado de sensores como Unidades de Medición Inercial (IMU: Inertial Measurement Unit) e incorporan en la mayoría de los casos un magnetómetro y sensores de temperatura o presión, sin embargo en este trabajo de tesis, este conjunto de sensores es denominado: unidad de medición (MU).

2.4.1. Acelerómetros

Los acelerómetros son dispositivos que generalmente cuentan con tres ejes, y son capaces de detectar fuerzas de aceleración estática y dinámica. La aceleración estática comprende la aceleración debido a la gravedad, y la aceleración dinámica comprende la aceleración debido a vibraciones o movimiento del cuerpo al que el sensor se encuentra solidariamente unido. Una manera muy usual en física de representar y describir la aceleración, es como una tasa de cambio de la velocidad con respecto al tiempo.

$$a = \frac{dv}{dt} \quad (2.26)$$

Se dice que los acelerómetros miden la fuerza específica en el sistema de referencia del cuerpo \mathbf{E}^b , dada la segunda ley de Newton donde la aceleración de un cuerpo con masa constante es proporcional a la fuerza que actúa sobre este mismo, este enunciado se representa de la forma siguiente:

$$F = ma \quad (2.27)$$

Las mediciones de un acelerómetro pueden entonces ser integradas para obtener información de velocidad y si se integra esta última se obtiene información de desplazamiento [19], sin embargo como se mencionó en los apartados 2.1.1 y 2.1.3, se necesitan conocer las condiciones iniciales, en este caso velocidad y posición.

Debido al bajo costo, estos sensores sufren de una serie de errores que comprometen la fiabilidad de sus mediciones, los errores más comunes según [29] y [19] son:

- Error de bias lentamente variante en el tiempo
- Error de factor de escala
- Ruido del sensor

Entonces la medición de un acelerómetro se puede representar como la suma de la aceleración lineal del vehículo, la aceleración debido a la gravedad, un componente de error de bias y una componente de ruido blanco [29],[16]:

$$a_m = a_b + ge_3 + b_a + \eta_a \quad (2.28)$$

Donde:

- $a_b \in \mathbb{R}^3$, es la aceleración específica expresada en el sistema de referencia del cuerpo \mathbf{E}^b .
- $ge_3 \in \mathbb{R}^3$, es la aceleración debido a la gravedad, donde $e_3 = (0, 0, 1)$, ya que sólo actúa sobre la componente vertical.
- $b_a \in \mathbb{R}^3$, es el error de bias del sensor.
- $\eta_a \in \mathbb{R}^3$, representa el ruido blanco gaussiano del sensor.

Como se observa en la ecuación (2.28), el acelerómetro proporciona información de la aceleración lineal del cuerpo y del componente de gravedad, es por ello que como se indica en la Figura 2.2, la componente de gravedad debe de ser removida para obtener únicamente la aceleración lineal del cuerpo, en orden de estimar la posición.

2.4.2. Girómetros

Los girómetros utilizados en navegación inercial son dispositivos de tres ejes que proporcionan mediciones de la velocidad angular del cuerpo al que se encuentran solidariamente unidos. Esto es, dan sus mediciones en el sistema de referencia del cuerpo \mathbf{E}^b , con respecto a una referencia inercial \mathbf{E}^f . Estos al igual que los acelerómetros presentan los mismos errores que caracterizan a las unidades de medición inercial y comprometen la fiabilidad de las mediciones, a continuación se mencionan los errores más comunes en los girómetros [19]:

- Error de bias lentamente variante en el tiempo
- Error de factor de escala
- Ruido del sensor

Así las mediciones de un girómetro pueden se descritas como, la suma de la velocidad angular del cuerpo, la componente de error de bias y la componente de ruido blanco gaussiano del sensor. Como se representa a continuación [15],[29]:

$$\omega_g = \omega_b + b_g + \eta_g \quad (2.29)$$

Donde:

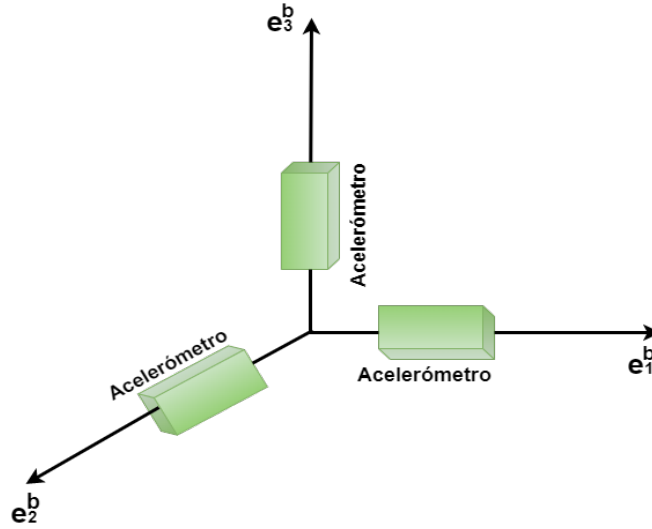


Figura 2.8: Sistema de coordenadas empleado para el acelerómetro

- $\omega_g \in \mathbb{R}^3$, es la velocidad angular, expresada en el marco de referencia del cuerpo.
- $b_g \in \mathbb{R}^3$, es el error de bias del sensor.
- $\eta_g \in \mathbb{R}$, es el ruido blanco Gaussiano.

2.4.3. Magnetómetro

El magnetómetro tiene la capacidad de medir el campo magnético local de la tierra, el campo magnético es una cantidad vectorial, es por ello que es representado en un sistema de 3 dimensiones. La medición de campo magnético puede ser empleado para determinar la dirección, rotación y el ángulo de los campos magnéticos, así como la ubicación de objetos dentro de ellos. Este sensor no presenta el error de deriva que presentan el acelerómetro y girómetro, sin embargo se ve afectado por el ruido blanco gaussiano y por perturbaciones originadas por objetos metálicos cercanos al sensor, o por campos magnéticos externos, generados por motores, imanes, etc. Es común que un vehículo contenga materiales metálicos, es por ello que deberá considerarse esta situación al momento de fijar el sensor al cuerpo del vehículo en cuestión para evitar disturbios en las mediciones. Por consiguiente la expresión que describe las mediciones de un magnetómetro está dada por la siguiente ecuación [29],

$$B_m = E_b + D_m + \eta_m \quad (2.30)$$

Donde:

- $E_b \in \mathbb{R}^3$, es el campo magnético de la tierra, expresado en el marco de referencia del cuerpo.
- $D_m \in \mathbb{R}^3$, es el campo magnético debido a perturbaciones.
- $\eta_m \in \mathbb{R}^3$, es el ruido blanco Gaussiano.

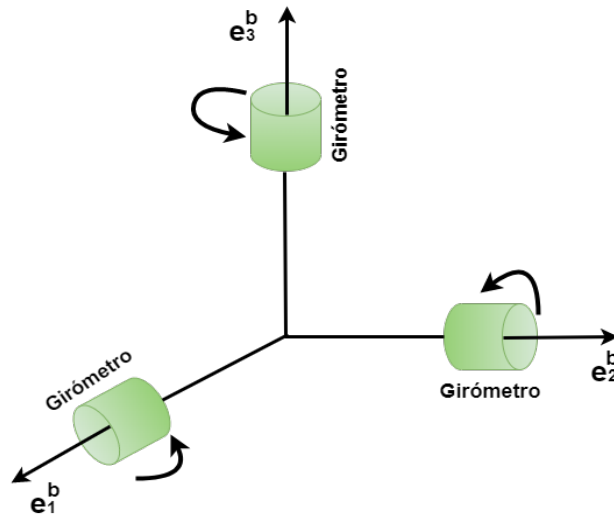


Figura 2.9: Sistema de coordenadas empleado para el girómetro

2.4.4. Sistema de Posicionamiento Global (*GPS*)

El sistema de posicionamiento global es ampliamente conocido como una de las mejores maneras de obtener información de: posicionamiento tridimensional, fecha y tiempo en casi cualquier parte del globo terrestre [1]. El GPS está basado en el sistema satelital *NAVSTAR*, que es mantenido por el Departamento de Defensa de los Estados Unidos.

El sistema de posicionamiento global se conforma de tres partes, para proporcionar la información antes mencionada (Fuente: <https://www.gps.gov/systems/gps/spanish.php>):

- Sistema satelital
- Centro de control terrestre
- Receptor del usuario

El sistema satelital *NAVSTAR*, consiste en 24 satélites, orbitando alrededor del globo terrestre divididos en seis planos con al menos 4 satélites cada uno de ellos, inclinados a unos 55° del ecuador y separados a unos 60° de longitud (ver Figura 2.10) [21].

El centro de control terrestre consiste en una serie de estaciones fijas distribuidas en todo el mundo y se encargan de monitorear y actualizar información de navegación y efemérides de los satélites. El receptor del usuario consiste en un dispositivo capaz de recibir señales del GPS, para proporcionar información de posición, fecha y tiempo.

En el presente trabajo no se pretende cubrir el funcionamiento del Sistema de Posicionamiento Global, más bien se busca mencionar de manera general las condiciones que afectan el desempeño del GPS, y el protocolo de comunicación que utilizan los receptores GPS.

El receptor GPS, como se mencionó en secciones anteriores, es ampliamente utilizado en los sistemas de navegación en conjunto con los sensores inerciales, ya que se complementan. En el caso del receptor GPS ofrece información absoluta de posición a una baja tasa de muestreo, de 1Hz a

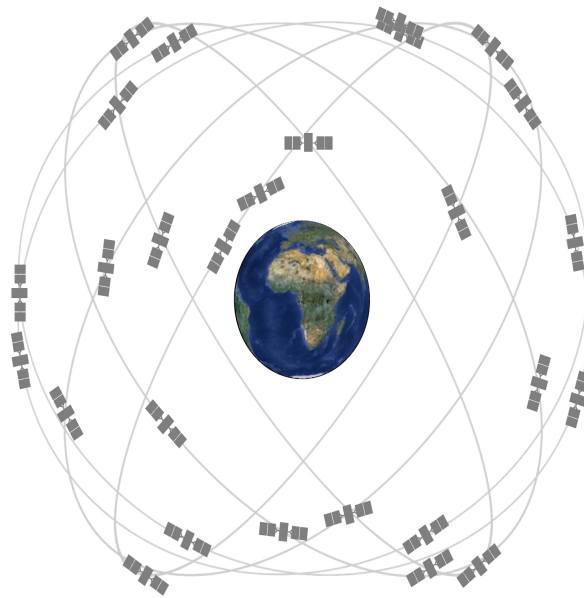


Figura 2.10: Constelación de satélites *NAVSTAR* (Fuente: <https://www.gps.gov/spanish.php>)

5Hz generalmente, al contrario de los sensores inerciales que que ofrecen esta información a una alta tasa de muestreo 100Hz, pero con errores asociados al proceso de estimación, que requieren ser actualizados con mediciones absolutas.

El GPS pertenece al método de navegación por referenciamiento absoluto, es decir, que depende de señales externas para proporcionar su información. Debido a esto el rendimiento del GPS puede verse afectado por diversos factores como pueden ser [1]:

- Condiciones ambientales
- Obstáculos entre satélite y receptor GPS
- Precisión en la transmisión satelital
- Características del receptor GPS

De los cuatro factores que podrían afectar el rendimiento del GPS, las características del receptor GPS es la única que el usuario podría llegar a controlar, al seleccionar receptores con buenas características, las más comunes son:

1. Máxima tasa de muestro
2. Tipo de receptor
3. Exactitud
4. Límite de operación (velocidad)
5. Protocolo
6. Sensibilidad de adquisición

El protocolo de comunicación que emplean la mayoría de los receptores GPS, es el protocolo NMEA (*National Marine Electronics Association*), el cual es un protocolo serial, y se han desarrollado muchas variantes de esta, sin embargo las más utilizada es el NMEA-0183 [1]. Este protocolo soporta un modo *ASCII* de comunicación, basado en un *talker* (receptor GPS) y uno o más *listeners* (computadoras), que reciben una cadena del protocolo llamadas sentencias [1]. A continuación se describe de manera general la estructura del protocolo NMEA-0 – 183.

La estructura de la sentencia consiste en una cadena de valores, separadas por comas y bien

identificadas por la posición que ocupan dentro de la cadena, toda cada NMEA, empieza por el signo '\$' seguida del contenido del mensaje y finaliza con el checksum.

En la primera posición de la cadena se encuentra el identificador del mensaje, el cual se divide en dos partes, las dos primeras letras de izquierda a derecha denotan el *Talker ID*, la cual hace referencia a la constelación. Las siguientes tres letras hacen referencia al contenido del mensaje, el cuál puede variar en longitud (ver Figura 2.11). En <https://brandidowns.com/?p=77> y <http://freenmea.net/docs>, puede encontrarse una descripción más detallada de estos parámetros y del protocolo en general.

Para casos de uso en el presente trabajo se menciona el formato en el que se recibe el tiempo, latitud y longitud.

Tiempo: el formato es *hhmmss.ss* de acuerdo al *Universal Time Coordinated* (UTC).

Latitud: es dada en grados decimales (*ddmm.mmmm*).

Longitud: es dada en grados decimales (*ddmm.mmmm*).

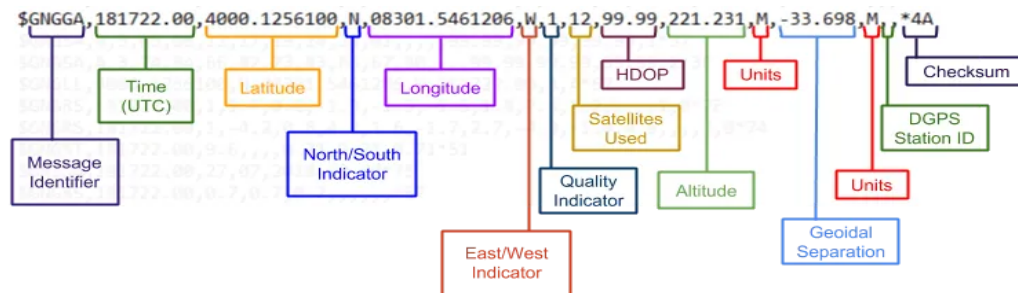


Figura 2.11: Ejemplo de una sentencia NMEA-0183 (Fuente: <https://brandidowns.com/?p=77>)

2.5. Fusión de Datos multi-sensorial

La fusión multi-sensorial, es la combinación de información procedente de distintas fuentes de sensores, con la finalidad de mejorar las características en general del sistema de sensado, como pueden ser:

- Mejorar la fiabilidad de la medición
- Mejorar la tasa de muestreo
- Mejorar la respuesta del sistema de sensado (velocidad)
- Complementar mediciones

Nosotros los humanos, estamos provistos de múltiples sensores (tacto, vista, gusto, olfato, oído) y de un potente procesador (cerebro), a diario hacemos uso de la mayoría de los sensores de manera individual o en conjunto, por ejemplo supongamos que un jugador profesional de basquetbol realiza tiros libres a ojos cerrados, únicamente tiene el conocimiento previo de la ubicación de la canasta y sabe en promedio que fuerza y ángulo debe de darle a su brazo para realizar un buen tiro, sin embargo tiene altas probabilidades de equivocarse al no estar viendo la canasta. Por el contrario si hiciera uso del sentido de la vista, podría tener información visual, con la cual podría calcular de mejor manera el ángulo y la fuerza que debe aplicar a su brazo para realizar un tiro exitoso. De esta manera se mejora la fiabilidad de los tiros. A menudo realizamos fusión de los diversos sentidos, como por ejemplo: gusto y olfato, vista y oído, vista y tacto, etc.

La fusión sensorial ofrece además un sistema tolerante a fallos, es decir que si uno de los sensores presenta algún tipo de fallo, el sistema seguiría funcionando sin ningún problema al recibir información de las demás fuentes de sensores, para ello conviene que algunos sensores provean el mismo tipo de información, aunque estos sean de diferentes características.

La fusión sensorial es aplicada actualmente en diversos campos como los son: automatización industrial, aplicaciones militares, aplicaciones biomédicas, robots inteligentes, sistemas de navegación, etc.

Para realizar la fusión de diversos sensores, como es el caso del presente trabajo de tesis, donde se busca fusionar las mediciones de sensores inerciales, magnetómetro y GPS, para obtener los parámetros de navegación, debe utilizarse alguno de los algoritmos de fusión de datos que existen y han sido ampliamente estudiados por diversos investigadores. Además, el algoritmo de fusión de datos debe de tener la capacidad de lidiar con las incertidumbres de las mediciones de los sensores (ruido asociado al sensor, bias del sensor) y del proceso mismo, con la finalidad de obtener un resultado de alta fiabilidad.

Uno de los primeros algoritmos desarrollado por diversos investigadores son los algoritmos deterministas, los cuales hacen uso de mediciones de referencia bien conocidas, como por ejemplo: la dirección del vector de gravedad, la dirección y rumbo del polo norte magnético. Otro algoritmo ampliamente utilizado en el campo de la fusión multi-sensorial o fusión de datos, es el Filtro de Kalman, el cual ha sido ampliamente estudiado y aplicado en diversas áreas, entre las que se destacan el *tracking* y sistemas de navegación. Otro algoritmo que en los últimos años ha tenido gran desarrollo, son los observadores no lineales, los cuales empiezan a ser bastante utilizados, debido a la menor carga computacional que presentan. Antes de dar una breve descripción del Filtro de Kalman y de los Observadores no lineales, se dará un breve repaso del concepto de Observador, el cuál es la base de los algoritmos de fusión de datos, que se encargan de fusionar los datos de sensores y estimar el vector de estado de un sistema.

2.5.1. Algoritmos deterministas para la estimación de la orientación

La estimación del attitude es clave en diversos campos de aplicaciones de control y robótica donde el conocimiento de la orientación es imprescindible en la toma de acciones, como por ejemplo, vehículos autónomos aéreos, terrestres y marinos, satélites, aplicaciones de seguimiento de trayectorias, etc. Es por ello por lo que desde hace años se han venido desarrollando principalmente dos métodos de determinación del attitude, los cuales son: mediante algoritmos de estimación como por ejemplo las diversas variaciones del Filtro de Kalman y los observadores No Lineales, y el otro método es mediante algoritmos deterministas. Entre las primeras técnicas deterministas desarrolladas se encuentran las puramente algebraicas, como el TRIAD, Davenport's q-method, the QUEST algorithm, el algoritmo SVD [15], y el algoritmo LQS (*Learning Quaternion Sequence*) [13], [32], estos dos últimos algoritmos han sido reportados recientemente, haciendo uso de un modelo de mediciones en base a vectores de observación.

Usualmente los algoritmos deterministas hacen uso de al menos un par de sensores que proporcionan información en manera de vectores unitarios de dirección bien conocidos, como por ejemplo la dirección del vector de gravedad terrestre dado por el acelerómetro, sensores que detectan la dirección del sol, el vector de rumbo del campo magnético dado por magnetómetros, o un vector apuntando a una estrella conocida. Estos sensores son conocidos en el campo de la estimación de la orientación como sensores de vector de referencia.

A continuación se describe un algoritmo recursivo determinista, el cuál será descrito utilizando las notaciones de los trabajos originales [13], [32], este algoritmo llamado LQS, sirve como base

en la estimación de la orientación de un cuerpo rígido y también forma parte del algoritmo de consenso dinámico propuesto en este trabajo de tesis.

Consideré la representación de un vector, \vec{x}_i , con respecto a \mathbf{E}^f y \mathbf{E}^b y denotados por \vec{r}_i y \vec{b}_i , respectivamente. El vector \vec{r}_i , es llamado vector de referencia y en general es bien conocido, como por ejemplo el vector de gravedad terrestre, $ge_3 \in \mathbb{R}^3$, donde $e_3 = (0, 0, 1)$; o la ubicación del norte magnético el cual únicamente tendrá componentes en $(mx, 0, mz)$ y puede ser obtenido a partir de un magnetómetro. Los vectores del cuerpo, son también conocidos como vectores de observación y estos son obtenidos a partir de las mediciones de sensores como por ejemplo acelerómetros y magnetómetros que se encuentran solidarios al cuerpo rígido en cuestión. Estas dos representaciones están vinculadas a través de la matriz de rotación $R \in SO(3)$:

$$\vec{b}_i = R\vec{r}_i \quad (2.31)$$

y denotando como $\mathbf{Q}(b_i)$ y $\mathbf{Q}(r_i)$ a los cuaterniones puros asociados a los vectores \vec{b}_i y \vec{r}_i :

$$\mathbf{Q}(b_i) = \begin{pmatrix} 0 & \vec{b}_i^T \end{pmatrix}^T \quad \mathbf{Q}(r_i) = \begin{pmatrix} 0 & \vec{r}_i^T \end{pmatrix}^T \quad (2.32)$$

de manera que una rotación de $\mathbf{Q}(r_i)$ a $\mathbf{Q}(b_i)$, puede ser representada de la siguiente manera, como se indica en (2.21):

$$\mathbf{Q}(b_i) = \mathbf{Q}^{-1} \otimes \mathbf{Q}(r_i) \otimes \mathbf{Q} \quad (2.33)$$

A partir de la relación (2.33) y utilizando la definición de producto cuaternión, dada en (2.18), los autores de los trabajos [13], [32], han desarrollado el siguiente modelo de medición,

$$\begin{pmatrix} 0 & -(\vec{b}_i - \vec{r}_i)^T \\ \vec{b}_i - \vec{r}_i & -[(\vec{b}_i + \vec{r}_i) \times] \end{pmatrix} \mathbf{Q} = H\mathbf{Q} = 0 \quad (2.34)$$

donde $H \in \mathbb{R}^{4 \times 4}$, el modelo de medición (2.34) forma un sistema de ecuaciones lineales que al solucionarlo permitirá obtener el cuaternión que mapea del sistema de referencia móvil \mathbf{E}^b al sistema de referencia inercial \mathbf{E}^f . Los autores generalizan (2.34) para m vectores de observación en (2.35), y al menos dos vectores de observación no colineales entre si deberá estar disponible, con la finalidad de estimar el cuaternión denotado como \mathbf{Q} .

$$\bar{H} = \begin{pmatrix} 0 & -(\vec{b}_1 - \vec{r}_1)^T \\ \vec{b}_1 - \vec{r}_1 & -[(\vec{b}_1 + \vec{r}_1) \times] \\ \vdots & \\ 0 & -(\vec{b}_m - \vec{r}_m)^T \\ \vec{b}_m - \vec{r}_m & -[(\vec{b}_m + \vec{r}_m) \times] \end{pmatrix} = \bar{H}(t)\mathbf{Q} = \begin{pmatrix} \varphi_1^T(t) \\ \vdots \\ \varphi_m^T(t) \end{pmatrix} \mathbf{Q} = 0, \quad \bar{H} \in \mathbb{R}^{4m \times 4} \quad (2.35)$$

El cuaternión desconocido \mathbf{Q} , el cual será llamado \mathbf{Q}_{ps} puede ser estimado minimizando la siguiente función de costo,

$$J(\mathbf{Q}) = \frac{1}{2} \sum_{i=1}^{4m} (\varphi_i^T(t)\mathbf{Q})^T (\varphi_i^T(t)\mathbf{Q}) = \frac{1}{2} \|\bar{H}(t)\mathbf{Q}\|^2 \quad (2.36)$$

Con la finalidad de dar solución al sistema de ecuaciones lineales formado en (2.35), se hace uso del método Kaczmarz, el cual consiste en un algoritmo iterativo para solucionar sistemas de ecuaciones lineales de la forma $Ax = b$. Este algoritmo opera solamente en una fila de la matriz \bar{H} por iteración, y en cada iteración se obtiene para este caso particular un cuaternión, donde el \mathbf{Q}_{ps} será dado por la última iteración del algoritmo al operar sobre el último renglon de la matriz \bar{H} .

$$\mathbf{Q}_{ps}^i = \left[\prod_{i=1}^{4m} \left(I_4 - \frac{\gamma \varphi_i(t) \varphi_i^T(t)}{\alpha + \varphi_i^T(t) \varphi_i(t)} \right) \right] \mathbf{Q}_{ps}^{i-1} \quad (2.37)$$

donde $\alpha \geq 0$ y $0 < \gamma < 2$, α sintoniza la velocidad de convergencia del algoritmo (2.37), propuesto en [13], [32].

2.5.2. Observadores

La representación en el espacio de estado, permite modelar un proceso, buscando tener en cuenta todas las variables que intervienen en la dinámica del mismo [33]. Por lo general, cuando se trabaja con variables de estado y se requiere realizar el control sobre alguno de los estados del sistema, este suele realizarse mediante retroalimentación de estados, sin embargo no todas las variables asociadas a la dinámica del sistema llegan a ser físicamente medibles, debido a que son variables internas del sistema o porque los sensores son demasiados caros o estos aún no existen. Por lo cual se hace necesario realizar una estimación de dichas variables, a partir de la información disponible en la entrada y salida del proceso. Esto permite tener información de la dinámica del sistema sin necesidad de medir todas las variables del vector de estado $\mathbf{x}(t)$ [33].

Los observadores, también llamados estimadores o filtros, permiten la reconstrucción o estimación del vector de estado de un sistema [34]. Si consideramos el siguiente sistema,

$$\begin{aligned} \dot{\mathbf{x}} &= A\mathbf{x} + B\mathbf{u}(t) \\ y &= C\mathbf{x}(t) \end{aligned} \quad (2.38)$$

El observador más básico conocido que se puede implementar, es el observador de Luenberger, el cual queda descrito por la ecuación (2.39) .

$$\dot{\hat{\mathbf{x}}} = A\hat{\mathbf{x}} + B\mathbf{u} + L(y - C\hat{\mathbf{x}}) \quad (2.39)$$

Haciendo $\hat{y} = C\hat{\mathbf{x}}$, tenemos

$$\dot{\hat{\mathbf{x}}} = A\hat{\mathbf{x}} + B\mathbf{u} + L(y - \hat{y}) \quad (2.40)$$

donde $\hat{\mathbf{x}}$, es el vector de estado estimado a partir de la información disponible en la salida y entrada del sistema, el término $L(y - C\hat{\mathbf{x}})$ es la ganancia del observador, la cual actúa como señal de corrección haciendo que el error de observación tienda asintóticamente a cero [34].

Analizando la dinámica del error, es posible visualizar la condición necesaria para que el observador converja al estado real. Haciendo,

$$e = \mathbf{x} - \hat{\mathbf{x}} \quad (2.41)$$

$$\begin{aligned}
 \dot{e} &= \dot{\mathbf{x}} - \dot{\hat{\mathbf{x}}} \\
 \dot{e} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} - \mathbf{A}\hat{\mathbf{x}} - \mathbf{B}\mathbf{u} - \mathbf{L}(\mathbf{C}\mathbf{x} - \mathbf{C}\hat{\mathbf{x}}) \\
 \dot{e} &= \mathbf{A}(\mathbf{x} - \hat{\mathbf{x}}) - \mathbf{L}\mathbf{C}(\mathbf{x} - \hat{\mathbf{x}}) \\
 \dot{e} &= \mathbf{A}e - \mathbf{L}\mathbf{C}e \\
 \dot{e} &= (\mathbf{A} - \mathbf{L}\mathbf{C})e
 \end{aligned} \tag{2.42}$$

Del análisis realizado en la ecuación (2.42), se observa que el observador converge si el error de estimación e tiende a cero [34], para que esto ocurra debe de cumplirse que el polinomio característico del sistema resultante de $(\mathbf{A} - \mathbf{L}\mathbf{C})$, debe de tener raíces negativas o con parte real negativa [34].

En la Figura 2.12, se observa el diagrama de bloques general de un observador, donde el observador de estados, se encuentra encerrado por las líneas discontinuas.

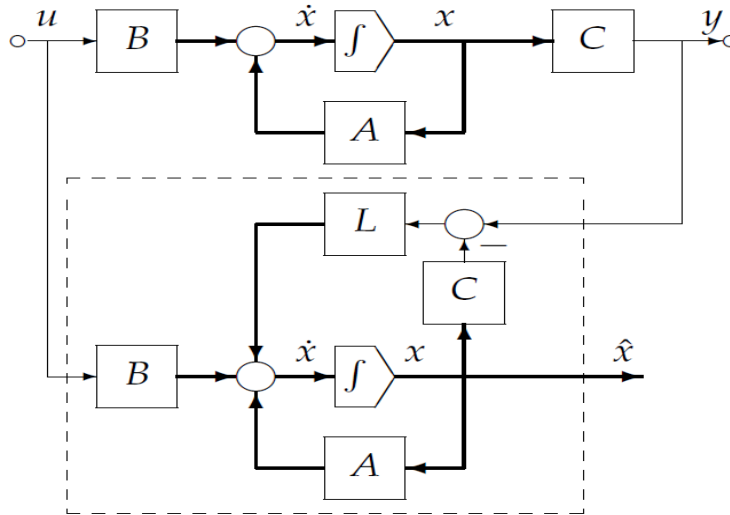


Figura 2.12: Estructura general del observador (Fuente: [35])

Antes de llevar a cabo el diseño de un observador de estados, debe de verificarse el criterio de observabilidad, el cual nos dice si es posible reconstruir el vector de estados a partir de la información dada por la salida y la entrada.

Considerando nuevamente el sistema de la ecuación (2.38), donde $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{C} \in \mathbb{R}^{r \times n}$, es observable si y solo si la matriz de observabilidad \mathcal{O} tiene rango n [34], esto es,

$$\text{rango}(\mathcal{O}) = \text{rango} \left(\begin{bmatrix} \mathbf{C} \\ \mathbf{C}\mathbf{A} \\ \mathbf{C}\mathbf{A}^2 \\ \vdots \\ \mathbf{C}\mathbf{A}^{n-1} \end{bmatrix} \right) = n \tag{2.43}$$

Como se observó en la ecuación (2.38), este modelo, representa un modelo ideal de un sistema, es decir que no es afectado por perturbaciones externas al sistema, por ruido o defectos en los sensores, o por alguna dinámica no tomada en cuenta en el modelo matemático [35]. A continuación se describe el Filtro de Kalman, el cual es un estimador óptimo, ya que toma en cuenta la

incertidumbre asociadas al proceso y a las mediciones, desarrollando de esta manera un modelo matemático más preciso del sistema.

2.5.3. Filtro de Kalman

El Filtro de Kalman (KF del inglés *Kalman Filter*), es un algoritmo matemático, ampliamente utilizado en sistemas de control moderno o control en el espacio de estados, ya que permite abordar la problemática de estimación del vector de estado (estado interno) del sistema, teniendo en cuenta las incertidumbres asociadas a la medición y al proceso (ver figura 2.13).

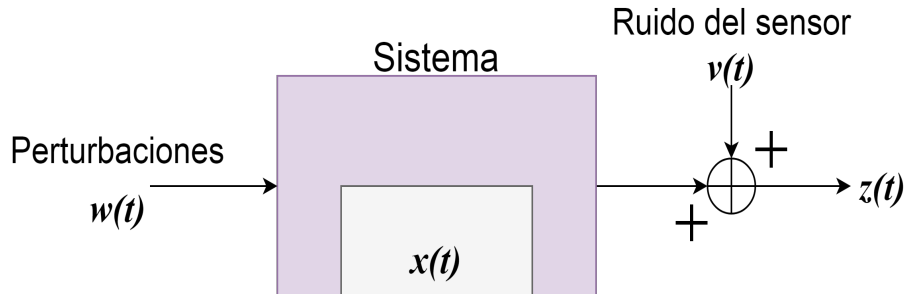


Figura 2.13: Sistema con perturbación en la entrada y ruido en la medición (Fuente: <http://mocha-java.uccs.edu/ECE5550/index.html>)

El filtro de Kalman, toma en cuenta en el modelo matemático las perturbaciones que el sistema puede sufrir, permitiendo de esta manera tomar mejores acciones de control, y toma también en cuenta el ruido asociado a la medición, el cual es causado por las características de los sensores, esto es clave en el proceso de estimación de las variables de estado, ya que permite la fusión de los datos de múltiples sensores con la finalidad de obtener una estimación precisa del estado interno del sistema.

Los estados del sistema, dependerán del sistema mismo, esto es, se podrían tener como variables de estado: posición, velocidad, aceleración y orientación de un vehículo aéreo, terrestre o marino [26] y cada uno de estos ambientes tendrán sus propias fuentes de incertidumbre o perturbaciones (viento, olas), que suelen modelarse como procesos aleatorios.

Como se mencionó anteriormente, la representación en espacio de estados, permite la implementación del filtro de Kalman, teniendo en cuenta el modelo matemático del sistema, de la medición y sus incertidumbres asociadas. La ecuación (2.44) [36], es una representación general del modelo en espacio de estado considerando el ruido del proceso y el ruido de la medición.

$$\begin{aligned} \mathbf{x}_k &= \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \\ \mathbf{z}_k &= \mathbf{C}_k\mathbf{x}_k + \mathbf{v}_k \end{aligned} \quad (2.44)$$

Donde el significado de la notación usada, basada en [36], se da en la tabla 2.1.

Cuando se estudia por primera vez algún curso de control moderno, y se trabaja en la estimación de estados con observadores, algunos alumnos, suelen tener la idea equivocada de que el vector de estado \mathbf{x} , es igual a la matriz de observación \mathbf{C} , sin embargo en la matriz de observación, se tienen las mediciones disponibles en la salida del proceso, expresadas en términos de las variables de estado y en el vector de estado se tienen las variables que serán estimadas por el filtro. Por ejemplo, si se tiene la medición disponible de aceleración y a partir de esta se requiere

Variable	Descripción	Dimensión
\mathbf{x}	Vector de estado	$n_x \times 1$
\mathbf{z}	Vector de salida	$n_y \times 1$
\mathbf{u}	Vector de entrada	$n_u \times 1$
\mathbf{w}	Vector de ruido del proceso	$n_x \times 1$
\mathbf{v}	Vector de de la medición	$n_y \times 1$
\mathbf{A}	Matriz de estado del sistema	$n_x \times n_x$
\mathbf{B}	Matriz de entrada del sistema	$n_x \times n_u$
\mathbf{C}	Matriz de Observación	$n_y \times n_x$

Tabla 2.1: Definición y dimensiones de variables

determinar la posición, se requerirá también de la estimación de la velocidad, por lo que el vector de estado quedaría como,

$$\mathbf{x} = \begin{bmatrix} p \\ v \\ a \end{bmatrix}$$

El vector de salida, que corresponde a la medición disponible, junto al ruido asociado de la medición sería,

$$\mathbf{z} = a + \mathbf{v}$$

Donde $a \in \mathbb{R}$. La medición disponible debe expresarse en términos de las variables de estado, con fines de comparación y así poder realizar correcciones, la matriz de observación se definiría como,

$$\mathbf{C} = [0 \quad 0 \quad 1]$$

Los vectores \mathbf{w} y \mathbf{v} que corresponden a los ruidos del proceso y de la medición respectivamente, son modelados como variables aleatorias, a estos vectores se les asocia una matriz de covarianza, que es donde se resumen la variabilidad del ruido del proceso y medición.

Algoritmo del Filtro de Kalman

Una vez introducidos conceptos y notaciones básicas de representación en espacio de estados, a continuación se presenta el algoritmo del filtro de Kalman, el cuál es un algoritmo iterativo, que consiste en dos fases, la fase de predicción y la fase de corrección. Para el diseño del filtro de Kalman, debe de cumplirse el criterio de observabilidad, del par (A, C) .

En la Figura 2.14, se observa el algoritmo iterativo del filtro de Kalman, el cual consiste en el cálculo de un conjunto de ecuaciones que involucran a la dinámica del sistema, así como a las mediciones disponibles. A continuación se da una breve explicación de cada paso en ambas fases [36]:

- **(1a): Predicción del vector de estado**, en este paso de la fase de predicción, se realiza la estimación del vector de estado actual, a partir de mediciones anteriores, en este paso se

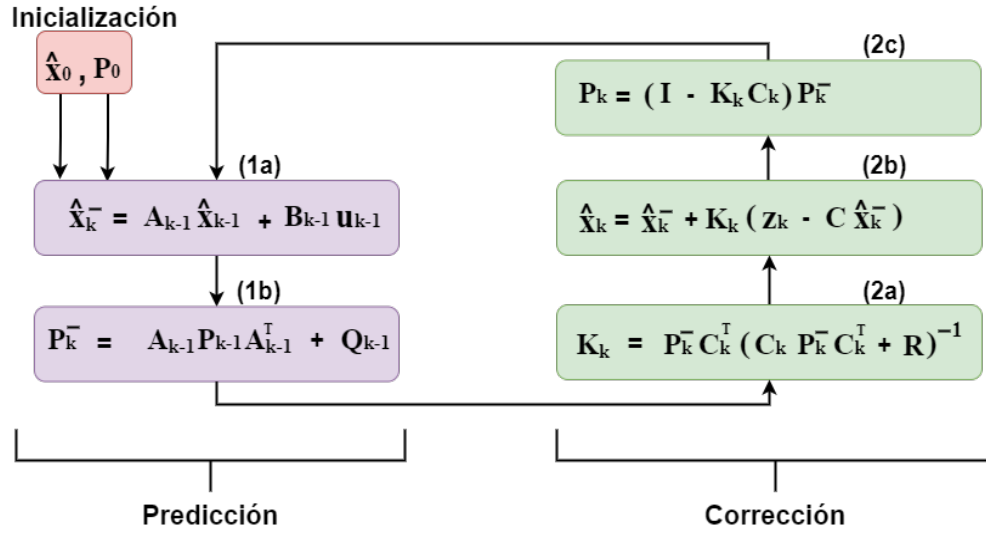


Figura 2.14: Algoritmo del filtro de Kalman (Basado en: <http://mocha-java.uccs.edu/ECE5550/index.html>)

tienen en cuenta las matrices \mathbf{A} y \mathbf{B} , que definen la dinámica del sistema, y el vector de entrada \mathbf{u}

- **(1b): Predicción de la matriz de covarianza de error del estado**, en este segundo paso de la fase de predicción, se estima la matriz de covarianza de error del estado, a partir de mediciones anteriores, para ello se tienen en cuenta la matriz \mathbf{A} del sistema, la matriz \mathbf{Q} que es la matriz de covarianza del proceso, la cuál se supone conocida.
- **(2a): Cálculo de la matriz de ganancia Kalman**, la ganancia de Kalman \mathbf{K} , es uno de los parámetros más importantes en este algoritmo, ya que define el factor de la innovación o corrección que se realiza en un paso posterior, para calcular esta ganancia, se toman en cuenta la matriz de covarianza \mathbf{P} , la matriz de observación \mathbf{C} y la matriz de covarianza \mathbf{R} , asociada al ruido de la medición.
- **(2b): Actualización del vector de estado**, en este paso de la fase de corrección, se actualiza el vector de estado, a partir del cálculo de la innovación la cual consiste en la diferencia de la medición disponible y su estimación. Esta innovación es multiplicada por la ganancia de Kalman y definen la corrección necesaria del vector de estado.
- **(2c): Actualización de la matriz de covarianza de error del estado**, finalmente, se actualiza la matriz de covarianza \mathbf{P} , la cual debería de decrementar la incertidumbre del estado estimado.

El Filtro de Kalman ha sido ampliamente estudiado y aplicado en diversos campos de investigación para la fusión y estimación de datos. El algoritmo de la Figura 2.14, es aplicable a sistemas lineales, sin embargo la mayor parte de los sistemas reales son de comportamiento no lineal, para este tipo de sistemas se han desarrollado algoritmos como el Filtro de Kalman Extendido y los Observadores no Lineales. Donde el sistema a tener en cuenta puede tener la forma general de las ecuaciones (2.45) y (2.46) [25],

$$\dot{\mathbf{x}} = f(\mathbf{x}) + \mathbf{B}\mathbf{u} + \mathbf{E}\mathbf{w} \quad (2.45)$$

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v} \quad (2.46)$$

Donde $f(x)$ es un vector de campo no lineal. El Filtro de Kalman Extendido, hace uso de la matriz Jacobiana, para realizar una linealización del sistema, evaluando dicha matriz en la trayectoria estimada [26]. Como se ha estudiado el Filtro de Kalman, pertenece a la categoría de estimadores estocásticos, por lo que aunque es un algoritmo de estimación muy utilizado, no hay manera de garantizar la convergencia del observador [37].

Fusión de datos de múltiples unidades de medición inercial

Con la finalidad de mejorar la precisión y fiabilidad del vector de estimación de alguna aplicación en específico o para tener un sistema redundante y tolerante a fallos, diversos grupos de investigación han incursionado en el campo de la fusión sensorial [38]. Un sistema tolerante a fallos es crucial en la gran mayoría de aplicaciones que utilizan sensores, debido a que en la práctica son los sensores los que usualmente presentan algún tipo de fallo y llegan a provocar diversos tipos de errores o accidentes en el proceso [39], una manera de solucionar esto es implementando un sistema tolerante a fallos mediante el uso de sensores redundantes, este sistema sería capaz de detectar el sensor defectuoso y aislarlo [40], de este modo la señal estimada de los demás sensores podría reemplazar sin ningún problema la señal del sensor que ha fallado, este proceso es conocido en la literatura como *Dead Reckoning* y puede funcionar hasta que el sensor defectuoso sea cambiado.

Algunos investigadores han propuesto utilizar múltiples unidades de medición inercial (*MIMU*, *Multiple Inertial Measurement Unit*), cuya información es fusionada utilizando principalmente el Filtro de Kalman. Por ejemplo en [41], los autores proponen tres arquitecturas basadas en el Filtro de Kalman Extendido, para llevar a cabo la fusión de la información procedente de cuatro unidades de medición inercial, en este trabajo los autores realizan la fusión sensorial en la etapa de predicción del filtro, además de que no realizan la estimación del bias del girómetro, con la finalidad de demostrar que con el uso de múltiples *IMUs* se puede obtener una respuesta de alta fidelidad sin necesidad de estimar el bias del girómetro. En [38], los autores presentan dos arquitecturas para fusionar datos de múltiples unidades de medición inercial, mencionan que la fusión de datos puede ser dividida en dos dominios: el dominio de la observación y el dominio de la estimación.

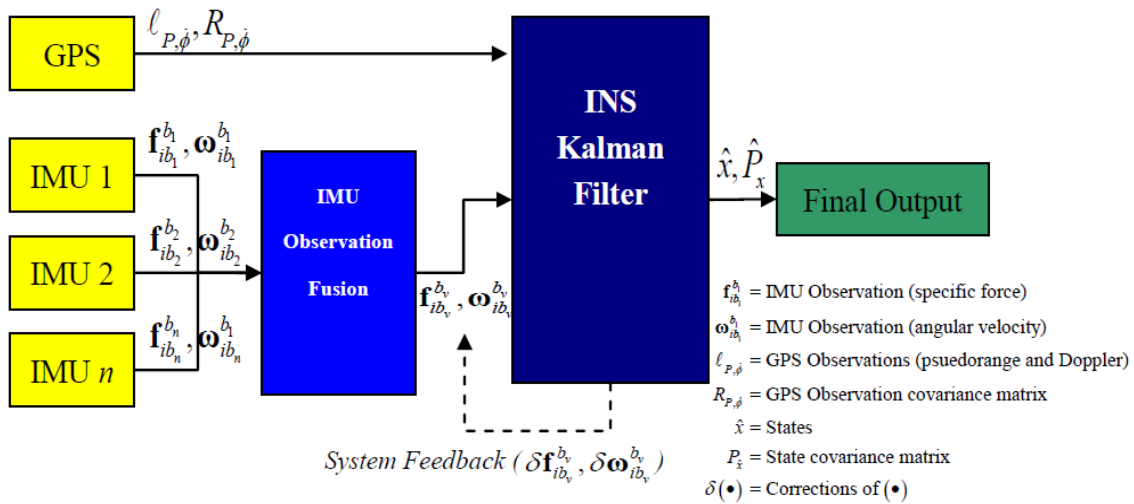


Figura 2.15: Arquitectura de fusión, dominio de la observación (Fuente: [38])

En la Figura 2.15, se muestra la arquitectura en el dominio de la observación, también llamada *Virtual IMU (VIMU)* [38], donde los datos sin procesar de múltiples unidades de medición inercial son fusionadas en un marco común. Los autores, también hacen mención de que promediar las observaciones de cada unidad de medición inercial es la manera más sencilla de generar un *VIMU*, sin embargo en las aplicaciones de navegación las unidades de medición inercial suelen colocarse en diferentes puntos del vehículo, por lo que los datos de los sensores dependerán de su ubicación y no es recomendable promediarlos [38].

La arquitectura de la Figura 2.16, corresponde al dominio de la estimación, la cual hace uso de un filtro centralizado para fusionar los datos de múltiples unidades de medición inercial. En [38], los autores proponen un filtro centralizado, el cual esta conformado de múltiples filtros locales, uno para cada *IMU*, estos filtros locales son los bloques en azul, mostrados en la Figura 2.16.

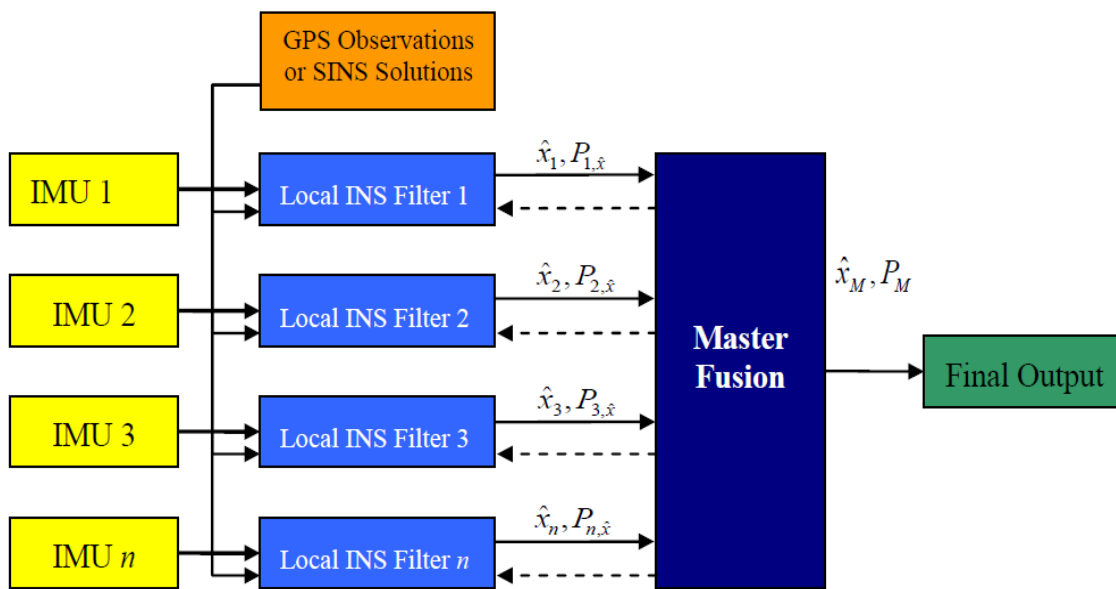


Figura 2.16: Arquitectura de fusión, dominio de la estimación (Fuente: [38])

Al trabajar con sistemas redundantes, los sensores pueden ser agrupados en diferentes configuraciones geométricas[40], las configuraciones más usuales son la ortogonal y la no ortogonal esta última más conocida como configuración *skewed*. En la investigación realizada en [40], el autor menciona que haciendo uso de la configuración *skewed*, se puede lograr un uso más eficiente de las mediciones de los sensores inerciales, además de que la geometría de configuración esta ligada al número de sensores a utilizar. Otro trabajo realizado en el diseño de sistemas tolerantes a fallas en el área de sistemas de navegación, es el realizado en [42], en este trabajo los autores reportan la comparación de dos procedimientos para el diseño de sistemas tolerantes a fallas, basados en observadores no lineales. El primer método es basado en la paridad de espacios, en este método, las mediciones de acelerómetros y girómetros son fusionadas en una sola medición. El segundo método el cual es una extensión usando cuaterniones, únicamente las mediciones de los acelerómetros son fusionados y las mediciones de los girómetros son enviadas cada uno a un estimador de orientación no lineal y al final son fusionadas en un observador de movimiento traslacional.

2.5.4. Observadores No Lineales

El diseño de observadores no lineales (*Non Linear Observer: NLO*) para aplicación a sistemas de navegación, se ha incrementado en los últimos años, ya que requieren un procesamiento de cómputo más reducido en comparación con el Filtro de Kalman Extendido, el cual requiere diversas operaciones con matrices. Otra ventaja del uso de observadores no lineales, es que son diseñados en base a la teoría de estabilidad no lineal, por lo cual es posible garantizar la convergencia del observador, aplicando el análisis de la teoría de estabilidad de Lyapunov[37].

El diseño de observadores no lineales, usualmente son basados en el filtro complementario para la determinación de orientación del vehículo [37], [29]. En general los observadores no lineales reportados en la literatura, hacen uso de la ecuación cinemática del cuerpo rígido definida en (2.23), donde se busca obtener el cuaternión que parametriza la rotación de un cuerpo rígido en función del vector de velocidad angular, esta ecuación diferencial es integrada numéricamente dada una condición inicial del cuaternión. Sin embargo debido al modelo que describe la medición dada por el girómetro (2.29), se puede deducir que la solución ofrecida por la ecuación cinemática tenderá a variar con respecto al tiempo debido al término de bias que afecta en la medición del girómetro.

El uso de un filtro complementario no lineal, permite la compensación o corrección del bias del girómetro [25],[43]. Un modelo de bias frecuentemente utilizado, consiste en el modelo de Markov [39]:

$$\dot{b} = -T_b^{-1}b + E_b w_b, \quad (2.47)$$

donde $b \in \mathbb{R}^3$ es el bias en la medición, $w_b \in \mathbb{R}^3$ es el vector de ruido blanco Gaussiano, $T_b \in \mathbb{R}^{3 \times 3}$ es una matriz de constantes de tiempos de bias y $E_b \in \mathbb{R}^{3 \times 3}$ es una matriz diagonal de escalado. Además el uso de un filtro complementario como base de un observador no lineal, permite la incorporación de otras fuentes de medición, con la finalidad de complementar la solución ofrecida por el girómetro para este caso específico, sensores adicionales como: acelerómetros y magnetómetros pueden ser incorporados para mejorar la estimación de la orientación, debido a que estos últimos ofrecen una buena solución a largo plazo y el girómetro una buena solución a corto plazo.

En los trabajos reportados en [13], [15], [32], los autores proponen el uso de un observador no lineal, basado en la ecuación cinemática del cuerpo rígido y en el uso de vectores de observación con las finalidades de calcular el cuaternión de error, lo que permite hacer una estimación del bias haciendo uso del modelo de Markov, esto conlleva a una estimación del cuaternión de orientación a partir de la ecuación cinemática sin deriva. Estas soluciones tienen la ventaja de tener bajo costo computacional, en especial la solución representada por la ecuación (2.48) [13], [32]. Es por ello que este observador será tomado como base en este trabajo, en donde se realizarán las modificaciones correspondientes a este observador, con la finalidad de tener un sistema de observadores distribuidos haciendo uso para ello de al menos tres unidades de medición (MU) lo que resultará en un sistema redundante en cuanto a mediciones inerciales y del magnetómetro, con la finalidad de mejorar la estimación de la orientación y tener un sistema robusto ante la posible falla de alguno de los sensores.

$$\mathcal{O} = \begin{cases} \dot{\hat{\mathbf{Q}}} = \frac{1}{2}\Xi(\hat{\mathbf{Q}}(t)) \left(\vec{\omega}_G(t) - \hat{\vec{v}}(t) + K_1 \vec{q}_e(t) \right) \\ \dot{\hat{\vec{v}}} = -T^{-1}\hat{\vec{v}}(t) - K_2 \vec{q}_e(t) \end{cases} \quad (2.48)$$

2.6. Teoría de grafos

La interacción de un sistema *multiagente* puede ser representada mediante grafos. La teoría de grafos es un recurso que permite definir la topología de comunicación entre los *Agentes* y desde el punto de vista de control, permite establecer propiedades de estabilidad.

Considere el grafo $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ (Fig. 2.17) el cual consiste en un conjunto de vértices (o nodos) $\mathcal{V} = 1, \dots, n$ y enlaces $\mathcal{E} \in \mathcal{V} \times \mathcal{V}$, donde cada nodo representa un agente. Si hay un enlace (i, l) entre los nodos i y l , con $1 \leq i \leq n$ y $1 \leq l \leq n$, entonces i y l se llaman adyacentes, es decir, $\mathcal{E} = \{(i, l) \in \mathcal{V} \times \mathcal{V} : i, j \text{ adyacente}\}$. \mathcal{G} es llamado no dirigido o bidireccional si $(i, l) \in \mathcal{E} \Leftrightarrow (l, i) \in \mathcal{E}$, en caso contrario el término asignado es grafo dirigido.

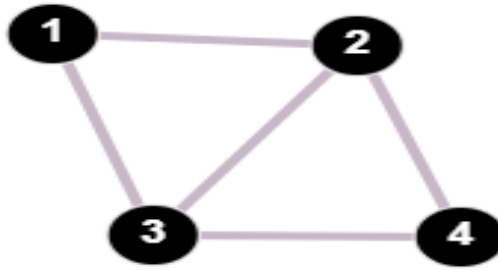


Figura 2.17: Grafo \mathcal{G} .

La matriz de adyacencia \mathbf{A} esta definida por los pesos $a_{il} > 0$ si i y l son adyacentes, y $a_{il} = 0$ en caso contrario. Si los pesos no son relevantes $a_{il} = 1$, si i y l son adyacentes como se muestra en la matriz de adyacencia 2.49 del grafo \mathcal{G} .

$$\mathbf{A} = \begin{matrix} & \begin{matrix} l_1 & l_2 & l_3 & l_4 \end{matrix} \\ \begin{matrix} i_1 \\ i_2 \\ i_3 \\ i_4 \end{matrix} & \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \end{matrix} \quad (2.49)$$

Un camino desde i hasta l es una secuencia de diferentes nodos, iniciando desde i y terminando con l , de modo que cada par de nodos consecutivos es adyacente. Si hay un camino de i a l , entonces i y l son llamados conectados. Si todos los pares de nodos en \mathcal{G} son conectados y \mathcal{G} es no dirigido, entonces se habla de un grafo fuertemente conectado.

Para grafos bidireccionales, el término *fuertemente* se omite. La distancia $d(i, l)$ entre dos nodos es el número de enlaces de la ruta más corta desde i hasta l . El diámetro \mathbf{d} de \mathcal{G} es la máxima distancia $d(i, l)$ sobre todos los pares de nodos. La matriz de grados \mathbf{D} de \mathcal{G} es la matriz diagonal con elementos d_i igual a la cardinalidad del conjunto vecino del nodo i , $\mathcal{N}_i = \{l \in \mathcal{V} : (i, l) \in \mathcal{E}\}$. Sea $\mathcal{M}_i = \mathcal{N}_i \cup \{i\}$ el conjunto de vecinos inclusivos del nodo i . La matriz Laplaciana \mathcal{L} de \mathcal{G} es definida como $\mathcal{L} = \mathbf{D} - \mathbf{A}$. Para grafos no direccionados, \mathcal{L} es simétrica y semi-definida positiva, es decir, $\mathcal{L} = \mathcal{L}^T \geq 0$. La suma de las filas de \mathcal{L} son cero.

En el siguiente capítulo se presentan de manera más detallada la descripción e implementación práctica del algoritmo *LQS* (*Learning Quaternion Sequence*) dado por las ecuaciones (2.35), (2.36) y (2.37). Se presenta también el Observador No Lineal descrito por la ecuación (2.48) y su implementación. Se presentan resultados prácticos para la estimación de la orientación de un cubo de rubik, donde se instrumenta haciendo uso de una MU que consta de un acelerómetro, magnetómetro y girómetro, todos de tres ejes.

Capítulo 3

Estimación de la orientación: el caso individual

El capítulo anterior se dio una introducción a los sistemas de coordenadas, las diferentes parametrizaciones de la orientación, el modelo matemático de los sensores inerciales. Ahora aplicando estos conceptos se pretende diseñar un sistema de *AHRS* (del inglés: *Attitude and Heading Reference System*). Un AHRS es un sistema electrónico basado en sensores inerciales, magnetómetros y un sistema de cómputo embebido sobre el cual se implementa un algoritmo capaz de proporcionar información completa de la orientación del vehículo sobre el que se encuentra abordo, a partir de las mediciones de los girómetros, acelerómetros y magnetómetros [13], como se observa en la Figura 3.1.

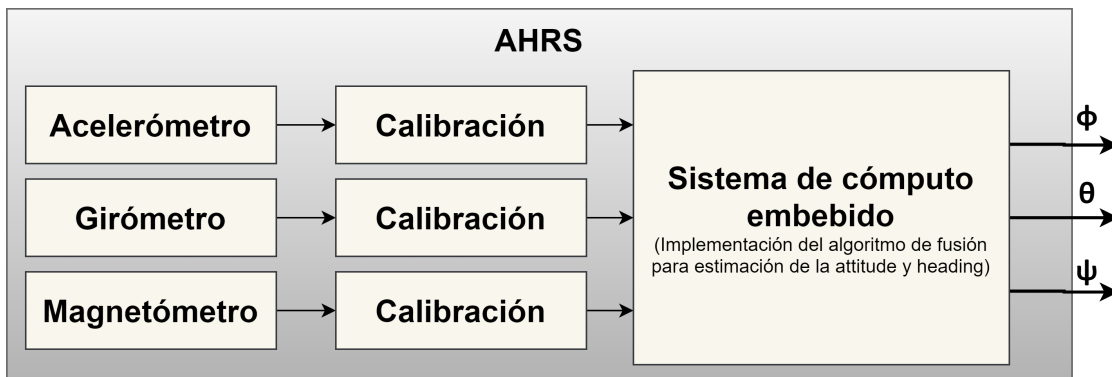


Figura 3.1: Diagrama de bloques de un Attitude and Heading Reference System

En términos de los ángulos de Euler, la representación de la orientación viene dada por: ϕ , θ que son llamados *attitude* (inclinación) y por el ángulo ψ llamado *heading* (dirección) [44]. Existen diversas maneras de determinar la orientación a partir de sensores inerciales y magnetómetros, una manera sencilla de determinar la *attitude* es a partir de la medición de la proyección del vector de gravedad en el sistema de referencia del vehículo y que es proporcionada por los acelerómetros, los ángulos ϕ y θ , son calculados haciendo uso de funciones trigonométricas en base a la proyección del vector de gravedad, este método es conocido como solución por arco tangente. Sin embargo este método requiere que el vehículo se encuentre en un estado casi estacionario, ya que las aceleraciones afectarían la estimación de estos ángulos, además de que los acelerómetros son

muy sensibles a vibraciones, por estas razones este método es poco recomendado en aplicaciones que implican movimiento constante.

El *heading* puede ser calculado a partir de las mediciones del campo magnético terrestre y haciendo uso de las mediciones del *attitude*, sin embargo es importante considerar que esta medición puede verse corrompida por campos magnéticos generados por aparatos eléctricos y por metales cercanos al sensor. El girómetro ofrece una solución completa para la determinación del *attitude* y *heading*, mediante la integración del vector de velocidad angular, sin embargo como se revisó en la sección de sensores, los girómetros sufren de deriva que es variante en el tiempo por lo que la estimación de los ángulos ϕ, θ, ψ se hace ineficiente pasado un determinado tiempo.

Considerando que la estimación de la orientación se ve comprometida si se utilizan acelerómetros-magnetómetros y girómetros por separado debido a las diversas deficiencias que estos presentan, es una buena práctica realizar la integración o fusión de las mediciones de los tres sensores. Los girómetros ofrecen una buena respuesta en movimientos de alta frecuencia, es decir no se ven afectados por vibraciones, sin embargo, tienden a la deriva en su estimación a corto plazo por el contrario los acelerómetros tienen mejor respuesta a bajas frecuencias y no presentan problemas de deriva, sin embargo, son sensibles a fuentes de vibración, lo que afectaría la estimación de la orientación. Se podrá notar que ambos sensores se pueden complementar para que de alguna manera se puedan solventar las deficiencias que presenta cada uno de ellos, mediante un esquema de fusión de la estimación de la orientación ofrecida por cada triada de sensores.

El AHRS implementa este tipo de fusión sensorial, donde se combinan la información provista por los acelerómetros, magnetómetros y por el girómetro [45]. Dadas las mediciones características de los acelerómetros y magnetómetros, que son capaces de medir el vector de aceleración gravitacional y el vector de campo magnético terrestre, para el presente trabajo, los sensores se dividirán en dos categorías:

1. **Sensor de velocidad angular:** El sensor de velocidad angular consiste en el girómetro, el cual ofrece mediciones en sus tres ejes sensibles mutuamente ortogonales, el modelo matemático para este sensor fue presentado en la ecuación (2.29), pero es reescrita nuevamente aquí utilizando la notación dada en [13].

$$\vec{\omega}_G = \vec{\omega} + \vec{v} + \vec{\eta}_G \quad (3.1)$$

$$\dot{\vec{v}} = -\mathbf{T}^{-1}\vec{v} + \vec{\eta}_v \quad (3.2)$$

Donde $\vec{\eta}_G$ y $\vec{\eta}_v \in \mathbb{R}^3$, son ruido blanco asociado al sensor y al modelo del bias respectivamente, $\mathbf{T} \in \mathbb{R}^{3 \times 3}$ es una matriz diagonal de constantes de tiempos de bias. La ecuación (3.2), corresponde al modelo del bias, debido a que este es variante con el tiempo deberá de ser estimado para después poder eliminarlo de la medición del sensor.

2. **Sensores de vectores de referencia:** Como se mencionó anteriormente, estos se basan en las mediciones de sensores capaces de medir el vector de dirección de referencias bien conocidas, como el vector de gravedad, y el vector de campo magnético.

Como se estudió en la sección 2.3 del Capítulo 2, existen diversos métodos de parametrizar la orientación de un cuerpo rígido, sin embargo, en este trabajo se ha decidido utilizar la parametrización por cuaterniones, debido a que es un método ampliamente utilizado y recomendado ya que evita las singularidades que los ángulos de Euler sufren, y requiere de un menor costo computacional a comparación de la matriz de rotación, donde deben de calcularse los nueve elementos de la matriz. A partir de los cuaterniones es posible obtener los ángulos de Euler que son más intuitivos y fáciles de comprender, por lo que esto representa una ventaja en el diseño y desarrollo del AHRS.

3.1. Planteamiento del problema

Dado un sistema autónomo como por ejemplo la boya marina que se instrumentará en este trabajo de tesis, la determinación de la orientación de la boya es relevante debido a que estas señales serán utilizadas en otros bloques de guiado y control del sistema, que contribuirán a la autonomía de la misma. Por lo que el objetivo en primer lugar, es diseñar un observador no lineal para la estimación de la orientación de un cuerpo rígido (boya marina), haciendo uso de sensores de bajo costo, en este caso: acelerómetro, magnetómetro y girómetro todos de tres ejes. El observador no lineal permitirá fusionar las buenas características de un algoritmo determinista que hace uso de vectores de observación y ofrece una buena solución a largo plazo, a la vez que se hace uso del girómetro empleando para ello la ecuación cinemática, que ofrece una buena solución a corto plazo.

Como se reviso en secciones anteriores, el modelo que describe las mediciones de cada uno de los sensores antes mencionados está contaminado con ruido y bias, lo que resulta en una divergencia lenta al integrar la señal del sensor como es el caso del girómetro. Por lo tanto el observador no lineal deberá de tomar en cuenta el modelo del bias con la finalidad de estimar esta señal para restarla de la medición del girómetro y de esta manera evitar la divergencia de la solución final.

3.2. Estimación de la orientación con algoritmo determinista

Con el propósito de determinar la orientación de un cuerpo rígido haciendo uso de vectores de observación que se encuentren solidarios a dicho cuerpo, se toma como base el algoritmo descrito en la sección 2.5.1 y reportado en [13], [32]. Por consiguiente, considere nuevamente el modelo de medición (2.35), y recuerde que los vectores de observación empleados en esta tesis son las mediciones dadas por el acelerómetro y magnetómetro ambos de tres ejes, por lo que el modelo de medición para este par de vectores de observación viene dado por,

$$\bar{H} = \begin{pmatrix} 0 & -(\vec{b}_A - \vec{r}_A)^T \\ \vec{b}_A - \vec{r}_A & -[(\vec{b}_A + \vec{r}_A)^\times] \\ 0 & -(\vec{b}_M - \vec{r}_M)^T \\ \vec{b}_M - \vec{r}_M & -[(\vec{b}_M + \vec{r}_M)^\times] \end{pmatrix} = \begin{pmatrix} \varphi_1^T(t) \\ \vdots \\ \varphi_8^T(t) \end{pmatrix} \mathbf{Q} = 0, \quad \bar{H} \in \mathbb{R}^{8 \times 4} \quad (3.3)$$

donde \vec{b}_A y \vec{b}_M , corresponden a los vectores de observación dados por las mediciones del acelerómetro y magnetómetro respectivamente y \vec{r}_A y \vec{r}_M son los vectores de referencia, que corresponden al sistema de coordenadas inercial \mathbf{E}^f expresado en el sistema NED . Con la finalidad de implementar el algoritmo, se considera el siguiente origen de sistema \mathbf{E}^f , localizado en: Chiapas, México (GPS: latitud 16.169901, longitud -93.084054, altitud 551.6 m), luego, el vector de referencia $\vec{r}_A = [0 \ 0 \ 1]^T$ corresponde al vector de gravedad normalizado expresado en \mathbf{E}^f ; el vector de referencia $\vec{r}_M = [0.28 \ 0 \ 0.1950]$ corresponde al vector de campo magnético normalizado expresado en \mathbf{E}^f . Por lo tanto, considerando nuevamente el modelo de medición (3.3), se retoma la ecuación (2.37) y se obtiene el siguiente algoritmo iterativo para solucionar el sistema de ecuaciones lineales sobre-determinado dado por la matriz \bar{H} :

$$\mathbf{Q}_{ps}^i = \left[\prod_{i=1}^8 \left(I_4 - \frac{\gamma \varphi_i(t) \varphi_i^T(t)}{\alpha + \varphi_i^T(t) \varphi_i(t)} \right) \right] \mathbf{Q}_{ps}^{i-1} \quad (3.4)$$

Como se explico anteriormente, el método Kaczmarz opera en la i -ésima fila de la matriz \bar{H} durante la i -ésima iteración, además de que requiere de una condición inicial para iniciar el

3.3.3. ESTIMACIÓN DE LA ORIENTACIÓN IMPLEMENTANDO UN OBSERVADOR NO LINEAL

ciclo de iteración. Una forma sencilla de implementar el algoritmo iterativo (3.4), es desarrollando un pequeño programa en MATLAB, el cuál se encuentra orientado al trabajo con matrices, el pseudocódigo 1 resume la implementación del algoritmo LQS, para encontrar el cuaternión \mathbf{Q}_{ps} que mapea del sistema de coordenadas \mathbf{E}^b al sistema de coordenadas \mathbf{E}^f .

Algorithm 1 LQS algorithm

Input: $\vec{b}_A, \vec{r}_A, \vec{b}_M, \vec{r}_M, \mathbf{Q}_{est}$
Output: q_{est}
 $\vec{b}_A := \frac{\vec{b}_A}{\|\vec{b}_A\|}$ and $\vec{b}_M := \frac{\vec{b}_M}{\|\vec{b}_M\|}$
 $\vec{r}_A := \frac{\vec{r}_A}{\|\vec{r}_A\|}$ and $\vec{r}_M := \frac{\vec{r}_M}{\|\vec{r}_M\|}$
Form the matrix \bar{H} applying equation (3.3)
 $\mathbf{Q}_{ps} = \text{zeros}(4, 9); \% \text{ is a matrix of } 4 \times 9$
 $\mathbf{Q}_{ps}(:, 1) := \mathbf{Q}_{est};$
 $\alpha := 0.8, \gamma := 0.8;$
 $I_4 := \text{eye}(4);$
for $i = 2:9$ **do**
 $\mathbf{Q}_{ps}(:, i) := \left(I_4 - \frac{\gamma * \bar{H}(i-1,:)^T * \bar{H}(i-1,:)}{\alpha + \bar{H}(i-1,:) * \bar{H}(i-1,:)^T} \right) * \mathbf{Q}_{ps}(:, i-1);$
end for
 $\mathbf{Q}_{est} := \mathbf{Q}_{ps}(:, 9);$
 $\mathbf{Q}_{est} := \frac{\mathbf{Q}_{est}}{\|\mathbf{Q}_{est}\|};$

$\mathbf{Q}_{ps}(:, 1)$ puede ser inicializado con un valor de $[1 \ 0 \ 0 \ 0]$ luego \mathbf{Q}_{est} es usado en el siguiente tiempo de muestreo para asignar el nuevo valor de $\mathbf{Q}_{ps}(:, 1)$.

El pseudocódigo 1, fue implementado en Simulink haciendo uso de bloques de función de MATLAB (*MATLAB Function*). Los resultados y detalles de la implementación son mostrados en la sección 3.4 que corresponden a los resultados del capítulo. En la siguiente sección se aborda la implementación del observador no lineal, donde el bias del girómetro es estimado y el cuaternión que parametriza la orientación del cuerpo rígido es estimado realizando para ello una comparación entre el $\hat{\mathbf{Q}}$ dado por la ecuación cinemática y el \mathbf{Q}_{ps} dado por el algoritmo LQS, donde el cuaternión de error es obtenido y utilizado para la convergencia del $\hat{\mathbf{Q}}$ y la estimación del bias $\hat{\nu}$.

3.3. Estimación de la orientación implementando un observador no lineal

Como se menciona anteriormente, el objetivo del observador no lineal, es hacer una estimación del cuaternión a partir de la integración de la ecuación cinemática del cuerpo rígido, a la vez que hace una comparación con el cuaternión obtenido del algoritmo *LQS*, para obtener el cuaternión de error, el cual es usado en la estimación del bias del girómetro lo que resulta en una solución sin divergencia del cuaternión estimado.

Considere nuevamente el observador no lineal presentado en (2.48), el cual es reescrito a continuación:

$$\mathcal{O} = \begin{cases} \dot{\hat{\mathbf{Q}}}(t) = \frac{1}{2} \Xi(\hat{\mathbf{Q}}(t)) \left(\vec{\omega}_G(t) - \hat{\nu}(t) + K_1 \vec{q}_e(t) \right) \\ \dot{\hat{\nu}}(t) = -T^{-1} \hat{\nu}(t) - K_2 \vec{q}_e(t) \end{cases} \quad (3.5)$$

donde $\hat{\mathbf{Q}}$ es el cuaternión de orientación estimado en el instante t , el cual es obtenido mediante la

integración de la ecuación cinemática que hace uso del vector de velocidad angular $\vec{\omega}_G \in \mathbb{R}^3$ dado por las mediciones del girómetro a las que se le restan el bias estimado $\hat{\vec{v}} \in \mathbb{R}^3$ y se le suma un término de error \vec{q}_e que resulta de la comparación de $\hat{\mathbf{Q}}$ y \mathbf{Q}_{ps} , K_1 y K_2 son constantes positivas y $T \in \mathbb{R}^{3 \times 3}$ es la matriz de constantes de tiempos.

Recuerde que,

$$\dot{\mathbf{Q}}(t) = \frac{1}{2} \mathbf{Q} \otimes \mathbf{Q}(\omega)(t) = \frac{1}{2} \begin{pmatrix} -\vec{q}^T \\ I_3 q_0 + [\vec{q}^\times] \end{pmatrix} \vec{\omega} = \frac{1}{2} \Xi(\mathbf{Q}) \vec{\omega} \quad (3.6)$$

$$\mathbf{Q}_e = \hat{\mathbf{Q}}^{-1} \otimes \mathbf{Q}_{ps} = (q_{e0} \ \vec{q}_e^T)^T \quad (3.7)$$

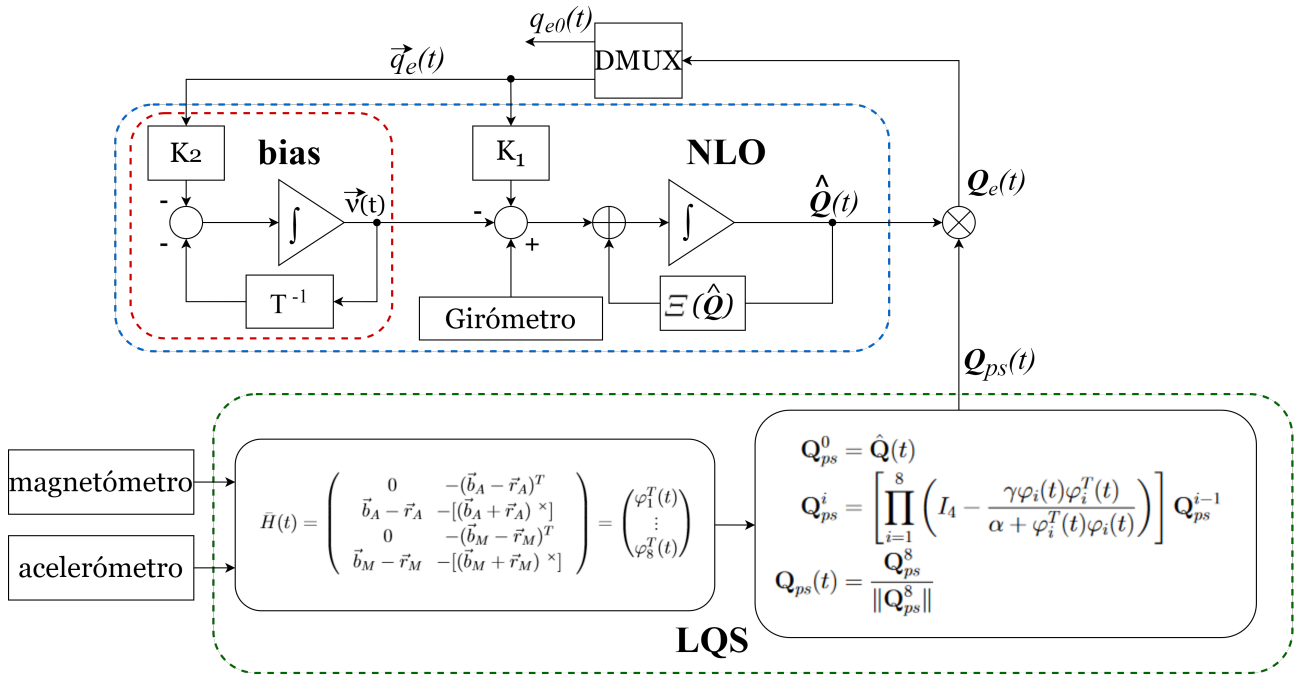


Figura 3.2: Diagrama de bloques del observador no lineal (3.5) (\otimes representa la multiplicación cuaternión y \oplus la multiplicación matricial).

La Figura 3.2, ilustra a manera de diagramas de bloques la ecuación (3.5) que corresponde al observador no lineal, aquí puede observarse de manera clara como se da el proceso de interacción entre la integración de la ecuación cinemática, la estimación del bias, la estimación de \mathbf{Q}_{ps} a partir del algoritmo *LQS* y el cálculo del cuaternión de error \mathbf{Q}_e . El pseudocódigo 2 únicamente indica la asignación de valores iniciales y constantes, así como la formación de algunas funciones necesarias para formar el diagrama de bloques como aparece en la Figura 3.2. A continuación en la sección 3.4, se presentan los resultados obtenidos en la estimación del cuaternión de orientación a partir del algoritmo *LQS*, la integración de la ecuación cinemática (3.6) y el observador no lineal.

3.4. Resultados del capítulo

En esta sección se presenta el procedimiento de calibración de los sensores utilizados en un AHRs y la evaluación de los métodos de estimación de orientación vistos en este capítulo y que

Algorithm 2 Non-linear observer (Simulink implementation)

Input: $\vec{\omega}_G$ **Output:** $\hat{\mathbf{Q}}$ Form the matrix Ξ applying equation (3.6)Assign $K_1 := 1.75$ Assign $K_2 := 0.75$ Assign $T := 100$ and form the diagonal matrixForm the function to calculate q_e applying equations (2.17), (2.19) and (3.7)Assign $\hat{\mathbf{Q}}^0 := [1 \ 0 \ 0 \ 0]$ For LQS algorithm assign $\mathbf{Q}_{ps}^0 := \hat{\mathbf{Q}}(t)$ Obtain \mathbf{Q}_{ps}

serán la base para los siguientes. En particular se evaluará el algoritmo *LQS*, el cual hace uso solo de las medidas de acelerómetros y magnetómetros, la integración de la ecuación cinemática, que hace solo uso de los girómetros y finalmente el observador no lineal, que permite fusionar los tres tipos de sensores. También, se presenta el desarrollo e implementación de una plataforma experimental, la cual facilitó la prueba experimental y análisis de los algoritmos de estimación propuestos en esta tesis. Cabe mencionar que los resultados de este capítulo son experimentales.

3.4.1. Descripción de la plataforma experimental

La instrumentación usada durante el procedimiento experimental para la evaluación de la estimación del cuaternión, se compone de: un microcontrolador basado en la tarjeta de desarrollo STM32F411 Nucleo-64. El microcontrolador STM32F411RET6 tiene un alto desempeño con un buen balance en consumo energético, y fue dispuesto a una frecuencia de 100 MHz. Contiene capacidades de procesador digital de señales (DSP), tiene una arquitectura de 32-bits y cuenta con unidad de punto flotante. Cuenta con los periféricos necesarios (I²C y USART) para establecer comunicación con los dispositivos sensores y la PC. El conjunto de sensores esta basado en el módulo Gy-87, que consta de un sensor MPU6050, con acelerómetro y girómetro ambos de tres ejes y un magnetómetro basado en el compás digital HMC5883L, donde cada uno de ellos se comunica vía I²C.

3.4.2. Calibración de los sensores

A continuación se describe el procedimiento utilizado para calibrar los diferentes sensores utilizados en este trabajo de tesis. Además, como se observa en la Figura 3.1, las mediciones de los sensores deberán de ser calibradas antes de realizar cualquier procesamiento de las mismas, ya que como se reviso en la sección 2.4, los sensores inerciales de bajo costo sufren de una serie de errores que comprometen la fiabilidad de sus mediciones, entre estos errores, los más típicos son, error en el factor de escala y el error de offset.

Calibración del girómetro

Para el caso del girómetro utilizado, presenta un offset considerable el cual requiere ser reducido como se observa en la Figura 3.3a, por lo que el procedimiento de calibración del offset esta basado en el trabajo de tesis realizado en [46], el cual consiste en tomar un número alto de muestras de las mediciones del girómetro en cada uno de sus ejes, mientras este se encuentra estático, es decir sin sufrir movimiento alguno, para posteriormente realizar un promedio aritmético de las mediciones correspondientes a cada eje, el resultado de los promedios obtenidos, serán los offsets

3. 3.4. RESULTADOS DEL CAPÍTULO

correspondientes a cada eje, por lo que habrá que restarlos de las mediciones posteriores del girómetro. En la ecuación (3.8), se resume el proceso para encontrar el offset correspondiente a cada eje del girómetro.

$$offset_X = \sum_{k=1}^n \frac{\omega_{x,k}^*}{n}, \quad offset_Y = \sum_{k=1}^n \frac{\omega_{y,k}^*}{n}, \quad offset_Z = \sum_{k=1}^n \frac{\omega_{z,k}^*}{n} \quad (3.8)$$

Donde k es una muestra en un instante dado de tiempo y n es el número total de muestras.

Como se observa en la Figura 3.3b, el offset del girómetro es ampliamente reducido, hasta aproximadamente un factor de $\frac{1}{10}$.

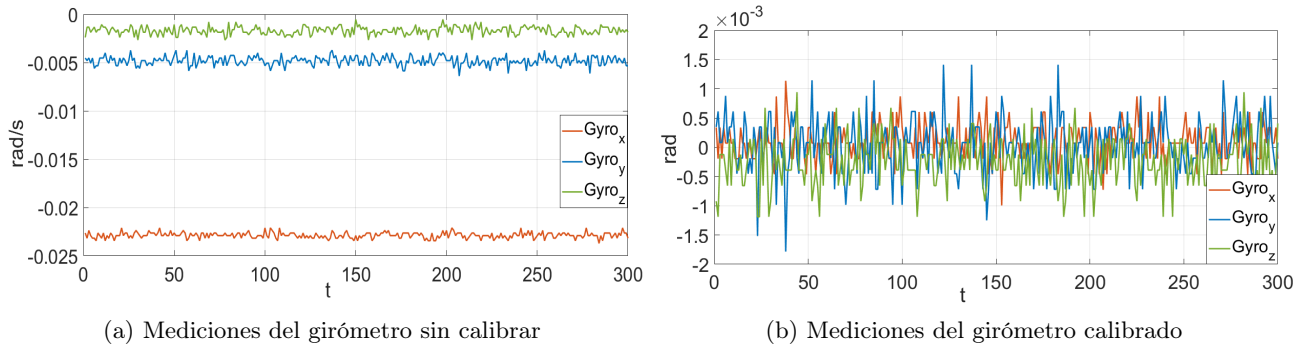


Figura 3.3: Mediciones del girómetro antes y después del proceso de calibración

Descripción	Especificaciones
Sensor	MPU6050 (Girómetro)
Grados de libertad	3
Rango	$250^\circ/s$, $500^\circ/s$, $1000^\circ/s$, $2000^\circ/s$
Comunicación	I ² C
Salida	digital
V. operación	$3V/3.3V \approx 5V$ DC
Dirección I ² C	0x68

Tabla 3.1: Características del girómetro.

Como se observa en la tabla 3.1, el fabricante proporciona las mediciones del girómetro en grados por segundo ($^\circ/s$), sin embargo para el presente trabajo se requiere que las mediciones estén representadas en radianes por segundo (rad/s). Por consiguiente, se requiere realizar una conversión de las mediciones brutas dadas por el girómetro a radianes antes de realizar la calibración, como rango de medición del girómetro se eligió ($2000^\circ/s$), por lo tanto el proceso de calibración para uno de los ejes estaría dado por:

$$\begin{aligned}\omega_x^* &= \left(\omega_g * \frac{\pi}{180} * \frac{1}{16.4}\right) \\ \omega_x &= \omega_x^* - offset_X\end{aligned}\tag{3.9}$$

donde ω_x^* es la conversión a radianes de la medición del girómetro, ω_g es la medición del girómetro y ω_x es la medición del girómetro después del proceso de calibración.

Calibración del acelerómetro

Para el proceso de detección de los errores de escala y offset, y el proceso de calibración, se utilizo el procedimiento desarrollado en [46], donde nos indica que para saber si el acelerómetro tiene problemas de offset, se deben de comparar las mediciones positivas y negativas correspondientes a cada eje alineado con el vector de aceleración gravitacional en condiciones estáticas, en el caso de no existir offset, estas mediciones deberán de ser iguales en valor absoluto, de lo contrario se asume que existe un offset presente en la medición. Para el factor de escala, este es fácilmente detectable al observar las mediciones del acelerómetro en cada uno de sus ejes, las cuales en condiciones estáticas, deberían de llegar a un aproximado de $1g = 9.81m/s^2$, en el caso de estar muy por debajo o encima de este valor, se considera que existe un error de escala en el sensor.

Para el sensor utilizado MPU6050, se detectaron problemas de escala y offset, por lo cual requieren de ser compensados. Para el caso del offset, el proceso de calibración como se indica en [46], consiste en tomar una alta cantidad de muestras en cada uno de los ejes positivos y negativos, alineados con el vector de aceleración gravitacional, por lo que se realizan un total de 6 mediciones. De estas muestras se extraen el valor más grande, medido en cada uno de los ejes positivos y el valor más pequeño medido en cada uno de los ejes negativos, y se realiza el procedimiento descrito en la ecuación (3.10). Los valores obtenidos deberán de ser restados de las mediciones del acelerómetro en su correspondiente eje.

$$offset_X = \frac{g_{maxx} + g_{minx}}{2}, \quad offset_Y = \frac{g_{maxy} + g_{miny}}{2}, \quad offset_Z = \frac{g_{maxz} + g_{minz}}{2}\tag{3.10}$$

Una vez calculado el offset correspondiente a cada eje, el factor de escala correspondiente a cada eje es calculado como se indica en la ecuación (3.11).

$$sf_X = \frac{g}{g_{maxx} - offset_X}, \quad sf_Y = \frac{g}{g_{maxy} - offset_Y}, \quad sf_Z = \frac{g}{g_{maxz} - offset_Z}\tag{3.11}$$

Una vez compensados los offsets para la medición de cada eje del acelerómetro, esta deberá de ser multiplicada por su correspondiente factor de escala. En la Figura 3.4, se muestran las mediciones tomadas antes del proceso de calibración y después del mismo, para el eje z positivo, coincidente con el vector de aceleración gravitacional.

Calibración del magnetómetro

El magnetómetro, al igual que el acelerómetro, es caracterizado por errores estáticos, los cuales pueden ser calculados, para posteriormente ser compensados en la medición. El proceso de calibración del magnetómetro es muy similar al del acelerómetro, pero a diferencia de este último, durante la adquisición de las muestras de medición del magnetómetro, este deberá de ser rotado en todas las direcciones posibles. Este proceso es indicado en [46], las fórmulas para el cálculo

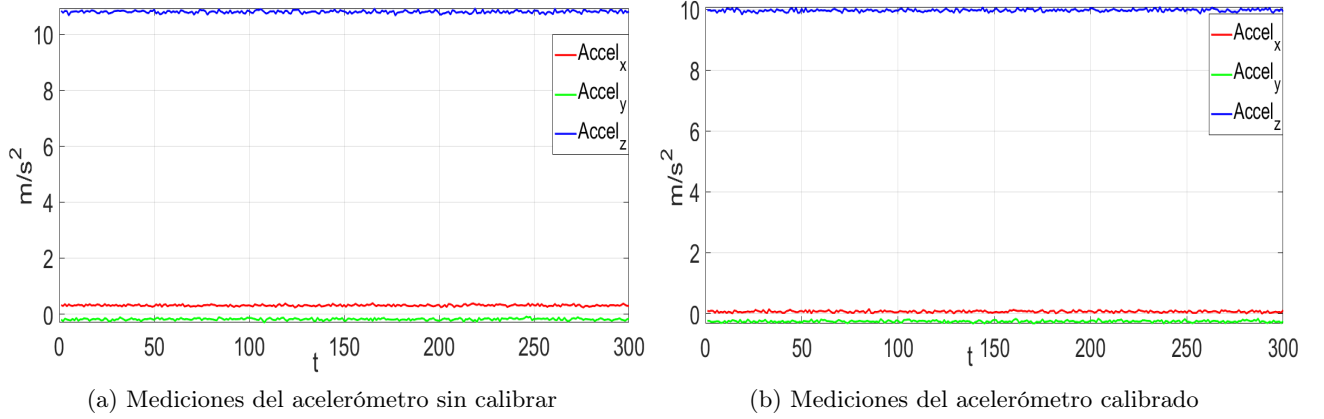


Figura 3.4: Mediciones del acelerómetro antes y después del proceso de calibración

Descripción	Especificaciones
Sensor	MPU6050 (Acelerómetro)
Grados de libertad	3
Rango	2g, 4g, 8g, 16g
Comunicación	I ² C
Salida	digital
V. operación	3V/3.3V ≈ 5V DC
Dirección I ² C	0x68

Tabla 3.2: Características del acelerómetro.

del offset y factor de escala correspondiente a cada eje se indica en las ecuaciones (3.12) y (3.13) correspondientemente.

$$offset_X = \frac{E_{maxx} + E_{minx}}{2}, \quad offset_Y = \frac{E_{maxy} + E_{miny}}{2}, \quad offset_Z = \frac{E_{maxz} + E_{minz}}{2} \quad (3.12)$$

$$sf_X = \frac{m}{E_{maxx} - offset_X}, \quad sf_Y = \frac{m}{E_{maxy} - offset_Y}, \quad sf_Z = \frac{m}{E_{maxz} - offset_Z} \quad (3.13)$$

donde E es el campo magnético de la Tierra medido por los sensores en su correspondiente eje y m es el máximo valor del campo magnético terrestre correspondiente a la zona del estado de Chiapas, México que es de aproximadamente 0.40 Gauss . Al realizar el proceso de calibración, el magnetómetro deberá estar alejado de objetos como computadoras, metales, etc. que puedan causar perturbaciones en las mediciones. Al igual que con el acelerómetro, a las mediciones del magnetómetro habrá primero que restarles el offset y posteriormente se multiplicará por su correspondiente factor de escala.

En la Figura 3.5, se observan las mediciones del magnetómetro tomadas antes de la calibración, la esfera que se forma a partir de las mediciones, debería estar centrada con origen $(0, 0, 0)$, sin embargo existe un pequeño desplazamiento que habrá que compensar. En la Figura 3.6, se muestran

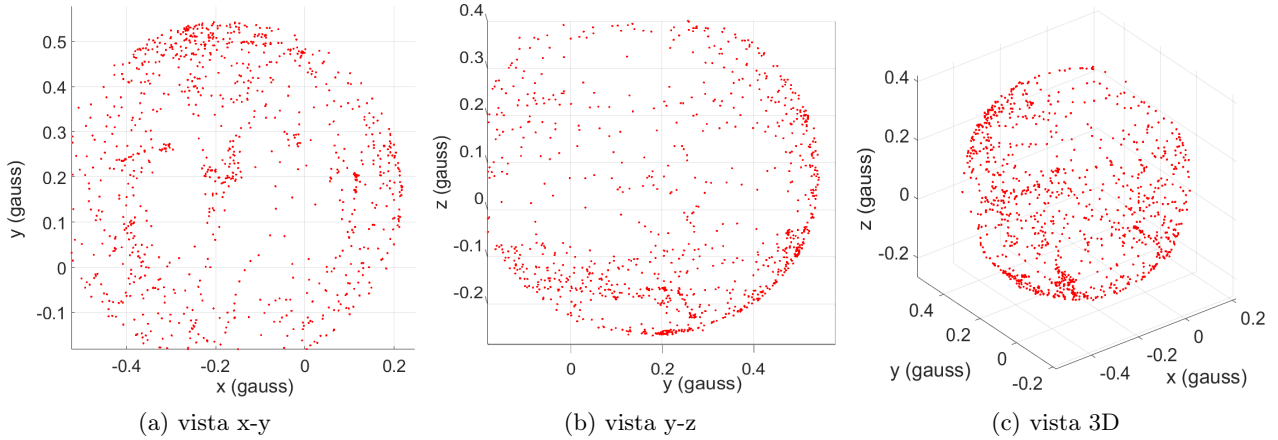


Figura 3.5: Mediciones del magnetómetro sin calibrar

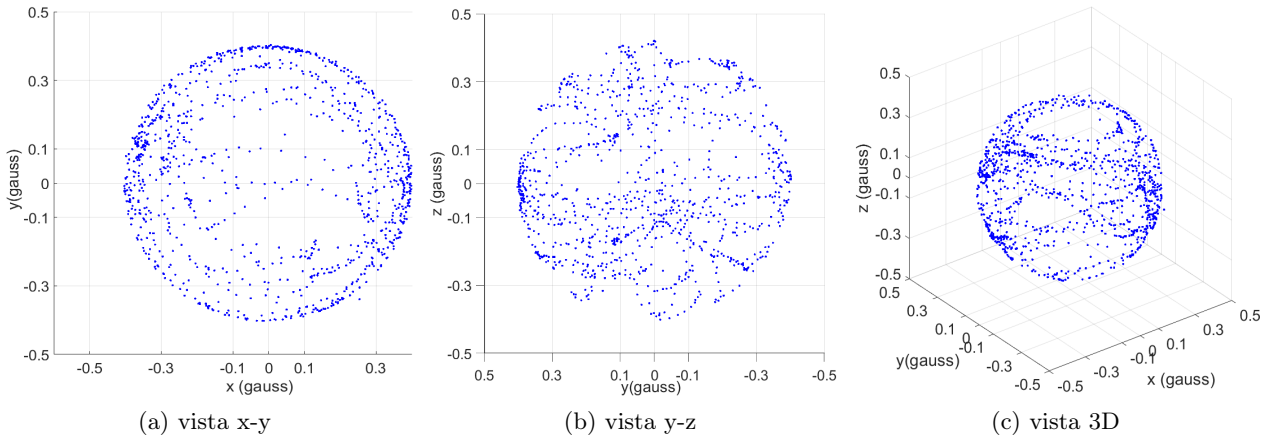


Figura 3.6: Mediciones del magnetómetro calibradas

las mediciones del magnetómetro tomadas después del proceso de calibración, en estas figuras se puede apreciar como el desplazamiento ha sido minimizado y la esfera tiene un origen muy cercano al $(0, 0, 0)$.

Es importante tener en cuenta que el magnetómetro deberá de ser recalibrado al ser utilizado en una zona geográfica diferente a donde había sido calibrado anteriormente, debido a que la magnitud máxima de campo magnético medible varía dependiendo de la zona geográfica.

3.4.3. Implementación de los algoritmos de estimación de la orientación

En esta subsección se presenta una implementación práctica del algoritmo *LQS*, la ecuación cinemática del cuerpo rígido y del observador no lineal, antes presentados. Esta implementación es muy importante, debido a que permite comprobar el funcionamiento de los algoritmos antes mencionados de una manera rápida y eficiente, haciendo uso de la información obtenida de sensores reales, que es procesada por un microcontrolador y luego las mediciones son transmitidas a una computadora donde se ejecuta cada algoritmo en el ambiente de MATLAB/Simulink. Esta prueba experimental, es lo que se conoce como **HIL** (del inglés, *Hardware In the Loop*), ya que nos

3. 3.4. RESULTADOS DEL CAPÍTULO

Descripción	Especificaciones
Sensor	HMC5883L (Magnetómetro)
Grados de libertad	3
Rango	$\pm 0.88Ga, \pm 1.3Ga, \pm 1.9Ga, \pm 2.5Ga, \pm 4.0Ga, \pm 4.7Ga, \pm 5.6Ga, \pm 8.1Ga$
Comunicación	I ² C
Salida	digital
V. operación	3.3V DC
Dirección I ² C	0x1E

Tabla 3.3: Características del magnetómetro.

permite probar un sistema complejo con datos reales, y hacer las modificaciones necesarias antes de llegar a su implementación física.

Algoritmo LQS

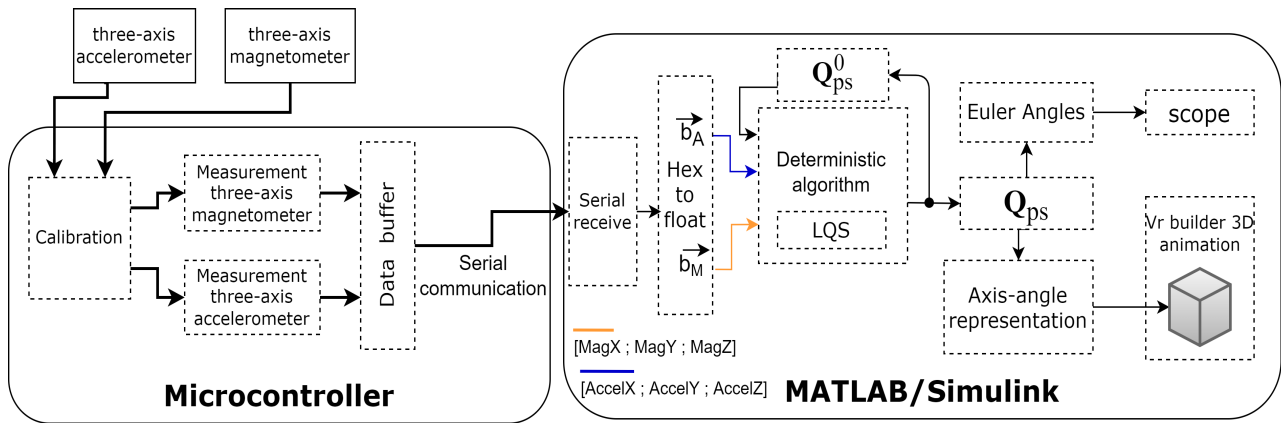


Figura 3.7: Diagrama de bloques de la implementación del algoritmo LQS en Simulink.

La Figura 3.7, presenta a manera de diagrama de bloques la implementación del algoritmo *LQS*, haciendo uso del entorno MATLAB/Simulink para la adquisición de datos y de un microcontrolador para transmisión de las mediciones de cada sensor. Cada bloque interno de la interfaz MATLAB/Simulink fue desarrollado haciendo uso de bloques de funciones, donde los bloques *Serial receive*, *scope* y *Vr builder 3D animation* son herramientas propias de Simulink. El bloque *LQS* se refiere al pseudocódigo 1, *Euler Angles* es el bloque que transforma el cuaternión a la representación de ángulos de Euler, el bloque *Axis-angle representation* convierte el cuaternión a una representación llamada eje-ángulo la cuál es usada por el bloque *Vr builder 3D animation*, finalmente el bloque *Hex to float* convierte la representación en bytes de cada medición a su equivalente en punto flotante, cada uno de estos bloques son detallados en el apéndice B.

La Figura 3.8, muestra los ángulos obtenidos a partir de una secuencia de rotaciones bien definida, donde en un inicio el sistema de referencia móvil \mathbf{E}^b coincide con el sistema de referencia inercial \mathbf{E}^f , en consecuencia los ángulos estimados son: $\phi = 0$, $\theta = 0$ y $\psi = 0$. Luego, el sistema

de referencia móvil es rotado un ángulo de $-\frac{\pi}{2}$ rad en ψ , seguido por una rotación de $-\frac{\pi}{2}$ rad en ϕ , luego por una rotación de $\frac{\pi}{2}$ rad en ψ y finalmente una rotación de $\frac{\pi}{2}$ rad en ϕ .

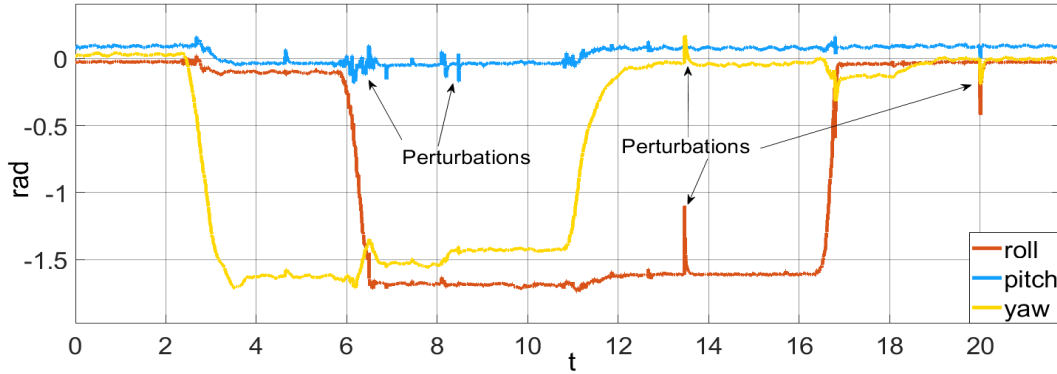


Figura 3.8: Ángulos de Euler obtenidos de una secuencia de rotaciones bien definida.

En cada una de las tres señales puede observarse una pequeña perturbación debida posiblemente a la sensibilidad que presenta el acelerómetro frente a vibraciones.

Ecuación cinemática

En la Figura 3.9, se presenta el diagrama de bloques de la implementación de la ecuación cinemática (2.24), con la finalidad de obtener el cuaternión que representa la orientación del cuerpo rígido. Los bloques restantes son los mismos descritos para la Figura 3.8, que son detallados en el apéndice B.

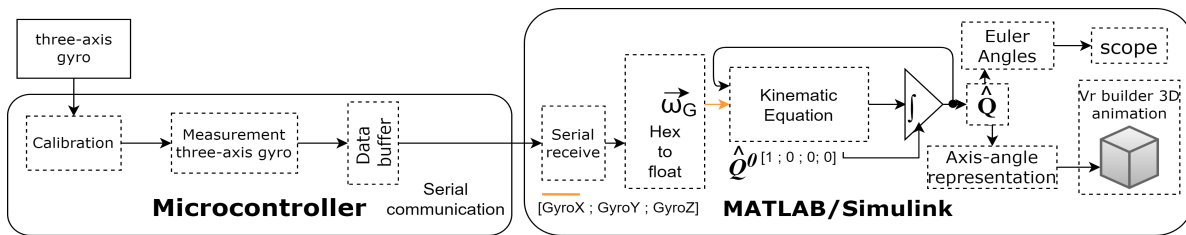


Figura 3.9: Diagrama de bloques de la implementación de la ecuación cinemática en Simulink.

En la Figura 3.10a, se observa la deriva en los ángulos de Euler debido a la integración del bias que afecta a cada eje del girómetro, como consecuencia, los ángulos van cambiando (derivando), conforme pasa el tiempo a pesar de que el cuerpo rígido se encuentre en reposo. Esto afectará en la estimación de los ángulos a largo plazo, como se observa en la Figura 3.10b, donde se realizan las mismas series de rotaciones hechas para el algoritmo *LQS*. Sin embargo hay que tener en cuenta que al utilizar la ecuación cinemática para la estimación del cuaternión, únicamente se utilizan las mediciones del girómetro, por lo que el sistema de referencia inercial \mathbf{E}^f estará definido de igual manera al sistema de referencia propio del girómetro. Para la Figura 3.10b, se observa ausencia de ruido en cada una de las señales, sin embargo, es evidente como la estimación de los ángulos de Euler son deficientes debido al termino de bias que conforme transcurre el tiempo afecta en la estimación, esto es evidente al retornar el cuerpo rígido a su posición inicial, donde los ángulos ϕ, θ, ψ deberían de ser todos iguales a cero, lo cual no ocurre. Otro aspecto importante a considerar

3. 3.4. RESULTADOS DEL CAPÍTULO

al utilizar la ecuación cinemática, es que suponemos que la condición inicial es conocida, lo cuál en la práctica esto no es cierto, de modo que para fines prácticos el uso de únicamente la ecuación cinemática presenta graves desventajas en la estimación de la orientación.

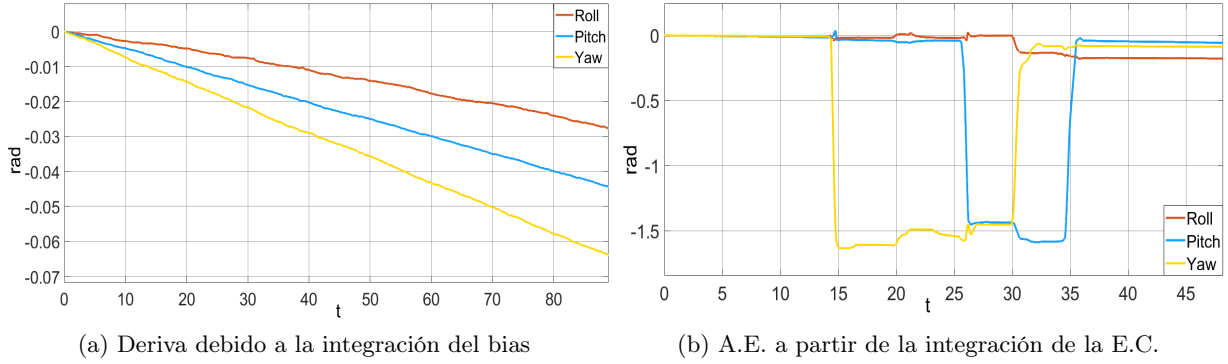


Figura 3.10: Deriva y ángulos de Euler a partir de la integración de la ecuación cinemática.

Observador no lineal

En la Figura 3.11, se muestra el diagrama de bloques de la implementación del observador no lineal en Simulink, para la implementación del observador no lineal deberán de ser consideradas las asignaciones y creaciones de funciones descritas en el pseudocódigo 2, además de que se cuentan con bloques de integración lo que permite una fácil implementación de la ecuación cinemática para la estimación del cuaternión y del bias. Los bloques restantes son los mismos descritos para la Figura 3.8, que son detallados en el apéndice B.

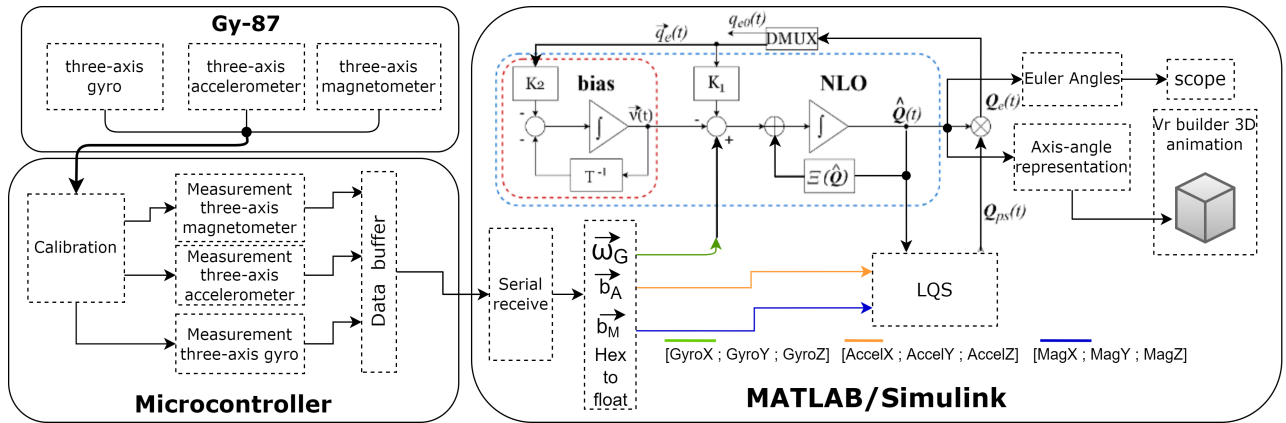


Figura 3.11: Diagrama de bloques de la implementación del observador no lineal en Simulink.

En la Figura 3.12, se muestran los ángulos de rotación obtenidos a partir de una serie de rotaciones bien definidas, esta secuencia de rotaciones consiste en la misma secuencia aplicada para el algoritmo LQS , con la finalidad de realizar una comparación entre ambos resultados. Nuevamente en un inicio el sistema de referencia móvil \mathbf{E}^b coincide con el sistema de referencia inercial \mathbf{E}^f , en consecuencia los ángulos estimados son: $\phi = 0$, $\theta = 0$ y $\psi = 0$. Luego, el sistema de referencia móvil es rotado un ángulo de $-\frac{\pi}{2}$ rad en ψ , seguido por una rotación de $-\frac{\pi}{2}$ rad en ϕ , luego por

una rotación de $\frac{\pi}{2}$ rad en ψ y finalmente una rotación de $\frac{\pi}{2}$ rad en ϕ .

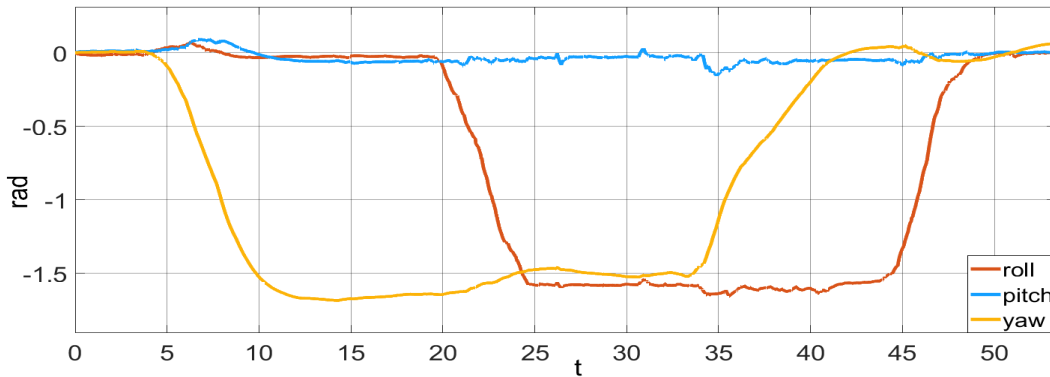


Figura 3.12: Ángulos de Euler obtenidos de una secuencia de rotaciones bien definida a partir del observador no lineal.

De los resultados obtenidos en la Figura 3.12, se observa que cada una de las señales que representan las diferentes direcciones de rotación del cuerpo rígido no son afectadas por ruido o perturbaciones, esto gracias a la acción de filtrado que desarrolla el observador no lineal. El contar con señales libre de ruido y perturbaciones, contribuirá en mejorar la calidad de los sistemas de guiado y control de la boya marina.

En la Figura 3.13, se presenta la estructura general del buffer de datos que contiene las mediciones del acelerómetro, magnetómetro y girómetro, expresados en bytes lo que facilita la transmisión de múltiples datos a través del USART del microcontrolador hacia la computadora (PC). El buffer consta de 38 bytes totales, donde el primer byte corresponde al encabezado, los 12 bytes siguientes corresponden a la medición del acelerómetro en cada uno de sus ejes, seguido por los 12 bytes correspondiente a la medición del magnetómetro en cada uno de sus ejes, finalmente se colocan los 12 bytes que corresponden a la medición del girómetro en cada eje, el último byte corresponde a un indicador de final.

Finalmente, en la Figura 3.14, se muestra el sistema instrumentado por medio del cual fueron obtenidos los resultados mostrados en este capítulo. Este sistema consta de un cubo de rubik, el módulo Gy-87 y la tarjeta de desarrollo STM32F411 Nucleo-64.

En el siguiente capítulo se aborda la problemática de estimación de orientación distribuida, donde el algoritmo *LQS* y el observador no lineal presentados en este capítulo son descritos de manera más general, considerando el uso de múltiples unidades de medición (MU's) que pertenecen a una red de sensores y comparten información entre ellos, además se propone un nuevo algoritmo de consenso dinámico con la finalidad de obtener un único cuaternión a partir del promedio de los cuaterniones estimados por cada MU de la red. Una característica muy importante del nuevo algoritmo de consenso dinámico propuesto, es la capacidad de proveer robustez al sistema, es decir, que ante el posible fallo de alguno de los sensores de la i -ésima MU, el sistema pueda seguir funcionando y estimando de manera correcta el cuaternión de orientación. Finalmente se presentan una serie de simulaciones y resultados experimentales en base al nuevo algoritmo propuesto, con la finalidad de verificar su buen funcionamiento, tanto en condiciones normales (todos los sensores funcionando de manera correcta), como en condiciones cruciales (perdida de uno o más sensores en la red).

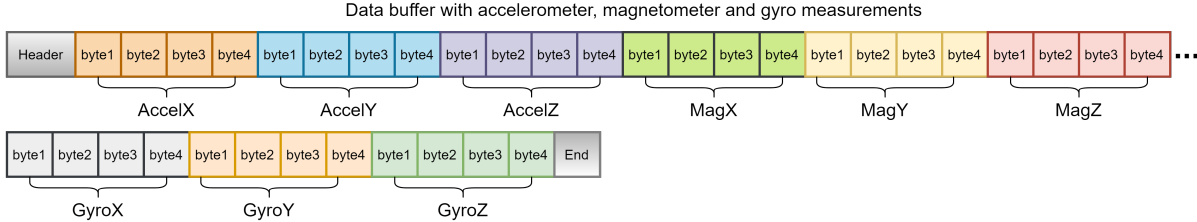


Figura 3.13: Estructura general del buffer de datos.

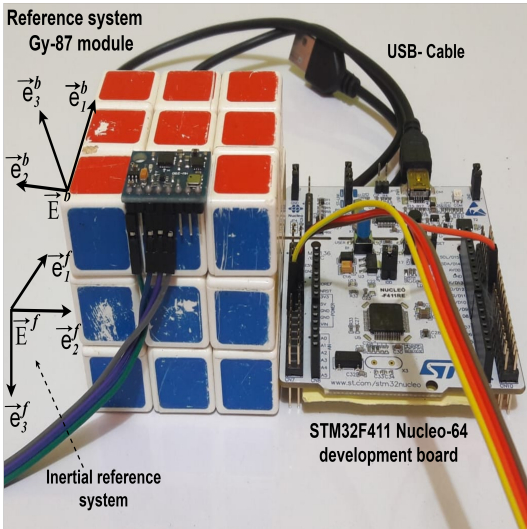


Figura 3.14: Sistema instrumentado para la estimación de la orientación.

Capítulo 4

Estimación distribuida de la orientación

En este capítulo se aborda el problema de la estimación de la orientación de manera distribuida, a partir de un consenso dinámico que promedia la estimación individual realizada por un conjunto de agentes con una topología de comunicación dada por un grafo. Cada agente es capaz de estimar la orientación con respecto a su propio sistema de referencia del mismo cuerpo rígido, y de ejecutar un algoritmo de consenso dinámico, con respecto al sistema de referencia del cuerpo. El algoritmo de consenso dinámico recibe información de las estimaciones individuales y del promedio realizado por los agentes “vecinos”, la estimación obtenida del consenso dinámico, es considerada como la correcta dada por el sistema completo. Otro aspecto importante que contempla la estimación distribuida, es la redundancia en los sensores, lo que contribuye a tener un sistema robusto a fallas, por lo tanto se consideran algunos posibles escenarios de fallas en los sensores con el objetivo de probar la robustez del sistema.

En este capítulo se hace uso de tres sistemas de referencia: el sistema de referencia inercial \mathbf{E}^f , el sistema de referencia del cuerpo \mathbf{E}^b y el sistema de referencia de la unidad de medición (MU) \mathbf{E}^{mu} , cada uno de estos sistemas es representado en la Figura 4.1 con sus respectivos componentes.

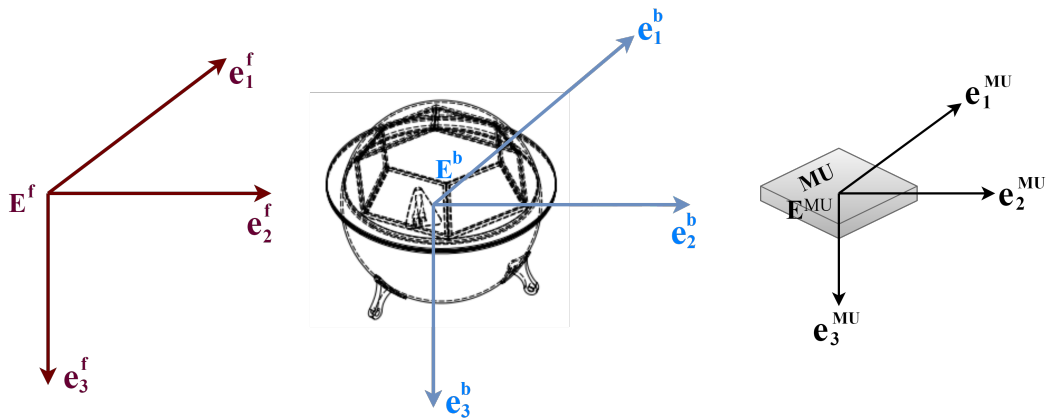


Figura 4.1: Sistemas de referencia \mathbf{E}^f , \mathbf{E}^b y \mathbf{E}^{mu}

En el capítulo anterior se considero que los sistemas de referencia \mathbf{E}^{mu} y \mathbf{E}^b eran coincidentes en cada uno de sus respectivos componentes y en el origen, esta es la forma más usual de posicionar el sistema de referencia de los sensores con respecto al sistema de referencia del cuerpo cuando únicamente se hace uso de una unidad de medición. Esta coincidencia puede ser apreciada en la Figura 4.2, sin embargo, cuando se tiene más de una unidad de medición con la finalidad de estimar la orientación del cuerpo sobre el que se encuentran posicionados, podría ser mejor práctica orientar y posicionar cada uno de los sistemas de referencia \mathbf{E}^{mu_i} , con $i = 1, 2, \dots, n$, de manera que no coincidan entre ellos ni con el sistema de referencia \mathbf{E}^b (ver Figura 4.3). Con el objetivo de obtener una mejor observación de la orientación del cuerpo, y haciendo uso de teoría de grafos se puede realizar un promedio de los diferentes cuaterniones estimados para obtener una mejor estimación de la orientación.

A continuación se describen las diferentes transformaciones entre los tres sistemas de referencia con la finalidad de realizar una simulación de un sistema de estimación de orientación distribuido.

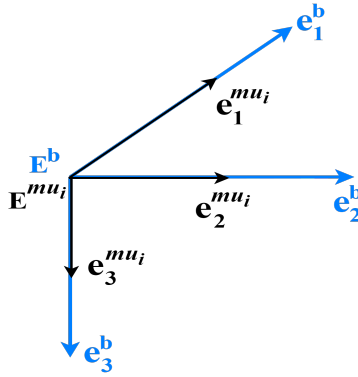


Figura 4.2: Sistemas de referencia \mathbf{E}^{mu_i} y \mathbf{E}^b alineados

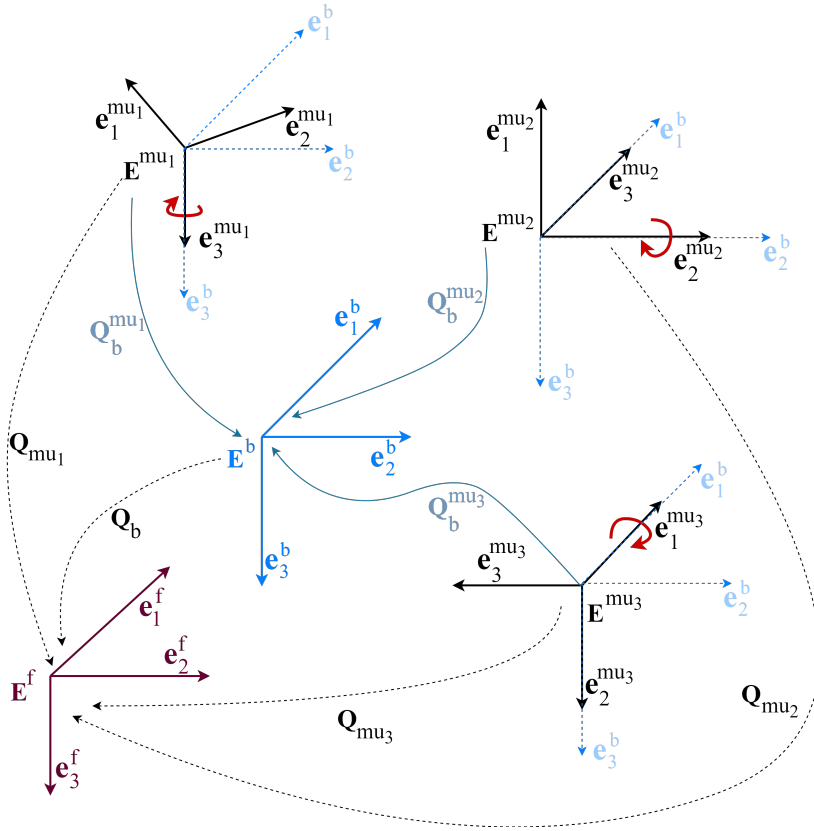
4.1. Rotación del sistema \mathbf{E}^{mu_i} con respecto al sistema \mathbf{E}^b

Los diferentes sistemas de referencia antes descritos pueden ser relacionados mediante matrices de rotación o haciendo uso de cuaterniones, ambas opciones serán desarrolladas a continuación.

4.1.1. Mapeo con matrices de rotación

De la sección 2.3.1, se tiene que las matrices de rotación expresan la rotación de un sistema de referencia $\{a\}$ a un sistema de referencia $\{b\}$ y es escrita de manera general como: \mathbf{R}_a^b , es posible también mapear un vector \mathbf{r}^a descrito en el sistema de referencia $\{a\}$ en su equivalente \mathbf{r}^b dado en el sistema de referencia $\{b\}$.

Observación 4.1.1 se cuenta con el cuaternión \mathbf{Q}_b , que relaciona \mathbf{E}^b con respecto a \mathbf{E}^f , del cual puede obtenerse su equivalente \mathbf{R}_b^f . Los vectores unitarios \vec{r}_j , \vec{b}_j y $\vec{b}_j^{mu_i}$ están representados con respecto a \mathbf{E}^f , \mathbf{E}^b , \mathbf{E}^{mu_i} . Los vectores \vec{r}_j son llamados “vectores de referencia” y son bien conocidos. Los vectores $\vec{b}_j^{mu_i}$ son conocidos como “vectores de observación” y son obtenidos de la i -ésima unidad de medición (MU). Finalmente la matriz de rotación $\mathbf{R}_{mu_i}^b$ que relaciona \mathbf{E}^{mu_i} con respecto a \mathbf{E}^b es bien conocida.


 Figura 4.3: Relación entre los sistemas de referencia \mathbf{E}^{mu_i} , \mathbf{E}^b y \mathbf{E}^f

Estas representaciones vectoriales están vinculadas a través de la rotación de vectores usando matrices de rotación, como se indica a continuación:

$$\vec{b}_j = (\mathbf{R}_b^f)^T \vec{r}_j \quad (4.1)$$

$$\vec{b}_j^{mu_i} = (\mathbf{R}_{mu_i}^b)^T \vec{b}_j \quad (4.2)$$

$$\vec{b}_j^{mu_i} = (\mathbf{R}_{mu_i}^b)^T (\mathbf{R}_b^f)^T \vec{r}_j = (\mathbf{R}_b^f \mathbf{R}_{mu_i}^b)^T \vec{r}_j = (\mathbf{R}_{mu_i}^f)^T \vec{r}_j \quad (4.3)$$

de la ecuación (4.3), se tiene que $\mathbf{R}_{mu_i}^f = \mathbf{R}_b^f \mathbf{R}_{mu_i}^b$. Es posible entonces mapear un vector de referencia (aceleración gravitacional, campo magnético, etc.) a su correspondiente vector de observación, el cual podría ser utilizado como medidas simuladas de sensores (acelerómetros, magnetómetros, etc.). Las medidas simuladas pueden ser utilizadas por algún algoritmo determinista basado en vectores de observación para la estimación de la orientación como el algoritmo *LQS*, el cual esta basado en cuaterniones, pero es posible transformar el cuaternión estimado a su correspondiente matriz de rotación (ver apéndice B). Luego, la matriz de rotación estimada por el algoritmo *LQS* será $\mathbf{R}_{mu_i}^f$, que relaciona los sistemas de referencia \mathbf{E}^{mu_i} con respecto a \mathbf{E}^f , esto es importante tenerlo en cuenta dado que la matriz de rotación deseada es \mathbf{R}_b^f que es la que relaciona

los sistemas de referencia \mathbf{E}^b con respecto al sistema \mathbf{E}^f . De la ecuación (4.3), se tiene que

$$\mathbf{R}_{mu_i}^f = \mathbf{R}_b^f \mathbf{R}_{mu_i}^b \quad (4.4)$$

por lo tanto la matriz de rotación deseada se obtiene a partir de

$$\mathbf{R}_b^f = \mathbf{R}_{mu_i}^f \left(\mathbf{R}_{mu_i}^b \right)^{-1} \quad (4.5)$$

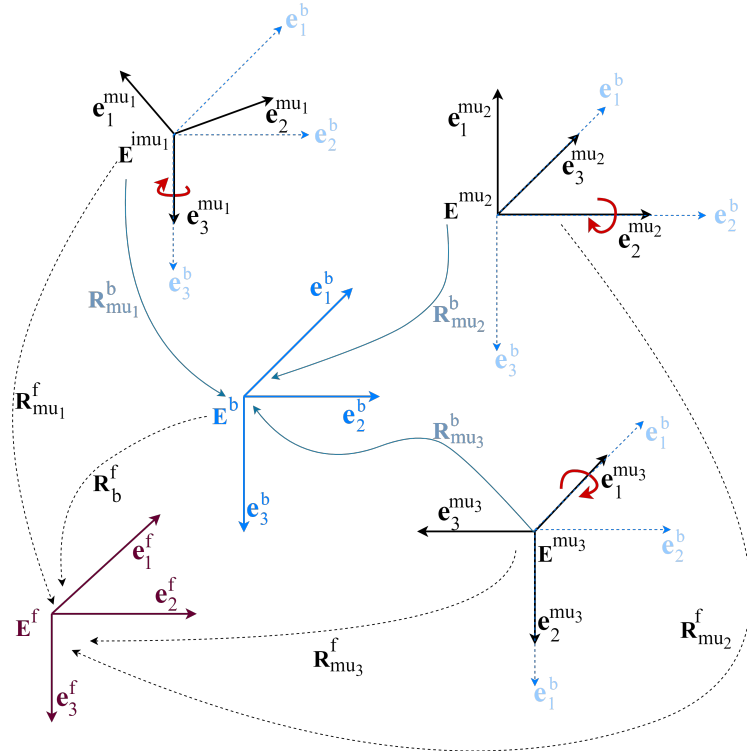


Figura 4.4: Relación de los sistemas de referencia \mathbf{E}^f , \mathbf{E}^b y \mathbf{E}^{mu_i} mediante matrices de rotación

La Figura 4.4 muestra la relación entre los diferentes sistemas de referencia, haciendo uso de las matrices de rotación, cada sistema de referencia \mathbf{E}^{mu_i} tiene una rotación de 90° con respecto a la componente marcada con una flecha. Hasta aquí se podrían contar con las medidas simuladas del acelerómetro y magnetómetro, en consecuencia, se hace uso del algoritmo *LQS* para la estimación de la orientación. Sin embargo se requiere el uso del observador no lineal para obtener una mejor estimación como se mostró en el capítulo anterior, esto conlleva a hacer uso del girómetro, por lo que se toma en cuenta lo siguiente:

Observación 4.1.2 Dado un vector de velocidad angular $\vec{\omega}$, expresado en el sistema de referencia \mathbf{E}^b , es posible mapear $\vec{\omega}$ a su correspondiente vector $\vec{\omega}^{mu_i}$, expresado en el sistema de referencia \mathbf{E}^{mu_i} para generar las medidas simuladas del i -ésimo girómetro.

En consecuencia se tiene que,

$$\vec{\omega}^{mu_i} = \left(\mathbf{R}_{mu_i}^b \right)^T \vec{\omega} \quad (4.6)$$

Ahora se tienen las medidas simuladas del i -ésimo acelerómetro, i -ésimo magnetómetro y del i -ésimo girómetro, lo que permite hacer uso del observador no lineal implementado anteriormente, donde la matriz de rotación deseada estará dada por (4.5). Hasta este punto, las transformaciones fueron realizadas con matrices de rotación, sin embargo, el algoritmo LQS es desarrollado haciendo uso de cuaterniones, por lo que a continuación se aborda la misma problemática en el ámbito de los cuaterniones.

4.1.2. Mapeo con cuaterniones

La parametrización por cuaterniones al igual que con las matrices de rotación, es ampliamente utilizado en robótica y navegación para representar la orientación de un cuerpo y permite realizar operaciones de rotación para mapear vectores expresados en un sistema de referencia hacia otro.

Observación 4.1.3 *se cuenta con el cuaternión \mathbf{Q}_b , que relaciona \mathbf{E}^b con respecto a \mathbf{E}^f . Los vectores unitarios \vec{r}_j , \vec{b}_j y $\vec{b}_j^{mu_i}$ están representados con respecto a \mathbf{E}^f , \mathbf{E}^b , \mathbf{E}^{mu_i} . Los vectores \vec{r}_j son llamados “vectores de referencia” y son bien conocidos. Los vectores $\vec{b}_j^{mu_i}$ son conocidos como “vectores de observación” y son obtenidos del i -ésimo magnetómetro y acelerómetro que forman parte de la i -ésima unidad de medición (MU). Finalmente el cuaternión $\mathbf{Q}_b^{mu_i}$ que relaciona el sistema de referencia \mathbf{E}^{mu_i} con respecto al sistema de referencia \mathbf{E}^b es bien conocido por el diseñador.*

Estas representaciones vectoriales están vinculadas a través de la rotación de vectores, haciendo uso de cuaterniones. A continuación se reescriben los vectores unitarios en su representación como cuaternión:

$$\mathbf{Q}(b_j^{mu_i}) = [0, (\vec{b}_j^{mu_i})^T]^T \quad \mathbf{Q}(b_j) = [0, \vec{b}_j^T]^T \quad \mathbf{Q}(r_j) = [0, \vec{r}_j^T]^T \quad (4.7)$$

Luego estos cuaterniones, están relacionados por los cuaterniones de rotación \mathbf{Q}_b , $\mathbf{Q}_b^{mu_i}$, y \mathbf{Q}_{mu_i} como sigue:

$$\mathbf{Q}(b_j) = (\mathbf{Q}_b)^{-1} \otimes \mathbf{Q}(r_j) \otimes \mathbf{Q}_b \quad (4.8)$$

$$\mathbf{Q}(b_j^{mu_i}) = (\mathbf{Q}_b^{mu_i})^{-1} \otimes \mathbf{Q}(b_j) \otimes \mathbf{Q}_b^{mu_i} \quad (4.9)$$

y como consecuencia

$$\begin{aligned} \mathbf{Q}(b_j^{mu_i}) &= (\mathbf{Q}_b^{mu_i})^{-1} \otimes [(\mathbf{Q}_b)^{-1} \otimes \mathbf{Q}(r_j) \otimes \mathbf{Q}_b] \otimes \mathbf{Q}_b^{mu_i} \\ &= (\mathbf{Q}_{mu_i})^{-1} \otimes \mathbf{Q}(r_j) \otimes \mathbf{Q}_{mu_i} \end{aligned} \quad (4.10)$$

de la ecuación (4.10), se tiene que $\mathbf{Q}_{mu_i} = \mathbf{Q}_b \otimes \mathbf{Q}_b^{mu_i}$. Es posible entonces mapear un vector de referencia (aceleración gravitacional, campo magnético, etc.) a su correspondiente vector de observación en el sistema de referencia \mathbf{E}^{mu_i} , el cual podría ser utilizado como medidas simuladas de sensores (acelerómetros, magnetómetros, etc.). Las medidas simuladas pueden ser utilizadas por algún algoritmo determinista basado en vectores de observación para la estimación de la orientación como el algoritmo LQS . Luego, el cuaternión estimado por el algoritmo LQS será \mathbf{Q}_{mu_i} , que relaciona los sistemas de referencia \mathbf{E}^{mu_i} con respecto a \mathbf{E}^f , sin embargo el cuaternión

deseado es \mathbf{Q}_b que es el que relaciona los sistemas de referencia \mathbf{E}^b con respecto al sistema \mathbf{E}^f . De la ecuación (4.10), se tiene que,

$$\mathbf{Q}_b = \mathbf{Q}_{mu_i} \otimes (\mathbf{Q}_b^{mu_i})^{-1} \quad (4.11)$$

Teniendo en cuenta el uso del observador no lineal para obtener una estimación de la orientación con niveles de ruido reducidos, es necesario hacer uso de un girómetro que mida la velocidad angular en cada uno de los ejes del sistema de referencia del cuerpo. Considerando nuevamente la Observación 4.1.2, se tiene que:

$$\mathbf{Q}(\vec{\omega}^{mu_i}) = (\mathbf{Q}_b^{mu_i})^{-1} \otimes \mathbf{Q}(\vec{\omega}) \otimes \mathbf{Q}_b^{mu_i} \quad (4.12)$$

Donde $\mathbf{Q}(\vec{\omega})$ es el vector de velocidad angular expresado como un cuaternión puro. Realizar las transformaciones entre los diferentes sistemas de referencia haciendo uso de la multiplicación de cuaterniones representa una menor carga computacional debido a que se operan con vectores de $\mathbb{R}^{4 \times 1}$ y porque no existe la necesidad de realizar transformaciones extras como en el caso de matrices de rotación, donde se debe de transformar de matriz de rotación a cuaternión y viceversa. Por consiguiente, durante el resto del trabajo se hará uso de la multiplicación de cuaterniones para realizar las diferentes transformaciones entre los sistemas de referencia.

De manera general la ecuación (2.34) puede ser reescrita como:

$$\begin{pmatrix} 0 & -(\vec{b}_j^{mu_i} - \vec{r}_j)^T \\ \vec{b}_j^{mu_i} - \vec{r}_j & -[(\vec{b}_j^{mu_i} + \vec{r}_j) \times] \end{pmatrix} \mathbf{Q}_{mu_i} = H_j^i \mathbf{Q}_{mu_i} = 0 \quad (4.13)$$

$H_j^i \in \mathbb{R}^{4 \times 4}$ es llamada la matriz de observación y puede verse que (4.13) es un sistema de ecuaciones lineales bien estructurado.

4.2. Planteamiento del problema

De acuerdo a (4.8) y (4.9), el objetivo del proceso de estimación de la orientación, consiste en estimar la orientación \mathbf{E}^b con respecto a \mathbf{E}^f , es decir, el cuaternión \mathbf{Q}_b . Este proceso se llevará acabo vía la estimación del cuaternión \mathbf{Q}_{mu_i} , el cual mapea los vectores de observación dentro de los conocidos vectores de referencia. Es decir, dados los vectores de observación $\vec{b}_j^{mu_i}$ y los vectores de referencia \vec{r}_j , con $j \in \{1, 2 \dots m\}$ el número de vectores de referencia, y $i \in \{1, 2 \dots n\}$ el número de unidades de medición (enseguida denotado como agentes), el objetivo es encontrar \mathbf{Q}_{mu_i} en cada agente i tal que:

$$H_j^i \mathbf{Q}_{mu_i} = 0 \quad (4.14)$$

Una vez que cada agente ha estimado el cuaternión de orientación \mathbf{Q}_{mu_i} , este es compartido a su agente “vecino” (ver Figura 4.5), para luego realizar un promedio dinámico de estos cuaterniones en el sistema de referencia \mathbf{E}^b con el algoritmo propuesto en el presente trabajo.

Una vez introducidos los conceptos de teoría de grafos, rotación entre los diferentes sistemas de coordenadas y algoritmos de estimación de orientación, en la siguiente subsección se aborda el diseño del algoritmo de consenso dinámico para los agentes de la red.

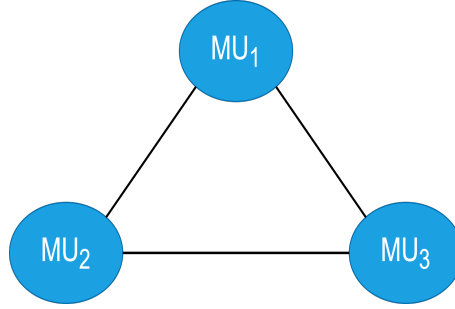


Figura 4.5: Grafo de comunicación entre cada MU

4.3. Diseño del algoritmo de consenso dinámico

Considere un conjunto de n agentes, cada uno conformado por una unidad de medición (MU). Se asume que cada agente es capaz de proveer un cuaternión de orientación $\hat{\mathbf{Q}}_{mu_i}$ con $i \in \{1, 2, \dots, n\}$. A continuación se explica como el cuaternión es estimado.

El observador no lineal descrito por la ecuación (3.5), es reescrito para obtener el cuaternión estimado por el i -ésimo agente como sigue:

$$\mathcal{O}_i = \begin{cases} \dot{\hat{\mathbf{Q}}}_{mu_i}(t) = \frac{1}{2}\Xi(\hat{\mathbf{Q}}_{mu_i}(t)) \left(\vec{\omega}_{G_i}(t) - \hat{\vec{v}}_i(t) + K_1 \vec{q}_{e_i}(t) \right) \\ \dot{\hat{\vec{v}}}_i(t) = -T^{-1} \hat{\vec{v}}_i(t) - K_2 \vec{q}_{e_i}(t) \end{cases} \quad (4.15)$$

Además de la ecuación (4.13), la matriz de observación para el i -ésimo agente en función de m par de vectores $[\vec{b}_j^{mu_i}, \vec{r}_j]$ con $j \in \{1, 2, \dots, m\}$ es reescrito como sigue:

$$H_m^i = \begin{pmatrix} 0 & -(\vec{b}_1^{mu_i} - \vec{r}_1)^T \\ \vec{b}_1^{mu_i} - \vec{r}_1 & -[(\vec{b}_1^{mu_i} + \vec{r}_1) \times] \\ & \vdots \\ 0 & -(\vec{b}_m^{mu_i} - \vec{r}_m)^T \\ \vec{b}_m^{mu_i} - \vec{r}_m & -[(\vec{b}_m^{mu_i} + \vec{r}_m) \times] \end{pmatrix} \in \mathbb{R}^{4m \times 4} \quad (4.16)$$

tal que,

$$H_m^i \mathbf{Q}_{mu_i} = \begin{pmatrix} \varphi_1^T(t) \\ \vdots \\ \varphi_{4m}^T(t) \end{pmatrix} \mathbf{Q}_{mu_i} = 0 \quad (4.17)$$

De manera que el algoritmo LQS funcionaría de la siguiente manera: considere una matriz de observación $H_m^i(t)$ en el instante t y un cuaternión inicial denotado \mathbf{Q}_{ps}^0 (de aquí en adelante el cuaternión \mathbf{Q}_{ps} , es denotado como $\tilde{\mathbf{Q}}$, para simplificar notaciones). Luego una estimación de la orientación del i -ésimo agente es dado por

$$\tilde{\mathbf{Q}}_{mu_i} = \left[\prod_{\kappa=1}^{4m} \left(\vec{I}_4 - \frac{\gamma \varphi_{\kappa}(t) \varphi_{\kappa}^T(t)}{\alpha + \varphi_{\kappa}^T(t) \varphi_{\kappa}(t)} \right) \right] \tilde{\mathbf{Q}}_{mu_i}^0 \quad (4.18)$$

donde $\alpha \geq 0$ y $0 < \gamma < 2$. Una vez que (4.18) es desarrollado, el cuaternión es normalizado

$$\tilde{\mathbf{Q}}_{mu_i} = \frac{\tilde{\mathbf{Q}}_{mu_i}^{Am}}{\|\tilde{\mathbf{Q}}_{mu_i}^{Am}\|} \quad (4.19)$$

Observación 4.3.1 Note que (4.18) representa un algoritmo iterativo que es ejecutado en cada instante de tiempo t . Luego, para $t = 0$ el cuaternión inicial $\tilde{\mathbf{Q}}_{mu_i}^0$ en (4.18) puede ser escogido arbitrariamente. En la secuencia, para toda $t > 0$, $\tilde{\mathbf{Q}}_{mu_i}^0$ será obtenido de la salida del algoritmo de consenso dinámico, es decir:

$$\tilde{\mathbf{Q}}_{mu_i}^0 = \begin{cases} \text{cuaternión unitario arbitrario, } \forall t = 0 \\ \tilde{\mathbf{Q}}_i, \forall t > 0 \end{cases} \quad (4.20)$$

Observación 4.3.2 El cuaternión $\hat{\mathbf{Q}}_{mu_i}$ representa la estimación de la orientación con respecto al sistema de referencia \mathbf{E}^{mu_i} . Es posible representar esta orientación en el sistema de referencia \mathbf{E}^b , por medio de (4.11), esto es:

$$\hat{\mathbf{Q}}_{b_i} = \hat{\mathbf{Q}}_{mu_i} \otimes (\mathbf{Q}_b^{mu_i})^{-1} \quad (4.21)$$

donde $\mathbf{Q}_b^{mu_i}$ son bien conocidos.

Cada agente es formado por una unidad de medición, donde cada sensor cuenta con sus propias características de ruido. Como consecuencia, el cuaternión estimado por cada agente será diferente en términos de confiabilidad. La idea es tomar cada cuaternión estimado $\hat{\mathbf{Q}}_{b_1}, \hat{\mathbf{Q}}_{b_2}, \dots, \hat{\mathbf{Q}}_{b_n}$ y desarrollar un consenso dinámico distribuido (promedio) con respecto al sistema de referencia \mathbf{E}^b .

Considere los siguientes vectores de estado $\bar{\mathbf{Q}} = [\bar{\mathbf{Q}}_1^T, \dots, \bar{\mathbf{Q}}_n^T]^T$ y $\Upsilon = [(\hat{\mathbf{Q}}_{b_1})^T, (\hat{\mathbf{Q}}_{b_2})^T, \dots, (\hat{\mathbf{Q}}_{b_n})^T]^T$. Por lo tanto se propone el siguiente Algoritmo de Consenso Dinámico (*Dinamic Consensus Algorithm, DCA*):

$$\dot{\bar{\mathbf{Q}}}_i(t) = \sum_{l \in \mathcal{N}_i} a_{il} (\bar{\mathbf{Q}}_l(t) - \bar{\mathbf{Q}}_i(t)) + \sum_{l \in \mathcal{M}_i} a_{il} (\hat{\mathbf{Q}}_{b_l}(t) - \bar{\mathbf{Q}}_i(t)) \quad (4.22)$$

donde a_{il} son elementos de la matriz de adyacencia \mathbf{A} de la red de sensores, \mathcal{N}_i y \mathcal{M}_i son el conjunto de vecinos y el conjunto de vecinos inclusivos del nodo i , respectivamente. Luego uno tiene los siguientes resultados:

Proposición 4.3.1 El algoritmo de consenso dinámico en (4.22) resulta en un filtro de consenso con la siguiente dinámica colectiva:

$$\dot{\bar{\mathbf{Q}}} = -\Xi \bar{\mathbf{Q}} + B \Upsilon \quad (4.23)$$

donde $\Xi = (\vec{I}_{4n} + \mathbf{D} + \mathcal{L})$ y $B = \vec{I}_{4n} + \mathbf{A}$, con \mathbf{D} y \mathcal{L} las matrices de grado y Laplaciana respectivamente, \vec{I}_{4n} es la matriz identidad.

Proposición 4.3.2 El filtro de consenso descrito en (4.23) es un filtro pasa bajos estable distribuido, con una función de transferencia como sigue:

$$H(s) = (s\vec{I}_{4n} + \Xi)^{-1} B \quad (4.24)$$

Prueba 4.3.1 Aplicando el teorema de Gersgorin a la matriz Ξ , se garantiza que todos los polos de $H(s)$ son estrictamente negativos. Como consecuencia el filtro es estable. Además, $H(s)$ es una función de transferencia propia MIMO, lo que significa que es un filtro pasa bajos [47].

En la Figura 4.6 se muestra el diagrama de bloques del proceso de estimación del cuaternión $\hat{\mathbf{Q}}_{mu_i}$ y del cuaternión obtenido del consenso dinámico $\bar{\mathbf{Q}}_i$.

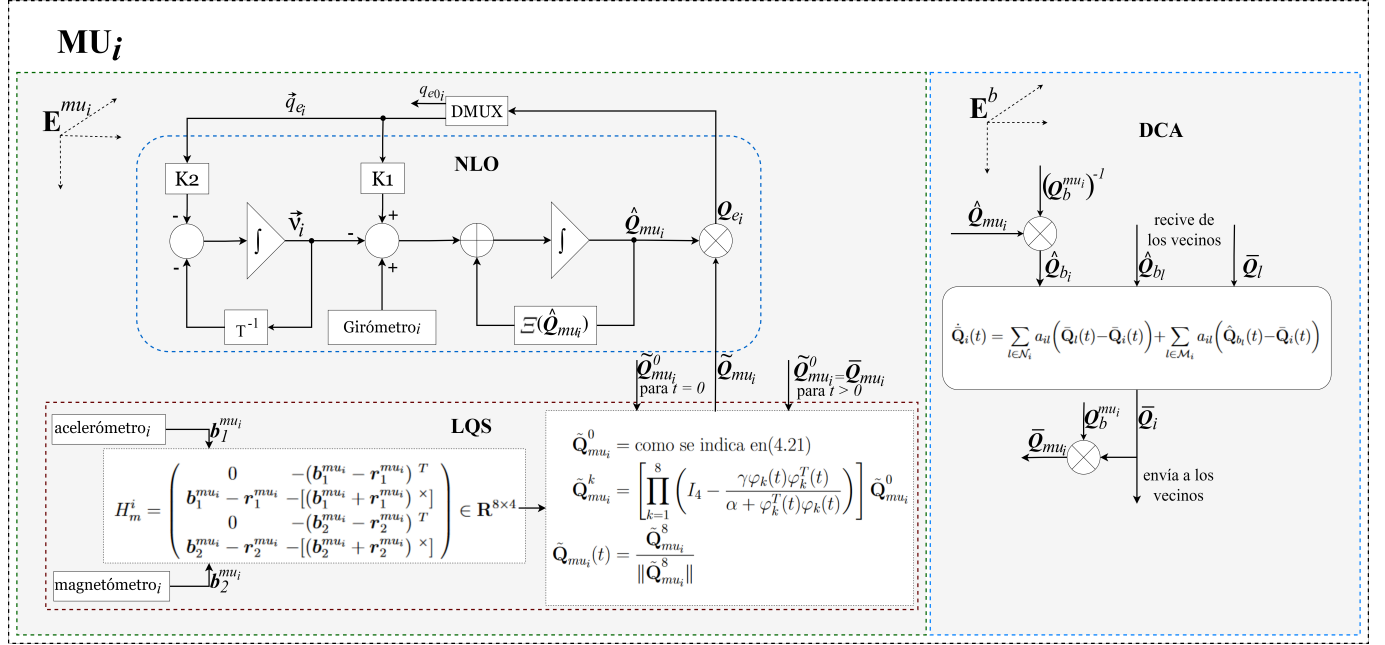


Figura 4.6: Diagrama de bloques de un agente en la red de sensores

4.4. Resultados basados en simulación

Para propósitos de simulación, la orientación de referencia (orientación verdadera) y las medidas de los sensores de cada unidad de medición serán generadas por los siguientes sistemas:

$$\dot{\vec{\omega}} = J^{-1} (-\vec{\omega} \times J\vec{\omega} + \Gamma) \quad (4.25)$$

$$\dot{\mathbf{Q}}_b(t) = \frac{1}{2} \begin{pmatrix} -\vec{q}^T \\ q_0 \vec{I}_3 + [\vec{q}^\times] \end{pmatrix} \vec{\omega} \quad (4.26)$$

Cuando la ecuación dinámica (4.25) es resuelta, una velocidad angular $\vec{\omega}$ expresada en el sistema de referencia \mathbf{E}^b es obtenida, con respecto a $\Gamma \in \mathbb{R}^3$, el cual es el vector de torque de entrada para cada eje del cuerpo rígido. J representa la matriz de inercia del cuerpo rígido. Luego $\vec{\omega}$ es utilizada por la ecuación (4.26), que al solucionarla nos da un cuaternión \mathbf{Q}_b (orientación verdadera) expresado en el sistema de referencia \mathbf{E}^b .

Ahora considere los siguientes vectores unitarios de referencia escrito como cuaterniones: $\mathbf{Q}(r_1) = [0; 0; 0; 1]$ (aceleración gravitacional) y $\mathbf{Q}(r_2) = [0; 0.337; 0; 0.348]$ (campo magnético terrestre), los cuales están descritos en el sistema de referencia \mathbf{E}^f , estos cuaterniones pueden

ser mapeados al sistema de referencia \mathbf{E}^{mu_i} , por medio de la ecuación (4.10). Es importante notar que el cuaternión $\mathbf{Q}_b^{mu_i}$ es bien conocido por el diseñador y \mathbf{Q}_b es dado por el sistema (4.26), por lo que es posible obtener los nuevos cuaterniones $\mathbf{Q}(b_1^{mu_i})$ y $\mathbf{Q}(b_2^{mu_i})$, que corresponden al mapeo de los vectores de aceleración gravitacional y campo magnético respectivamente. Las medidas simuladas son tomadas de la parte vectorial del correspondiente cuaternión $\mathbf{Q}(b_j^{mu_i})$, esto es: $\vec{b}_1^{mu_i}$ y $\vec{b}_2^{mu_i}$, que corresponden a las medidas simuladas para el i -ésimo acelerómetro y para el i -ésimo magnetómetro respectivamente. Finalmente se obtiene la medida simulada para el i -ésimo girómetro, a partir del vector $\vec{\omega}$, obtenido del sistema (4.25) y aplicando la ecuación (4.12).

Una vez que se cuentan con los valores de referencia \mathbf{Q}_b y $\vec{\omega}$ y con las medidas simuladas $\vec{b}_1^{mu_i}$, $\vec{b}_2^{mu_i}$ y $\vec{\omega}^{mu_i}$, es posible estimar la orientación del cuerpo rígido de manera distribuida aplicando los algoritmos descritos en el diagrama de bloques de la Figura 4.6.

En orden de obtener el error de desajuste entre $\hat{\mathbf{Q}}_{mu_i}$ y \mathbf{Q}_b (cuaternión de referencia) y entre $\hat{\mathbf{Q}}_i$ y \mathbf{Q}_b , se obtiene el cuaternión de error aplicando (4.27) y luego se calcula el coseno inverso de la parte escalar del cuaternión \mathbf{Q}_e , esto es, $\beta_e = \arccos q_{e0}$.

$$\mathbf{Q}_e := \mathbf{Q}_b \otimes \hat{\mathbf{Q}}^{-1} = [q_{e0}, \vec{q}_e^T]^T \quad (4.27)$$

4.4.1. Robustez tolerante a fallas

Es importante resaltar que el algoritmo *LQS* hace uso de vectores de referencia \vec{r}_j bien conocidos y de vectores de observación $\vec{b}_j^{mu_i}$, como se define en la matriz de observación (4.16). Sin embargo, para cada agente, esta matriz es función de los datos que son sensados. Si algunos de los sensores de la MU i falla, la matriz cambia con el par de vectores $[\vec{b}_j^{mu_i}, \vec{r}_j]$ que falla.

Para el caso de los vectores de observación $\vec{b}_1^{mu_i}$ y $\vec{b}_2^{mu_i}$, se tiene la siguiente matriz de observación:

$$H_m^i = \begin{pmatrix} 0 & -(\vec{b}_1^{mu_i} - \vec{r}_1^{mu_i})^T \\ \vec{b}_1^{mu_i} - \vec{r}_1^{mu_i} & -[(\vec{b}_1^{mu_i} + \vec{r}_1^{mu_i}) \times] \\ 0 & -(\vec{b}_2^{mu_i} - \vec{r}_2^{mu_i})^T \\ \vec{b}_2^{mu_i} - \vec{r}_2^{mu_i} & -[(\vec{b}_2^{mu_i} + \vec{r}_2^{mu_i}) \times] \end{pmatrix} \in \mathbf{R}^{8 \times 4} \quad (4.28)$$

si el vector de aceleración gravitacional falla, en uno de los agentes, la matriz (4.28), podría ser modificada en ese agente, como sigue:

$$H_m^i = \begin{pmatrix} 0 & -(\vec{b}_2^{mu_i} - \vec{r}_2^{mu_i})^T \\ \vec{b}_2^{mu_i} - \vec{r}_2^{mu_i} & -[(\vec{b}_2^{mu_i} + \vec{r}_2^{mu_i}) \times] \end{pmatrix} \in \mathbf{R}^{4 \times 4} \quad (4.29)$$

cuando el vector de campo magnético falla, la matriz (4.28) se modifica como sigue:

$$H_m^i = \begin{pmatrix} 0 & -(\vec{b}_1^{mu_i} - \vec{r}_1^{mu_i})^T \\ \vec{b}_1^{mu_i} - \vec{r}_1^{mu_i} & -[(\vec{b}_1^{mu_i} + \vec{r}_1^{mu_i}) \times] \end{pmatrix} \in \mathbf{R}^{4 \times 4} \quad (4.30)$$

Con la finalidad de analizar el desempeño del sistema de estimación distribuido, descrito por la Figura 4.6, cuando alguno de los sensores del i -ésimo agente falla, se hace uso del criterio de la integral del error cuadrático ISE (del inglés *Integral Square-Error*) [48], que se define a continuación:

$$ISE = \int_0^T \tilde{x}^2(t) dt \quad (4.31)$$

Luego, el criterio ISE fue aplicado al ángulo β_e como sigue:

$$ISE_{\beta_e} = \int_0^T \tilde{\beta}_e^2(t) dt \quad (4.32)$$

Este criterio de desempeño, nos da una medida cuantitativa cuando alguno de los sensores falla en uno de los agentes, ya que al estarse integrando el cuadrado del error, esta magnitud se incrementa cuando una falla se presenta y se mantiene estable cuando no hay fallas.

$\mathbf{Q}_b^{mu_i}$ asociado a cada MU		
mu_i	$\mathbf{Q}_b^{mu_i}$	Ángulos de Euler
1	$[0.3827; 0.9239; 0; 0]^T$	$\phi = 135^\circ, \theta = 0^\circ, \psi = 0^\circ$
2	$[0.3827; 0; 0.9239; 0]^T$	$\phi = 0^\circ, \theta = 135^\circ, \psi = 0^\circ$
3	$[0.3827; 0; 0; 0.9239]^T$	$\phi = 0^\circ, \theta = 0^\circ, \psi = 135^\circ$

Tabla 4.1: Rotación aplicada a cada MU con respecto al sistema de referencia \mathbf{E}^b

La tabla 4.1, muestra el cuaternión $\mathbf{Q}_b^{mu_i}$ que relaciona los sistemas de referencia \mathbf{E}^{mu_i} y \mathbf{E}^b a través de una rotación, y es definido por el diseñador. Haciendo uso de tres agentes, los algoritmos descritos en el diagrama de bloques de la Figura 4.6 son simulados, en orden de obtener una única estimación de la orientación del cuerpo rígido. Las condiciones iniciales para $t = 0$ son: $\bar{\mathbf{Q}}_i^0 = [0.933; 0.25; 0.25; -0.06699]$ y $\hat{\mathbf{Q}}_{mu_i}^0 = [1; 0; 0; 0]$.

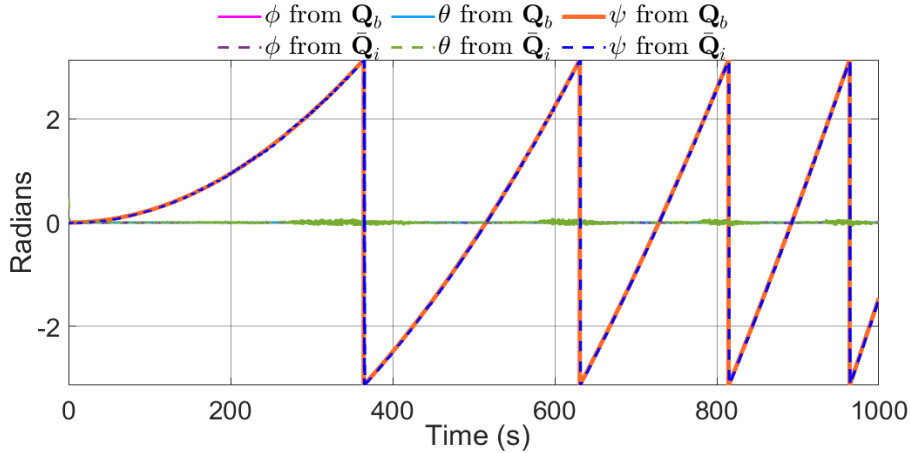


Figura 4.7: Comparación entre los ángulos de Euler obtenidos de \mathbf{Q}_b y los ángulos de Euler obtenidos de \mathbf{Q}_i

En la Figura 4.7, las líneas continuas corresponden a los ángulos de Euler obtenidos del cuaternión de referencia \mathbf{Q}_b y las líneas discontinuas a los ángulos de Euler obtenidos del cuaternión del consenso dinámico $\bar{\mathbf{Q}}_i$. Estos resultados corresponden a una simulación donde cada agente de la red de sensores estima de manera correcta su correspondiente cuaternión $\hat{\mathbf{Q}}_{mu_i}$, es decir, no se presentan fallas de los sensores en el i -ésimo agente.

Escena	Tiempo (s)	MU ₁		MU ₂		MU ₃		MU	
		\vec{b}_1	\vec{b}_2	\vec{b}_1	\vec{b}_2	\vec{b}_1	\vec{b}_2	\vec{b}_1	\vec{b}_2
1	0 – 200	✓	✓	✓	✓	✓	✓	✓	✓
2	200 – 400	✓	X	✓	✓	✓	✓	✓	✓
3	400 – 500	X	✓	X	✓	✓	✓	X	✓
4	500 – 700	✓	✓	✓	X	X	✓	✓	✓
5	700 – 900	X	✓	✓	X	✓	X	✓	X
6	900 – 1100	X	✓	✓	X	X	✓	✓	✓
7	1100 – 1300	✓	✓	X	X	X	✓	X	✓
8	1300 – 1500	X	X	X	X	✓	✓	✓	✓

Tabla 4.2: Operación de los sensores del i -ésimo agente durante el proceso de simulación de la Figura 4.8

En la Tabla 4.2, se muestran 8 posibles escenarios de simulación, con el comportamiento de los vectores de observación $b_j^{mu_i}$ asociado a la i -ésima MU que forman parte de la red de sensores, en la última columna superior de la Tabla 4.2, MU, representa una unidad de medición independiente, es decir, no pertenece a alguna red de sensores, por lo que únicamente ejecuta el algoritmo presentado en el diagrama de bloques de la Figura 3.2. Una ✓ indica un buen funcionamiento del correspondiente sensor y una X indica una falla del correspondiente sensor, por lo que la matriz (4.28) se modifica, según el sensor que haya fallado.

La Figura 4.8, muestra los resultados de simulación asociados a los escenarios de la Tabla 4.2. La Figura 4.9, muestra la evolución en el tiempo de la parte escalar de los cuaterniones Q_b , \hat{Q}_{mu_i} y \hat{Q}_i , asociado a la simulación de la Tabla 4.2.

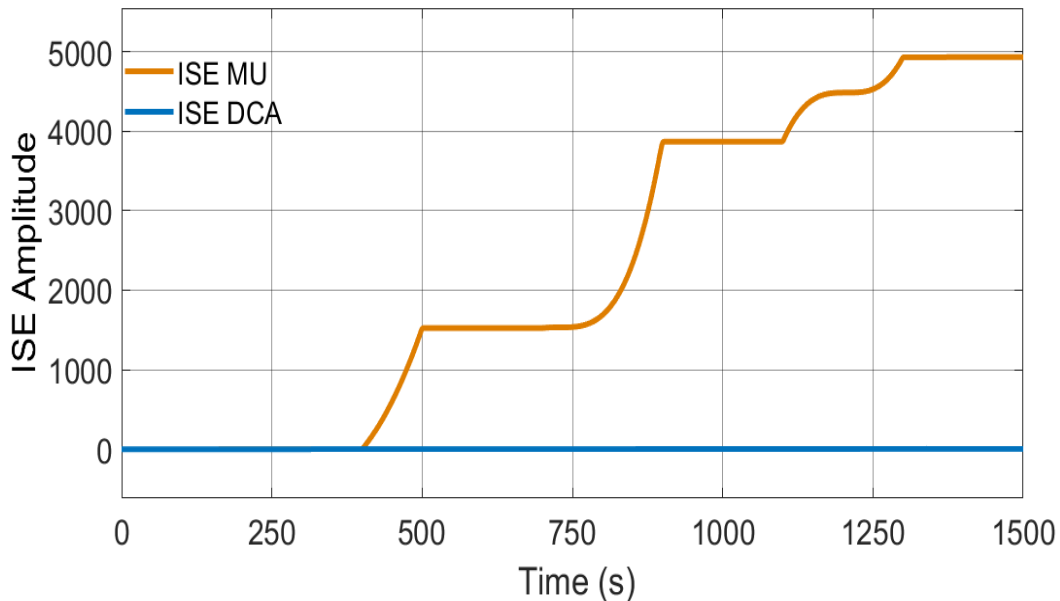


Figura 4.8: Criterio de desempeño ISE aplicado a β_e

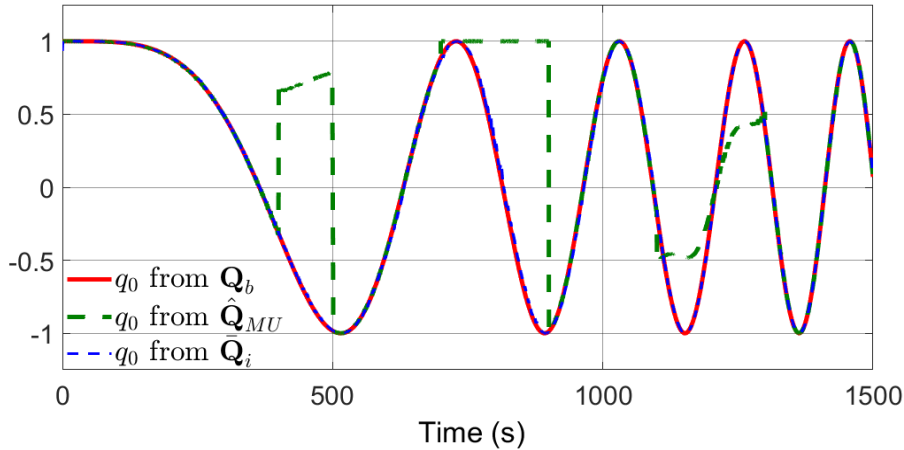


Figura 4.9: Evolución en el tiempo de la parte escalar de \mathbf{Q}_b , $\hat{\mathbf{Q}}_{MU}$ y $\bar{\mathbf{Q}}_i$

4.5. Resultados experimentales

Las simulaciones tienen un rol muy importante en el proceso de investigación y desarrollo científico, debido a las amplias consideraciones y escenarios que pueden ser tomadas en cuenta durante dicho proceso, además de evaluar objetivamente si el proceso de investigación va por buen camino y de brindar una base sólida de los resultados prácticos que podrían esperarse. Una vez que la etapa de simulación ha sido efectuada de manera exitosa, se procede a las pruebas experimentales siempre teniendo en cuenta muchas de las consideraciones realizadas en el proceso de simulación.

En esta sección se presentan los resultados experimentales del algoritmo de consenso dinámico, con el objetivo de validar su buen funcionamiento, para ello se hace uso de un sistema de Optitrack el cual entrega el cuaternión de orientación de un cuerpo rígido, este cuaternión entregado por el sistema Optitrack es tomado como el cuaternión de referencia y es comparado con el cuaternión estimado por el algoritmo de consenso dinámico, además también se comparan los ángulos de Euler de ambos, referencia y del consenso dinámico. Dos casos son presentados en las pruebas experimentales:

- *Caso 1:* Las mediciones de los sensores son enviados a MATLAB en donde el algoritmo de consenso dinámico es ejecutado.
- *Caso 2:* El algoritmo de consenso dinámico es ejecutado a nivel de microcontrolador y el cuaternión obtenido es enviado a MATLAB para efectos de comparación.

4.5.1. Algoritmo de consenso dinámico caso 1

En esta sección son presentados los resultados experimentales obtenidos de la implementación del algoritmo de consenso dinámico a nivel práctico, para ello se diseñó una estructura que permitió tener un grado de libertad para la rotación de los sensores, es decir uno de los sensores tenía la posibilidad de rotar sobre el eje x , el segundo sensor sobre el eje y y el tercer sensor sobre el eje z , la estructura fue diseñada con piezas de lego de manera que los sensores coincidieran en origen pero con un pequeño desplazamiento en altura. La Figura 4.10 muestra el diagrama de bloques de disposición de los sensores sobre el robot Turtlebot el cual fue utilizado como cuerpo rígido durante las pruebas experimentales, sobre el robot se colocaron pequeños marcadores esféricos con los cuales se definió un cuerpo rígido en el software de motive, las cámaras detectan los marcadores

y a partir de ellos determinan la orientación y posición del robot con respecto a un sistema de referencia definido durante el proceso de calibración del sistema Optitrack.

El diagrama de bloques de la Figura 4.11 muestra la interacción de los diferentes elementos que conforman la prueba experimental, las cámaras del sistema Optitrack se comunican con el software motive vía ethernet, la información de posición y orientación alojada en motive puede ser transmitida vía streaming local a MATLAB, en el cual se ejecuta el algoritmo de consenso dinámico y se realizan las comparaciones del cuaternión obtenido del consenso dinámico \bar{Q}_{mu_i} vs el cuaternión de referencia Q_b (obtenido del Optitrack), las mediciones obtenidas de cada unidad de medición son enviadas vía USB a MATLAB para ejecutar el algoritmo de consenso dinámico con datos reales, además es posible controlar el Turtlebot de manera remota por wifi haciendo uso de ROS (*Robot Operating System*), el cual puede ejecutarse en una máquina virtual con Linux.

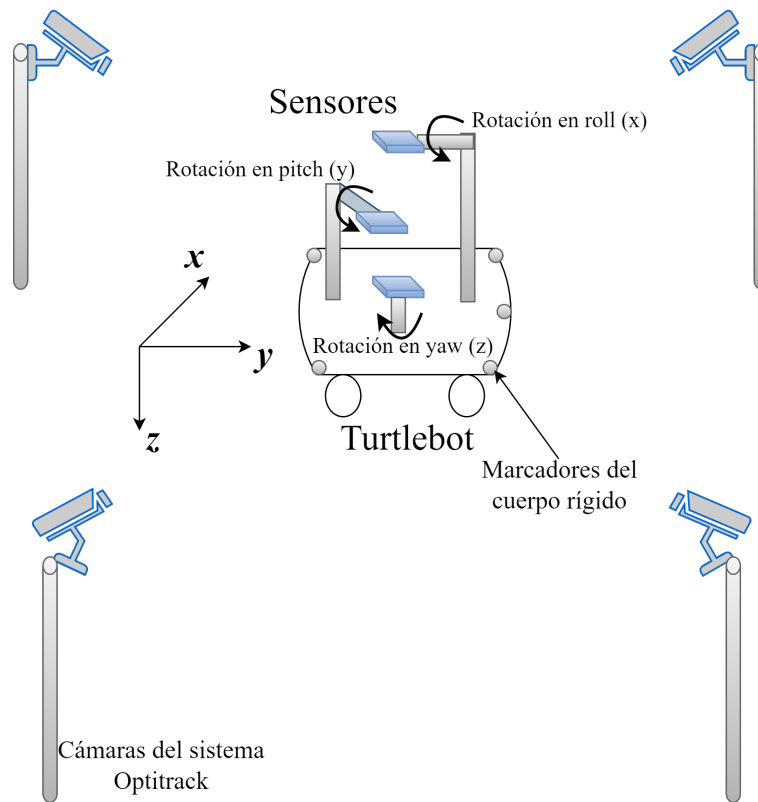


Figura 4.10: Esquema del funcionamiento de las pruebas experimentales para el caso 1

La Figura 4.12 muestra el Turtlebot instrumentado con las MU's, se observa también la tarjeta de desarrollo STM32F411RE la cual fue utilizada para adquisición de las mediciones de cada MU y para transmisión de las mismas a una PC con MATLAB y finalmente se dispusieron diez marcadores de cuerpo rígido sobre el Turtlebot.

La Figura 4.13, muestra el espacio de trabajo que corresponde a la sala de captura de movimiento, la cual se encuentra ubicada en el cubículo L1.14 del INSA (*Institut National des Sciences Appliquées*) en Estrasburgo, Francia y cuenta con cuatro cámaras Optitrack.

4. 4.5. RESULTADOS EXPERIMENTALES

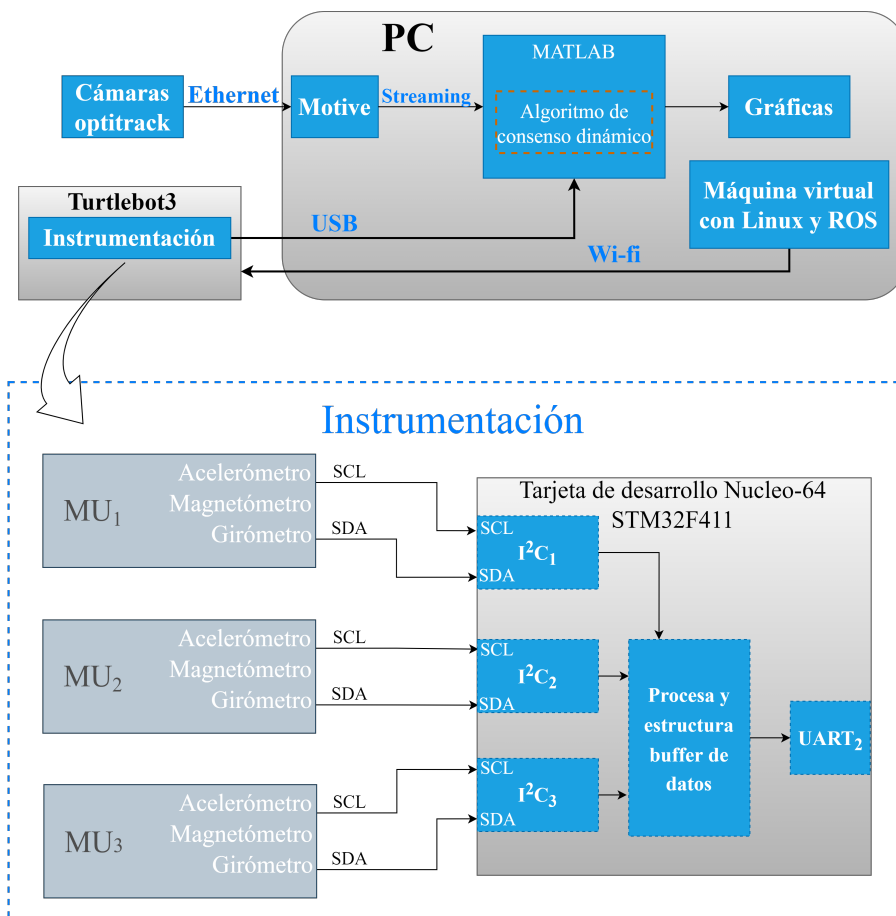


Figura 4.11: Diagrama de bloque de la interacción de los elementos de la Figura 4.10

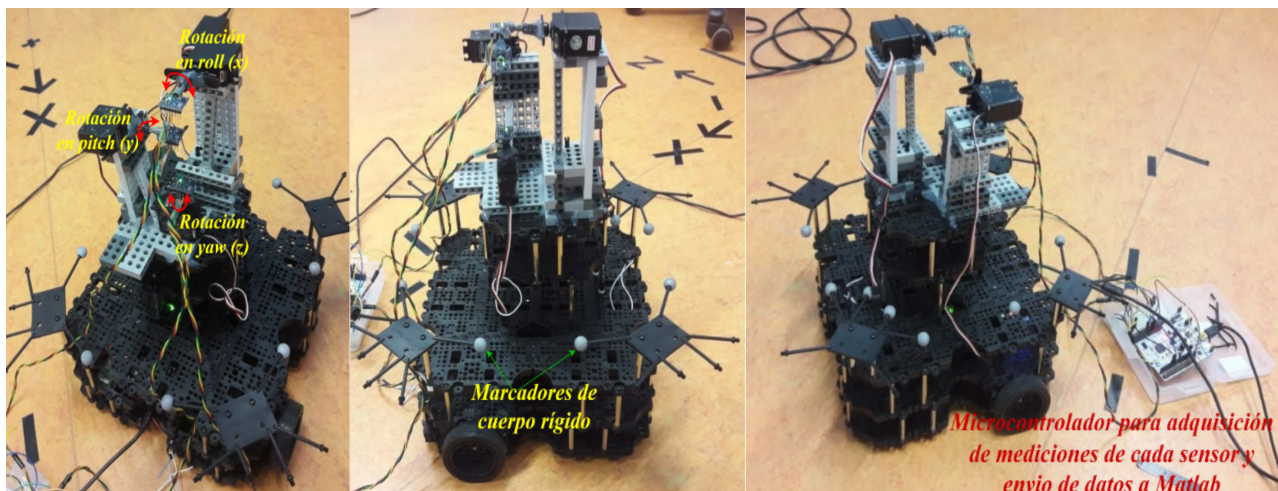


Figura 4.12: Turtlebot instrumentado con unidades de medición y marcadores de cuerpo rígido para estimación de la orientación



Figura 4.13: Sala de captura de movimiento con cuatro cámaras Optitrack

Observación 4.5.1 *El vector de referencia de campo magnético tuvo que ser modificado para hacerlo coincidir con el sistema de referencia definido por el sistema Optitrack. Al rotar los sensores es necesario modificar nuevamente un poco el vector de referencia magnético asociado a cada sensor, esto debido a que no son completamente coincidentes en origen, es decir que existe un pequeño desplazamiento entre sus sistemas de referencia.*

En la Tabla 4.3, se presenta el cuaternión $\mathbf{Q}_b^{mu_i}$ y su equivalente en ángulos de Euler aplicado a cada MU, este cuaternión es definido por el diseñador y es utilizado para relacionar los sistemas de coordenadas \mathbf{E}^{mu_i} y \mathbf{E}^b .

Rotación aplicada a cada MU con respecto al sistema de referencia \mathbf{E}^b		
MU	$\mathbf{Q}_b^{mu_i}$	Ángulos de Euler
1	[0.3827; -0.9239; 0; 0]	$(\phi = -135^\circ, \theta = 0^\circ, \psi = 0^\circ)$
2	[0; 0; 0; 0]	$(\phi = 0^\circ, \theta = 0^\circ, \psi = 0^\circ)$
3	[0.9239; 0; 0; 0.3827]	$(\phi = 0^\circ, \theta = 0^\circ, \psi = 45^\circ)$

Tabla 4.3: Cuaternión de rotación aplicada a cada MU en su correspondiente eje con grado de libertad

Resultados caso 1

Los resultados, primeramente se presentan en ángulos de Euler, porque son más intuitivos y de mejor comprensión, sin embargo, posteriormente en esta misma sección se presentan los resultados usando el cuaternión como parametrización. Las Figuras 4.14, 4.15 y 4.16 muestran los ángulos

4. 4.5. RESULTADOS EXPERIMENTALES

de Euler asociados al cuaternión de referencia \mathbf{Q}_b (Optitrack) y los ángulos de Euler relacionados al cuaternión estimado mediante el algoritmo de consenso dinámico $\bar{\mathbf{Q}}_{mu_i}$.

En un primer escenario el Turtlebot es colocado con una rotación inicial de $45^\circ\psi$, en la Figura 4.14a queda evidenciado como la estimación obtenida del consenso dinámico se apega a la referencia, aproximadamente con un tiempo de convergencia de 1s considerando que las mediciones de los sensores son transmitidas cada 0.02s. La Figura 4.14b corresponde al escenario donde el Turtlebot tiene una posición inicial de $-45^\circ\psi$, se observa como la estimación no alcanza en su totalidad a la referencia, existiendo una diferencia aproximada de 0.07 radianes entre el ángulo *yaw* de referencia y el estimado por el consenso.

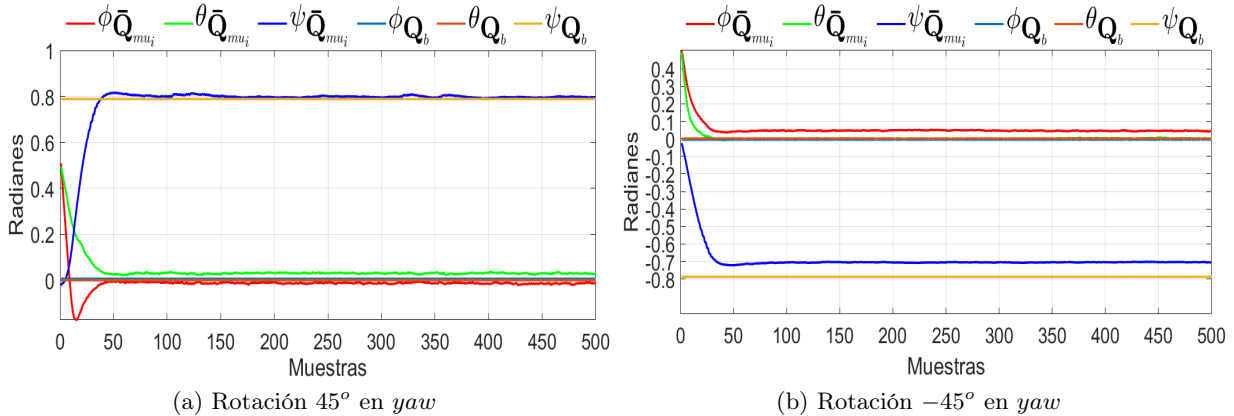


Figura 4.14: Turtlebot con una posición inicial de $+45^\circ$ y -45°

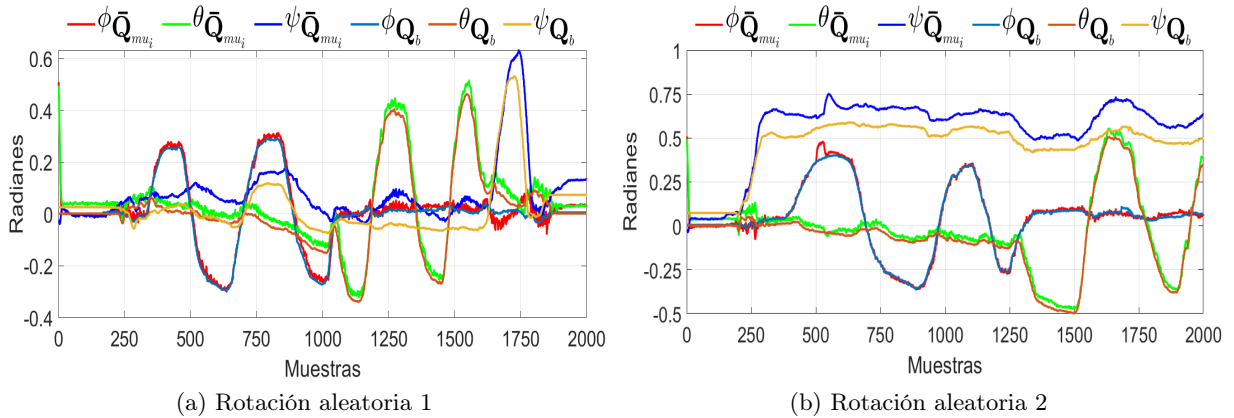


Figura 4.15: Ángulos de Euler obtenidos al mover aleatoriamente el Turtlebot

Considerando que el cuerpo rígido donde será dispuesto el sistema de estimación de orientación será una boya marina, se aplicaron movimientos aleatorios al Turtlebot simulando movimientos sobre un cuerpo de agua. La Figura 4.15 muestra los resultados obtenidos, se puede observar como la estimación obtenida del consenso dinámico sigue con gran exactitud a la referencia en los ángulos *roll* y *pitch*, el ángulo *yaw* tiene una buena aproximación a la referencia con una diferencia aproximada de 0.05 radianes por encima de la referencia.

4. 4.5. RESULTADOS EXPERIMENTALES

Haciendo uso de *ROS*, se envió una secuencia de movimientos al Turtlebot, los cuales consistían en hacerlo rotar un ángulo *yaw*, los resultados son mostrados en la Figura 4.16, puede observarse que la estimación obtenida del consenso dinámico, sigue con fidelidad a la referencia.

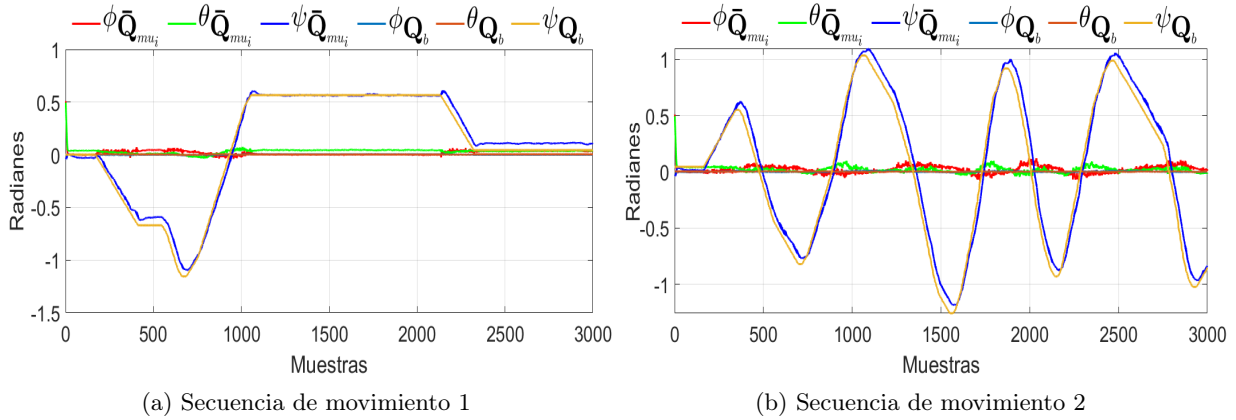


Figura 4.16: Ángulos de Euler obtenidos al mover al Turtlebot enviando comandos por ROS

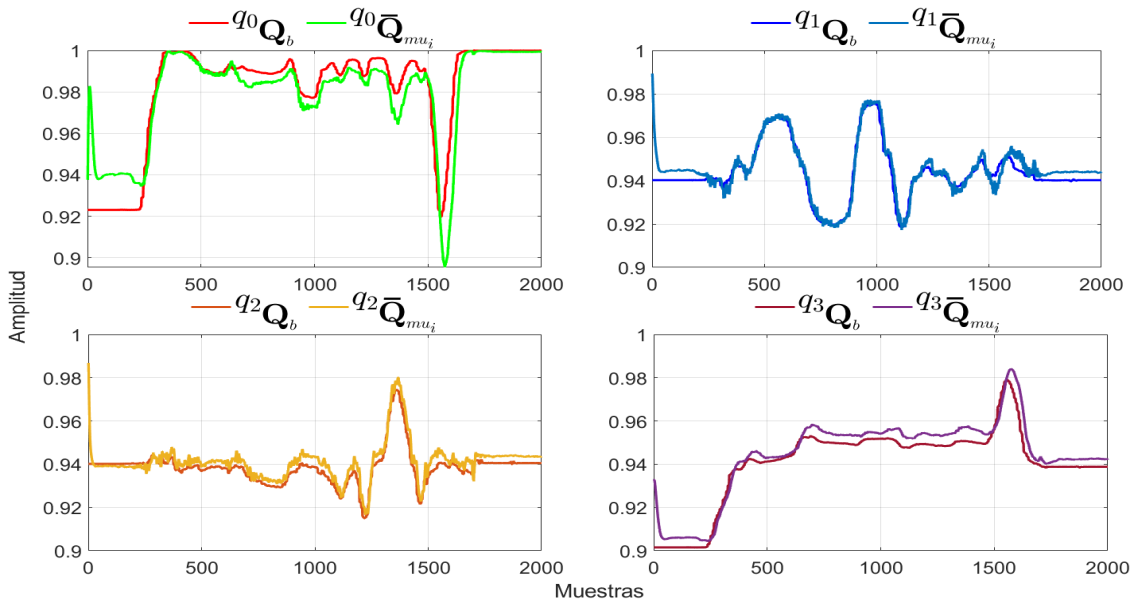


Figura 4.17: Cuaternión de referencia (Obtenido del Optitrack) y cuaternión estimado por el consenso dinámico, con movimientos aleatorios aplicados al Turtlebot

En las Figuras 4.17 y 4.18 se muestran los cuatro componentes correspondientes al cuaternión de referencia \mathbf{Q}_b y al cuaternión estimado por el consenso dinámico $\bar{\mathbf{Q}}_{mu_i}$, donde q_0 corresponde a la parte escalar y q_1, q_2, q_3 corresponden a la parte vectorial del cuaternión denotado en este documento como \vec{q} . Para la Figura 4.17, se aplico una serie de movimientos aleatorios manualmente al Turtlebot, mientras que para el resultado de la Figura 4.18, el movimiento fue realizado por el Turtlebot al enviarle comandos haciendo uso de *ROS*, en ambos casos la estimación dada por el

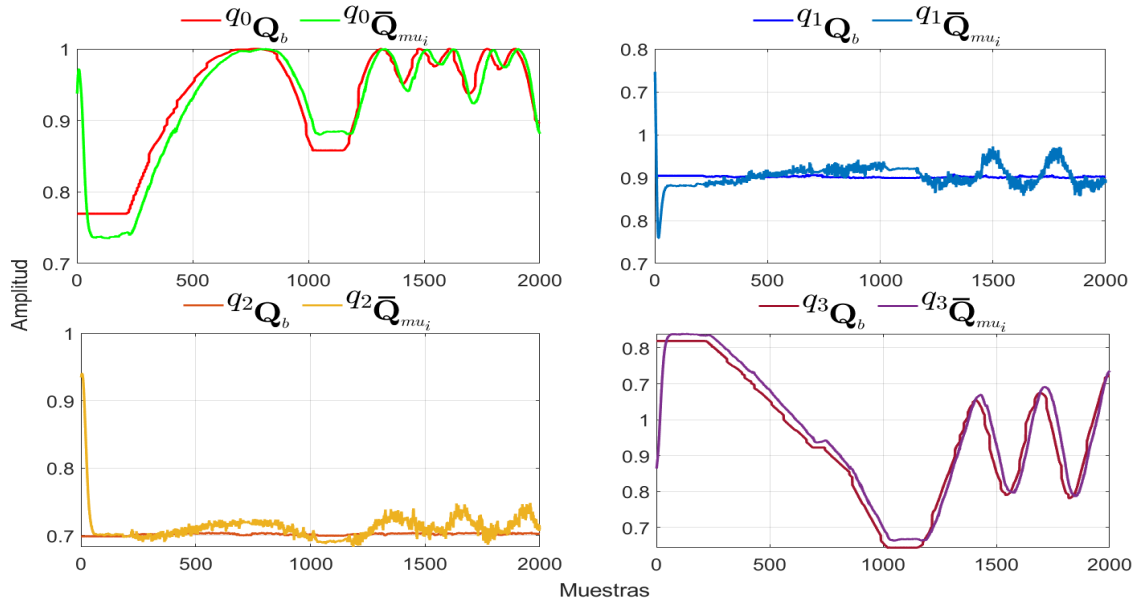


Figura 4.18: Cuaternión de referencia (Obtenido del Optitrack) y cuaternión estimado por el consenso dinámico, con movimientos realizados por el Turtlebot mediante comandos de *ROS*

consenso dinámico es capaz de seguir a la señal de referencia dada por el Optitrack.

Resultados: Robustez tolerante a fallas

Un aspecto importante de la estimación distribuida es garantizar la robustez del sistema, es decir que ante la falla de alguno de los sensores de cualesquiera de los nodos, el sistema tenga la capacidad de seguir realizando una buena estimación de la orientación del cuerpo rígido. Durante las pruebas experimentales se consideraron algunos posibles escenarios de fallos de sensores y son mostrados y explicados a continuación.

Escenario	Figura	MU ₁			MU ₂			MU ₃			Rotación
		\vec{b}_1	\vec{b}_2	$\vec{\omega}_G$	\vec{b}_1	\vec{b}_2	$\vec{\omega}_G$	\vec{b}_1	\vec{b}_2	$\vec{\omega}_G$	$\mathbf{Q}_b^{mu_i}$
–	–										
1	4.19a y 4.19b	X	✓	✓	✓	✓	✓	✓	✓	✓	[1; 0; 0; 0]
2	4.20a y 4.20b	X	✓	✓	✓	✓	✓	✓	✓	✓	ver Tabla 4.3
3	4.21a y 4.21b	✓	✓	✓	✓	X	✓	✓	✓	✓	[1; 0; 0; 0]
4	4.22a y 4.22b	✓	✓	✓	✓	X	✓	✓	✓	✓	ver Tabla 4.3
5	4.23a y 4.23b	✓	✓	✓	✓	✓	✓	✓	✓	X	[1; 0; 0; 0]
6	4.24a y 4.24b	✓	✓	✓	✓	✓	✓	✓	✓	X	ver Tabla 4.3
7	4.25a y 4.25b	X	✓	✓	✓	X	✓	✓	✓	X	ver Tabla 4.3
8	4.25c y 4.25d	X	✓	✓	✓	X	✓	✓	✓	X	ver Tabla 4.3

Tabla 4.4: Operación de los sensores del i -ésimo agente durante el proceso experimental

• Caso falla de alguno de los vectores de observación

Considere nuevamente la información presentada en la subsección 4.4.1, donde se presenta la matriz de observación y como está cambia en función del par de vectores $[\vec{b}_j^{mu_i}, \vec{r}_j]$ que falla.

La ecuación (4.29), presenta la forma que tomaría la matriz de observación en el caso de que el i -ésimo acelerómetro ($\vec{b}_1^{mu_i}$) fallará en el i -ésimo agente, y la ecuación (4.30), presenta la forma que tomaría la matriz de observación en el caso de que el i -ésimo magnetómetro ($\vec{b}_2^{mu_i}$) fallará en el i -ésimo agente.

Las modificaciones a la matriz \bar{H} indicadas por las ecuaciones (4.29) y (4.30), fueron realizadas durante las pruebas experimentales, esto considerando que el diseñador conoce cual de los sensores ha presentado fallas, ya en la práctica y como trabajo futuro será necesario el desarrollo e implementación de un algoritmo de detección de fallas que permita saber si alguno de los vectores de observación ha fallado para realizar las modificaciones correspondientes en la matriz \bar{H} de manera automática.

- **Caso falla del girómetro**

Otro escenario considerado durante las pruebas experimentales fue el caso de falla del girómetro en una de las MU's, en este caso, el vector de velocidad angular $\vec{\omega}_G$ es integrado como se indica en la ecuación (3.5), sin embargo al no contar con la medición $\vec{\omega}_G$ en el nodo correspondiente al fallo, el observador no lineal dejaría de ser ejecutado en dicho nodo y únicamente se tendría el cuaternión \bar{Q}_{mu_i} estimado por el algoritmo *LQS*, el cual puede ser utilizado para realizar el consenso dinámico junto a la estimación de los otros nodos. Estos cambios correspondientes en los algoritmos de estimación fueron realizados durante las pruebas experimentales, siempre considerando que el diseñador sabe que sensor ha fallado y en que nodo, a nivel práctico y como trabajo futuro al igual que con los vectores de observación, será necesario desarrollar un algoritmo de detección de fallas para saber si algún sensor se ha dañado y en que nodo con la finalidad de realizar los cambios correspondientes de manera autónoma.

La Tabla 4.4, muestra algunos posibles escenarios de funcionamiento de los sensores asociados a cada unidad de medición, y se presenta el cuaternión $\mathbf{Q}_b^{mu_i}$ asociado a cada MU y se asocia la Figura con los resultados para dicho escenario de funcionamiento de los sensores.

En la Figura 4.19, es interesante observar como la estimación se realiza de manera correcta, sin embargo al no aplicar movimientos al Turtlebot, como en el caso de la Figura 4.19a y en la parte final de la Figura 4.19b, el ángulo *yaw* empieza a sufrir una pequeña deriva conforme pasa el tiempo. Por lo que al tener los sistemas de referencia \mathbf{E}^{mu_i} y \mathbf{E}^b coincidentes en origen y alineados, podría ser perjudicial al presentarse el fallo del acelerómetro.

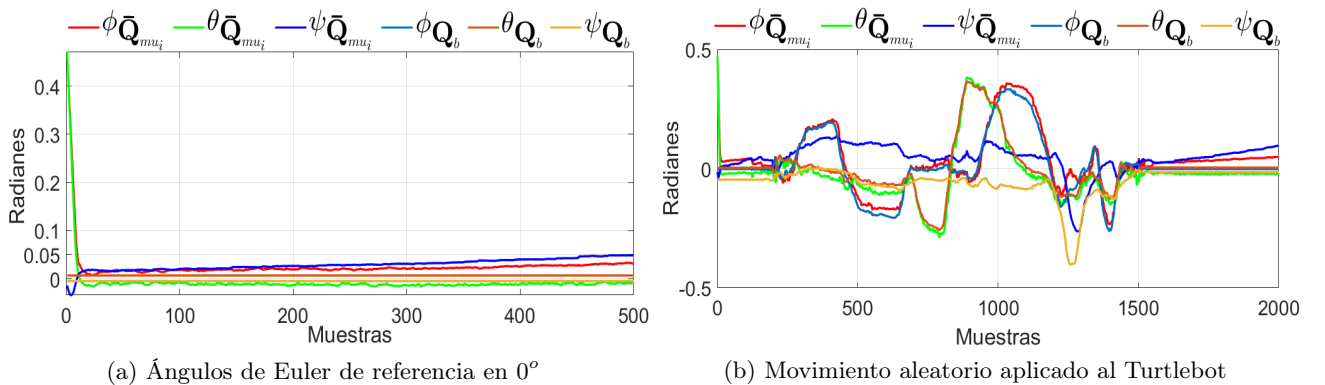


Figura 4.19: Ángulos de Euler obtenidos del consenso dinámico, asociado al escenario 1

En la Figura 4.20, se observa que en este caso la falla del acelerómetro no representa un

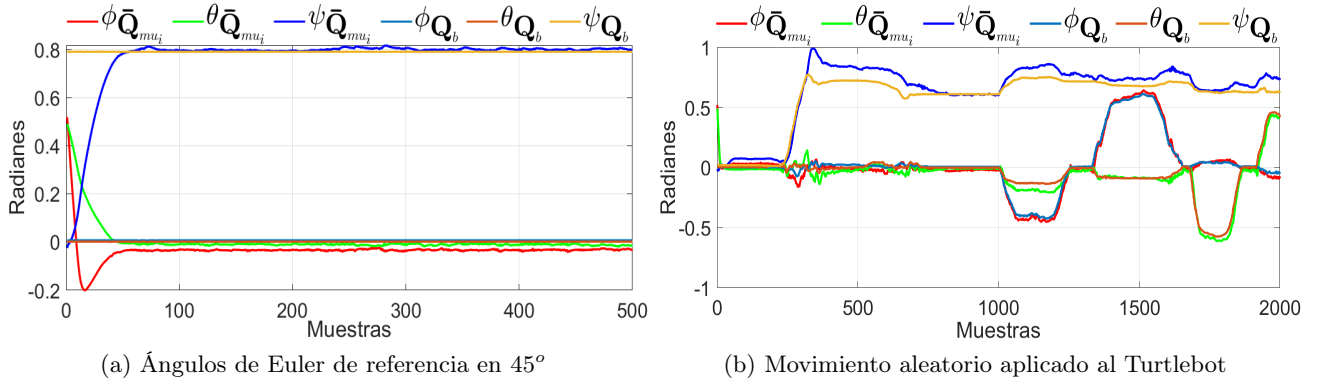


Figura 4.20: Ángulos de Euler obtenidos del consenso dinámico, asociado al escenario 2

problema significativo en la estimación de la orientación del Turtlebot, y el ángulo *yaw* no deriva en ningún momento a diferencia de los resultados de la Figura 4.19.

En las Figuras 4.21 y 4.22, los muestran que la perdida de uno de los magnetómetros no afecta gravemente la estimación de la orientación del cuerpo rígido, por lo que el sistema podría seguir considerándose como funcional.

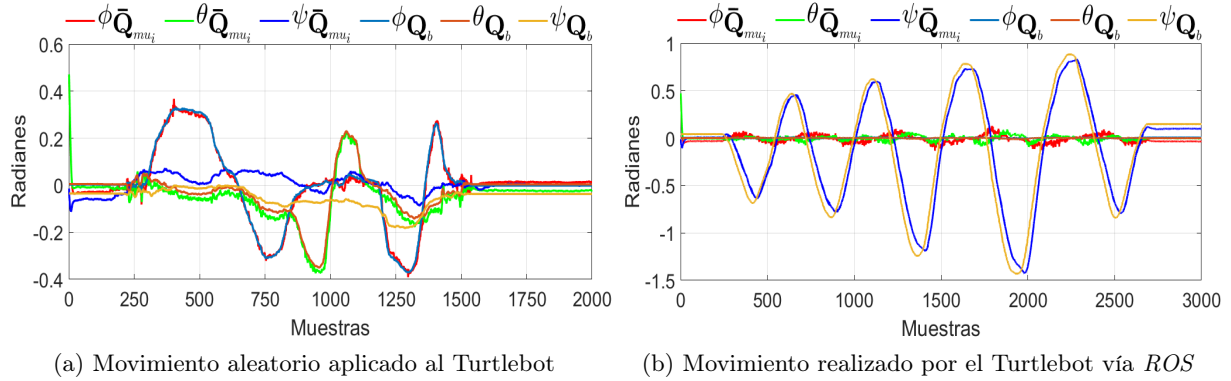


Figura 4.21: Ángulos de Euler obtenidos del consenso dinámico, asociado al escenario 3

Las Figuras 4.23 y 4.24 muestran los resultados tras aplicar un movimiento aleatorio, el ángulo ψ es el que presenta un pequeño error de estimación mientras que ϕ y θ logran estimar y seguir la referencia sin ningún problema. En el caso de las Figuras 4.24a y 4.24b, puede observarse como la estimación dada por el consenso dinámico sigue siendo altamente efectiva.

En la Figura 4.25a, se observa como la estimación realizada por el consenso dinámico tiene un mayor rango de error con respecto a la referencia en el ángulo ψ , además de que el tiempo en que la estimación pasa a un estado estable es mayor, aproximadamente de 3s, a comparación de cuando se cuentan con todos los sensores funcionales o únicamente uno de los nodos pierde un sensor. La Figura 4.25b, corresponden a un movimiento aleatorio aplicado al Turtlebot, se observa como la estimación de los ángulos ϕ y θ converge hacia la referencia con un ligero error en el pico de cada ángulo, el ángulo ψ estimado es el que presenta el error un poco más pronunciado al realizar los movimientos aleatorios. Las Figuras 4.25c y 4.25d, corresponden a movimientos realizados por el

4. 4.5. RESULTADOS EXPERIMENTALES

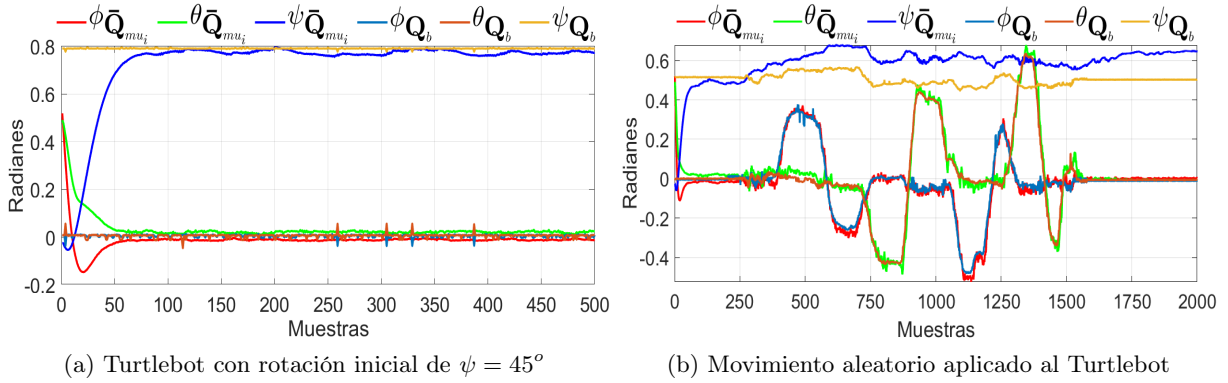


Figura 4.22: Ángulos de Euler obtenidos del consenso dinámico, asociado al escenario 4

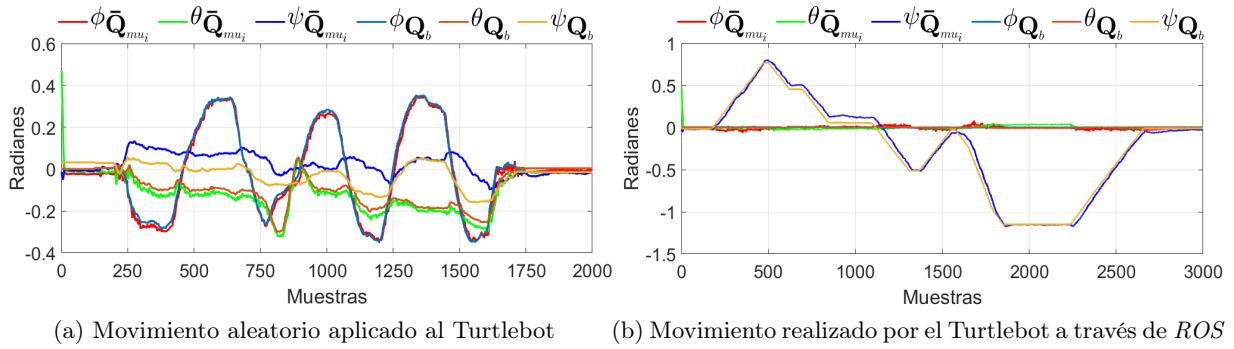


Figura 4.23: Ángulos de Euler obtenidos del consenso dinámico, asociado al escenario 5

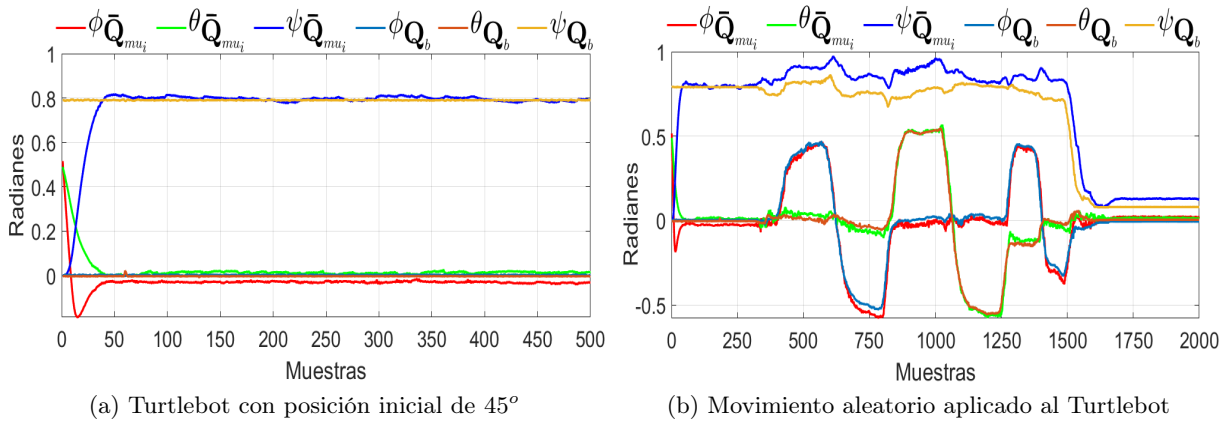


Figura 4.24: Ángulos de Euler obtenidos del consenso dinámico, asociado al escenario 6

Turtlebot tras el envío de una serie de comandos mediante ROS, en estos casos el Turtlebot realiza movimientos más suaves, por lo que la estimación del ángulo ψ se ve mejorada mientras esta en movimiento, cuando el Turtlebot se queda un tiempo en un determinado ángulo, la estimación presenta un pequeño error con respecto a la referencia.

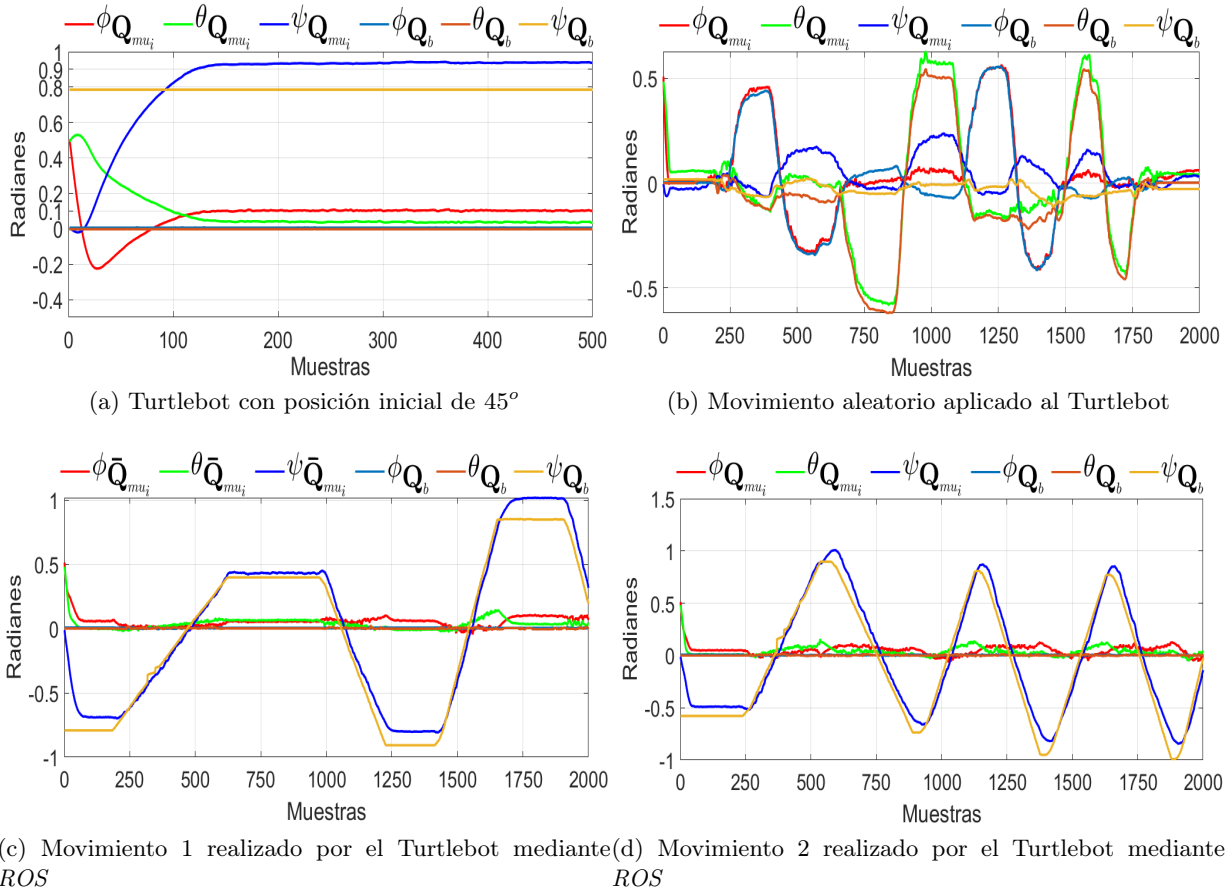


Figura 4.25: Ángulos de Euler obtenidos del consenso dinámico, para los escenarios 7 y 8

4.5.2. Algoritmo de consenso dinámico caso 2

Debido a la naturaleza misma de la aplicación donde se pretende hacer uso del sistema de estimación de orientación es importante que todo el algoritmo se ejecute en un sistema basado en microcontrolador, ya que son económicamente más accesibles, tienen buena capacidad de cómputo y cuentan con todo lo necesario para realizar comunicaciones entre diversos microcontroladores y otros dispositivos. En orden de ejecutar el algoritmo de estimación de orientación y consenso dinámico en un sistema de cómputo embebido, se ha elegido trabajar con la tarjeta de desarrollo NUCLEO-64 F411RE, la cual hace uso del microcontrolador STM32F411RE de la empresa ST-Microelectronics y cuenta con las características mostradas en la Figura 4.26.

Se hizo uso de tres de estas tarjetas de desarrollo, con la finalidad de generar una pequeña red de comunicación con tres nodos (ver Figura 4.27), cada nodo tiene las siguientes características:

4. 4.5. RESULTADOS EXPERIMENTALES

- Capacidad de comunicación con su nodo vecino, vía UART
- Capacidad de adquirir datos de una MU a través de su puerto I²C
- Capacidad de ejecutar el algoritmo de estimación de orientación
- Capacidad de ejecutar el algoritmo de consenso dinámico

Con la finalidad de comparar el cuaternión obtenido del consenso dinámico en este nivel de implementación, el nodo 1 envía el \bar{Q}_1 vía UART a una PC donde se tiene el software de MATLAB quien recibe el \bar{Q}_1 y recibe también el cuaternión del sistema Optitrack para realizar las comparaciones correspondientes tanto en ángulos de Euler como en cuaterniones.

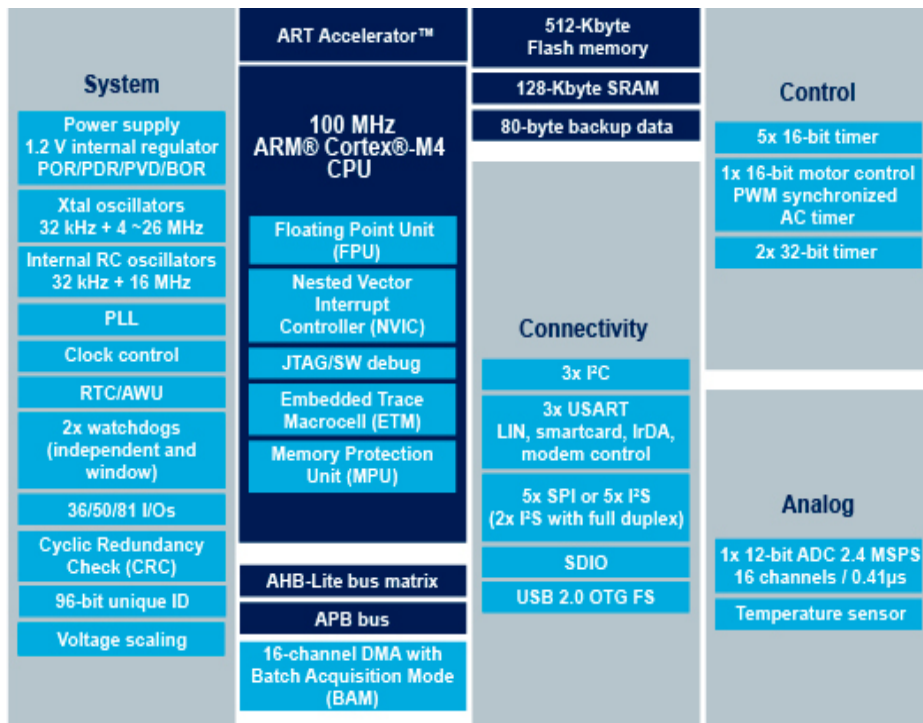


Figura 4.26: Características del microcontrolador utilizado para ejecutar los diferentes algoritmos de estimación (Fuente: <https://www.st.com/en/microcontrollers-microprocessors/stm32f411re.html>)

La Figura 4.28 corresponde a un diagrama de bloques donde se observa cada nodo de la red, los procesos de cálculo que desarrolla cada uno y la forma en que se comunican, en la parte inferior se observa como se conforma la trama de datos que cada nodo comparte con su vecino, esta trama se compone de 19 bytes, donde el primer byte corresponde al encabezado de la trama, los 16 bytes siguientes corresponden al cuaternión, el byte 18 corresponde a una suma de comprobación para verificar la correcta recepción de la trama y el último byte corresponde a una indicación de final de trama.

Resultados caso 2

Los resultados, primeramente se presentan en ángulos de Euler, porque son más intuitivos y de mejor comprensión, sin embargo, posteriormente en esta misma sección se presentan los resultados

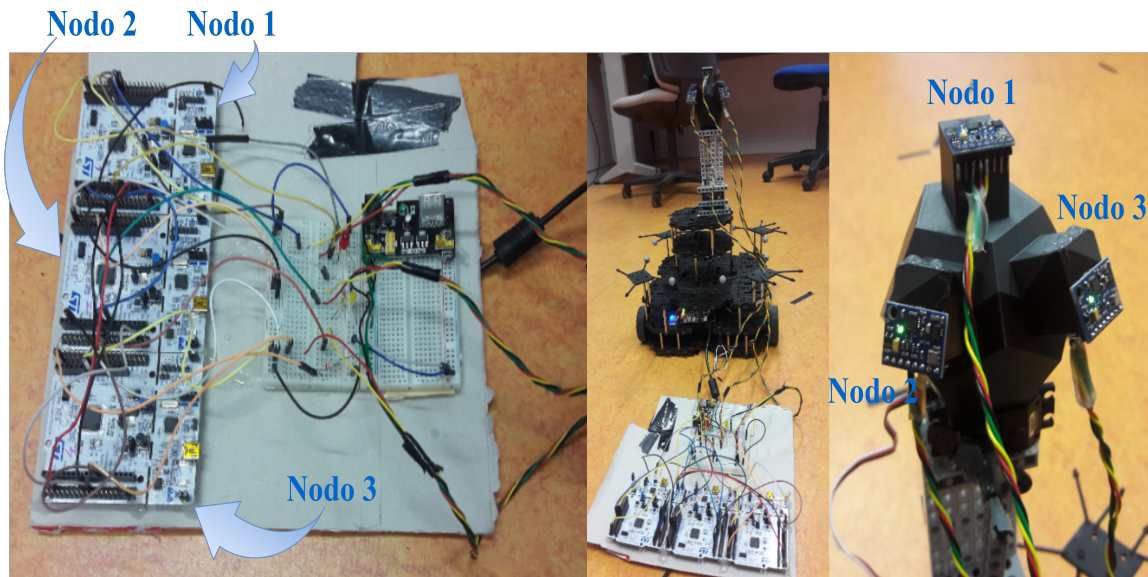


Figura 4.27: Implementación de la red de sensores, para la estimación de la orientación de manera distribuida

usando el cuaternión como parametrización.

La Figura 4.29, muestra los ángulos de Euler obtenidos al procesar los algoritmos de cada nodo en su respectivo microcontrolador. Las Figuras 4.29a y 4.29b muestran los ángulos obtenidos al aplicar una serie de movimientos aleatorios al Turtlebot, se observa que los tres ángulos estimados por el consenso dinámico son muy cercanos a los ángulos de referencia dados por el sistema Optitrack. Las Figuras 4.29c y 4.29d, corresponden a una serie de movimientos realizados por el Turtlebot al enviarle comandos haciendo uso de ROS, en este caso, como puede observarse únicamente existe movimiento de rotación en uno de los ejes del robot, debido a la configuración del mismo. La estimación obtenida muestra un buen seguimiento con respecto a la señal de referencia del Optitrack. Es importante resaltar que durante la obtención de estos resultados el algoritmo *LQS* debería recibir el cuaternión inicial del consenso dinámico para toda $t > 0$, como se muestra en la Figura 4.6, sin embargo, esto no fue posible implementarlo, probablemente debido al tiempo consumido en las comunicaciones y ejecución de los algoritmos, por lo que el cuaternión inicial para el algoritmo *LQS* fue proporcionado por el observador no lineal.

Las Figuras 4.30 y 4.31, corresponden a las gráficas de los componentes del cuaternión estimado por el consenso dinámico ejecutado por el nodo microcontrolador, en ambos casos se aplica una serie de movimientos aleatorios al Turtlebot y la estimación es comparada con el cuaternión de referencia dado por el sistema Optitrack. En general el cuaternión estimado presenta una buena exactitud y respuesta dinámica a las rotaciones aplicadas.

Implementar los diferentes algoritmos de estimación de orientación que son modelados matemáticamente por ecuaciones diferenciales, fue posible gracias al uso de métodos numéricos de solución de Ecuaciones Diferenciales Ordinarias (*ODE*). La solución numérica para el observador

4. 4.5. RESULTADOS EXPERIMENTALES

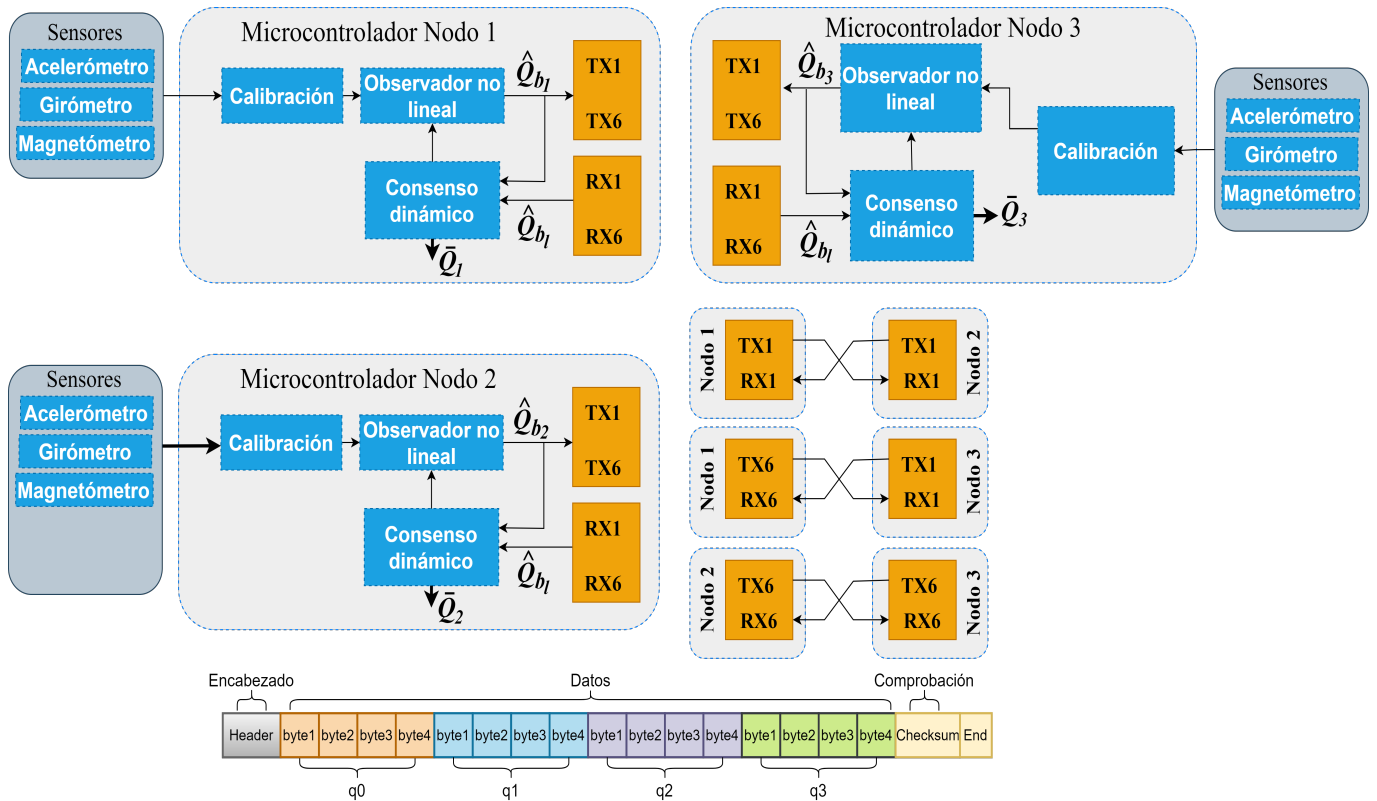


Figura 4.28: Diagrama de bloques del intercambio de información entre los tres nodos

no lineal y el algoritmo de consenso dinámico fue realizada haciendo uso del método Runge Kutta de orden 4, el cual es presentado en el apéndice A. En el siguiente capítulo se considera la estimación de la posición de un móvil, haciendo uso del GPS, se introduce las ventajas de fusionar las medidas del GPS con las estimaciones de posición realizadas por un acelerómetro. Luego se describe el procedimiento de transformación del sistema de coordenadas \mathbf{E}^{EC-EF} al \mathbf{E}^f y finalmente se presentan resultados experimentales de la transformación de los sistemas de coordenadas mencionados.

4. 4.5. RESULTADOS EXPERIMENTALES

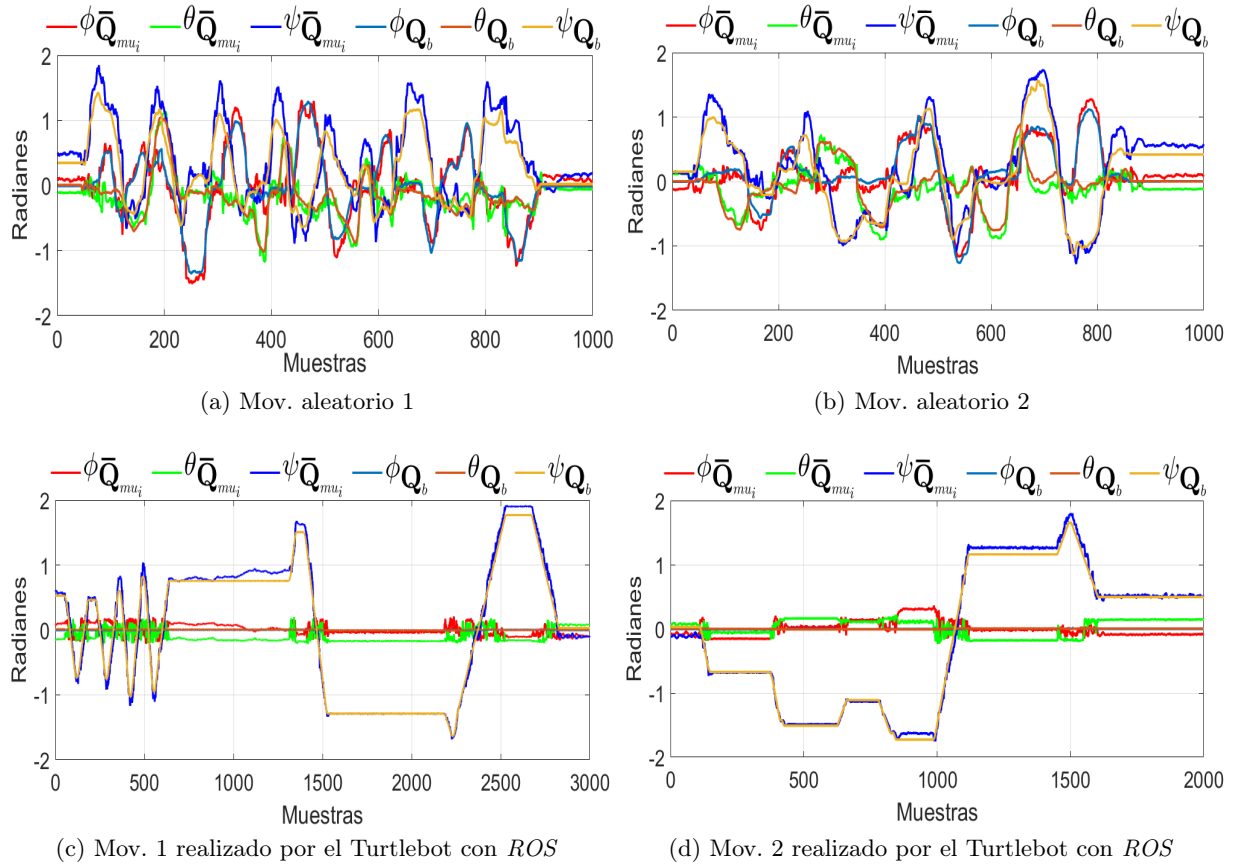


Figura 4.29: Ángulos de Euler obtenido del consenso dinámico procesado a nivel del microcontrolador.

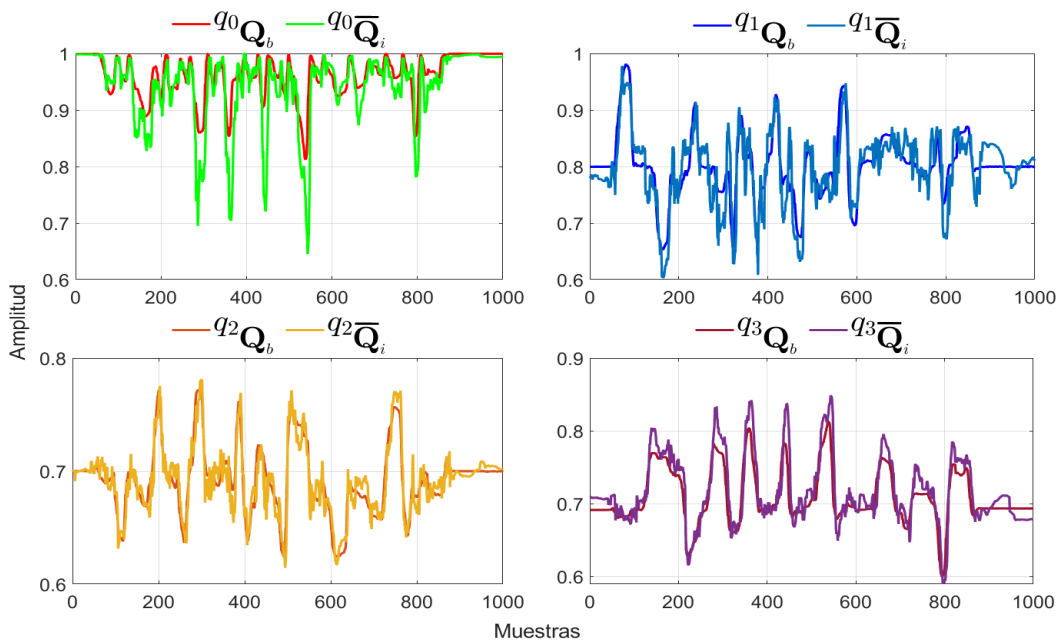


Figura 4.30: Cuaternión 1 obtenido del consenso dinámico implementado en la tarjeta de desarrollo NUCLEO-64.

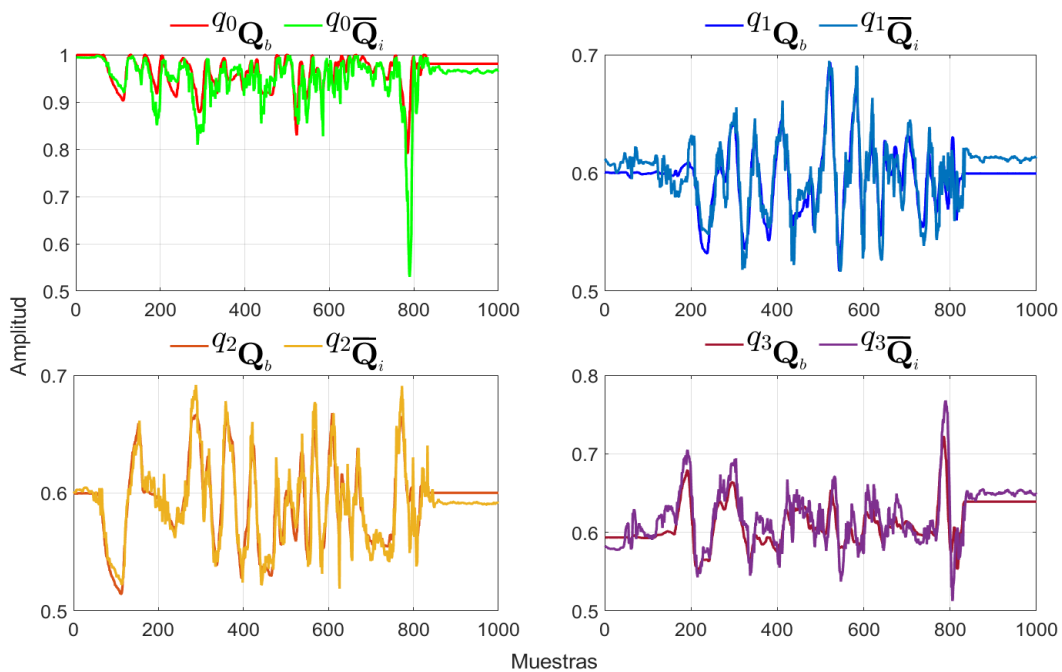


Figura 4.31: Cuaternión 2 obtenido del consenso dinámico implementado en la tarjeta de desarrollo NUCLEO-64.

Capítulo 5

Integración GPS/INS

En robótica y en muchas otras aplicaciones, contar con información de posición del móvil (robot, vehículo, etc.) en general es de gran importancia por cuestiones de seguridad o porque esta información forma parte de las señales de control del móvil. En los vehículos no tripulados es muy usual contar con un receptor GPS para la determinación de la posición del propio vehículo, sin embargo como se ha visto en capítulos anteriores, el receptor GPS tiene una tasa de muestreo muy baja que va de 1Hz a 5Hz aproximadamente, esto conlleva a que vehículos con alta dinámica se vean perjudicados por la baja tasa de muestreo del receptor GPS. Suponga que una boya viaja a una velocidad de 10 m/s y hace uso de un receptor GPS a una tasa de 1Hz para dibujar la ruta por la que pasa, durante los primeros 20 metros la boya viaja en línea recta, pero en los siguientes 10 metros la boya dibuja una serie de curvas en su trayectoria y luego regresa a la ruta recta que venía siguiendo desde el inicio (obsérvese la Figura 5.1). La trayectoria curva realizada por la boya no quedaría registrada por el receptor GPS debido a su baja tasa de muestreo y como consecuencia quedaría fuera de la ruta trazada al final del trayecto.

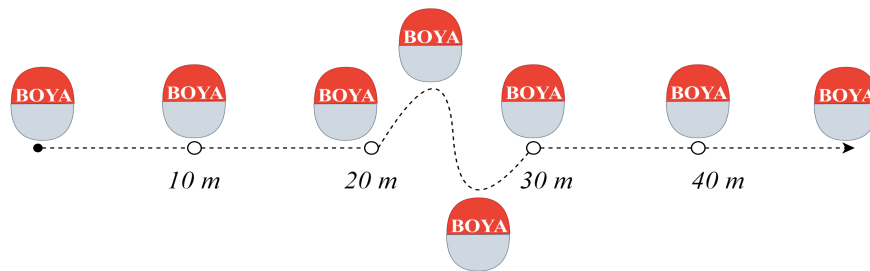


Figura 5.1: Trayectoria recorrida por la boya, las líneas discontinuas representan datos de posición obtenidos a partir de la integración de las medidas de un acelerómetro

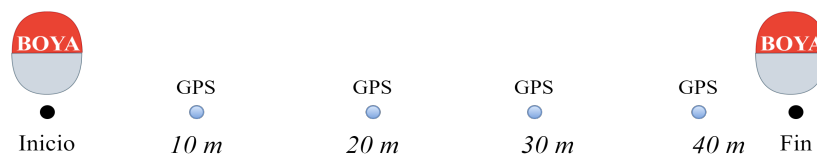


Figura 5.2: Trayectoria de la boya registrada por el receptor GPS

En la trayectoria final hecha por el receptor GPS indicaría un desplazamiento en línea recta por parte de la boya como se indica en la Figura 5.2, lo que podría representar un problema en el control de la boya del cual no se percataría el usuario, si se espera que únicamente avance en línea recta. Es por ello que un buen equipo que brinde información de posición de un vehículo con alta dinámica debe tener una frecuencia de muestreo alta para lograr registrar el desplazamiento total realizado por el vehículo. En este trabajo nos enfocaremos únicamente en el uso del receptor GPS para obtener información de la posición de la boya marina con respecto a un sistema de referencia inercial \mathbf{E}^f , con el objetivo de colocar la información brindada por el receptor GPS en un sistema de referencia en común al utilizado por los acelerómetros, donde en un trabajo futuro sería posible integrar las mediciones que ofrece un acelerómetro y un GPS con la finalidad de hacer uso de un observador de posición con una tasa de muestreo alta.

5.1. Coordenadas del GPS

El receptor GPS ofrece un conjunto completo de información, como se introdujo en el capítulo 2 sección 2.4.4. Sin embargo, la trama de datos que más interesa en este trabajo es la GRPS en la cual se encuentra la posición del receptor GPS sobre la tierra. Esta posición se encuentra expresada en el sistema de referencia \mathbf{E}^{EC-EF} el cual puede expresarse en dos maneras, en un sistema de coordenadas rectangulares (X^e, Y^e, Z^e) y en un sistema de coordenadas geodésico o curvilíneo (λ, φ, h) , donde λ es la latitud, φ es la longitud y h la altura con respecto a el nivel del mar. Es en este último sistema de coordenadas como el receptor GPS despliega la información de su propia ubicación sobre el globo terrestre. La latitud tiene un rango de -90° a $+90^\circ$, un ángulo negativo implica coordenadas del hemisferio sur y un ángulo positivo coordenadas del hemisferio norte. La longitud tiene un rango de -180° (oeste) a $+180^\circ$ (este) con respecto al meridiano de Greenwich. Por lo tanto el vector de posición dado por el receptor GPS expresado en el sistema de referencia \mathbf{E}^{EC-EF} se denota como:

$$\mathbf{P}^e = \begin{pmatrix} \lambda \\ \varphi \\ h \end{pmatrix} \quad (5.1)$$

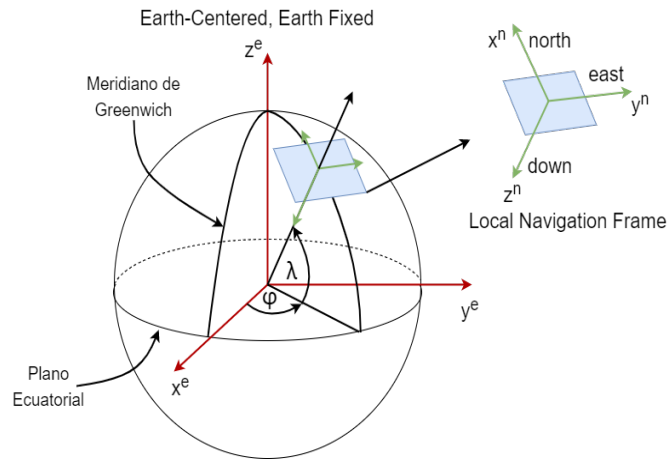


Figura 5.3: Sistema de referencia centrado y fijo en la Tierra \mathbf{E}^{EC-EF} , sistema de referencia local \mathbf{E}^{NED}

5.2. Cambio de coordenadas

Realizar una integración o fusión de las mediciones del receptor GPS y del sistema de navegación inercial implica situar todas las mediciones en un marco de referencia común. En este trabajo de tesis el sistema de navegación desarrollado será utilizado en una boya marina, por lo que se considera que su área de movimiento o desplazamiento estará restringida y como consecuencia la longitud y latitud pueden considerarse constantes, en este caso el sistema de referencia \mathbf{E}^{NED} puede ser utilizado para la navegación y control, y puede considerarse un sistema inercial, el cual es denotado como \mathbf{E}^f y donde las leyes de Newton son aplicables.

Dada las consideraciones anteriores, las mediciones del receptor GPS deberán ser transformadas al sistema \mathbf{E}^f , para ello se realiza una primera transformación del sistema coordenadas curvilíneo al sistema de coordenadas rectangular del sistema de referencia \mathbf{E}^{EC-EF} , como se indica a continuación [23]:

$$\mathbf{P}^e = \begin{pmatrix} x^e \\ y^e \\ z^e \end{pmatrix} = \begin{pmatrix} (R_c + h) \cos \lambda \cos \varphi \\ (R_c + h) \cos \lambda \sin \varphi \\ [R_c(1 - e^2) + h] \sin \lambda \end{pmatrix} \quad (5.2)$$

Donde R_c es el radio de curvatura de la Tierra, en base al modelo elipsoidal WGS-84,

$$R_c = \frac{a}{\sqrt{1 - e^2 \sin^2(\lambda)}} \quad (5.3)$$

$e = 0.0818$ es la excentricidad del elipsoide y $a = 6378137 \text{ m}$ es el radio ecuatorial. La posición expresada en el sistema de referencia \mathbf{E}^f viene dada por la siguiente ecuación:

$$\mathbf{P}^f = \begin{pmatrix} x^f \\ y^f \\ z^f \end{pmatrix} = \mathbf{R}_e^f (\tilde{\mathbf{P}}^e - \mathbf{P}_O^e) \quad (5.4)$$

Donde $\tilde{\mathbf{P}}^e$ es la posición dada por el receptor GPS expresado en el sistema de coordenadas rectangular, \mathbf{P}_O^e es el origen del sistema de referencia \mathbf{E}^f , expresado en coordenadas rectangulares del sistema de referencia \mathbf{E}^{EC-EF} y el cual puede ser obtenido mediante un receptor GPS. \mathbf{R}_e^f es la matriz de rotación que lleva del sistema \mathbf{E}^{EC-EF} al sistema \mathbf{E}^f y viene dada por

$$\mathbf{R}_e^f = \begin{pmatrix} -\sin \lambda_o \cos \varphi_o & -\sin \lambda_o \sin \varphi_o & \cos \lambda_o \\ -\sin \varphi_o & \cos \varphi_o & 0 \\ -\cos \lambda_o \cos \varphi_o & -\cos \lambda_o \sin \varphi_o & -\sin \lambda_o \end{pmatrix} \quad (5.5)$$

Donde λ_o y φ_o representan la latitud y longitud del origen respectivamente.

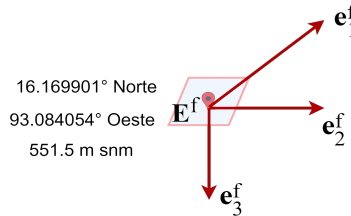


Figura 5.4: Origen del sistema de referencia \mathbf{E}^f

El origen del sistema de referencia \mathbf{E}^f al momento es $\lambda = 16.169901^\circ$, $\varphi = -93.084054^\circ$ y $h = 551.5m$, como se ilustra en la Figura 5.4. Por lo tanto el origen del sistema de referencia \mathbf{E}^f expresado en coordenadas rectangulares puede obtenerse aplicando (5.2), quedando de la siguiente manera:

$$\mathbf{P}_O^e = \begin{pmatrix} x_o^e \\ y_o^e \\ z_o^e \end{pmatrix} = \begin{pmatrix} -2813766.504 \\ -5443744.680 \\ 1764946.724 \end{pmatrix} m \quad (5.6)$$

5.3. Resultados del capítulo

Con la finalidad de generar un sistema de referencia inercial \mathbf{E}^f donde nosotros lo requiramos, y que este sirva como referencia a partir del cual nuestro vehículo pueda registrar su desplazamiento, se realizó un escenario práctico en la ciudad de Puebla, México, donde un dron recorrió los extremos de un parque y registro su trayectoria dada por un GPS, la información fue trasladada al ambiente de MATLAB y se realizaron las operaciones correspondientes a las ecuaciones (5.2) - (5.5).

5.3.1. Resultados de la conversión de coordenadas EC-EF a coordenadas cartesianas

El origen del sistema de referencia \mathbf{E}^f es: $\lambda_o = 18.9848^\circ$, $\varphi_o = -98.2028^\circ$ y $h = 2121msnm$, ahora, hay que convertir de grados a radianes las medidas de la latitud y longitud, como sigue:

$$\begin{aligned} \lambda_{rad} &= \lambda_o * \left(\frac{\pi}{180}\right) \\ \varphi_{rad} &= \varphi_o * \left(\frac{\pi}{180}\right) \end{aligned} \quad (5.7)$$

Una vez expresadas la latitud y longitud en radianes, se transforma del sistema de coordenadas curvilíneo al sistema de coordenadas rectangular, aplicando la ecuación (5.2), para el origen, el vector \mathbf{P}_O^e quedaría como sigue:

$$\mathbf{P}_O^e = \begin{pmatrix} x_o^e \\ y_o^e \\ z_o^e \end{pmatrix} = \begin{pmatrix} -861100.9125952375 \\ -5973595.384956398 \\ 2062452.616892105 \end{pmatrix} m \quad (5.8)$$

La matriz de rotación que transforma del sistema de coordenadas \mathbf{E}^{EC-EF} al sistema de coordenadas \mathbf{E}^f estará en función de λ_o y φ_o , finalmente deberá obtenerse el vector $\tilde{\mathbf{P}}^e$ en cada instante de tiempo t , para obtener el vector \mathbf{P}^f .

La Figura 5.5, muestra la trayectoria recorrida por el dron, con respecto al sistema de referencia \mathbf{E}^{EC-EF} en coordenadas curvilíneas, como puede observarse esta gráfica no ofrece información muy útil al usuario, acerca de cuantos metros recorrió el dron. La Figura 5.6 corresponde a la misma trayectoria del dron pero con respecto al sistema de coordenadas rectangular, donde el origen de este sistema es el centro de la Tierra, al igual que las coordenadas curvilíneas, las coordenadas rectangulares del sistema \mathbf{E}^{EC-EF} , no ofrece información útil al usuario en áreas restringidas de movimiento.

La Figura 5.7, muestra la trayectoria realizada por el dron, con respecto a un sistema de referencia inercial \mathbf{E}^f , este sistema es el que se formo a partir del vector de posición \mathbf{P}_O^e , como puede observarse graficar la información dada por el receptor GPS con respecto al nuevo sistema

5. 5.3. RESULTADOS DEL CAPÍTULO

\mathbf{E}^f es útil al usuario, ya que la gráfica nos da información más intuitiva y fácil de comprender acerca del recorrido.

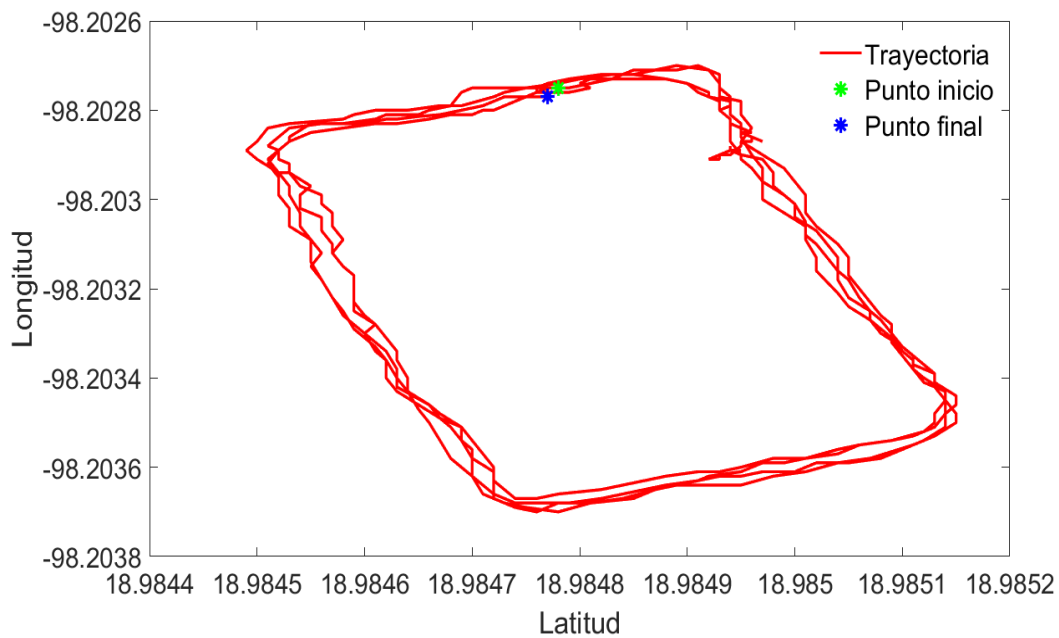


Figura 5.5: Trayectoria del dron, con respecto al sistema de coordenadas curvilíneo \mathbf{E}^{EC-EF}

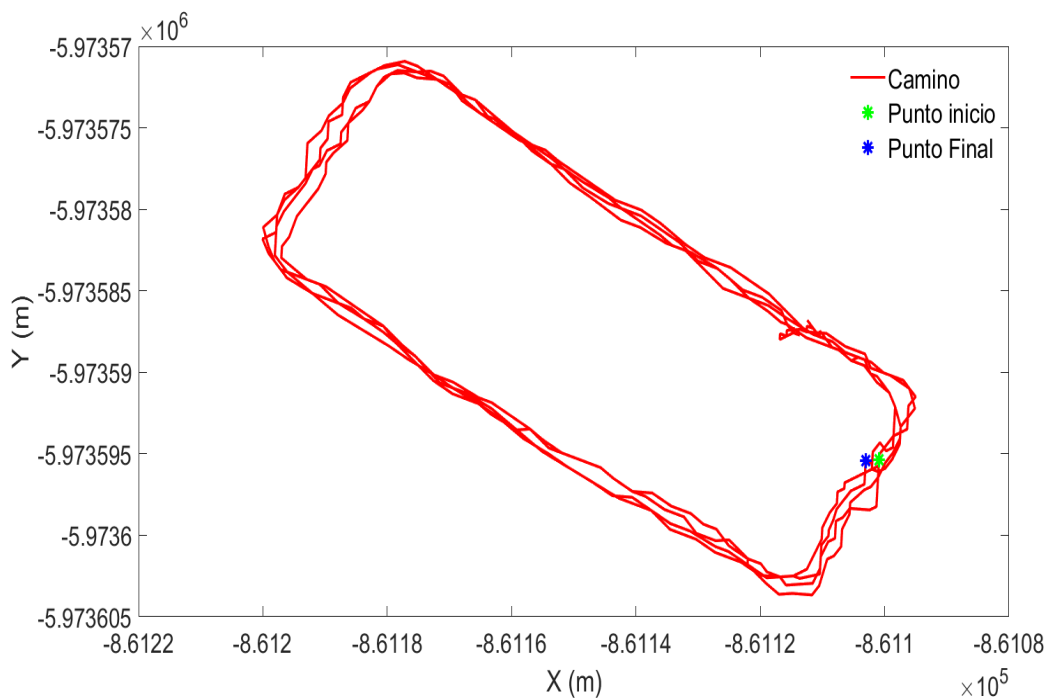


Figura 5.6: Trayectoria del dron, con respecto al sistema de coordenadas rectangulares \mathbf{E}^{EC-EF}

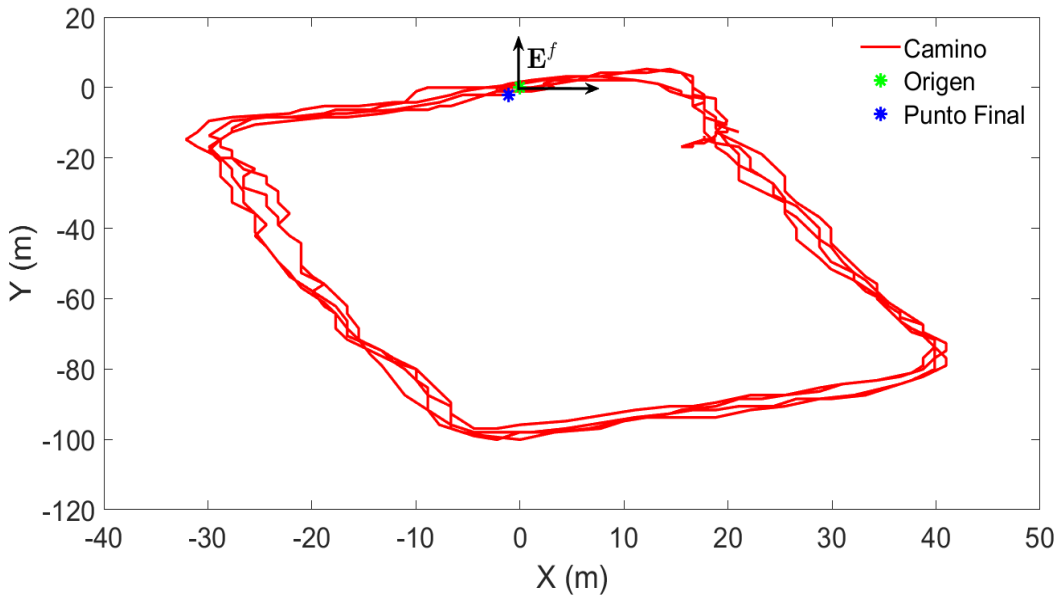


Figura 5.7: Trayectoria del dron, con respecto a un nuevo sistema de referencia E^f , el cual puede ser definido por el usuario

5.3.2. Plataforma experimental (Hardware In the Loop)

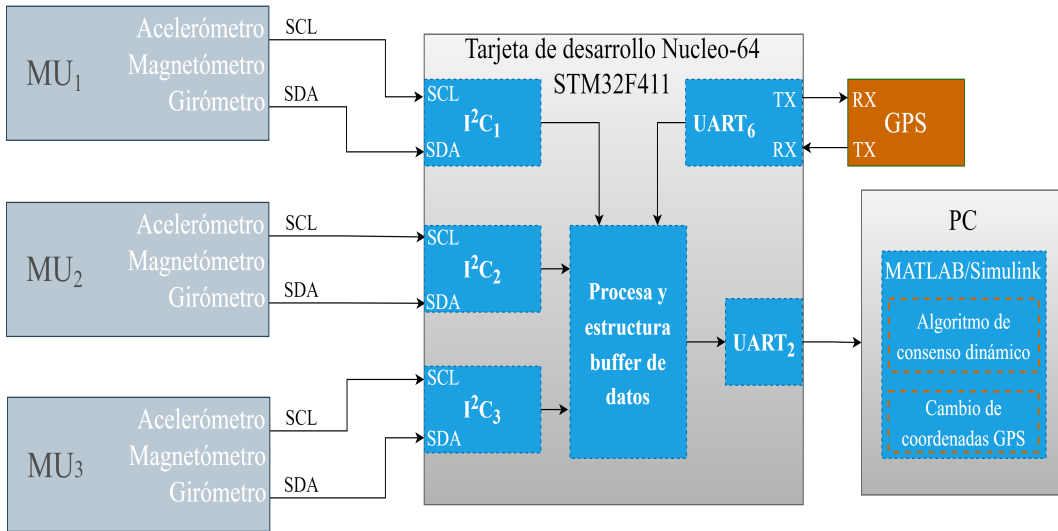


Figura 5.8: Diagrama de bloques de la plataforma experimental implementada

La Figura 5.8, describe de manera gráfica la plataforma experimental implementada con fines de probar de manera fácil y eficiente el algoritmo de consenso desarrollado, así como el cambio de coordenadas realizado a las mediciones del GPS. El algoritmo de consenso dinámico a ejecutarse en MATLAB/Simulink corresponde al presentado en la Figura 4.6, tres de estos algoritmos estarían ejecutándose, donde a cada uno le corresponde una MU, y para el cambio de coordenadas se ejecutan las ecuaciones (5.2) - (5.5). En la Figura 5.9, se muestra la implementación de la plata-

5. 5.3. RESULTADOS DEL CAPÍTULO

forma experimental, relacionada al diagrama de bloques de la Figura 5.8, y en la Figura 5.10, se muestran los resultados obtenidos de la plataforma experimental al aplicar una serie de rotaciones bien conocidas con respecto al ángulo ψ , al permanecer el cuerpo en un punto fijo, es decir, no existe desplazamiento.

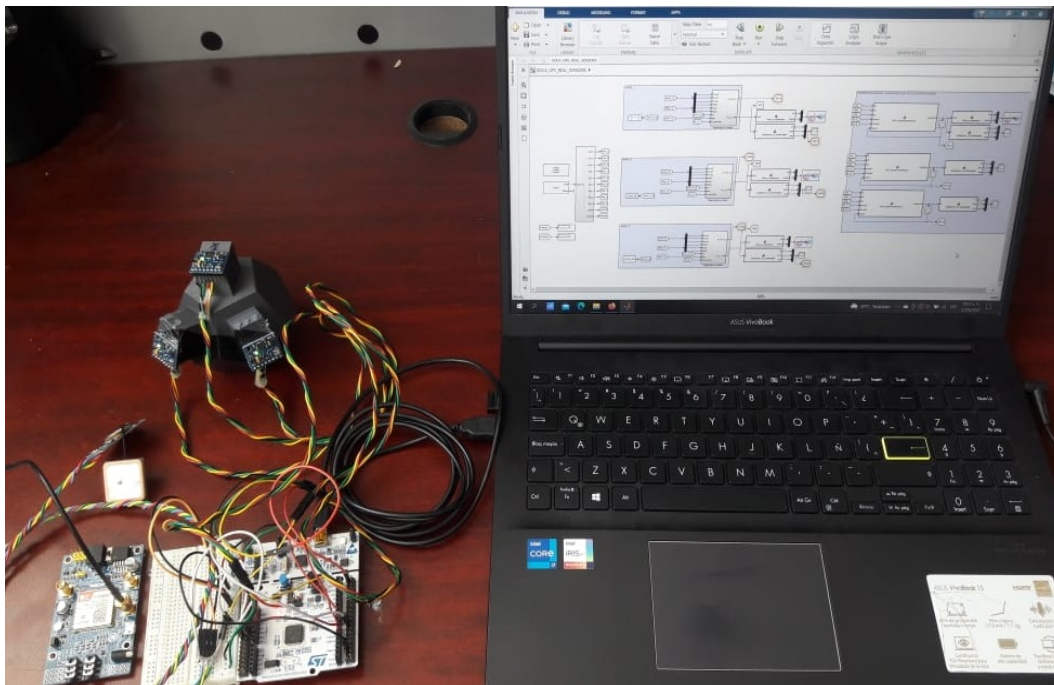


Figura 5.9: Plataforma experimental implementada

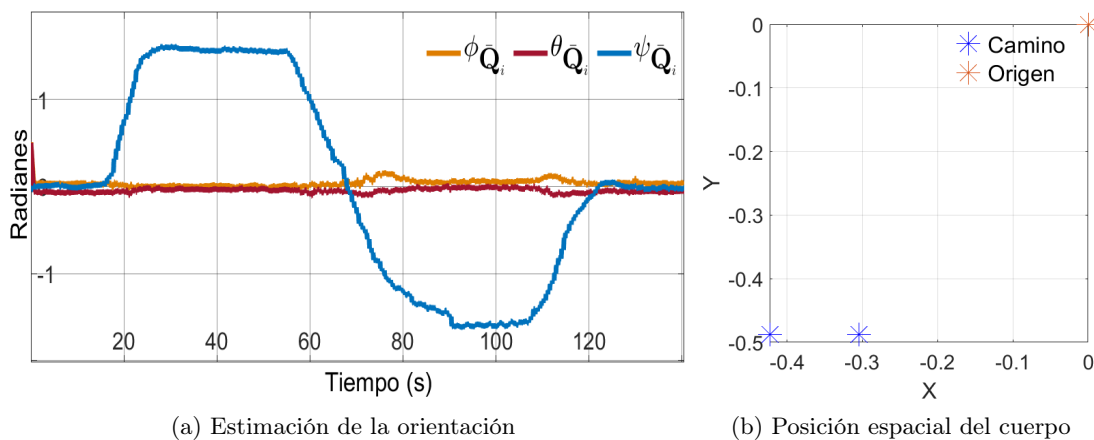


Figura 5.10: Rotación bien conocida con respecto al ángulo ψ y obtención de la posición expresado en el sistema cartesiano

Conclusiones

En los últimos años, se han estado desarrollando vehículos terrestres, aéreos y marinos autónomos, ya sea con fines de investigación, transporte público o para realizar trabajos peligrosos. En este tipo de aplicaciones, la estimación de la orientación del vehículo es de vital importancia ya que forma parte crucial del sistema de navegación que le da la autonomía necesaria para realizar los trabajos asignados, asegurar el continuo y buen funcionamiento del sistema de navegación se vuelve relevante. En este trabajo de tesis se implementó un estimador de orientación haciendo uso de un algoritmo determinista y de un observador no lineal cuya información fue fusionada para obtener la mejor parte de ambas estimaciones, se diseñó y propuso un nuevo algoritmo de consenso dinámico para cuaterniones basado en teoría de grafos, con el objetivo de obtener un único cuaternión que represente la orientación del vehículo, además de proveer de robustez al sistema y mejorar la exactitud de este. A continuación se presentan algunos puntos concluyentes del trabajo realizado.

- La estimación de la orientación haciendo uso de vectores de observación y vectores de referencia (algoritmo determinista), resulta tener muy buena respuesta, sin embargo, la estimación se ve un poco afectada con ruido, debido a las características de los acelerómetros y magnetómetros.
- Fusionar la información obtenida del algoritmo determinista con la información obtenida de un observador no lineal, el cual hace uso de las medidas obtenidas de un girómetro, mejora la estimación de la orientación del vehículo, ya que se reducen los niveles de ruido presentados en la estimación al ser filtrados por el observador no lineal.
- Distribuir espacialmente las MU's sobre el móvil ayuda a obtener una mejor observación de su orientación.
- Establecer como condición inicial el cuaternión estimado del consenso dinámico al algoritmo *LQS*, provee de robustez al sistema, haciéndolo tolerante ante el posible fallo de alguno de sus sensores.
- Implementar en un microcontrolador los diferentes algoritmos de estimación de orientación es más eficiente si se hace uso de la parametrización con cuaterniones, ya que generan menor carga computacional y se evitan las singularidades que presentan los ángulos de Euler.
- Transformar las mediciones obtenidas de un GPS a un sistema de coordenadas inercial, ofrece mayor información útil al usuario, esto considerando áreas restringidas de movimiento. Además de que el origen del nuevo sistema de coordenadas puede ser elegido estratégicamente por el diseñador.
- La transformación de coordenadas presentada en el capítulo 5, sitúa las nuevas medidas del GPS en un sistema de referencia en común con la información que podría obtenerse al integrar las mediciones de un acelerómetro, lo que permitiría la fusión de ambos sensores, con la finalidad de obtener las mejores características de cada uno de ellos.

Trabajo futuro

Como trabajo futuro y mejoras al sistema diseñado, se proponen los siguientes puntos:

- Fusionar la estimación de posición dada por un acelerómetro, con las mediciones de un GPS.
- Diseñar un algoritmo de detección de fallas para los sensores.
- Diseñar el hardware dedicado para el sistema desarrollado.
- Optimizar el procedimiento de intercambio de información entre los nodos de la red.

Apéndice A

Métodos numéricos

A.1. Métodos numéricos en la solución de Ecuaciones diferenciales ordinarias

Las ecuaciones diferenciales ordinarias ODEs *Ordinary Differential Equations*, son una herramienta matemática muy poderosa, debido a que muchos problemas del mundo real pueden ser modelados matemáticamente, y la solución a dicho problema consiste en resolver la ecuación diferencial asociada a su modelo [49]. Un ejemplo claro de estas aplicaciones, se tienen en este trabajo, se toma la ecuación cinemática del observador no lineal para la estimación de orientación en el tiempo t al encontrar su solución. También se tiene el algoritmo de consenso dinámico propuesto el cual consiste en un conjunto de ecuaciones diferenciales ordinarias y que al darles solución se obtiene un promedio de la solución de cada nodo de la red.

En sistemas de control, robótica y otras disciplinas a fines, es común trabajar con modelos matemáticos que implican ecuaciones diferenciales con condiciones iniciales de la forma:

$$\dot{y} = f(x, y(x)), \quad y(x_0) = y_0 \quad (\text{A.1})$$

Estas ecuaciones, en la mayoría de los casos deberán ser resueltas a nivel de computadores o microcomputadores, como por ejemplo sistemas basados en microcontroladores, por lo que para solucionar la EDO se utilizan métodos numéricos que ofrecen una muy buena aproximación a la solución. Entre los métodos numéricos más conocidos, se encuentran el método de Euler, que consiste en aproximar la solución del problema de valor inicial en un intervalo $[a, b]$, dividiendo este intervalo en $m + 1$ espacios iguales, como se muestra a continuación,

$$y_{n+1} = y_n + hf(x_n, y_n), \quad h = \frac{b - a}{m} \quad (\text{A.2})$$

La exactitud de este método viene de la mano con el tamaño del paso h , por lo que para incrementar dicha exactitud deberá reducirse el tamaño del paso, que corresponde al tiempo de muestreo.

Otro método numérico de amplio uso en diversas disciplinas y que es implementado en este trabajo para encontrar la solución a los modelos matemáticos de la ecuación cinemática y del consenso dinámico, es el Runge-Kutta de orden cuatro (RK4), el cual es un método iterativo de

un solo paso, que consiste en evaluar cuatro veces la función a resolver en un punto determinado, en la siguiente ecuación se describe el algoritmo que sigue el método RK4,

$$\begin{aligned}
 k_1 &= hf(x_n, y_n) \\
 k_2 &= hf\left(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right) \\
 k_3 &= hf\left(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right) \\
 k_4 &= hf(x_n + h, y_n + k_3) \\
 y_{n+1} &= y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)
 \end{aligned} \tag{A.3}$$

h , al igual que en el método de Euler corresponde al paso de integración, y se alcanza una mejor exactitud al disminuir este paso.

A.1.1. Integración numérica del Observador no lineal

De la ecuación (3.5), se tiene que Ξ es una matriz de tamaño 4×3 que esta en función de q_n que es el cuaternión estimado en el tiempo de muestreo n ,

$$\Xi(q_n) := \begin{pmatrix} -\vec{q}_n^T \\ I_3 q_{0n} + [\vec{q}_n^\times] \end{pmatrix} \tag{A.4}$$

Aplicando el método RK4 para integrar $\dot{\hat{q}}$ y $\dot{\hat{v}}$ de la ecuación (3.5), se obtienen las ecuaciones (A.5) y (A.6).

$$\begin{aligned}
 k_1 &= \frac{1}{2}\Xi(q_n)(\omega_{G_n} - \vec{v}_n + K_1\vec{q}_{e_n})\Delta T \\
 k_2 &= \frac{1}{2}\Xi\left(q_n + \frac{1}{2}k_1\right)(\omega_{G_n} - \vec{v}_n + K_1\vec{q}_{e_n})\Delta T \\
 k_3 &= \frac{1}{2}\Xi\left(q_n + \frac{1}{2}k_2\right)(\omega_{G_n} - \vec{v}_n + K_1\vec{q}_{e_n})\Delta T \\
 k_4 &= \frac{1}{2}\Xi(q_n + k_3)(\omega_{G_n} - \vec{v}_n + K_1\vec{q}_{e_n})\Delta T \\
 q_{n+1} &= q_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)
 \end{aligned} \tag{A.5}$$

$$\begin{aligned}
 k_1 &= (-T^{-1}\vec{v}_n - K_2\vec{q}_{e_n})\Delta T \\
 k_2 &= \left[-T^{-1}\left(\vec{v}_n + \frac{1}{2}k_1\right) - K_2\vec{q}_{e_n}\right]\Delta T \\
 k_3 &= \left[-T^{-1}\left(\vec{v}_n + \frac{1}{2}k_2\right) - K_2\vec{q}_{e_n}\right]\Delta T \\
 k_4 &= [-T^{-1}(\vec{v}_n + k_3) - K_2\vec{q}_{e_n}]\Delta T \\
 \vec{v}_{n+1} &= \vec{v}_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)
 \end{aligned} \tag{A.6}$$

donde ΔT , corresponde al paso de integración.

Las Figuras A.1 y A.2, muestran la implementación de las ecuaciones (A.5) y (A.6) en Matlab, con $\Delta T = 0.02s$.

```
function quaternion = RK4_Q(bias, omega, Q_error, Q)

N = 100;
h = 0.02;
Q = Q/norm(Q);

for i=1:N
k1 = f(bias, omega, Q_error, Q)*h;
k2 = f(bias, omega, Q_error, Q + k1/2)*h;
k3 = f(bias, omega, Q_error, Q + k2/2)*h;
k4 = f(bias, omega, Q_error, Q + k3)*h;
Q = Q + (k1+2*k2+2*k3+k4)/6;
quaternion = Q;
quaternion = quaternion/norm(quaternion);
end

function dotQuaternion = f(bias, omega, Q_error, quat)

Q_hat = quat;
Q_hat = Q_hat/norm(Q_hat);
k = 1.25;

suma = omega - bias + k*Q_error;
suma = [suma(1);suma(2);suma(3)];
I = eye(3);

p1 = I*Q_hat(1);
skew_Qv = [0 -Q_hat(4) Q_hat(3); Q_hat(4) 0 -Q_hat(2); -Q_hat(3) Q_hat(2) 0];
sum = p1 + skew_Qv;
Qv_neg = [-Q_hat(2) -Q_hat(3) -Q_hat(4)];

y = 0.5*[Qv_neg; sum];

dotQuaternion = y*suma;
```

Figura A.1: RK4 correspondiente a la ecuación (A.5) implementado en Matlab.

```
function bias_hat = RK4_bias(bias, qe)
N = 100;
h = 0.02;

for i=1:N
k1 = f(qe, bias)*h;
k2 = f(qe,bias + k1/2)*h;
k3 = f(qe, bias + k2/2)*h;
k4 = f(qe, bias + k3)*h;
bias = bias + (k1+2*k2+2*k3+k4)/6;
bias_hat = bias;
end

function bias_dot = f(qe, bias)

K = 0.07;
T = [1/100 0 0;0 1/100 0;0 0 1/100];

Tv = T*bias;
Kqe = K*qe;

bias_dot = -Tv -Kqe;
```

Figura A.2: RK4 correspondiente a la ecuación (A.6) implementado en Matlab.

Apéndice B

Conversiones entre sistemas de referencias

Este apéndice provee un conjunto de algoritmos que permiten pasar de una representación de la rotación a otra deseada.

La conversión de matriz de rotación a eje-ángulo se lleva a cabo como se indica en el algoritmo 3. La representación eje-ángulo es muy útil debido a que vr builder 3D animation en Simulink hace uso de esta representación para las rotaciones.

La parametrización de rotaciones a través de cuaterniones, es ampliamente usada en varios campos de aplicaciones tales como: robótica, computación gráfica, navegación, etc. El cuaternión puede ser obtenido a partir de la matriz de rotación haciendo uso del algoritmo 4.

Algorithm 3 rotation matrix to axis-angle

Input: $R = R_b^f$

Output: β, \vec{e}

$R := \frac{R}{\|R\|};$

$aux := (trace(R) - 1)/2;$

$\beta := -acos(aux);$

$vec := [R_{32} - R_{23}; R_{13} - R_{31}; R_{21} - R_{12}];$

if $\beta \neq 0$ **then**

$\vec{e} := (1/2 * sin(\beta) * vec);$

$\vec{e} := \frac{\vec{e}}{\|\vec{e}\|};$

else

$\vec{e} := [1; 0; 0];$

end if

Debido a que los ángulos de Euler ofrecen una forma más intuitiva de representar rotaciones, estos pueden ser obtenidos del cuaternión. En [25], los ángulos de Euler vienen dados por el algoritmo 5.

Donde R_q es la matriz de rotación del cuaternión, para más detalles ver [25] capítulo 2. Sin embargo es bien sabido que los ángulos de Euler sufren de singularidad cuando $\theta = \pm 90^\circ$, por lo que esto deberá de ser tomado en cuenta al momento de realizar rotaciones en las cercanías de $\theta = \pm 90^\circ$, debido a que se pueden presentar errores.

En el algoritmo 6 se presenta la conversión de la representación eje-ángulo a partir del cuater-

Algorithm 4 rotation matrix to Quaternion**Input:** R **Output:** Q

$$R := \frac{R}{\|R\|};$$

$$s0 := R_{11} + R_{22} + R_{33};$$

$$s1 := R_{11} - R_{22} - R_{33};$$

$$s2 := -R_{11} + R_{22} - R_{33};$$

$$s3 := -R_{11} - R_{22} + R_{33};$$
if $s0 > -0.1$ **then**

$$q0 := (\text{sqrt}(1 + s0))/2;$$

$$q1 := (R_{32} - R_{23})/4/q0;$$

$$q2 := (R_{13} - R_{31})/4/q0;$$

$$q3 := (R_{21} - R_{12})/4/q0;$$
else if $s1 > -0.1$ **then**

$$q1 := (\text{sqrt}(1 + s1))/2;$$

$$q0 := (R_{32} - R_{23})/4/q1;$$

$$q2 := (R_{21} + R_{12})/4/q1;$$

$$q3 := (R_{31} + R_{13})/4/q1;$$
else if $s2 > -0.1$ **then**

$$q2 := (\text{sqrt}(1 + s2))/2;$$

$$q0 := (R_{13} - R_{31})/4/q2;$$

$$q1 := (R_{21} + R_{12})/4/q2;$$

$$q3 := (R_{23} + R_{32})/4/q2;$$
else

$$q3 := (\text{sqrt}(1 + s3))/2;$$

$$q0 := (R_{21} - R_{12})/4/q3;$$

$$q1 := (R_{31} + R_{13})/4/q3;$$

$$q3 := (R_{23} + R_{32})/4/q3;$$
end if $Q := [q0 ; q1 ; q2 ; q3];$ **Algorithm 5** quaternion to Euler angles**Input:** R_q **Output:** ϕ, θ, ψ $\phi := \text{atan2}(R_{q32}, R_{q33})$ $\theta := -\sin^{-1}(R_{q31}) = -\tan^{-1}\left(\frac{R_{q31}}{\sqrt{1-R_{q31}^2}}\right), \theta \neq \pm 90^\circ$ $\psi := \text{atan2}(R_{q21}, R_{q11})$

ni3n dado, lo cu3al es importante cuando se desarrollan animaciones 3D usando vr-builder. Donde β es el 3ngulo de rotaci3n y \vec{e} es el eje de rotaci3n.

En el algoritmo 7 se presenta la conversi3n para obtener la representaci3n en n3mero de coma flotante a partir de sus cuatro bytes dados. La transmisi3n de datos en formato de bytes hexadecimales es importante cuando un conjunto de datos ser3 transmitido, adem3s de que es m3s f3cil para el microcontrolador transmitir datos de esta forma.

Algorithm 6 quaternion to axis-angle

Input: Q **Output:** β, \vec{e}

$$Q \triangleq [q0 ; q1 ; q2 ; q3]$$

$$\beta := 2 * \text{acos}(q0);$$

$$e_1 := \frac{q1}{\sin(\beta/2)};$$

$$e_2 := \frac{q2}{\sin(\beta/2)};$$

$$e_3 := \frac{q3}{\sin(\beta/2)};$$

$$\vec{e} := [e_1 ; e_2 ; e_3];$$

Es posible obtener la matriz de rotación a partir de su cuaternión asociado, como se muestra a continuación [25]:

$$\mathbf{R}_b^f(\mathbf{Q}_b) = \begin{pmatrix} 1 - 2(q_2^2 + q_3^2) & 2(q_1q_2 - q_3q_0) & 2(q_1q_3 + q_2q_0) \\ 2(q_1q_2 + q_3q_0) & 1 - 2(q_1^2 + q_3^2) & 2(q_2q_3 - q_1q_0) \\ 2(q_1q_3 - q_2q_0) & 2(q_2q_3 + q_1q_0) & 1 - 2(q_1^2 + q_2^2) \end{pmatrix} \quad (\text{B.1})$$

Algorithm 7 Conversion hexadecimal to float

Input: $\text{byte1}, \text{byte2}, \text{byte3}, \text{byte4}$ **Output:** value

$$\text{num} := \text{zeros}(1, 4, \text{'uint8'});$$

$$\text{num} := \text{uint8}([\text{byte4}, \text{byte3}, \text{byte2}, \text{byte1}]);$$

$$\text{num} := \text{fliplr}(\text{num});$$

$$\text{value} := \text{zeros}(1, 1, \text{'single'});$$

$$\text{value} := \text{typecast}(\text{num}, \text{'single'}); \%float$$

Apéndice C

Artículos publicados

C.1. Artículo 1

Como parte de los resultados de las investigaciones realizadas durante el trabajo de tesis, se publicaron dos artículos, uno de ellos publicado en la *International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE 2021)*, con sede en Cuernavaca, Morelos. El segundo artículo fue publicado en la Conferencia Nacional de Control Automático, organizado por la Asociación de México de Control Automático A.C. (AMCA 2022), con sede en Tuxtla Gutiérrez, Chiapas.



The screenshot shows the IEEE Xplore digital library interface. At the top, there are navigation links for IEEE.org, IEEE Xplore, IEEE SA, IEEE Spectrum, and More Sites. The IEEE Xplore logo is prominent on the left. A search bar is located in the center, with a dropdown menu set to 'All' and a search button on the right. Below the search bar, the breadcrumb trail reads 'Conferences > 2021 International Conference...'. The main title of the article is 'Evaluation of Deterministic Attitude Estimation Algorithms in Real-Time Using Low-Cost Hardware'. Below the title, it indicates the publisher is IEEE and provides buttons for 'Cite This' and 'PDF'. The authors listed are Osberto A. Betanzos-Ramirez, Alejandro Sanchez-Mendoza, J. Fermi Guerrero-Castellanos, and Lizeth Torres. On the left side, there is a 'Document Sections' table of contents with the following items: 1. Introduction, 2. Mathematical Background, 3. Statement Problem, 4. Deterministic Attitude Estimation, and 5. Experimental Set-Up. The abstract text is visible on the right, describing the implementation of three deterministic algorithms for attitude estimation. At the bottom, there is a 'Published in' section listing the conference: '2021 International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE)'. Other metadata includes the date of the conference (22-26 November 2021), the date added to IEEE Xplore (26 September 2022), the DOI (10.1109/ICMEAE55138.2021.00041), the publisher (IEEE), and the conference location (Cuernavaca, Mexico).

Figura C.1: Artículo enviado a congreso internacional ICMEAE.

Evaluation of Deterministic Attitude Estimation Algorithms in Real-Time Using Low-Cost Hardware

Osberto A. Betanzos-Ramirez, Alejandro Sanchez-Mendoza, J. Fermi Guerrero-Castellanos
Benemérita Universidad Autónoma de Puebla (BUAP)
Faculty of Electronics,
Ciudad Universitaria, Puebla, 72570, Mexico
(osberto.betanzos; alejandro.sanchezme)@alumno.buap.mx, fermi.guerrero@correo.buap.mx

Lizeth Torres
Universidad Autónoma de México (UNAM), Instituto de Ingeniería, CdMx, 04510, Mexico
ltorreso@iingen.unam.mx

Abstract

This paper describes the implementation of three deterministic algorithms for the attitude estimation of a rigid body in a pragmatic fashion. For that, a tri-axis accelerometer and a tri-axis magnetometer, and a low-cost micro-controller are used. The algorithms implementation is carried out using MATLAB/Simulink. The proposed methodology allows a fast evaluation of any deterministic algorithm. Then, it aids the subsequent development of embedded implementations. Experimental results validate the proposed methodology.

1. Introduction

Obtaining attitude information requires precise instrumentation and algorithms to ensure its reliability. Rigid body attitude measurement has become an opportunity area that brought huge interest during the past few decades, and it has been widely studied. High-Cost applications in the aerospace environment such as spacecrafts or satellites matches with the rigid body framework and demands accurate acquisition of attitude data because of the nature of its application [4].

The development of applications in which micro-electro-mechanical systems (MEMS) are used has increased in the last decade. Therefore, attitude estimation became important on new areas including terrestrial and aerial robotics [9][5] and virtual reality [6]. In 2020 Amazon gained the approval from the USA Federal Aviation Administration (FAA) to deploy delivery services using unmanned aerial vehicles, typically called drones, other companies such as

UPS and Wing got this permission as well [2] this is a representative case of the growth of low-cost applications in which the attitude estimation got a major role for systems operation, precedents that motivates the study of attitude measurement.

Since the attitude cannot be directly measured, it is typically estimated through inertial and magnetic sensors, particularly, three-axis gyro, three-axis accelerometer and three-axis magnetometers orthogonally placed inside the system somehow the sensors frame axes match with the rigid body frame axes. In the main, these sensors can be classified by the nature of the information they offer, angular velocity sensors measure the angular velocity of the body using as reference some other inertial frame, at the same time, the coordinates of a fixed vector in the mobile frame are obtained with reference vector sensors [11].

Deterministic algorithms can be used as solution for attitude estimation problem. TRIAD, SVD and LQS-req techniques are mainly used examples, these algorithms are already studied and implemented in different scientific works and areas, as shown in [3],[1]. The main goal of the present work is to offer an option for and easier understanding of deterministic algorithms for attitude estimation and explain as simple as possible the rigid body orientation problem using quaternions. This paper includes solved examples for each deterministic algorithm presented, in the way to help in the development of other attitude estimation systems.

This paper is organized as follows. In section 2, a mathematical background for attitude parametrization is provided. Section 3 presents the statement problem. Deterministic Attitude Estimation is presented in section 4. The experimental Set-Up is presented in section 5. Section 6 the experimental attitude estimation is introduced. The experimental

results is presented in section 7. Finally section 8 presents the conclusions and next the acknowledgments.

2. Mathematical Background

2.1. Unit Quaternions

Consider two orthogonal, right-handed coordinate frames: the body coordinate frame, $\mathbf{E}^b = [\vec{e}_1^b, \vec{e}_2^b, \vec{e}_3^b]$, located at the center of mass of the rigid body, and the inertial coordinate frame, $\mathbf{E}^f = [\vec{e}_1^f, \vec{e}_2^f, \vec{e}_3^f]$, located at some fixed point in space. The rotation of the body frame, \mathbf{E}^b , with respect to the fixed frame, \mathbf{E}^f , is represented by the attitude matrix $R \in SO(3) = \{R \in \mathbb{R}^{3 \times 3} : R^T R = I, \det R = 1\}$. The cross product between two vectors, $\vec{\chi}, \vec{\xi} \in \mathbb{R}^3$, is represented by a matrix multiplication $[\vec{\xi}^\times] \vec{\chi} = \vec{\xi} \times \vec{\chi}$, where:

$$[\vec{\xi}^\times] = \begin{pmatrix} 0 & -\xi_3 & \xi_2 \\ \xi_3 & 0 & -\xi_1 \\ -\xi_2 & \xi_1 & 0 \end{pmatrix}$$

The n -dimensional unit sphere embedded in \mathbb{R}^{n+1} is denoted as $\mathbb{S}^n = \{x \in \mathbb{R}^{n+1} : x^T x = 1\}$. Hence, a unit quaternion, $q \in \mathbb{S}^3$, is defined as:

$$q := \begin{pmatrix} \cos \frac{\beta}{2} \\ \vec{e} \sin \frac{\beta}{2} \end{pmatrix} = \begin{pmatrix} q_0 \\ \vec{q} \end{pmatrix} \in \mathbb{S}^3 \quad (1)$$

$\vec{q} = [q_1 \ q_2 \ q_3]^T \in \mathbb{R}^3$ and $q_0 \in \mathbb{R}$ are known as the vector and scalar parts of the quaternion, respectively. q represents an element of $SO(3)$ through the map, $\mathcal{R} : \mathbb{S}^3 \rightarrow SO(3)$, defined as:

$$\mathcal{R} := (q_0^2 - \vec{q}^T \vec{q})I - 2q_0[\vec{q}^\times] + 2\vec{q}\vec{q}^T \quad (2)$$

Note that $R = \mathcal{R}(q) = \mathcal{R}(-q)$ for each $q \in \mathbb{S}^3$, i.e., quaternions q and $-q$ represent the same physical attitude. The inverse unit quaternion is given by $q^{-1} := [q_0 \ -\vec{q}^T]^T$, and the quaternion product is defined by:

$$q_1 \otimes q_2 := \begin{pmatrix} q_{10} & -\vec{q}_1^T \\ \vec{q}_1 & I_3 q_{10} + [\vec{q}_1^\times] \end{pmatrix} \begin{pmatrix} q_{20} \\ \vec{q}_2 \end{pmatrix} \quad (3)$$

The attitude error is used to quantify the mismatch between two attitudes. If q defines the true attitude quaternion and \hat{q} is the estimated quaternion, then the error quaternion that represents the attitude error between the true orientation and the estimated one is given by:

$$q_e := \hat{q}^{-1} \otimes q = (q_{e0} \ \vec{q}_e^T)^T \quad (4)$$

In the case that the true quaternion and the estimated one coincide, the quaternion error becomes:

$$q_e = (\pm 1 \ 0^T)^T \quad (5)$$

2.2. Attitude measurement models

Let us consider the representation of a unit vector \vec{x}_i with respect to \mathbf{E}^f and \mathbf{E}^b , and denoted by \vec{r}_i and \vec{b}_i respectively. The vectors \vec{r}_i are also called the ‘‘reference vectors’’, and in general, are known quite accurately. The body vectors \vec{b}_i are

known as ‘‘vector observations’’ and are obtained from on-board sensors (for instance, magnetometers, star trackers, accelerometers). These two representations are linked up through the rotation matrix $R \in SO(3)$:

$$\vec{b}_i = R \vec{r}_i \quad (6)$$

Furthermore, vectors \vec{b}_i and \vec{r}_i can be written as a quaternion as follows:

$$b_{iq} = (0 \ \vec{b}_i^T)^T \quad r_{iq} = (0 \ \vec{r}_i^T)^T \quad (7)$$

then, these quaternions are related by the quaternion rotation q such that

$$b_{iq} = q^{-1} \otimes r_{iq} \otimes q \quad (8)$$

from (8) is easy to proof that $q \otimes b_{iq} - r_{iq} \otimes q = 0$, then using the definition of quaternion product (3), one obtains the following relation

$$\begin{pmatrix} 0 & -(\vec{b}_i - \vec{r}_i)^T \\ \vec{b}_i - \vec{r}_i & -[(\vec{b}_i + \vec{r}_i)^\times] \end{pmatrix} q = Hq = 0 \quad (9)$$

with $H \in \mathbb{R}^{4 \times 4}$, hence (9) is a well-structured linear system of equations.

3. Statement Problem

According to (6) and (9), the goal of the deterministic attitude estimation process is to estimate the rotation matrix R or quaternion q , denoted in the rest of the paper as \hat{R} and \hat{q} , respectively, which maps the vector observations into the known reference vectors. That is, given the vector observations \vec{b}_i for $i \in \{1, 2, \dots, m\}$ to find \hat{R} or \hat{q} such that

$$\vec{b}_i = \hat{R} \vec{r}_i \quad \text{or} \quad H \hat{q} = 0 \quad (10)$$

Remark 1 *Although the observation vector is a three dimensional vector, the unit-length constraint implies that only two independent attitude degrees of freedom can be observed with this measurement. As a consequence, the minimum number of vector observation must be two ($m \geq 2$), and these vectors must be non-collinear.*

Remark 2 *In this paper, the vector observations are obtained from accelerometers and magnetometers. These measures are normalized since one is interested in the direction and not in the magnitude. The reference vectors are the normalized gravity vector and the normalized magnetic field vector, both expressed in the NED frame.*

In the following sections, one briefly presents the well-known TRIAD algorithm as well as two deterministic algorithms developed by our group in the last years to show the proposed methodology to implement and assess attitude algorithms easily and quickly using low-cost hardware and the widely known MATLAB environment.

4. Deterministic Attitude Estimation

4.1. TRIAD

The TRIAD algorithm provides a simple solution for the attitude estimation problem [10]. It uses a pair of reference vectors \vec{r}_i and its corresponding vector observations \vec{b}_i with $i = 1, 2$. The goal is to construct two triad frames from these vectors, the first expressed in the body frame and the second in the inertial frame. Then the rotation matrix \hat{R} is estimated using the relation between these vectors. Without loss of generality, let us assume that \vec{b}_1 is the more accurate vector observation, then the \mathcal{T} -frame axes expressed in the body frame are constructed as follows

$$\vec{t}_{1_b} = \vec{b}_1, \vec{t}_{2_b} = \frac{\vec{b}_1 \times \vec{b}_2}{\|\vec{b}_1 \times \vec{b}_2\|}, \vec{t}_{3_b} = \vec{t}_{1_b} \times \vec{t}_{2_b} \quad (11)$$

Similarly, given the known reference vectors components of these two vectors observations, the \mathcal{T} -frame axes are expressed in the inertial frame \mathbf{E}^f using

$$\vec{t}_{1_f} = \vec{r}_1, \vec{t}_{2_f} = \frac{\vec{r}_1 \times \vec{r}_2}{\|\vec{r}_1 \times \vec{r}_2\|}, \vec{t}_{3_f} = \vec{t}_{1_f} \times \vec{t}_{2_f} \quad (12)$$

The \mathcal{T} -frame axes expressed in the inertial frame \mathbf{E}^f and the \mathcal{T} -frame axes defined in the inertial frame \mathbf{E}^f are linked up through the rotation matrix \hat{R} as follows

$$[\vec{t}_{1_b} \ \vec{t}_{2_b} \ \vec{t}_{3_b}] = \hat{R} [\vec{t}_{1_f} \ \vec{t}_{2_f} \ \vec{t}_{3_f}] \quad (13)$$

thus the estimated rotation matrix can be assembled using

$$\hat{R} = [\vec{t}_{1_b} \ \vec{t}_{2_b} \ \vec{t}_{3_b}] [\vec{t}_{1_f} \ \vec{t}_{2_f} \ \vec{t}_{3_f}]^T \quad (14)$$

Remark 3 In practical applications, one seeks to construct the \mathcal{T} -frame defined in the inertial frame \mathbf{E}^f such that its axes correspond with the axes of \mathbf{E}^f .

The estimated rotation matrix can be readily converted to another attitude representation, for instance, Euler angles, quaternion, or axis-angle representation (see Appendix). Although the TRIAD method is simple to implement and evaluate, it does suffer of the issue that can only be used with two vector observations. Thus, even if more than two sensors are available these extra attitude sensing capability cannot be exploited with this algorithm. The two methods discussed in the following subsections attempt to address these problems using the best least-squares fit across two or more vector observations.

4.2. SVD

If at least two vector observations are non-collinear, the attitude quaternion can be estimated using these simultaneous vector measurements. For that, the measurement equation (9) is exploited. The observation matrix in function of

m pair vectors (\vec{b}_i, \vec{r}_i) with $i \in \{1, 2, \dots, m\}$ is written as follows

$$\bar{H} = \begin{pmatrix} 0 & -(\vec{b}_1 - \vec{r}_1)^T \\ \vec{b}_1 - \vec{r}_1 & -[(\vec{b}_1 + \vec{r}_1) \times] \\ \vdots & \vdots \\ 0 & -(\vec{b}_m - \vec{r}_m)^T \\ \vec{b}_m - \vec{r}_m & -[(\vec{b}_m + \vec{r}_m) \times] \end{pmatrix} \in \mathbb{R}^{4m \times 4} \quad (15)$$

In practice, the measurements are perturbed by noise, vibrations, etc.; the problem is estimating the attitude quaternion from a perturbed observation matrix \bar{H} , that is:

$$\hat{q} = \arg \min_q \left\{ \frac{1}{2} \|\bar{H}q\|^2 \right\} \quad (16)$$

subject to $\|\hat{q}\| = 1$

Then one has the following result from [4].

Lemma 1 Let $\bar{H} \in \mathbb{R}^{4m \times 4}$ the observation matrix formed by the m pair of vectors (\vec{b}_i, \vec{r}_i) with $i \in \{1, 2, \dots, m\}$. Then, there is an orthonormal matrix $U = (U_1 \ U_2 \ \dots \ U_{4m}) \in \mathbb{R}^{4m \times 4m}$, an orthonormal matrix $V = (V_1 \ V_2 \ V_3 \ V_4) \in \mathbb{R}^{4 \times 4}$ and an orthonormal matrix $S \in \mathbb{R}^{4m \times 4}$ with elements σ_k along the diagonal and zeros everywhere else, where $k = 1 : 4$, such that

$$\bar{H} = USV^T \quad (17)$$

Then, the attitude quaternion is given by the last column of the matrix V , that is

$$\hat{q} = V_4 \quad (18)$$

furthermore, the normality condition $\|\hat{q}\| = 1$ is satisfied.

4.3. LQS-req

The SVD method is a very robust approach to estimate the attitude quaternion. However, its numerical implementation is challenging. In [8], a recursive method was developed to find the attitude quaternion estimate to overcome this issue. The main advantage of the so-called LQS-req method is the speed at which the numerical computation can be carried out. To explain the method briefly, let us consider the measurement model (9) and the observation matrix \bar{H} given by (15). Then, the measurement model can be written as follows

$$\bar{H}(t)\hat{q} = \begin{pmatrix} \varphi_1^T(t) \\ \vdots \\ \varphi_n^T(t) \end{pmatrix} \hat{q} = 0 \quad (19)$$

The unknown quaternion \hat{q} , which be called q_{ps} can be estimated by minimizing the following cost function [8]:

$$J(q) = \frac{1}{2} \sum_{i=1}^{n-4} (\varphi_i^T(t)q)^T (\varphi_i^T(t)q) = \frac{1}{2} \|\bar{H}(t)q\|^2 \quad (20)$$

By applying the Kaczmarz's method, one can estimate the attitude quaternion by using vector observations [8]

$$q_{ps}^i = \left[\prod_{i=1}^{n-4} \left(I_4 - \frac{\gamma \varphi_i(t) \varphi_i^T(t)}{\alpha + \varphi_i^T(t) \varphi_i(t)} \right) \right] q_{ps}^{i-1} \quad (21)$$

where $\alpha \geq 0$ and $0 < \gamma < 2$.

5. Experimental Set-Up

In this section, one evaluates the three different deterministic algorithms. The instrumentation and the description for each algorithm is presented in a general way. The instrumentation is composed of one microcontroller based on the STM32F411 Nucleo-64 development board. Their 3x I²C and 3x USARTs peripherals allow communication with sensors and the PC. The sensor suite is based on a sensor board equipped with a three-axis accelerometer, a three-axis gyro, and a three-axis magnetometer; all sensors are part of the Gy-87 module. All sensors use the I²C peripheral communication. Figure 1 shows the development board Nucleo-64, Gy-87 sensor and the reference system for Gy-87 module also the inertial reference system (E^f) is defined. In this work, the sensors' measurements are processed using MATLAB/Simulink, which offers an easy form to evaluate the different deterministic algorithms, namely, TRIAD, SVD, and LQS-req. Next, the development and implementation of deterministic algorithms are described.

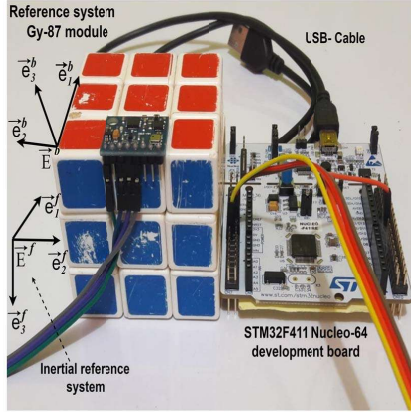


Figure 1: Inertial reference system E^f and body reference system E^b

The block diagram of the general system for the three algorithms is depicted in Figure 2. The first step consists of the sensor's calibration; this is very important because the low-cost sensors, for this case, accelerometer and magnetometer, are characterized by scale factor problem, a term of bias and sensor self-noise that affect the measurement of each sensor. The calibration aims to reduce or remove the bias and adjust the scale factor for each sensor. The calibration is performed in the microcontroller and the procedure followed for these calibrations is based on the thesis work developed in [7].

Now the sensor data are sent by serial communication from the microcontroller to MATLAB/Simulink. The data buffer consists of 1 byte for the header, 24 bytes for data measurements, and 1 byte for the end. This is illustrated in Figure 3. Next, in MATLAB/Simulink the data is received and processed. Once the measurements are in Simulink, they can be used by the deterministic algorithms, as explained in the following subsections. In order to provide the necessary tools for the implementation and testing of the algorithms, it is recommended to review the file available at <https://drive.google.com/file/d/1zkotdG6Xpy2tHcJ5jBX6y8G0ztpNevvQ/view?usp=sharing>, where a series of algorithms necessary in this work are provided.

6. Experimental attitude estimation

Since one is interested in deterministic attitude estimation algorithms, only the measurements issued of the three-axis accelerometer and the three-axis magnetometer are employed, which correspond to the vector observations, denoted here as \vec{b}_A and \vec{b}_M , respectively. Furthermore, the inertial coordinate frame E^f is the NED (North, East, Down) coordinate frame. The vector observations and the reference vectors are defined as follows:

- The vector observation \vec{b}_A corresponds to the accelerometer measurement, which measures the direction of the gravitational acceleration vector in the body frame. Then, the reference vector $\vec{r}_A = [0 \ 0 \ 1]^T$ corresponds to the normalized gravity vector expressed in the E^f frame.
- The vector observation \vec{b}_M corresponds to the magnetometer measurements. The reference vector $\vec{r}_M = [0.28 \ 0 \ 0.1950]^T$ corresponds to the normalized magnetic field vector expressed in the E^f frame.

Note that the gravity and magnetic magnitude, and direction in the NED frame depend on the longitude and latitude.

6.1. TRIAD Algorithm

For the TRIAD method, the \mathcal{T} -frame expressed in the body frame is created by means of the vector observations \vec{b}_A and \vec{r}_M and it is assembled such that its axes correspond to the axes of the frame E^b . Then, one has

$$\begin{aligned} \vec{t}_{1_b} &= \frac{\vec{b}_M - (\vec{b}_M^T \vec{b}_A) \vec{b}_A}{\|\vec{b}_M - (\vec{b}_M^T \vec{b}_A) \vec{b}_A\|} \\ \vec{t}_{2_b} &= \vec{b}_A \times \vec{t}_{1_b} \\ \vec{t}_{3_b} &= \vec{b}_A \end{aligned} \quad (22)$$

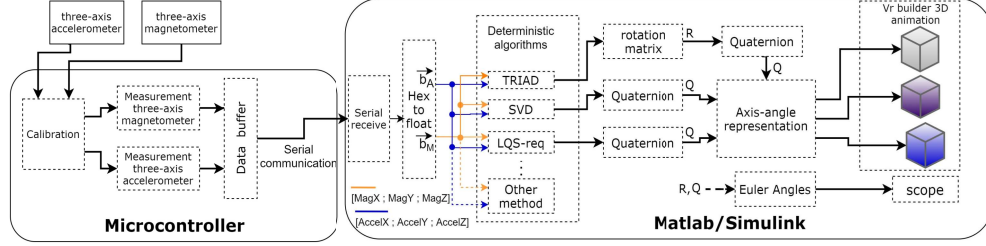


Figure 2: Block diagram of the general evaluation of deterministic algorithms

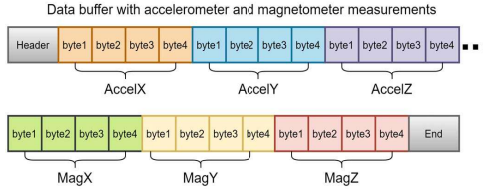


Figure 3: General structure of data buffer

Algorithm 1 SVD algorithm

Input: $\vec{b}_A, \vec{r}_A, \vec{b}_M, \vec{r}_M$

Output: q_{ps}

$$\vec{b}_A := \frac{\vec{b}_A}{\|\vec{b}_A\|} \text{ and } \vec{b}_M := \frac{\vec{b}_M}{\|\vec{b}_M\|}$$

Form the matrix \bar{H} applying equation (25)

$$[U, S, V] = svd(\bar{H})$$

q_{ps} is given by the last column in matrix V

$$q_{ps} := \frac{q_{ps}}{\|q_{ps}\|}$$

The \mathcal{T} -frame described in the inertial frame is such that its axes correspond to the axes of \mathbf{E}^f , thus

$$\vec{t}_{1_f} = (1 \ 0 \ 0)^T, \vec{t}_{2_f} = (0 \ 1 \ 0)^T, \vec{t}_{3_f} = (0 \ 0 \ 1)^T \quad (23)$$

clearly the estimated rotation matrix is

$$\hat{R} = [\vec{t}_{1_b} \ \vec{t}_{2_b} \ \vec{t}_{3_b}] \underbrace{[\vec{t}_{1_f} \ \vec{t}_{2_f} \ \vec{t}_{3_f}]^T}_{I \in \mathbb{R}^{3 \times 3}} \quad (24)$$

6.2. SVD Algorithm

From equation (15), it is possible to construct the measurement matrix $\bar{H} \in \mathbb{R}^{8 \times 4}$, and this becomes.

$$\bar{H} = \begin{pmatrix} 0 & -(\vec{b}_A - \vec{r}_A)^T \\ \vec{b}_A - \vec{r}_A & -[(\vec{b}_A + \vec{r}_A)^\times] \\ 0 & -(\vec{b}_M - \vec{r}_M)^T \\ \vec{b}_M - \vec{r}_M & -[(\vec{b}_M + \vec{r}_M)^\times] \end{pmatrix} \quad (25)$$

where \vec{b}_A, \vec{b}_M are the observation vectors given by the measurements of the accelerometer and magnetometer, respectively. Remember that \vec{r}_A and \vec{r}_M are the reference vectors, in this case gravitational vector and magnetic vector pointing north. The pseudo-code that summarizes this deterministic algorithm is given below,

6.3. LQS-req Algorithm

Consider again the vector observations and reference vectors obtained of the on-board sensors. The measurement

model is constructed by means of the matrix $\bar{H} \in \mathbb{R}^{8 \times 4}$ as follows

$$\begin{pmatrix} 0 & -(\vec{b}_A - \vec{r}_A)^T \\ \vec{b}_A - \vec{r}_A & -[(\vec{b}_A + \vec{r}_A)^\times] \\ 0 & -(\vec{b}_M - \vec{r}_M)^T \\ \vec{b}_M - \vec{r}_M & -[(\vec{b}_M + \vec{r}_M)^\times] \end{pmatrix} \hat{q} = \bar{H} \hat{q} = \begin{pmatrix} \varphi_1^T(t) \\ \vdots \\ \varphi_8^T(t) \end{pmatrix} \hat{q} = 0 \quad (26)$$

The quaternion can be estimated using measurement equation (26) and Kaczmarz's method, which consist in an iterative method for the solution of the overdetermined linear system. Then, one has

$$q_{ps}^i := \left[\prod_{i=1}^8 \left(I_4 - \frac{\gamma \varphi_i(t) \varphi_i^T(t)}{\alpha + \varphi_i^T(t) \varphi_i(t)} \right) \right] q_{ps}^{i-1} \quad (27)$$

with $\alpha = 0.8$ and $\gamma = 0.8$ which are heuristically selected. α tunes the convergence velocity. The pseudo-code that summarizes this deterministic algorithm is given below.

7. Experimental results

To evaluate the three deterministic algorithms, one uses a small Rubik's cube, where the suite of sensors is attached (see Figure 1). The Euler Angles representation depicts the results since it is more intuitive than quaternion or rotation matrix. However, TRIAD method use the matrix representation while SVD and LSQ-req use the quaternion parametrization. The Figure 4 shows the results, where the attitude estimation via the TRIAD, SVD, and LQS-req algorithm is depicted, respectively. Firstly, the body frame

Algorithm 2 LQS-req algorithm

Input: $\vec{b}_A, \vec{r}_A, \vec{b}_M, \vec{r}_M, q_{est}$
Output: q_{est}
 $\vec{b}_A := \frac{\vec{b}_A}{\|\vec{b}_A\|}$ and $\vec{b}_M := \frac{\vec{b}_M}{\|\vec{b}_M\|}$
 Form the matrix \bar{H} applying equation (26)
 $q_{ps} = \text{zeros}(4, 9)$; % is a matrix of 4×9
 $q_{ps}(:, 1) := q_{est}$;
 $\alpha := 0.8, \gamma := 0.8$;
 $I_4 := \text{eye}(4)$;
for $i = 2:9$ **do**
 $q_{ps}(:, i) := \left(I_4 - \frac{\gamma * \bar{H}(i-1,:) * \bar{H}(i-1,:)}{\alpha + \bar{H}(i-1,:) * \bar{H}(i-1,:)} \right) * q_{ps}(:, i-1)$;
end for
 $q_{est} := q_{ps}(:, 9)$;
 $q_{est} := \frac{q_{est}}{\|q_{est}\|}$;

E^b is coincident with the inertial frame, thus, the estimated angles are $\phi = 0 \text{ rad}$, $\theta = 0 \text{ rad}$ and $\psi = 0 \text{ rad}$. Next the body frame is rotated an angle of $-\frac{\pi}{2} \text{ rad}$ in ψ , followed by a rotation of $-\frac{\pi}{2} \text{ rad}$ in ϕ , next a rotation of $+\frac{\pi}{2} \text{ rad}$ in ψ and finally, a rotation of $+\frac{\pi}{2} \text{ rad}$ in ϕ comes back the body frame to the initial attitude. In general, the three algorithms respond similarly at the same time. These movements were performed carefully because the algorithms are sensitive to vibrations (remember that one uses the accelerometer). However, it is observed that the LQS-req algorithm presented less sensitivity to disturbance and vibrations by keeping a memory of past measurements.

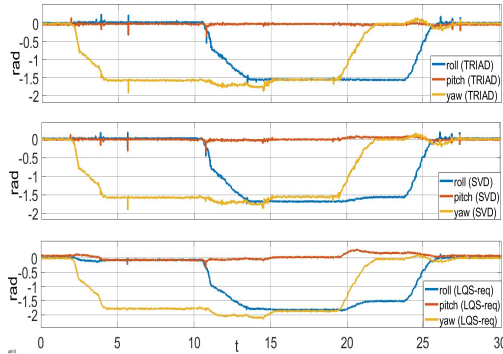


Figure 4: Euler angles obtained from well-known rotations applying deterministic algorithms

8. Conclusions

This paper presented a methodology including low-cost hardware and well-known software to evaluate three deterministic attitude estimation algorithms. Having a simple evaluating system for deterministic algorithms is essential

since this estimated attitude can be used as an initial input condition of other algorithms more complex like Non-Linear Observers and Extended Kalman Filter, which offers the best estimation of the attitude.

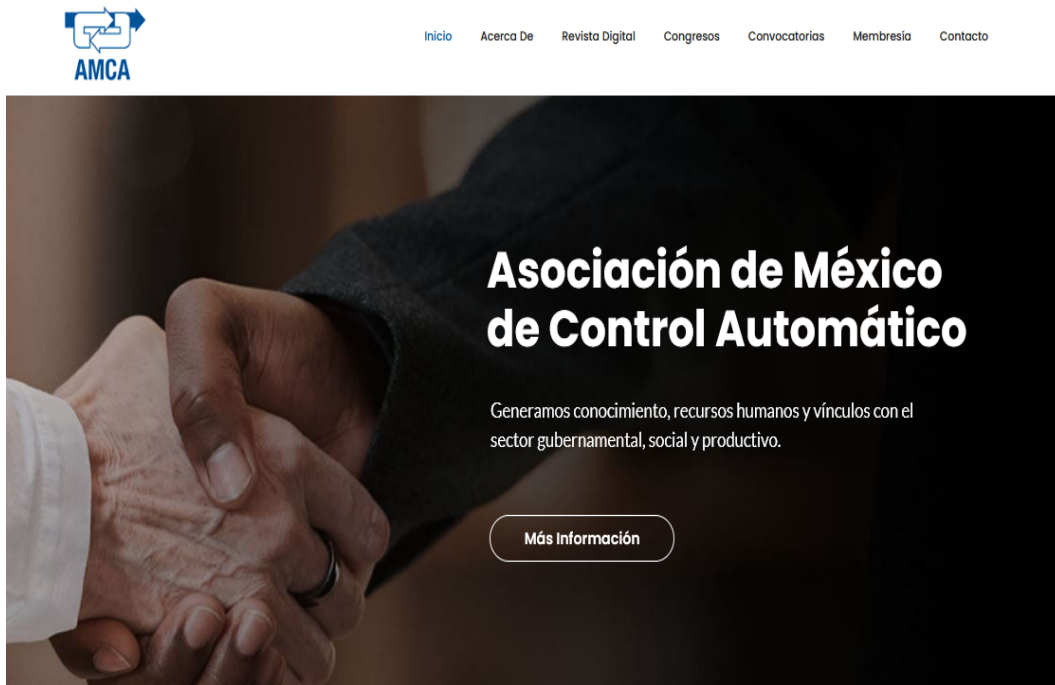
Acknowledgments

This contribution was partially financed by DGAPA-UNAM (PAPIIT TA100421), as well as by CONACYT (4730, Atención a Problemas Nacionales, convocatoria 2017) and VIEP-BUAP 100305333.

References

- [1] J. An and D. Wang. Efficient one-sided jacobi svd computation on amd gpu using opencl. In *2016 IEEE 13th International Conference on Signal Processing (ICSP)*, pages 491–495, 2016.
- [2] P. Annie. Amazon wins faa approval for prime air drone delivery fleet. Online, 2020. <https://www.cnbc.com/2020/08/31/amazon-prime-now-drone-delivery-fleet-gets-faa-approval.html>.
- [3] S.-W. Chang, Y.-T. Chiang, and F.-R. Chang. Slerp-based optimal triad algorithm. In *Proceedings of SICE Annual Conference 2010*, pages 331–335, 2010.
- [4] J. Guerrero-Castellanos, H. Madrigal-Sastre, S. Durand, L. Torres, and G. Munoz-Hernandez. A robust nonlinear observer for real-time attitude estimation using low-cost mems inertial sensors. *Sensors (Basel, Switzerland)*, 13:15138–58, 11 2013.
- [5] J. Guerrero-Castellanos, N. Marchand, A. Hably, S. Leseq, and J. Delamare. Bounded attitude control of rigid bodies: Real-time experimentation to a quadrotor mini-helicopter. *Control Engineering Practice*, 19, 08 2011.
- [6] J. D. Hol, T. B. Schon, F. Gustafsson, and P. J. Slycke. Sensor fusion for augmented reality. In *2006 9th International Conference on Information Fusion*, pages 1–6, 2006.
- [7] D. López-Montes. Fusión sensorial para la estimación del estado de un vehículo autónomo. Master's thesis, Escuela Técnica Superior de Ingenieros Industriales y de Telecomunicación, Universidad de Cantabria, 2016.
- [8] J. Ramos-G., J. Guerrero-Castellanos, V. Gonzales-Díaz, J. Linares-Flores, M. López-López, and S. Durand. Embedded implementation of a nonlinear-observer-based ahrs. *International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 363–371, 06 2018.
- [9] S. Roumeliotis, G. Sukhatme, and G. Bekey. Smoother based 3d attitude estimation for mobile robot localization. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, volume 3, pages 1979–1986 vol.3, 1999.
- [10] H. Schaub and J. L. Junkins. *Analytical Mechanics of Space Systems*. AIAA Education Series, Reston, VA, 2nd edition, October 2009.
- [11] J. R. Wertz. *Spacecraft Attitude Determination and Control*. Springer, Dordrecht, Torrance, CA, 1st edition, October 1978.

C.2. Artículo 2



Bienvenidos y Bienvenidas

La Asociación de México de Control Automático A.C. (AMCA), es una organización perteneciente a la Federación Internacional de Control Automático (IFAC, por sus siglas en inglés), que considera a las áreas de la Ingeniería y de la Ciencia relacionadas con el Control Automático en México.

Los objetivos de la Asociación están alineados con los de la Federación Internacional de Control Automático y son: promover la Ciencia y la Tecnología del Control en el más amplio sentido de la palabra, en todos los sistemas comprendidos en las áreas de la ingeniería, la física, la biología y las ciencias sociales y económicas, vistos como sistemas dinámicos, tanto desde un punto de vista teórico como práctico.

Figura C.2: Artículo enviado a congreso nacional AMCA.

Distributed Dynamic Consensus Algorithm for Attitude Estimation

Osberto A. Betanzos-Ramírez*
J. Fermi Guerrero-Castellanos* Sylvain Durand**
Lizeth Torres***

* *Faculty of Electronic, Autonomous University of Puebla (BUAP),
Ciudad universitaria, Puebla 72570, Mexico
(e-mail: fermi.guerrero@correo.buap.mx,
osberto.betanzos@alumno.buap.mx).*

** *ICube Laboratory, INSA Strasbourg, Strasbourg University, CNRS
UMR 7357, Strasbourg, France.*

*** *Engineering Institute, Autonomous University of Mexico (UNAM),
Circuito Escolar, Ciudad universitaria, Mexico D.F. 04510, Mexico.*

Abstract: This paper proposes a new dynamic consensus algorithm for the attitude estimation of a rigid body using n Measurement Units (MUs) spatially distributed on the body. The MUs are considered as agents in a sensor network, which is modeled using graph theory. It is assumed that the n th MU shares attitude information only with its neighbor agents. Then, a dynamic consensus algorithm is developed to obtain an average quaternion that describes the general attitude of the rigid body. The objective is to get a more accurate and reliable estimation of the attitude, further providing a fault-tolerant system in the event of failure sensors. This new dynamic consensus algorithm is validated in simulation using three MUs in the network. Results show a better precision and demonstrate that the proposed algorithm provides a good estimation of attitude even in the scenario of possible failure of any of the sensors.

Keywords: Dynamic consensus algorithm, sensor network, attitude estimation, fault tolerance.

1. INTRODUCTION

Attitude estimation has been of great interest in the past few decades (Guerrero-Castellanos et al., 2013), due to the wide range of applications in various fields of engineering such as: robotics, ships, spacecraft, biomechanics, etc. However, in some of these applications the accuracy and reliability are of great importance because of the nature of its application (Betanzos-Ramirez et al., 2021).

Since attitude estimation is an issue very important, this has been carried on with inertial sensors (three-axis gyro and three-axis accelerometer) and three axis magnetometers, these sensors form a sensor suite typically called IMU (Inertial Measurement Unit) or MIMU (Magnetic and Inertial Measurement Unit). In this paper, we use MU (Measurement Unit) to refer accelerometer and magnetometer like a sensor suit. Normally deterministic attitude estimators like TRIAD, SVD, ect. are carry out with vectors observations typically obtained from the MU and this is placed so that its reference system is matched with the reference system of the rigid body, therefore the attitude of the rigid body will be given directly by the orientation measured by the MU. In order to improve the accuracy for attitude estimation some algorithms

that fuse or averaging the quaternion attitude has been propose such as in (Markley et al., 2007) the authors propose an algorithm based on the minimization of a weighted sum of the Frobenius norm, in (Challis, 2020) the author mentions another form to average quaternion, this consists of an arithmetic average, that is to say, a sum of n quaternions divided by n .

Sensor network is another concept that make possible to obtain measurements of a sensor array and fuse information that each node generates in order to reach an agreement (Kia et al., 2018). The concept of network sensor has been utilized widely in control systems for consensus theory (Olfati-Saber and Shamma, 2006), where the communication between each node of the sensor network is given by a communication graph and each node is capable of develop local computations (Kia et al., 2018).

This paper is organized as follows. In section 2, a mathematical background for distributed attitude parameterization is provided and the statement problem is presented. Section 3, introduces the dynamic consensus algorithm design. The simulation results are presented in section 4. Finally section 5 presents the conclusions.

2. MATHEMATICAL BACKGROUND

2.1 Unit Quaternions

Consider the following, right-handed coordinate frames (see Fig. 1): the inertial coordinate frame, $\mathbf{E}^f = [e_1^f, e_2^f, e_3^f]$, located at some fixed point in space, the body coordinate frame, $\mathbf{E}^b = [e_1^b, e_2^b, e_3^b]$, located at the center of mass of the rigid body, and, the coordinate frame of the MU i denoted $\mathbf{E}^{mu_i} = [e_1^{mu_i}, e_2^{mu_i}, e_3^{mu_i}]$ which is located in the body. It is assumed that the origin of \mathbf{E}^b coincides with the origin of \mathbf{E}^{mu_i} . The rotation of a reference frame to another can be represented by the rotation matrix $R \in SO(3) = \{R \in \mathbb{R}^{3 \times 3} : R^T R = I, \det R = 1\}$.

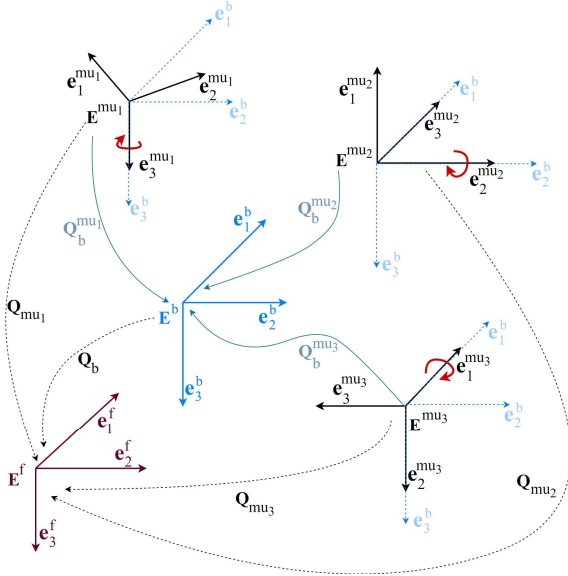


Fig. 1. Relationship between \mathbf{E}^b , \mathbf{E}^f and \mathbf{E}^{mu_i} reference systems.

The n -dimensional unit sphere embedded in \mathbb{R}^{n+1} is denoted as $\mathbb{S}^n = \{x \in \mathbb{R}^{n+1} : x^T x = 1\}$. Members of $SO(3)$ are often parameterized in terms of a rotation $\beta \in \mathbb{R}$ about a fixed axis $e \in \mathbb{S}^2$ by the map $\mathcal{U} : \mathbb{R} \times \mathbb{S}^2 \rightarrow SO(3)$ defined as:

$$\mathcal{U}(\beta, e) := I + \sin(\beta)[e^\times] + (1 - \cos(\beta))[e^\times]^2 \quad (1)$$

where $[e^\times]$ is the skew symmetric matrix defined by:

$$[e^\times] = \begin{pmatrix} 0 & -\xi_3 & \xi_2 \\ \xi_3 & 0 & -\xi_1 \\ -\xi_2 & \xi_1 & 0 \end{pmatrix}$$

Hence, a unit quaternion, $\mathbf{Q} \in \mathbb{S}^3$, is defined as:

$$\mathbf{Q} := \begin{pmatrix} \cos \frac{\beta}{2} \\ e \sin \frac{\beta}{2} \end{pmatrix} = \begin{pmatrix} q_0 \\ \mathbf{q} \end{pmatrix} \in \mathbb{S}^3 \quad (2)$$

where $\mathbf{q} = [q_1, q_2, q_3]^T \in \mathbb{R}^3$ and $q_0 \in \mathbb{R}$ are known as the vector and scalar parts of the quaternion respectively. The

inverse unit quaternion is given by $\mathbf{Q}^{-1} := [q_0, -\mathbf{q}^T]^T$, and the quaternion product is defined by (Markley et al., 2007):

$$\mathbf{Q}_1 \odot \mathbf{Q}_2 := \begin{pmatrix} q_{1_0} & -\mathbf{q}_1^T \\ \mathbf{q}_1 & I_3 q_{1_0} + [\mathbf{q}_1^\times] \end{pmatrix} \begin{pmatrix} q_{2_0} \\ \mathbf{q}_2 \end{pmatrix} \quad (3)$$

$$\mathbf{Q}_1 \otimes \mathbf{Q}_2 := \begin{pmatrix} q_{1_0} & -\mathbf{q}_1^T \\ \mathbf{q}_1 & I_3 q_{1_0} - [\mathbf{q}_1^\times] \end{pmatrix} \begin{pmatrix} q_{2_0} \\ \mathbf{q}_2 \end{pmatrix} \quad (4)$$

It follows of these definitions that:

$$\mathbf{Q}_1 \otimes \mathbf{Q}_2 = \mathbf{Q}_2 \odot \mathbf{Q}_1 \quad (5)$$

The attitude error is used to quantify the mismatch between two attitudes. If \mathbf{Q} and $\hat{\mathbf{Q}}$ define the true attitude quaternion and the estimated one, respectively, then attitude error is given by:

$$\mathbf{Q}_e := \mathbf{Q} \odot \hat{\mathbf{Q}}^{-1} = [q_{e_0}, \mathbf{q}_e^T]^T \quad (6)$$

In the case that the true quaternion and the estimated one coincide, the quaternion error becomes:

$$\mathbf{Q}_e = [\pm 1, \mathbf{0}^T]^T \quad (7)$$

where $\mathbf{0}$ is a vector of zeros.

2.2 Attitude Measurement Models

Let us consider the representation of a unit vector \mathbf{x}_j with respect to \mathbf{E}^f , \mathbf{E}^b , and \mathbf{E}^{mu_i} which is denoted by \mathbf{r}_j , \mathbf{b}_j , and $\mathbf{b}_j^{mu_i}$, respectively. The vectors \mathbf{r}_j are called the ‘‘reference vectors’’, and in general, are known quite accurately. The vectors $\mathbf{b}_j^{mu_i}$ are known as ‘‘observation vectors’’ and are obtained from the i th measurement unit (in our case, magnetometers and accelerometers).

These vector representations are linked up through the vector rotation using quaternions. Vectors $\mathbf{b}_j^{mu_i}$, \mathbf{b}_j and \mathbf{r}_j can be written as quaternions as follows:

$$\mathbf{Q}(\mathbf{b}_j^{mu_i}) = [0, (\mathbf{b}_j^{mu_i})^T]^T \quad \mathbf{Q}(\mathbf{b}_j) = [0, \mathbf{b}_j^T]^T \quad (8)$$

$$\mathbf{Q}(\mathbf{r}_j) = [0, \mathbf{r}_j^T]^T$$

Then, these quaternions are related by the rotation quaternions \mathbf{Q}_b , $\mathbf{Q}_b^{mu_i}$, and \mathbf{Q}_{mu_i} as follows:

$$\mathbf{Q}(\mathbf{b}_j) = (\mathbf{Q}_b)^{-1} \odot \mathbf{Q}(\mathbf{r}_j) \odot \mathbf{Q}_b \quad (9)$$

$$\mathbf{Q}(\mathbf{b}_j^{mu_i}) = (\mathbf{Q}_b^{mu_i})^{-1} \odot \mathbf{Q}(\mathbf{b}_j) \odot \mathbf{Q}_b^{mu_i} \quad (10)$$

and, as a consequence:

$$\begin{aligned} \mathbf{Q}(\mathbf{b}_j^{mu_i}) &= (\mathbf{Q}_b^{mu_i})^{-1} \odot [(\mathbf{Q}_b)^{-1} \odot \mathbf{Q}(\mathbf{r}_j) \odot \mathbf{Q}_b] \odot \mathbf{Q}_b^{mu_i} \\ &= (\mathbf{Q}_{mu_i})^{-1} \odot \mathbf{Q}(\mathbf{r}_j) \odot \mathbf{Q}_{mu_i} \end{aligned} \quad (11)$$

where $\mathbf{Q}_{mu_i} = \mathbf{Q}_b \odot \mathbf{Q}_b^{mu_i}$. Note that $\mathbf{Q}_b^{mu_i}$ is quite known by design. Then, if \mathbf{Q}_{mu_i} is estimated, it is possible to know \mathbf{Q}_b , *i.e.*, the attitude of \mathbf{E}^b with respect to \mathbf{E}^f by simple quaternion multiplication, that is:

$$\mathbf{Q}_b = \mathbf{Q}_{mu_i} \odot (\mathbf{Q}_b^{mu_i})^{-1} \quad (12)$$

Moreover, from (11) it is easy to prove that:

$$\mathbf{Q}_{mu_i} \odot \mathbf{Q}(\mathbf{b}_j^{mu_i}) = \mathbf{Q}(\mathbf{r}_j) \odot \mathbf{Q}_{mu_i} \quad (13)$$

$$\mathbf{Q}(\mathbf{b}_j^{mu_i}) \otimes \mathbf{Q}_{mu_i} = \mathbf{Q}(\mathbf{r}_j) \odot \mathbf{Q}_{mu_i}$$

Then using the definition of quaternion product (3) and (4), one obtains the following relation:

$$\begin{pmatrix} 0 & -(\mathbf{b}_j^{mu_i} - \mathbf{r}_j)^T \\ \mathbf{b}_j^{mu_i} - \mathbf{r}_j & -[(\mathbf{b}_j^{mu_i} + \mathbf{r}_j)^\times] \end{pmatrix} \mathbf{Q}_{mu_i} = H_j^i \mathbf{Q}_{mu_i} = 0 \quad (14)$$

$H_j^i \in \mathbb{R}^{4 \times 4}$ is called the observation matrix and one can see that (14) is a well-structured linear system of equations.

2.3 Graph Theory

The interaction of a multi-agent system can be represented by graphs properties. A graph is formally represented by the set $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ consisting of a set of vertices (or nodes) $\mathcal{V} = 1, \dots, n$ and edges $\mathcal{E} \in \mathcal{V} \times \mathcal{V}$. Each node represents an agent. If there is an edge (i, l) between nodes i and l , with $1 \leq i \leq n$ and $1 \leq l \leq n$, then i and l are called adjacent, *i.e.*, $\mathcal{E} = \{(i, l) \in \mathcal{V} \times \mathcal{V} : i, l \text{ adjacent}\}$. An entry of the adjacency matrix \mathbf{A} is defined by $a_{il} = 1$ if i and l are adjacent and $a_{il} = 0$ otherwise. Note that the diagonal elements of the adjacency matrix are all zero for a graph without any loop (as in the present paper). \mathcal{G} is called undirected if $(i, l) \in \mathcal{E} \Leftrightarrow (l, i) \in \mathcal{E}$. A path from i to l is a sequence of distinct nodes, starting from i and ending with l , such that each pair of consecutive nodes is adjacent. If there is a path from i to l , then i and l are called connected. If all pairs of nodes in \mathcal{G} are connected, then \mathcal{G} is called connected. The distance $d(i, l)$ between two nodes is the number of edges of the shortest path from i to l . The diameter \mathbf{d} of \mathcal{G} is the maximum distance $d(i, l)$ over all pairs of nodes. The degree (or valency) matrix \mathbf{D} of \mathcal{G} is a diagonal matrix whose diagonal elements d_i are equal to the cardinality of node i 's neighbor set $\mathcal{N}_i = \{l \in \mathcal{V} : (i, l) \in \mathcal{E}\}$. Let $\mathcal{M}_i = \mathcal{N}_i \cup \{i\}$ denote the set of inclusive neighbors of node i . The Laplacian matrix \mathcal{L} of \mathcal{G} is defined as $\mathcal{L} = \mathbf{D} - \mathbf{A}$. For undirected graphs, \mathcal{L} is symmetric and positive semi-definite, *i.e.* $\mathcal{L} = \mathcal{L}^T \geq 0$. The row sums of \mathcal{L} are zero.

2.4 Problem Statement

According to (9)-(10), the goal of the attitude estimation process is to estimate the attitude of \mathbf{E}^b with respect to \mathbf{E}^j , *i.e.* quaternion \mathbf{Q}_b . This process will be carried out via the estimation of quaternion \mathbf{Q}_{mu_i} , which maps the observation vectors into the known reference vectors. That is, given the observation vectors $\mathbf{b}_j^{mu_i}$ and the reference vectors \mathbf{r}_j , with $j \in \{1, 2 \dots m\}$ the number of reference vectors, and $i \in \{1, 2 \dots n\}$ the number of measurement units (denoted hereafter agents), the objective is to find \mathbf{Q}_{mu_i} in each agent i such that:

$$H_j^i \mathbf{Q}_{mu_i} = 0 \quad (15)$$

Once the attitude quaternion \mathbf{Q}_{mu_i} is estimated in each agent, we will propose an algorithm that averages these quaternions dynamically in the \mathbf{E}^b frame.

3. DESIGN OF THE DYNAMIC CONSENSUS ALGORITHM

Consider a set of n agents, each one conformed by a measurement unit (MU). One assumes that each agent is capable of providing a unit quaternion of attitude, *i.e.* \mathbf{Q}_{mu_i} with $i \in \{1, 2 \dots n\}$. Then, in this section one explains how this quaternion is estimated. After that, the dynamic average in the \mathbf{E}^b frame will be exposed.

From (14), the observation matrix for the i th agent in function of m pair vectors $[\mathbf{b}_j^{mu_i}, \mathbf{r}_j]$ with $j \in \{1, 2, \dots, m\}$ is written as follows:

$$H_m^i = \begin{pmatrix} 0 & -(\mathbf{b}_1^{mu_i} - \mathbf{r}_1)^T \\ \mathbf{b}_1^{mu_i} - \mathbf{r}_1 & -[(\mathbf{b}_1^{mu_i} + \mathbf{r}_1)^\times] \\ \vdots & \vdots \\ 0 & -(\mathbf{b}_m^{mu_i} - \mathbf{r}_m)^T \\ \mathbf{b}_m^{mu_i} - \mathbf{r}_m & -[(\mathbf{b}_m^{mu_i} + \mathbf{r}_m)^\times] \end{pmatrix} \in \mathbb{R}^{4m \times 4} \quad (16)$$

such that:

$$H_m^i \mathbf{Q}_{mu_i} = \begin{pmatrix} \varphi_1^T(t) \\ \vdots \\ \varphi_{4m}^T(t) \end{pmatrix} \mathbf{Q}_{mu_i} = 0 \quad (17)$$

Quaternion \mathbf{Q}_{mu_i} can be obtained by a deterministic method. In (Guerrero-Castellanos et al., 2013) a deterministic attitude estimator which we coin here as “*Learning Quaternion Sequence*” (LQS) is reported and its implementation with others deterministic attitude estimators are reported in (Betanzos-Ramirez et al., 2021). LQS is key in the distributed dynamic consensus for attitude estimation proposed here and then it is briefly presented. Consider a matrix observation $H_m^i(t)$ in the instant t and an initial quaternion denoted \mathbf{Q}_{ps}^0 . Then, a estimation of the attitude of the i th agent is given by

$$\hat{\mathbf{Q}}_{mu_i} = \left[\prod_{\kappa=1}^{4m} \left(\mathbf{I}_4 - \frac{\gamma \varphi_\kappa(t) \varphi_\kappa^T(t)}{\alpha + \varphi_\kappa^T(t) \varphi_\kappa(t)} \right) \right] \mathbf{Q}_{ps}^0 \quad (18)$$

where $\alpha \geq 0$ and $0 < \gamma < 2$. Once (18) is performed, one enforces the quaternion norm

$$\hat{\mathbf{Q}}_{mu_i} = \frac{\mathbf{Q}_{ps}^{4m}}{\|\mathbf{Q}_{ps}^{4m}\|} \quad (19)$$

Remark 3.1. Note that (18) represents an iterative algorithm to be performed for each instant t . Then, for $t = 0$ the initial quaternion \mathbf{Q}_{ps}^0 in (18) can be chosen arbitrarily. In the sequel, for all $t > 0$, \mathbf{Q}_{ps}^0 will be obtained from the output of the dynamic consensus. In the present collaborative framework, \mathbf{Q}_{ps}^0 it can be chosen as:

$$\mathbf{Q}_{ps}^0 = \begin{cases} \text{arbitrary unit quaternion, for } t = 0 \\ \bar{\mathbf{Q}}_i, \text{ for } t > 0 \end{cases} \quad (20)$$

Remark 3.2. Quaternion $\hat{\mathbf{Q}}_{mu_i}$ represents the attitude estimation of the \mathbf{E}^{mu_i} frame. This attitude can be represented in the \mathbf{E}^b frame by means of (12), that is:

$$\hat{\mathbf{Q}}_{b_i} = \hat{\mathbf{Q}}_{mu_i} \odot (\mathbf{Q}_b^{mu_i})^{-1} \quad (21)$$

It is deserving to be noted that $\mathbf{Q}_b^{mu_i}$ are known.

Each agent is formed by a MU whose sensors account with their own noise characteristic. As a consequence, the quaternion estimated by each agent will be different in term of reliability. The idea is to take each estimated quaternion $\hat{\mathbf{Q}}_{b_1}, \hat{\mathbf{Q}}_{b_2}, \dots, \hat{\mathbf{Q}}_{b_n}$ and perform a distributed dynamic consensus (average) in the \mathbf{E}^b frame.

Let us define the following state vectors $\bar{\mathbf{Q}} = [\bar{\mathbf{Q}}_1^T, \dots, \bar{\mathbf{Q}}_n^T]^T$ and $\Upsilon = [(\hat{\mathbf{Q}}_{b_1})^T, (\hat{\mathbf{Q}}_{b_2})^T, \dots, (\hat{\mathbf{Q}}_{b_n})^T]^T$. Thus, we propose the following Dynamic Consensus Algorithm (DCA):

$$\dot{\bar{\mathbf{Q}}}_i(t) = \sum_{l \in \mathcal{N}_i} a_{il} (\bar{\mathbf{Q}}_l(t) - \bar{\mathbf{Q}}_i(t)) + \sum_{l \in \mathcal{M}_i} a_{il} (\hat{\mathbf{Q}}_{b_l}(t) - \bar{\mathbf{Q}}_i(t)) \quad (22)$$

where a_{il} are elements of the adjacency matrix \mathbf{A} of the sensor network, \mathcal{N}_i and \mathcal{M}_i are the set of neighbors and the set of inclusive neighbors of node i , respectively, as already defined in subsection 2.3. Then, one has the following results.

Proposition 3.1. The distributive algorithm in (22) gives a consensus filter with the following collective dynamics:

$$\dot{\bar{\mathbf{Q}}} = -\Xi \bar{\mathbf{Q}} + B \Upsilon \quad (23)$$

where $\Xi = (\mathbf{I}_{4n} + \mathbf{D} + \mathcal{L})$ and $B = \mathbf{I}_{4n} + \mathbf{A}$, with \mathbf{D} and \mathcal{L} the degree and Laplacian matrices respectively, \mathbf{I}_{4n} is the identity matrix.

Proof 3.1. The proof follows from the definition of graph Laplacian reviewed in subsection 2.3. For further details, see (Olfati-Saber and Shamma, 2006).

Proposition 3.2. The consensus filter described in (23) is a distributed stable low-pass filter with transfer function given by:

$$H(s) = (s\mathbf{I}_{4n} + \Xi)^{-1} B \quad (24)$$

Proof 3.2. Applying Gersgorin theorem to matrix Ξ guarantees that all poles of $H(s)$ are strictly negative. Then the filter is stable. Furthermore, $H(s)$ is a proper MIMO transfer function, which means that it is a low-pass filter (Olfati-Saber and Shamma, 2006).

Figure 2 depicts a schema of the proposed approach. This configuration offers certain advantages, such as fault tolerance if any of the sensors associated with each MU fails, resulting in a fault-tolerant attitude estimator.

3.1 Fault-Tolerant Robustness

For simulations purpose, the reference attitude (true attitude) and the sensor measurement on each MU will be generated by the following system:

$$\dot{\boldsymbol{\omega}} = \mathbf{J}^{-1} (-\boldsymbol{\omega} \times \mathbf{J} \boldsymbol{\omega} + \boldsymbol{\Gamma}) \quad (25)$$

$$\dot{\mathbf{Q}}_b(t) = \frac{1}{2} \begin{pmatrix} -\mathbf{q}^T \\ q_0 \mathbf{I}_3 + [\mathbf{q} \times] \end{pmatrix} \boldsymbol{\omega} \quad (26)$$

When solving the dynamic equation (25), an angular velocity $\boldsymbol{\omega}$ expressed in the \mathbf{E}^b reference system is obtained, with respect to $\boldsymbol{\Gamma} \in \mathbb{R}^3$ which is the vector of input

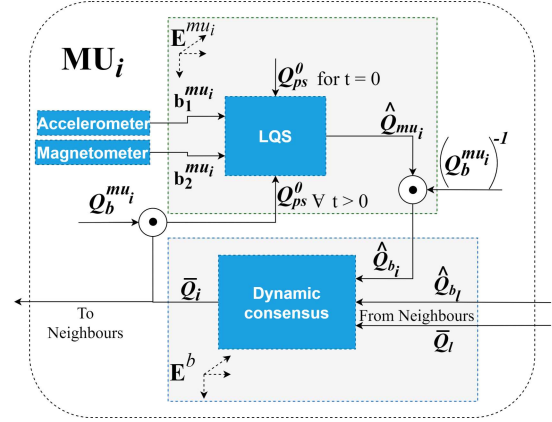


Fig. 2. Block diagram of an agent in the sensor network. \odot represents the quaternion product.

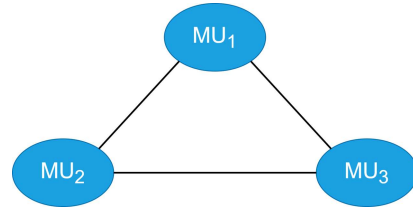


Fig. 3. Communication graph between each MU_i .

torque for each axis of the rigid body. \mathbf{J} represents the rigid body's inertia matrix. Then $\boldsymbol{\omega}$ is an input signal for the kinematic equation (26), that gives a quaternion expressed in the \mathbf{E}^b reference system when it is solved, i.e. \mathbf{Q}_b , with q_0 its vector and scalar parts. Both $\boldsymbol{\omega}$ and \mathbf{Q}_b are taken as reference (true) signals for the simulation process.

Now consider the following unit reference vectors written as quaternions: $\mathbf{Q}(r_1) = [0; 0; 0; 1]$ (gravitational acceleration) and $\mathbf{Q}(r_2) = [0; 0.337; 0; 0.348]$ (earth's magnetic field), which are described in the \mathbf{E}^f reference system, these new quaternions can be mapped to the \mathbf{E}^{mu_i} reference system through the equation (11). It is worthy to note that $\mathbf{Q}_b^{mu_i}$ is defined by the designer and \mathbf{Q}_b is given by the system (26), therefore it is possible to obtain the quaternions $\mathbf{Q}(b_1^{mu_i})$ and $\mathbf{Q}(b_2^{mu_i})$. So the simulated measurements for i th accelerometer and i th magnetometer can be taken from vector part of $\mathbf{Q}(b_1^{mu_i})$ and $\mathbf{Q}(b_2^{mu_i})$ respectively. Next LQS algorithm and dynamic consensus can be executed like in Figure 2.

Finally in order to obtain the mismatch between $\hat{\mathbf{Q}}_{mu_i}$ and \mathbf{Q}_b (reference/true quaternion) and between $\bar{\mathbf{Q}}_i$ and \mathbf{Q}_b , equation (6) is applied to these quaternions. Next the inverse cosine of the scalar part of the quaternion \mathbf{Q}_e is calculated, i.e., $\beta_e = \arccos q_{e_0}$.

It is important to highlight that the LQS algorithm makes use of well-known reference vectors \mathbf{r}_j and observation vectors $\mathbf{b}_j^{mu_i}$, as defined in the observation matrix H_m^i in (16). However, for each agent this matrix is function of data that are sensed. If any of the sensors fail in MU i , the matrix changes with a pair vector $[\mathbf{b}_j^{mu_i}, \mathbf{r}_j]$ that falls.

Obviously, in a practical application it will be necessary to develop a fault detection algorithm in order to identify the sensor that has failed and make the necessary changes. This is not considered here.

4. SIMULATIONS RESULTS

Since one is interested in distributed attitude estimation, a simulation was developed making use of three agents, so, the algorithms described in the block diagram of Figure 2 are simulated, with a graph communication given by Figure 3, where each node accounts with one MU with its own noise characteristics. First, two observation vectors are available, gravitational and magnetic vectors, which are obtained from $\mathbf{Q}(b_1^{mu_i})$ and $\mathbf{Q}(b_2^{mu_i})$, i.e. $\mathbf{b}_1^{mu_i}$, $\mathbf{b}_2^{mu_i}$, respectively, for $i \in \{1, 2, 3\}$. Then, the matrix H_m^i of the LQS algorithm, previously defined in its general form in (16), takes the form:

$$H_m^i = \begin{pmatrix} 0 & -(\mathbf{b}_1^{mu_i} - \mathbf{r}_1^{mu_i})^T \\ \mathbf{b}_1^{mu_i} - \mathbf{r}_1^{mu_i} & -[(\mathbf{b}_1^{mu_i} + \mathbf{r}_1^{mu_i}) \times] \\ 0 & -(\mathbf{b}_2^{mu_i} - \mathbf{r}_2^{mu_i})^T \\ \mathbf{b}_2^{mu_i} - \mathbf{r}_2^{mu_i} & -[(\mathbf{b}_2^{mu_i} + \mathbf{r}_2^{mu_i}) \times] \end{pmatrix} \in \mathbf{R}^{8 \times 4} \quad (27)$$

Remind that LQS algorithm is developed for each node, this block provides \mathbf{Q}_{mu_i} , i.e., attitude of the MU with respect to inertial frame. So, a transformation is required in order to obtain $\hat{\mathbf{Q}}_b$ which parameterizes the attitude of the rigid body. Each node, represented in Figures 3 and 2, runs the dynamic consensus algorithm (22).

Table 1 shows the quaternion $\mathbf{Q}_b^{mu_i}$ associated to each node of the network that relates the \mathbf{E}^{mu_i} and \mathbf{E}^b reference systems through a rotation. The resulting allocation of each MU over the rigid body can be translated to Figure 1, where the origin of \mathbf{E}^b and \mathbf{E}^{mu_i} reference systems are coincident, but their components are not aligned.

Table 1. Rotation applied to each MU with respect to \mathbf{E}^b reference system.

i	$\mathbf{Q}_b^{mu_i}$ associated to each MU	
	$\mathbf{Q}_b^{mu_i}$	Euler angles
1	$[0.3827; 0.9239; 0; 0]^T$	$\phi = 135^\circ, \theta = 0^\circ, \psi = 0^\circ$
2	$[0.3827; 0; 0.9239; 0]^T$	$\phi = 0^\circ, \theta = 135^\circ, \psi = 0^\circ$
3	$[0.3827; 0; 0; 0.9239]^T$	$\phi = 0^\circ, \theta = 0^\circ, \psi = 135^\circ$

Euler angles will be depicted and analysed since they are more intuitive. However, the algorithm is quaternion-based. The dynamic consensus algorithm is simulated together with the LQS algorithm in order to obtain a single estimate of the attitude of the rigid body. The initial condition for $t = 0$ are: $\bar{\mathbf{Q}}^0 = [0.933; 0.25; 0.25; -0.06699]$ and $\mathbf{Q}_{ps}^0 = [1; 0; 0; 0]$. Figure 4 shows the result obtained

from this simulation, where the continuous lines represent the Euler angles obtained from reference \mathbf{Q}_b , and the dashed lines the Euler angles obtained from the dynamic consensus algorithm. In this case, a torque in axis *yaw* of rigid body is applied.

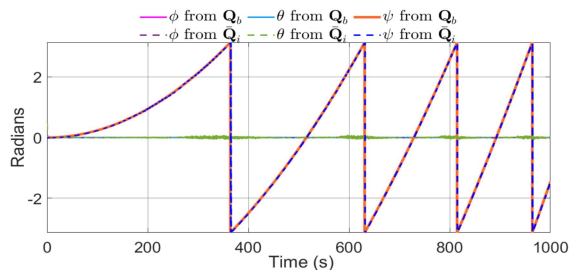


Fig. 4. Comparison between Euler angles from reference and Euler angles from dynamic consensus.

If the gravitational vector fails, in any of the nodes, the matrix (16) could be modified at that node. This matrix becomes:

$$H_m^i = \begin{pmatrix} 0 & -(\mathbf{b}_2^{mu_i} - \mathbf{r}_2^{mu_i})^T \\ \mathbf{b}_2^{mu_i} - \mathbf{r}_2^{mu_i} & -[(\mathbf{b}_2^{mu_i} + \mathbf{r}_2^{mu_i}) \times] \end{pmatrix} \in \mathbf{R}^{4 \times 4} \quad (28)$$

When the magnetic vector fails, the matrix (16) becomes:

$$H_m^i = \begin{pmatrix} 0 & -(\mathbf{b}_1^{mu_i} - \mathbf{r}_1^{mu_i})^T \\ \mathbf{b}_1^{mu_i} - \mathbf{r}_1^{mu_i} & -[(\mathbf{b}_1^{mu_i} + \mathbf{r}_1^{mu_i}) \times] \end{pmatrix} \in \mathbf{R}^{4 \times 4} \quad (29)$$

As mentioned before, the mismatch between \mathbf{Q}_b and $\hat{\mathbf{Q}}_i$ and $\bar{\mathbf{Q}}_i$ is $\beta_e = \arccos q_{e0}$, in order to analyze this mismatch the ISE index defined in (30) was used (Guerrero-Castellanos et al., 2021).

$$ISE = \int_0^T \tilde{x}^2(t) dt \quad (30)$$

Therefore, the ISE criterion was applied to angle β_e as follows,

$$ISE_{\beta_e} = \int_0^T \tilde{\beta}_e^2(t) dt \quad (31)$$

Table 2. Sensor operation during the simulation process of Figure 5

scene	time (s)	MU ₁		MU ₂		MU ₃		MU	
		\mathbf{b}_1	\mathbf{b}_2	\mathbf{b}_1	\mathbf{b}_2	\mathbf{b}_1	\mathbf{b}_2	\mathbf{b}_1	\mathbf{b}_2
-	-	✓	✓	✓	✓	✓	✓	✓	✓
1	0 – 200	✓	✓	✓	✓	✓	✓	✓	✓
2	200 – 400	✓	X	✓	✓	✓	✓	✓	✓
3	400 – 500	X	✓	X	✓	✓	✓	✓	✓
4	500 – 700	✓	✓	✓	X	X	✓	✓	✓
5	700 – 900	X	✓	✓	X	✓	X	✓	X
6	900 – 1100	X	✓	✓	X	X	✓	✓	✓
7	1100 – 1300	✓	✓	X	X	X	✓	✓	✓
8	1300 – 1500	X	X	X	X	✓	✓	✓	✓

Table 2 shows the operation of sensors for the sensor network, where MU₁, MU₂ and MU₃ are nodes or agents of

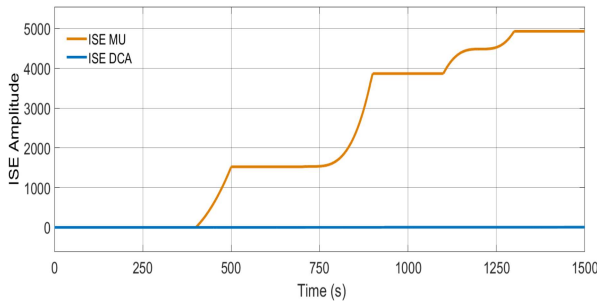


Fig. 5. ISE index of the quaternion error when a sensor failure occurs

the network and MU is independent of the sensor network. For this case \mathbf{b}_1 and \mathbf{b}_2 represent the vector observations obtained from accelerometer and magnetometer respectively, a \checkmark indicates that the correspond sensor is working correctly, X represent the case of failure of any sensor.

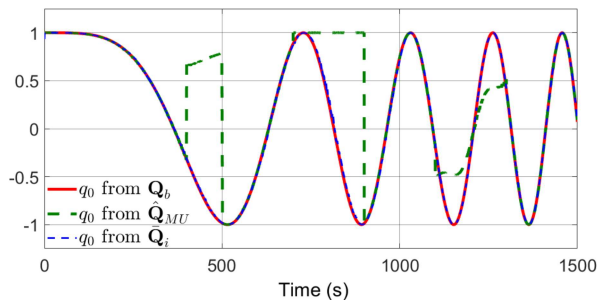


Fig. 6. Time evolution of the scalar part of \mathbf{Q}_b , $\hat{\mathbf{Q}}_{MU}$ and $\hat{\mathbf{Q}}_i$

Figure 5, shows the results of simulation associated to Table 2. Figure 6 shows the behavior of $q_0(t)$ for the \mathbf{Q}_b (true attitude), $\hat{\mathbf{Q}}_{MU}$ (single measurement unit) and for $\hat{\mathbf{Q}}_i$ (average quaternion), these behaviors are related with simulation of Figure 5.

5. CONCLUSION

The paper presents a dynamic consensus algorithm for distributed attitude estimation. Attitude estimation is fundamental in many applications, such as robotics, biomedical engineering, and aerospace. In most of them, knowing an accurate value of the attitude is critical. Performing a distributed estimation and developing the dynamic consensus algorithm ensures a more precise attitude estimation. Furthermore, with the proposed approach, the system remains robust in the event of any sensor failure. Simulation results were presented. In the future, real-time experiments will be performed.

ACKNOWLEDGEMENTS

This work is part of the French-Mexican TOBACCO project funded by the FORDECYT-PRONACES through the joint SEP-CONACYT-ANUIES-ECOS Nord program (MX-296702 & FR-M18M02) and dark-NAV project funded by the French National Research Agency. All the authors thank DGAPA-UNAM for the financial support given through the PAPIIT project TA100421: Control y Automatización de una Boya para el Monitoreo de variables Oceanográficas. This work was carried out thanks to the support granted by CONACYT (Atención a Problemas Nacionales, Convocatoria 2017, Proyecto 4730: Estaciones de diagnóstico y monitoreo para redes de distribución de agua con interconexión a internet). In addition, Osberto A. Betanzos-Ramirez with CVU 1075995 thanks CONACYT for providing financial support through his Master's scholarship .

REFERENCES

- Betanzos-Ramirez, O.A., Sánchez-Mendoza, A., Guerrero-Castellanos, J.F., and Torres, L. (2021). Evaluation of deterministic attitude estimation algorithms in real-time using low-cost hardware. In *2021 International Conference on Mechatronics, Electronics and Automotive Engineering ICMEAE 2021*, 225–230. doi:10.1109/ICMEAE55138.2021.00041.
- Challis, J. (2020). Quaternions as a solution to determining the angular kinematics of human movement. *BMC Biomedical Engineering*, 2. doi:10.1186/s42490-020-00039-z.
- Guerrero-Castellanos, J., Durand, S., Muñoz Hernández, G., Marchand, N., González-Romeo, L., Linares-Flores, J., Mino-Aguilar, G., and Guerrero-Sánchez, W. (2021). Bounded attitude control with active disturbance rejection capabilities for multirotor uavs. *Applied Sciences*, 11(13), 5960. doi:10.3390/app11135960. URL <http://publis.icube.unistra.fr/2-GDMM21>.
- Guerrero-Castellanos, J.F., Madrigal-Sastre, H., Durand, S., Torres, L., and Munoz-Hernandez, G. (2013). A robust nonlinear observer for real-time attitude estimation using low-cost MEMS inertial sensors. *Sensors (Basel, Switzerland)*, 13, 15138–58. doi: 10.3390/s131115138.
- Kia, S., Van Scoy, B., Cortes, J., Freeman, R., Lynch, K., and Martínez, S. (2018). Tutorial on dynamic average consensus: The problem, its applications, and the algorithms. *IEEE Control Systems*, 39. doi: 10.1109/MCS.2019.2900783.
- Markley, L., Cheng, Y., Crassidis, J., and Oshman, Y. (2007). Averaging quaternions. *Journal of Guidance, Control, and Dynamics*, 30, 1193–1196. doi: 10.2514/1.28949.
- Olfati-Saber, R. and Shamma, J. (2006). Consensus filters for sensor networks and distributed sensor fusion. In *44th IEEE Conference on Decision and Control*, volume 2005, 6698 – 6703. doi: 10.1109/CDC.2005.1583238.

Apéndice D

Conexión entre los software MATLAB y Motive

D.1. Introducción

En este capítulo se explica al lector como realizar una comunicación exitosa entre MATLAB y Motive.

Motive es un software que ofrece una forma simple y precisa de realizar el seguimiento de un cuerpo rígido. Permite la creación de cuerpos rígidos de una manera sencilla, así como también visualizar la posición y orientación del cuerpo rígido con respecto al sistema de referencia del optitrack. También permite la grabación de la trayectoria realizada, la información generada en el software motive puede ser exportada en diferente formatos, con la finalidad de realizar un análisis posterior en otro software. Esta forma de exportar la información que motive genera es llamada off-line, es decir, primero se genera la información en el software motive y luego es exportado a otro software como MATLAB. Sin embargo una útil e interesante forma de compartir la información generada por motive es a través de una comunicación online, es decir, transmitir la información en tiempo real de un software a otro. Esto es posible de hacer por medio de streaming, que es una herramienta que motive ofrece. Si el software donde se realizará el análisis se encuentra en la misma computadora que el software motive, el streaming es local, en caso de encontrarse en otra computadora, ambos equipos deberán estar conectados a la misma red de internet.

MATLAB es utilizado como software para el análisis de la información generada por motive, considerando un streaming local. Los pasos para realizar una conexión exitosa entre MATLAB y motive son descritos a continuación.

D.1.1. Instalación del Toolbox Optitrack

- Paso 1: Descargar el toolbox optitrack de la siguiente página: <https://la.mathworks.com/matlabcentral/fileexchange/55675-kutzer-optitracktoolbox>. Este toolbox es gratis y para su descarga, deberá iniciar sesión en su cuenta de MATLAB. Otra opción de descarga del toolbox es del repositorio de github del autor: <https://github.com/kutzer/OptiTrackToolbox>.
- Paso 2: Una vez descargado el toolbox, deberá descomprimir el archivo .zip, para acceder a la carpeta principal.
- Paso 3: Ejecute el software de MATLAB y cambie el directorio actual de trabajo a la dirección de la carpeta donde se encuentra el toolbox optitrack y que acaba de descomprimir.

La Figura D.1 muestra el directorio de trabajo con la carpeta `github_repo` como la carpeta de trabajo actual y se muestran los archivos dentro de la misma.

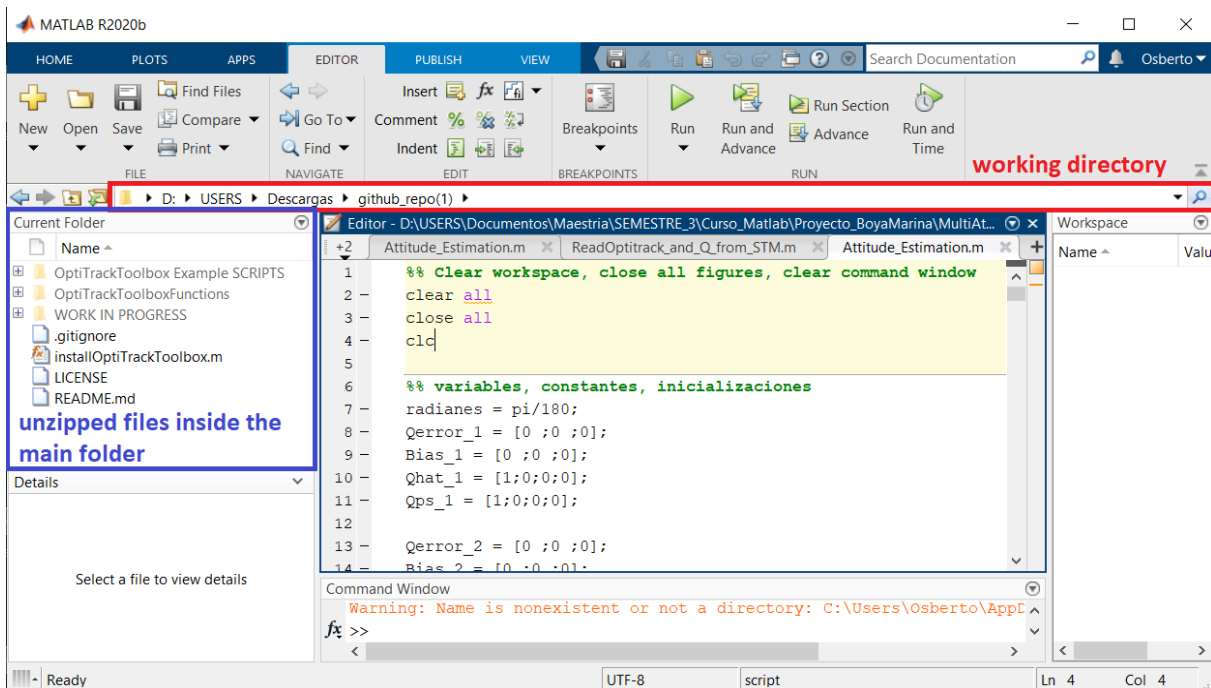


Figura D.1: Directorio de trabajo con el Toolbox optitrack para MATLAB

- Paso 4: Abra y ejecute el archivo `installOptiTrackToolbox.m` que se encuentra dentro de la carpeta principal. Si el toolbox no se encuentra instalado, el archivo se ejecutará hasta que todas las herramientas necesarias del toolbox sean instaladas. En caso contrario, MATLAB le notificará que el toolbox ya se encuentra instalado.

D.1.2. Configuración de Motive

Ahora, en orden de permitir el intercambio de información de motive a MATLAB, es necesario configurar algunos parámetros en el software motive. Es importante remarcar que ambos software motive y MATLAB, deberán estar instalados en la misma computadora. Los pasos necesarios son descritos abajo.

- Paso 1: Ejecute el software motive.
- Paso 2: Haga click sobre el **Streaming Pane**.

La Figura D.2 Muestra el streaming pane en el software motive. Note que la apariencia de la ventana de inicio de motive podría ser diferente a la que usted ejecuta, esto dependerá de la versión del software de motive que se tenga instalado.

- Paso 3: En el lado izquierdo de la ventana, podrás ver el siguiente menú: **OptiTrack Streaming Engine**, deberás **activar (turn on)** el Broadcast Frame Data y luego del menú Local Interface deberás escoger la opción: **loopback** para un streaming local. La Figura D.3 muestra el menú OptiTrack Streaming Engine y la configuración necesaria para un streaming local.

Estos son los únicos pasos necesarios para un intercambio exitoso de información entre motive y MATLAB, sin embargo algunas veces los componentes y y z del sistema de referencia que usamos

D. D.1. INTRODUCCIÓN

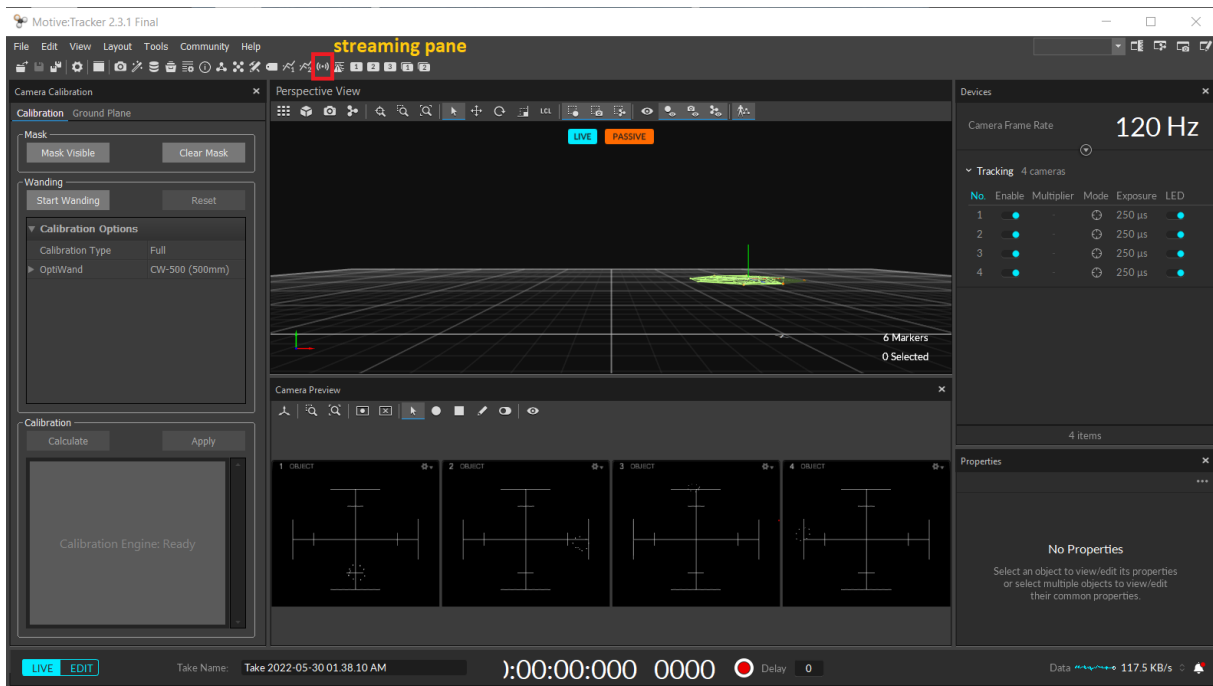


Figura D.2: Software Motive y streaming pane option

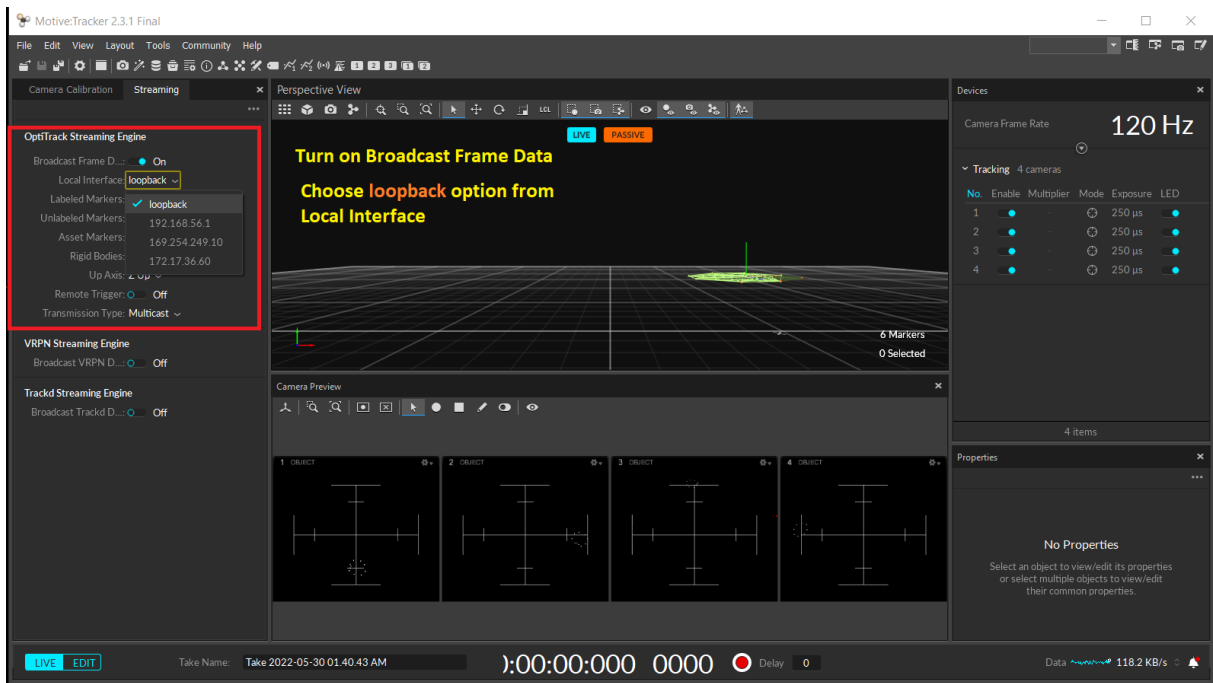


Figura D.3: Menú optiTrack Streaming Engine

no coincide con el sistema de referencia del motive, por lo tanto, motive permite realizar un ajuste en el eje vertical del sistema de referencia, esta configuración se encuentra en el menú: OptiTrack

5. D.1. INTRODUCCIÓN

Streaming Engine, damos clic en la opción: Up Axis y escogemos la configuración que mejor se ajuste a nuestra aplicación. La Figura D.4 muestra la configuración del eje vertical.

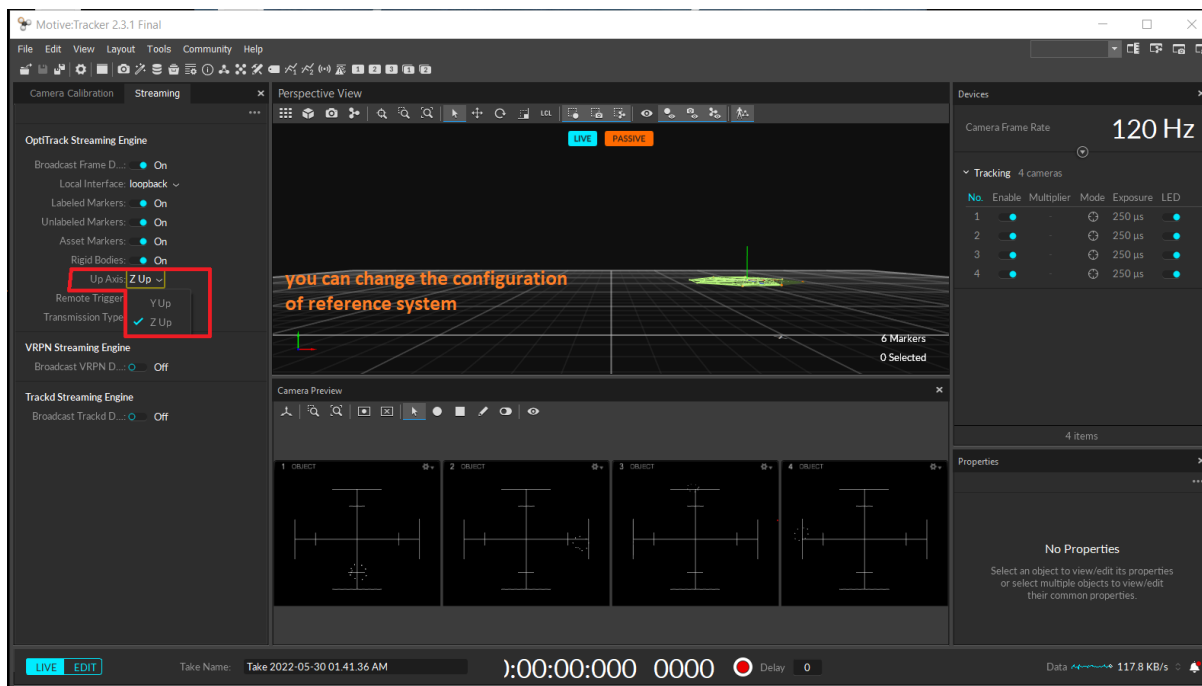


Figura D.4: Configuración del Up Axis

D.1.3. Ejemplo de comunicación entre MATLAB y Motive

El autor del toolbox de Optitrack para MATLAB, M. Kutzer ha desarrollado algunos ejemplos de comunicación entre ambos software MATLAB y Motive, estos ejemplos se encuentran alojados dentro de la carpeta OptiTrackToolbox Example SCRIPTS, puedes abrir y ejecutar el archivo SCRIPT_DisplayData.m file, es importante remarcar que el software motive se deberá de estar ejecutando y deberá tener seleccionado un cuerpo rígido para el seguimiento de su posición y orientación, como se muestra en la Figura D.5. La Figura D.6 muestra el archivo SCRIPT_DisplayData.m ejecutándose, la posición es dada en milímetros y la orientación haciendo uso de la parametrización de cuaterniones.

5. D.1. INTRODUCCIÓN

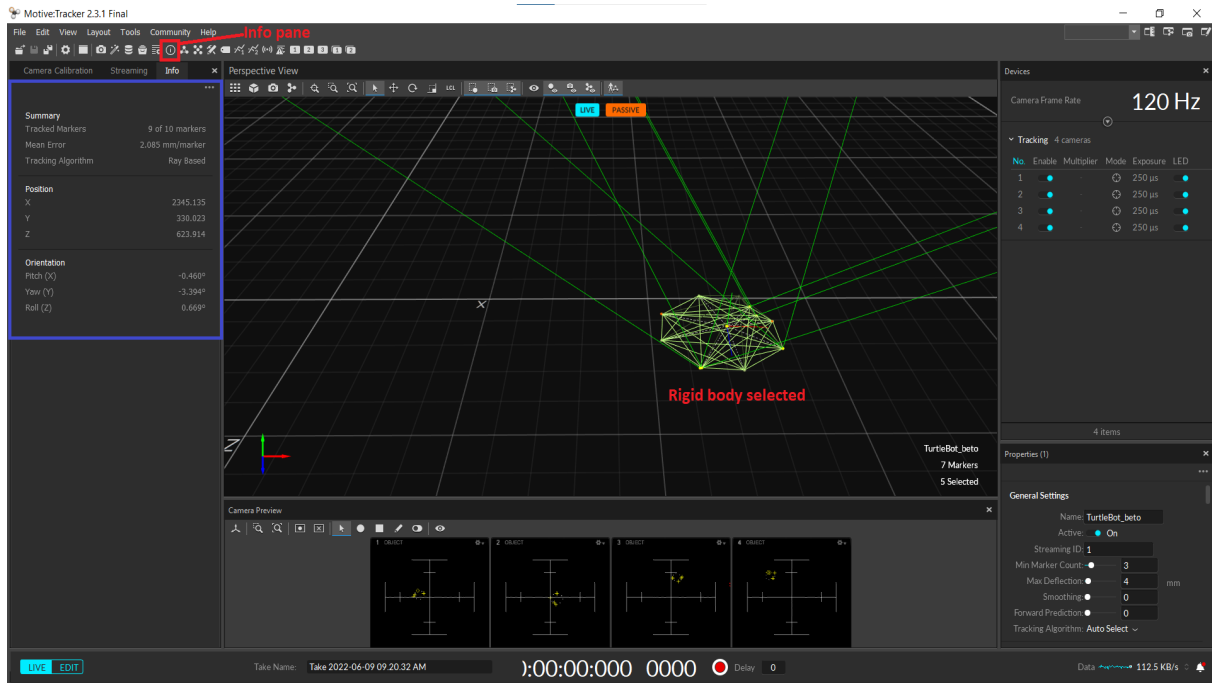


Figura D.5: Info pane para visualizar la información referente a posición y orientación del cuerpo rígido

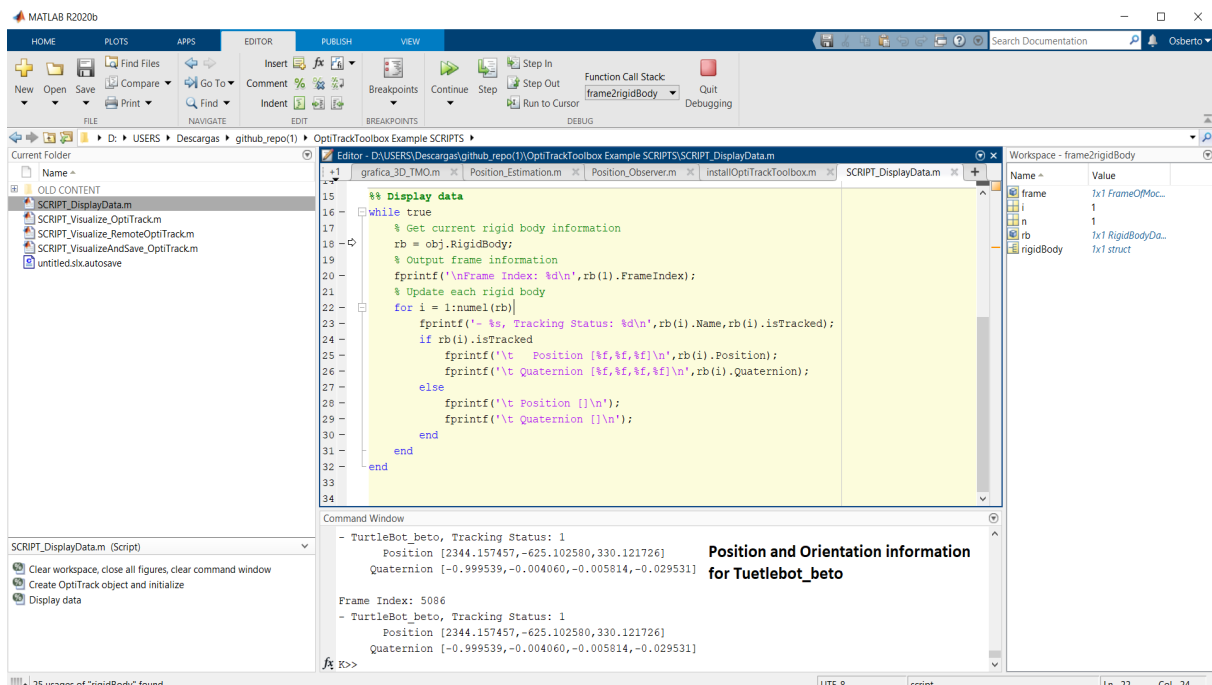


Figura D.6: SCRIPT_DisplayData.m file ejecutándose para adquirir la posición y orientación del cuerpo rígido en tiempo real

Apéndice E

Cronograma de Actividades

1. Revisión del estado del arte de las boyas marinas basadas en sistemas de navegación inercial.
2. Selección de la instrumentación electrónica: Procesador, sensores inerciales, sistema de comunicación.
3. Estudio de las Herramientas necesarias para el Desarrollo del Algoritmo de Fusión de Datos.
4. Desarrollo del Algoritmo de Fusión de Datos del AHRS utilizando MATLAB/Simulink.
5. Expansión del Algoritmo de Fusión de Datos para múltiples sensores inerciales y magnetómetro utilizando MATLAB/Simulink.
6. Desarrollo del Algoritmo de Fusión de Datos GPS-Sensores inerciales.
7. Desarrollo de una GUI para validar posición y orientación del sistema en base al Algoritmo de Fusión de Datos en MATLAB/Simulink.
8. Implementación del Algoritmo de Fusión de Datos en un microcontrolador.
9. Estancia.
10. Pruebas Experimentales del sistema de sensado de boya autónoma, trabajando en conjunto.
11. Elaboración del documento de tesis.

	2021						2021						2022					
	ENE	FEB	MAR	ABR	MAY	JUN	JUL	AGO	SEP	OCT	NOV	DIC	ENE	FEB	MAR	ABR	MAY	JUN
1	■	■	■	■	■	■												
2						■												
3				■	■	■	■											
4							■	■	■	■	■							
5										■	■	■						
6										■	■	■	■					
7												■	■	■				
8												■	■	■	■			
9															■	■	■	■
10															■	■	■	■
11			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

Tabla E.1: Tabla de actividades

Apéndice F

Tabla de avances

Objetivos	Porcentaje de la tesis	Porcentaje de avance del objetivo	Porcentaje de avance total
Estudio del estado del arte, de los sistemas de navegación y algoritmos de fusión de datos	15 %	100 %	15 %
Desarrollar un algoritmo de fusión de datos que permita estimar la orientación y posición de la boya, utilizando al menos 3 IMU (del inglés, Inertial Measurement Unit), 3 magnetómetros y receptor GPS.	40 %	100 %	40 %
Implementar el algoritmo de fusión de datos desarrollado, utilizando un sistema de cómputo embebido y los sensores correspondientes.	35 %	100 %	35 %
Validar el correcto funcionamiento de la boya, por medio de pruebas experimentales que permitan reconstruir la disposición espacial de la boya.	10 %	100 %	10 %
			Total: 100 %

Tabla F.1: Tabla de porcentaje de avance de tesis

Referencias

- [1] Antonelli, G.; Fossen, T. I. y Yoerger R., D., Underwater Robotics, Springer Handbook of Robotics; Siciliano Khatib Editors: **2008**, 987-1008.
- [2] Venkatesan, R.; Ramesh, K.; Kishor, A.; Vedachalam, N. y Atmanand, M. Best Practices for the Ocean Moored Observatories. *frontiers in Marine Science* **2018**, DOI: [10.3389/fmars.2018.00469](https://doi.org/10.3389/fmars.2018.00469).
- [3] Gattuso, J.-P.; Magnan, A.; Bopp, L. y col. Ocean Solutions to Address Climate Change and Its Effects on Marine Ecosystems. *frontiers in Marine Science* **2018**, DOI: [10.3389/fmars.2018.00337](https://doi.org/10.3389/fmars.2018.00337).
- [4] Posada-Restrepo, N. L.; Zuluaga-Toro, C. A.; Valencia-Cardona, R. A.; Correa-Rodríguez, J. C.; Flórez-Londoño, D. A. y Arias-Jaramillo, C. D. Instrumentation for the measurement of salinity, temperature and currents in oceans; Engineering education journal; Pontifical Bolivarian University, Medellin, Colombia, **2010**.
- [5] García, E.; Quiles, E.; Correcher, A. y Morant, F. Sensor Buoy System for Monitoring Renewable Marine Energy Resources. *Sensors* **2018**, DOI: [10.3390/s18040945](https://doi.org/10.3390/s18040945).
- [6] Ki-moon, B.; Touré, H. y Rancy, F. Radiocomunicaciones y cambio climático, Ginebra Suiza, **2012**.
- [7] Sendra, S.; Parra, L.; Lloret, J. y Jiménez, J. M. Oceanographic Multisensor Buoy Based on Low Cost Sensors for Posidonia Meadows Monitoring in Mediterranean Sea. *Hindawi* **2015**, DOI: [10.1155/2015/920168](https://doi.org/10.1155/2015/920168).
- [8] Zoss, B.; Mateo, D.; Kuan, Y.; Tokic, G.; Chamanbaz, M.; Goh, L.; Vallegra, F.; Bouffanals, R. y Yue, D. Distributed system of autonomous buoys for scalable deployment and monitoring of large waterbodies. *Springer Science* **2018**, DOI: [10.1007/s10514-018-9702-0](https://doi.org/10.1007/s10514-018-9702-0).
- [9] Moreno, A. M.; Saltarén, R.; Puglisi, L.; Carrera, I.; Cárdenas, P. y Álvarez, C. Robótica Submarina: Conceptos, Elementos, Modelado y Control. *ELSEVIER ScienceDirect* **2014**, DOI: [10.1016/j.riai.2013.11.001](https://doi.org/10.1016/j.riai.2013.11.001).
- [10] Kok, M.; Jeroen, D. H. y Schon, T. "Using Inertial Sensors for Position and Orientation Estimation", *Foundations and Trends in Signal Processing*. **2017**, *11*, 1-153.
- [11] Rico-Azagra, J.; Rico, R.; Maisterra, P. y Gil-Martínez Comparación de algoritmos de estimación de actitud; inf. téc., Jornadas de Automática; Grupo de Ingeniería de Control, Departamento de Ingeniería Eléctrica, Universidad de La Rioja, **2015**.
- [12] Rogne, R.; Bryne, T. H.; Fossen, T. y Johansen, T. A. On the Usage of Low-Cost MEMS Sensors, Strapdown Inertial Navigation and Nonlinear Estimation Techniques in Dynamic Positioning. *IEEE Journal of Oceanic Engineering* **2021**, *46*, DOI: [10.1109/JOE.2020.2967094](https://doi.org/10.1109/JOE.2020.2967094).

- [13] Ramos-G, J.; Guerero-Castellanos, J. F.; Gonzales-Díaz, V. R.; Linares-Flores, J.; López-López, M. y Durand, S. en *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, **2018**, 363-371.
- [14] Bryne, T. H.; Hansen, J. M.; Rogne, R. H.; Sokolova, N.; Fossen, T. I. y Johansen, T. A. Nonlinear Observers for Integrated INS/GNSS Navigation. *IEEE CONTROL SYSTEMS MAGAZINE* **2017**, DOI: [10.1109/MCS.2017.2674458](https://doi.org/10.1109/MCS.2017.2674458).
- [15] Guerrero-Castellanos, J.; Madrigal-Sastre, H.; Durand, S.; Torres, L. y Muñoz-Hernandez, G. A Robust Nonlinear Observer for Real-Time Attitude Estimation Using Low-Cost MEMS Inertial Sensors. *Sensors (Basel, Switzerland)* **2013**, *13*, 15138-58.
- [16] Ludwig, S. A. Investigation of Orientation Estimation of Multiple IMUs; inf. téc.; Fargo, ND, USA: Department of Computer Science, North Dakota State University, **2020**.
- [17] Zhao, L.; Yu, F.; Hou, J.; Wang, P. y Fan, T. The role of tsunami buoy played in tsunami warning and its application in South China Sea. *The Chinese Society of Theoretical and Applied Mechanics* **2013**, National Marine Environmental Forecasting Center (NMEFC), Key Laboratory of Research on Marine Hazard Forecasting (LoMF), DOI: [10.1063/2.1303202](https://doi.org/10.1063/2.1303202).
- [18] Trevathan, J. y Johnstone, J. Smart Environmental Monitoring and Assessment Technologies (SEMAT)—A New Paradigm for Low-Cost, Remote Aquatic Environmental Monitoring. *Sensors* **2018**, DOI: [10.3390/s18072248](https://doi.org/10.3390/s18072248).
- [19] España, M., Sistemas de Navegación Integrada con Aplicaciones, 2ª Edición; CONAE, Comisión Nacional de las Actividades Espaciales: **2019**.
- [20] Skog, I. y Händel, P. In-Car Positioning and Navigation Technologies—A Survey. *Intelligent Transportation Systems, IEEE Transactions on* **2009**, *10*, 4-21.
- [21] Groves, P. D., Principles of GNSS, Inertial, and Multisensor Integrated Navigation System, 2ª Edición; Artech House: **2008**.
- [22] Luna-Torres, J., Notas de clases, Modelos algebraicos para la estabilización de micro-satélites; Universidad Industrial de Santander, **2018**.
- [23] Farrell, A. J., AIDED NAVIGATION GPS with High Rate Sensors; Mc Graw Hill: **2008**.
- [24] Skog, I. A Low-Cost GPS Aided Inertial Navigation System for Vehicular Applications, Tesis de mtría., Stockholm, Sweden: KTH Royal Institute of Technology, **2005**.
- [25] Fossen, I. T., HANDBOOK OF MARINE CRAFT HYDRODYNAMICS AND MOTION CONTROL, 1ª Edition; John Wiley & Sons, Inc.: **2011**.
- [26] Grewal, M. S.; Weill, L. R. y Andrews, A. P., Global Positioning Systems, Inertial Navigation, and Integration, 2ª Edition; John Wiley & Sons, Inc.: **2007**.
- [27] Perez, T. y Fossen, T. I. Kinematics and Kinetics of Marine Vessels (module 3), Centre for Complex Dynamic System and Control, The University of Newcastle Australia and Department of Engineering Cybernetics NTNU Det skapende universitet, **2007**.
- [28] Nonami, K.; Kendoul, F.; Suzuki, S.; Wang, W. y Nakazawa, D., *Autonomous Flying Robots*; Springer: **2010**.
- [29] Hua, M.-D.; Ducard, G.; Hamel, T. y Mahony, R. Introduction to Nonlinear Attitude Estimation for Aerial Robotic Systems. *Journal AerospaceLab* **2014**, DOI: [10.12762/2014.AL08-04](https://doi.org/10.12762/2014.AL08-04).

- [30] Cova, W. J. D. Cuaterniones visualización de rotaciones de cuerpos rígidos; inf. téc., REUN, Red de editoriales de Universidades Nacionales de la Argentina; Universidad Tecnológica Nacional, Facultad Regional Cordova, **2019**.
- [31] Leonard, J. y Durrant Whyte, H. Mobile Robot Localization by Tracking Geometric Beacons. *IEEE Transaction on Robotics and Automation* **1991**, 376-382.
- [32] Guerrero-Castellanos, J.; Munoz-Hernandez, G.; Graciós, C. y Salmeron, B. Design and Implementation of Low-Cost Attitude Heading References Systems for Micro Aerial Vehicles. **2016**.
- [33] Quesada-Navarro, R. Acondicionamiento de las mediciones de sensores inerciales de bajo costo con fines de navegación; inf. téc.; Santa Clara: Universidad Central "Martha Abreu" de Las Villas, Facultad de Ingeniería Eléctrica, Departamento de Automática y Sistemas Computacionales, **2014**.
- [34] Dulhoste, J. F., Teoría de control; Escuela de Ingeniería Mecánica. Facultad de Ingeniería. Universidad de los Andes. Mérida-Venezuela.
- [35] Universidad Nacional de Quilmes Programa CAUT2," Newcastle.edu.au, 2021.<https://www.eng.newcastle.edu.au/~jhb519/teaching/caut2/Programa.html> (accessed Jun. 23, 2021). **2001**.
- [36] Rhudy, M. B.; Salguero, R. A. y Keaton, H. A Kalman Filtering Tutorial For Undergraduate Students. *International Journal of Computer Science & Engineering Survey (IJCSSES)* **2017**, DOI: 10.5121/ijcses.2017.8101.
- [37] Bryne, T. H.; Hansen, J. M.; Rogne, R. H.; Sokolova, N.; Fossen, T. I. y Johansen, T. A. Nonlinear Observers for Integrated INS/GNSS Navigation Implementation Aspects. *IEEE* **2016**, DOI: 10.1109/MCS.2017.2674458.
- [38] Bancroft, B. J. y Lachapelle, G. Data Fusion Algorithms for Multiple Inertial Measurement Units. *Sensors* **2011**, DOI: 10.3390/s110706771.
- [39] Sørensen, J. A. Marine Control Systems Propulsion and Motion Control of Ships and Ocean Structures; inf. téc.; **2013**.
- [40] Jafari, M. Optimal redundant sensor configuration for accuracy increasing in space inertial navigation system. *Elsevier* **2015**, DOI: 10.1016/j.ast.2015.09.017.
- [41] Rhudy, M.; Gross, J.; Gu, Y. y Napolitano, R. M. en American Institute of Aeronautics, Astronautics, Guidance, Navigation y Control Conference: **2012**.
- [42] Rogne, R. H.; Bryne, T. H.; Fossen, T. I. y Johansen, T. A. en Department of Engineering Cybernetics, Centre for Autonomous Marine Operations and Systems (NTNU AMOS), Norwegian University of Science and Technology.
- [43] Narkhede, P.; Poddar, S.; Walambe, R.; Ghinea, G. y Kotecha, K. Cascaded Complementary Filter Architecture for Sensor Fusion in Attitude Estimation. *Sensors* **2021**, DOI: 10.3390/s21061937.
- [44] Bingfei, F.; Qingguo, L. y Tao, L. How Magnetic Disturbance Influences the Attitude and Heading in Magnetic and Inertial Sensor-Based Orientation Estimation. *Sensors* **2017**, DOI: 10.3390/s18010076.
- [45] Munoz-Diaz, E.; de-Ponte-Müller, F.; R.-Jiménez, A. y Zampella, F. Evaluation of AHRS Algorithms for Inertial Personal Localization in Industrial Environments. *IEEE* **2015**, DOI: 10.1109/ICIT.2015.7125605.

- [46] López-Montes, D. Fusión sensorial para la estimación del estado de un vehículo autónomo, Tesis de mtría., Escuela Técnica Superior de Ingenieros Industriales y de Telecomunicación, Universidad de Cantabria, **2016**.
- [47] Olfati-Saber, R. y Shamma, J. en *44th IEEE Conference on Decision and Control*, **2006**; *2005*, 6698-6703.
- [48] Guerrero-Castellanos, J.; Durand, S.; Muñoz-Hernández, G.; Marchand, N.; González-Romeo, L.; Linares-Flores, J.; Mino-Aguilar, G. y Guerrero-Sánchez., W. Bounded Attitude Control with Active Disturbance Rejection Capabilities for Multirotor UAVs. *Applied Sciences* **2021**, *11*, 5960.
- [49] Henry, R., Ecuaciones diferenciales: una introducción moderna; Reverté: **2008**.