



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
INGENIERÍA ELÉCTRICA – INSTRUMENTACIÓN

DESARROLLO E INTEGRACIÓN DE INSTRUMENTACIÓN MECATRÓNICA
PARA VALIDAR EXPERIMENTALMENTE ARQUITECTURAS
RECONFIGURABLES DE CONTROL DE ORIENTACIÓN DE SATÉLITES
PEQUEÑOS

TESIS
QUE PARA OPTAR POR EL GRADO DE:
DOCTOR EN INGENIERÍA

PRESENTA:
MARIO ALBERTO MENDOZA BÁRCENAS

TUTOR (ES) PRINCIPAL(ES)
DR. ESAÚ VICENTE VIVAS, INSTITUTO DE INGENIERÍA, UNAM
COMITÉ TUTOR
DRA. GRACIELA VELASCO HERRERA, CCADET, UNAM
DR. HUGO RODRIGUEZ CORTÉS, CINVESTAV, IPN

MÉXICO, D. F. ABRIL, 2014.

JURADO ASIGNADO:

Presidente: DRA. TETYANA BAYDYK
Secretario: DR. JORGE PRADO MOLINA
Vocal: DR. ESAÚ VICENTE VIVAS
1^{er.} Suplente: DR. HUGO RODRÍGUEZ CORTÉS
2^{do.} Suplente: DR. LEONID FRIDMAN

Lugar o lugares donde se realizó la tesis:

- COORDINACIÓN DE ELÉCTRICA Y COMPUTACIÓN, INSTITUTO DE INGENIERÍA, UNAM
- LABORATORIO DE SISTEMAS INTELIGENTES, CCADET, UNAM
- SECCIÓN DE MECATRÓNICA, CINVESTAV, ZACATENCO, IPN

TUTOR DE TESIS:

DR. ESAÚ VICENTE VIVAS



FIRMA

“No importa si has fallado o si quedaste golpeado.

Lo que importa es que aprendas algo, te levantes y vuelvas a intentarlo.

Ganar es una buena sensación, pero ganar cuando nadie pensaba que podías lograrlo es una sensación... i n c r e í b l e”.

Anónimo.

Dedicatorias y agradecimientos

Antes que a nadie... gracias a Dios, que me ha permitido llegar hasta acá, con todo y aquello que he tenido que pasar para lograrlo. Fue una experiencia compleja, pero muy nutritiva...

*Esta tesis la dedico a los principales motores que me impulsan cada día para seguir adelante: **mi mamá** y el núcleo más querido e importante de mi familia (**Abuela Concepción, Marianita Christian “Chaparro”, Carlos, Sandra María, Tío Luis, Tía Gretell, Lulú, Aldo, Tía Salva y a mi hija/sobrina Chispa**). A mis padrinos **Silvestre y Juana** que estuvieron siempre al pendiente de esta fecha tan especial...*

*Mención especial y mi más profundo agradecimiento a dos hermosas personas que me ayudaron a que este proceso se concretara: **Lic. Susana Ramírez Romero y Dra. Teresa Ruíz**.*

*A todos los amigos que siempre tuvieron una palabra de aliento y muestras de apoyo: **Jafet y Aminadab (Core), Rodrigo Córdova, Paloma Pedrajas, Enkorcito, Miguel Angel Alvarado, Moriz B. (Mario), Juan Carlos Moctezuma e Isabel. A M. Gaytán, J.F. Pliego y Clara Romero de la AEDT-EETA**, porque me enseñaron que la vida es como una pista con muchos verticales y oxers, y retos como éste, siempre se tienen que afrontar viendo del otro lado del obstáculo...*

*Mi más profundo agradecimiento a la **doctora Graciela Velasco (CCADET)**, parte fundamental de éste logro alcanzado. Gracias doctora por todo su apoyo... ¡un abrazo!*

*Fundamental para el desarrollo de este trabajo, por su impulso y apoyo (moral y técnico), aún en los peores momentos, muchas, muchas gracias al **doctor Miguel Arias Estrada (INAOE)**.*

*Gracias también al maestro **Rafael Prieto Meléndez (CCADET)** por su gran apoyo en la parte inicial de esta investigación, sobre todo por presentarme a los FPGAs...*

*A mi equipo de trabajo en el IINGEN: **Ing. Eduardo Vizcaíno, Ing. José Francisco Osorio, Ing. Miguel Ángel Alvarado, Ing. Alejandro Castilla Gallardo y Alberto Muñoz “Betito”**. Gracias amigos, porque este pequeño logro también es de ustedes...*

*Al resto de mis compañeros y amigos en el IINGEN, particularmente al **M.I. Emilio Jiménez y al M.I. Rodrigo Alva Gallegos**.*

*A mi tutor, **Dr. Esaú Vicente**, muchas gracias por todas las oportunidades de crecimiento.*

*Gracias a los integrantes de mi comité tutorial: **Dr. Hugo Rodríguez y Doctora Graciela Velasco**, por todo su seguimiento, comentarios y enseñanzas durante mis estudios doctorales.*

*Muchas gracias al **doctor Jorge Prado**. El detalle en sus comentarios y lecciones depuraron este trabajo y lo hicieron mejor.*

*Gracias a mis sinodales: **Dr. Leonid Fridman, Dra. Tetyana Baydyk, Dr. Hugo Rodríguez, Dr. Jorge Prado** por sus comentarios y observaciones sobre este trabajo de investigación.*

*Gracias a la **Coordinación de estudios de Posgrado de la UNAM** por la beca otorgada durante mis estudios de doctorado. Sin duda, fue elemental para el desarrollo de este trabajo de investigación doctoral.*

CONTENIDO

Resumen

Capítulo 1. Introducción	1
1.1. Generalidades	1
1.2. Objetivos y metas de la investigación doctoral	2
1.3. Metodología	4
1.4. Productos académicos generados	5
1.5. Planteamiento del problema.....	7
1.5.1. Computadora a bordo basada en FPGA	9
1.5.2. Esquema de cosimulación HIL como base del simulador del Subsistema de Determinación y Control de Orientación	10
1.5.3. Operación del simulador MSA.....	11
1.5.4. Desarrollo de herramientas de validación en hardware y software	12
1.6. Organización de la tesis	13

Capítulo 2. Estado del arte en Sistemas de Control de Orientación satelital

y Simuladores.....	15
2.1. El Subsistema de Control de Orientación en satélites.....	15
2.2 Técnicas de simulación para validación de pruebas de control de Orientación satelital.....	19
2.2.1. Tipos de plataformas de simulación para pruebas de validación de control de orientación basadas en sistemas de cojinete de aire	23
2.2.2. Torques de perturbación en las Plataformas de Simulación Satelital de Vuelo Orbital	26
2.2.3. Casos destacados de Plataformas de simulación para pruebas de validación de control de orientación en el mundo.....	27
2.2.3.1. Plataforma de simulación de la Universidad Yonsei, Corea del Sur.....	28
2.2.3.2. Plataforma de simulación de la Escuela de Posgraduados Navales, Monterey, California	29
2.2.3.3. Plataforma de simulación del Instituto de Tecnología de la Fuerza Aérea de los E.E.U.U.	30
2.2.3.4. Plataforma de simulación BIRD de Astro-Feinwerktechnik, Alemania	31
2.2.3.5. Plataforma de simulación del Instituto de Geografía, UNAM	32
2.2.3.6. Plataforma de simulación del IINGEN-UNAM, México.....	34

Capítulo 3. Modelado dinámico de la plataforma de simulación MSA	38
3.1. Conceptos matemáticos	38
3.1.1. Concepto de cuerpo rígido	38
3.1.2. Representación de la orientación de un cuerpo rígido	39
3.1.3. Sistema de referencia y matriz de rotación.....	40
3.1.4. Utilización de ángulos de Euler para la orientación de un cuerpo rígido.....	42
3.1.5. Utilización de cuaterniones para la representación de la de la orientación de un cuerpo rígido	43
3.1.6. Energía cinética rotacional.....	44
3.1.7. Momento de inercia	44
3.1.8. Torque.....	45
3.1.9. Relación entre torque y velocidad angular	45
3.2. Descripción general de la plataforma de simulación MSA.....	47
3.3. Modelo dinámico de la MSA.....	48
3.3.1. Análisis de la cinemática y dinámica de la MSA.....	49
3.3.2. Cinemática de la MSA.....	51
3.3.3. Dinámica de la MSA.....	54
3.3.4. Tensores de inercia de la MSA.....	57
 Capítulo 4. Instrumentación de la MSA.....	 60
4.1. Introducción.....	60
4.2. Esquema de instrumentación de la MSA.....	62
4.3. Hardware de la MSA.....	65
4.3.1. Sensores de navegación inercial	65
4.3.2. Actuadores.....	73
4.3.3. Computadora de a bordo.....	76
4.3.3.1. Sistema mínimo (SMIN).....	78
4.3.4. Drivers y potencia	83
4.3.5. Comunicaciones	85
4.4. Software de la MSA.....	85
4.4.1. Software de la Plataforma Suspendida en aire	85
4.4.1.1 Sistemas Embebidos en FPGA	85
4.4.1.2. Desarrollo de Sistemas Embebidos basados en	

herramientas Xilinx®	88
4.4.1.3. Codiseño hardware/software para la integración de Sistemas Embebidos en FPGA	90
4.4.2. Sistema Embebido desarrollado para la MSA.....	93
4.4.3. Diseño del núcleo IP PWM	95
4.5. Software de la PC.....	97
4.5.1. Función principal	98
4.5.2. Función TRIAD	99
4.5.3. Función Filtro de Kalman Extendido discreto (EKF).....	101
4.5.4. Función CONTROL.....	104
 Capítulo 5. Experimentos de seguimiento de maniobras de la MSA	109
5.1. Introducción.....	109
5.2. Determinación de la orientación de un vehículo espacial.....	109
5.3. Seguimiento de maniobras del modelo en realidad virtual de la MSA	110
5.3.1. Descripción de elementos del seguimiento de la MSA.....	111
5.4. Detalles del SMSA desarrollado en Simulink.....	113
5.5. Bloque personalizado de comunicaciones seriales con <i>buffer</i> de recepción de longitud variable	115
5.6. Bloque personalizado para el algoritmo de determinación de la orientación	117
5.6.1. Transferencia de la lógica del TRIAD a un bloque personalizado de Simulink.....	117
5.6.2. Bloque de acondicionamiento: matriz de rotación a VRML	119
5.7. Descripción de la prueba de seguimiento virtual de la MSA.....	119
5.7.1. Posición inicial para el inicio de pruebas de seguimiento de la MSA.....	120
5.7.2. Maniobras en torno al eje X.....	120
5.7.3. Maniobras en torno al eje Y.....	121
5.7.4. Maniobras en torno al eje Z.....	122
5.8. Discusión de resultados.....	123
 Capítulo 6. Validación experimental de la MSA	126
6.1. Plataforma de simulación satelital de vuelo orbital MSA.....	126
6.2. Inicialización de la MSA.....	127
6.3. Pruebas de validación de algoritmos de control con la MSA	129

6.3.1. Definición de condiciones iniciales en los algoritmos del esquema de control.....	129
6.4. Experimentos de validación del control de orientación de la MSA por cada eje.....	131
6.5. Experimentos de validación del control de orientación de la MSA en tres ejes	133
6.6. Discusión de resultados de los experimentos de control de orientación con la MSA.....	139
Capítulo 7. Conclusiones y trabajo futuro.....	142
7.1. Conclusiones	142
7.2. Trabajo futuro.....	144
Referencias	146
Apéndice A. Software de operaciones del FPGA	152
Apéndice B. Software de operaciones en Matlab®	160

Resumen

En esta investigación se presenta el desarrollo e integración de una plataforma de simulación (software y hardware), para pruebas de validación experimental de esquemas de control de orientación satelital en tierra, denominada mesa suspendida en aire (MSA).

El desarrollo de la MSA se da en dos vertientes. Por un lado, el análisis teórico basado en el estudio de la cinemática y dinámica de la plataforma del simulador, considerándola como un grupo de cuerpos rígidos, unidos entre sí, por medio del cual se obtiene su ecuación de movimiento para validarla por medio de modelos y simulaciones numéricas, cuyos resultados son visualizados por medio de gráficas en 2D y 3D. Por otro lado, en términos de implementación física, considera la integración y desarrollo de elementos de su esquema de instrumentación, compuesto por sensores de navegación inercial (magnetómetros, acelerómetros y giróscopos), así como el diseño y construcción de actuadores activos basados en ruedas inerciales y de una computadora a bordo, la cual está integrada en una tarjeta electrónica denominada SMIN, basada en un dispositivo reconfigurable FPGA con capacidad de reconfigurarse de forma remota por medio de un microcontrolador.

El desarrollo del esquema de instrumentación tomó como base la estructura de la técnica de cosimulación denominada *hardware-in-the-loop*, por medio de la cual se aceleran resultados experimentales de implementación de un esquema de control de orientación. Al mismo tiempo que se evalúa el desempeño del hardware a bordo de la plataforma de la MSA, se depura y optimiza el software de operaciones que contiene a los algoritmos de determinación y control de orientación satelital.

El uso de una plataforma FPGA como base de la computadora a bordo, obedece principalmente al paradigma de desarrollo del dispositivo, particularmente en lo que se refiere a la flexibilidad para integrar arquitecturas computacionales polimórficas donde se programaron los algoritmos de control de orientación de la MSA.

Adicionalmente, se presentan resultados exitosos de un esquema de validación de la instrumentación de la MSA por medio de experimentos de seguimiento de maniobras manuales en la plataforma del simulador, visualizados por medio de un modelo en realidad virtual. También se presentan gráficas resultantes de pruebas experimentales de control de orientación conduciendo la MSA hacia objetivos definidos, utilizando un esquema de control integrado por algoritmos de

determinación de la orientación (TRIAD), estimación y filtrado (EKF discreto) y una ley de control lineal basado en un controlador proporcional-derivativo.

Capítulo 1

Introducción

1.1. Generalidades

La función del subsistema de determinación y control de la orientación (en inglés *Attitude Determination and Control System* o simplemente SDCO) de un satélite, es estabilizarlo y orientarlo hacia una posición deseada durante la misión en órbita ante la presencia de fuerzas perturbadoras actuando sobre él. Este procedimiento requiere que el vehículo espacial determine su orientación, expresada por medio de sus ejes de referencia locales, respecto a un marco de referencia inercial, generalmente fijo en el centro de la Tierra, usando para ello sensores y controlándolo por medio de actuadores. El SDCO frecuentemente interactúa con otros subsistemas de la misma plataforma satelital, especialmente con el subsistema de propulsión y con los sensores de navegación inercial [1].

El diseño del SDCO implica el manejo de conceptos físicos asociados como las propiedades de masa, torques de perturbación, torques de control, momento angular y sistemas coordenados de referencia. Particularmente las propiedades de masa del vehículo espacial, son un factor clave para el dimensionamiento del esquema de control y la determinación de los torques que afectarán su orientación. Típicamente se requiere conocer la localización de su centro de masa así como los elementos de su tensor de inercia [1].

Un cuerpo en el espacio está sujeto a una serie de pequeños pero persistentes torques de perturbación, agrupados en cuatro principales fuentes: gradiente gravitacional, radiación solar, campo magnético terrestre y aerodinámico. Éstos desvían el cuerpo del satélite respecto a su orientación deseada, a menos que exista algún medio que lo impida. El SDCO tiene la función de compensar dichas perturbaciones ya sea de forma pasiva, es decir, utilizando los efectos de los pares de fuerza naturales para mantener la estabilidad requerida del satélite o bien, de forma activa, por medio de la medición del movimiento angular satélite y aplicando torques correctores por medio de computadoras y actuadores que requieren energía eléctrica [1] y [2].

El SDCO es crítico para el desarrollo de satélites que puedan ofrecer productos y servicios para resolver grandes problemas de impacto social y económico asociados a desastres naturales (sismos,

inundaciones, huracanes, sequías, etc.), administración de recursos naturales (agricultura, agua, pesca, bosques, etc.), seguridad y comunicaciones (video, voz y datos).

Uno de los principales retos en el desarrollo del SDCO es la validación y verificación de maniobras de control de orientación satelital en Tierra. En la literatura reportada en el tema [3] se plantean diferentes vertientes metodológicas para el desarrollo de este sistema, siendo dos de ellas las más utilizadas. La primera emplea simulaciones numéricas y modelos matemáticos, los cuales se ejecutan en software comercial de análisis matemático; en estos casos, dichos modelos describen la órbita de vuelo, la dinámica y cinemática del satélite así como el funcionamiento de sensores de navegación inercial (giróscopos, magnetómetros, sensores finos de sol, etc.), actuadores (bobinas de torque magnético, ruedas inerciales, propulsores, etc.) y leyes de control. De igual forma usan modelos para emular numéricamente el campo magnético terrestre y algunas perturbaciones ambientales. Posteriormente, estos estudios permiten visualizar los resultados ya sea por medio de gráficas y/o modelos de realidad virtual, [4] y [5].

Otra de las vertientes utilizadas para el desarrollo del SDCO es mediante el uso de plataformas físicas de simulación, las cuales emulan en Tierra las condiciones del medio espacial, particularmente la ausencia de fricción y, debidamente instrumentadas, permiten la verificación física de maniobras de control de orientación satelital en vuelo orbital [6], [7] y [8].

La presente investigación doctoral se enfocó en la segunda vertiente de tecnología espacial (relativa al desarrollo de plataformas de simulación) por medio del desarrollo de un sistema de simulación en Tierra al cual se denominó Mesa Suspendida en Aire (MSA).

La MSA es una herramienta de instrumentación para la evaluación y desarrollo de hardware de orientación de vehículos espaciales, que por sus características de flexibilidad y escalabilidad, permite el diseño de estrategias de control de orientación satelital. Facilita además la integración e interacción operativa de elementos en hardware y software (sensores de navegación inercial, plataforma de cómputo a bordo, sistemas de comunicaciones y actuadores), al mismo tiempo que permite evaluar su desempeño en conjunto, con el fin de definir el esquema de instrumentación definitivo más adecuado a los requerimientos del SDCO.

1.2. Objetivos y metas de la investigación doctoral

Los objetivos que persigue esta investigación son:

1. Desarrollar e integrar un esquema de instrumentación mecatrónica para una plataforma de simulación para pruebas de control de orientación con tres grados de libertad, que permita implementar en hardware la lógica de algoritmos del SDCO de un satélite pequeño y verificar experimentalmente en laboratorio terrestre la ejecución de maniobras de control de orientación satelital.
2. Desarrollar el modelo matemático que describe la dinámica del simulador MSA.
3. Transferir a hardware polimórfico los algoritmos de control de orientación satelital del SDCO.
4. Desarrollar y validar el diseño de una computadora a bordo (de tamaño y potencia reducida) que incorpora tecnología FPGA (*Field Programmable Gate Array*) para implementar la lógica del SDCO a bordo del simulador satelital.

Respecto de las metas, se plantean las siguientes:

- a) Diseñar el esquema de instrumentación del simulador, integrado a partir de la técnica de cosimulación *hardware-in-the-loop* (HIL), para acelerar la obtención de resultados, evaluar hardware de orientación (sensores, plataforma de cómputo y actuadores) y optimizar el software de operaciones.
- b) Seleccionar sensores de navegación inercial para la instrumentación de la plataforma de simulación de vuelo orbital, con base en el desempeño, costo, dimensiones, consumo de potencia, protocolo de comunicaciones y disponibilidad comercial.
- c) Emplear una tarjeta comercial de desarrollo FPGA como plataforma para la integración y validación experimental de arquitecturas digitales y sistemas embebidos, orientados a la implementación de esquemas de control de orientación satelital.

d) Diseñar e integrar actuadores activos basados en ruedas inerciales, los cuales permitirán la realización de maniobras de control de orientación de la plataforma de simulación.

e) Realizar un análisis sobre la implementación en hardware de los algoritmos de determinación y control de orientación satelital, en términos de diagramas de bloques de sistemas interconectados, con énfasis en las operaciones aritméticas requeridas en los procesos internos de cada algoritmo.

f) Diseñar y validar con simulaciones y pruebas experimentales núcleos periféricos de propósito general y de coprocesamiento, desarrollados en lenguaje descriptor de hardware (HDL).

g) Diseñar y validar una plataforma de cómputo a la medida basada en un FPGA, para integrar arquitecturas reconfigurables y polimórficas que resuelvan algoritmos de control de orientación satelital.

h) Desarrollar en Matlab® un sistema de funciones personalizadas que contengan la lógica de los algoritmos de procesamiento de datos para la determinación (TRIAD), estimación (EKF) y control de la orientación. Dichas funciones se implementarán en una PC externa, la cual se enlazará inalámbricamente con la plataforma móvil del simulador para interactuar con la instrumentación a bordo.

i) Realizar pruebas experimentales de control de orientación definiendo diferentes puntos objetivo en la periferia de la plataforma móvil del simulador satelital para pruebas con 3 grados de libertad.

j) Desarrollar el modelo matemático de la dinámica de la plataforma de simulación para entender el comportamiento físico del sistema en operación, a partir de la conceptualización de la plataforma de simulador como un cuerpo rígido.

k) Proponer estrategias de control de orientación y medios alternativos de validación experimental basados en modelos de simulación numérica y visualización por medio de entornos en realidad virtual.

1.3. Metodología

La metodología que se empleó en esta tesis doctoral fue la siguiente:

- Se realizó una revisión bibliográfica de los temas principales asociados: algoritmos de determinación y control de orientación satelital, desarrollo de arquitecturas digitales, sistemas embebidos en FPGA y medios de simulación y validación de control de orientación en tierra.

- Se realizó una búsqueda de información sobre herramientas de desarrollo tecnológico en hardware y software, lo cual apoyó en la selección de la instrumentación de la plataforma de simulación (sensores de navegación inercial y actuadores). Esta búsqueda ayudó a definir el ambiente integrado de desarrollo (software) para el diseño y simulación de arquitecturas digitales e integración de sistemas embebidos en FPGAs, así como a seleccionar la plataforma hardware (tarjeta de desarrollo) para validación experimental.

- Se desarrollaron arquitecturas digitales utilizando lenguaje descriptor de hardware (HDL) que llevaron a la integración de un núcleo periférico en la computadora a bordo para generar señales moduladas en ancho de pulso (PWM), que es fundamental para el control de los actuadores de la plataforma de simulación.

- Se propusieron diseños de actuadores basados en ruedas inerciales. Se manufacturaron las ruedas inerciales, se seleccionaron motores de DC, se acoplaron mecánicamente a las ruedas y se caracterizó su operación para conocer su desempeño real durante las pruebas de control de orientación.

- Con base en los puntos anteriores, se definió un primer esquema de instrumentación para el control de orientación en un eje de la plataforma de simulación satelital.

- Basados en las pruebas de control de orientación en un eje, se ajustó el esquema inicial de instrumentación para considerar pruebas en tres ejes (X, Y, Z). Se optó por el uso de la técnica de cosimulación *hardware-in-the-loop* (HIL). Esta permitió la validación rápida del siguiente esquema en tres ejes y apoyó la evaluación de desempeño de elementos de la instrumentación, particularmente sensores de navegación inercial y actuadores.

- Con el fin de realizar un estudio analítico de las maniobras ejecutadas por el simulador y validar parcialmente el esquema de instrumentación integrado (hardware y software), se implementó un experimento de seguimiento por medio de un modelo virtual.

- Con base en los resultados obtenidos de la prueba de seguimiento virtual del simulador satelital, se realizaron experimentos de control de orientación. Se definió un esquema de control que consideró la integración de algoritmos de determinación de la orientación, estimación y filtrado así como el uso de una ley de control lineal.

1.4. Productos académicos generados

1. Memorias en congresos internacionales

2012

E.Vicente-Vivas, M.A.Mendoza-Bárceñas, et.al. “Ground Validation of 3-axis Stabilization and Attitude Control Algorithms for Small Satellites”. European Cubesat Symposium. Von Karman Institute for Fluid Dynamics. Bruselas, Bélgica.

M.A.Mendoza-Bárceñas, E.Vizcaíno-Torres, et.al. “Embedded attitude control system for the educative satellite SATEDU”. Congreso Internacional de Electrónica y Computación (CONIELECOMP) IEEE. Puebla, México.

Artículo citado en:

Malik, M.S.I., Asghar, S. “Inverse free steering law for small satellite attitude control law power tracking with VSCMGs” Advances in Space Research (Elsevier). Volume 53, issue 1, year 2014, pp. 97-109.

2. Memorias en congresos nacionales

2010

M.A.Mendoza-Bárceñas, et.al. “Development and preliminary results of one-axis FPGA based Attitude Control System”. Primer Workshop Mexicano de Cómputo Reconfigurable y sus Aplicaciones en Educación e Ingeniería. Universidad Politécnica de Victoria e INAOE.

2011

M.A.Mendoza-Bárceñas, R.Alva-Gallegos. “Diseño de ruedas inerciales para una plataforma satelital”. 1er Congreso de Ciencia y Tecnología Aeroespacial. Sociedad Mexicana de Ciencia y Tecnología Aeroespacial, A.C. Cholula, Puebla, México.

3. Publicación de artículo en revista indizada

M.A.Mendoza-Bárceñas, E.Vicente-Vivas, H.Rodríguez-Cortés. “Mechatronic design, dynamic modeling and results of a satellite flight simulator for experimental validation of satellite attitude determination and control schemes in 3-axis”. Journal of Applied Research and Technology. Sometida el 30 de Julio de 2013. Aceptada para publicación el 5 de noviembre de 2013.

4. Desarrollo tecnológico

- * Desarrollo e integración de la instrumentación para un simulador de pruebas de control de orientación de satélites basado en una plataforma suspendida en aire.
- * Diseño y desarrollo de una computadora a bordo basada en un FPGA, que integra un módulo de reconfiguración remota a bordo.
- * Integración de un sistema de seguimiento y simulación de movimientos durante maniobras de control de orientación satelital con la plataforma suspendida en aire, por medio de modelos desarrollados en realidad virtual.

5. Desarrollo de software

- * Desarrollo de un sistema de funciones personalizadas en MATLAB para implementar el esquema de determinación y control de orientación en tres ejes para satélites pequeños.
- * Desarrollo de un núcleo periférico personalizado para generación de señales PWM e integración de sistemas embebidos en FPGA.

1.5. Planteamiento del problema

El desarrollo e implementación física del subsistema satelital SDCO en un ambiente de laboratorio en tierra es un reto en términos técnicos y económicos, considerando que debe emular alguna característica del medio espacial (microgravedad, fricción cero), para validar operativamente el desempeño del SDCO.

En el IINGEN-UNAM se han alcanzado experiencias previas a esta tesis, en torno a la simulación computarizada de esquemas de control de orientación [5] en tres ejes de satélites pequeños. Dichas simulaciones incluyen el modelado y la descripción matemática de sensores de navegación inercial, actuadores activos basados en ruedas inerciales y bobinas de par magnético, algoritmos de determinación y estimación de la orientación, así como leyes de control y recursos que permiten también visualizar en tiempo real las maniobras de control satelital por medio de animaciones digitales de alta resolución realizadas en entornos de realidad virtual [9].

Por otro lado, el desarrollo e integración de un simulador satelital experimental que permita validar esquemas de control de orientación en Tierra para un satélite, constituye una herramienta indispensable con alto valor agregado, esencialmente en términos de desarrollo, optimización y depuración de software de estabilización y control de apuntamiento satelital. En el caso de los satélites, una vez puestos en órbita, es altamente complejo y costoso realizarles modificaciones o ajustes, por lo que el uso de sistemas de simulación experimental como el que plantea esta tesis doctoral, representa una oportunidad única para experimentar en torno a diversas estrategias de procesamiento de datos y control, así como con diferentes esquemas de instrumentación (sensores de navegación y actuadores), lo cual resulta prioritario para desarrollar el SDCO de satélites reales.

Por tales razones, esta tesis doctoral persiguió la integración, instrumentación y análisis de un simulador MSA, que permite desarrollar y validar SDCO para satélites reales.

La tesis comprende la integración del esquema de instrumentación de la MSA para pruebas de validación experimental de estrategias de control de orientación satelital en tierra. La MSA está basada en un sistema de cojinete de aire y se encuentra instrumentada con sensores, actuadores y una computadora a bordo basada en un dispositivo reconfigurable FPGA.

Una de las principales aportaciones de este trabajo doctoral, además de la que ya representa *per se* la implementación experimental de algoritmos asociados al esquema de control de orientación de satélites, está relacionada directamente con el desarrollo del hardware de la computadora de a bordo, el cual permite integrar arquitecturas de cómputo polimórficas que contendrán algoritmos de control de orientación. Estas arquitecturas se integran en un sistema embebido, evitando con ello la remanufactura del hardware físico de cada esquema de control bajo prueba, limitándose a su reconfiguración remota por software.

La computadora FPGA permite implementar arquitecturas digitales paso a paso, así como la evaluación de algoritmos de control de orientación de forma parcial, utilizando para ello técnicas

auxiliares de co-simulación (HIL). Para ello, se analiza previamente la forma (procesos y operaciones aritméticas necesarias) de cada uno de los algoritmos y se realiza el particionamiento lógico para su descripción en hardware y software embebido y su posterior integración dentro de un sistema computacional basado en un microprocesador *soft core* y núcleos periféricos.

La flexibilidad de la computadora expuesta en esta investigación, refuerza el carácter de plataforma de amplio espectro de aplicación, la cual se puede aplicar además como plataforma didáctica para: la enseñanza en el desarrollo de arquitecturas y sistemas digitales en FPGA, validación experimental de esquemas de control, desarrollo de sistemas electromecánicos para sistemas de control, comunicaciones y procesamiento de datos.

Por otro lado, en términos de análisis de SDCOs, en esta tesis se presenta el desarrollo y obtención del modelo dinámico de la plataforma de la MSA, la cual es una base teórica importante para describir analíticamente su movimiento, así como para tomar decisiones respecto a la mejor estrategia para la implementación del control de orientación en el simulador. El análisis teórico es importante para entender el planteamiento de nuevas técnicas y procedimientos de validación experimental de sistemas espaciales en Tierra.

Cabe señalar que el alcance de esta investigación doctoral se limita al desarrollo e integración de un primer prototipo de simulador, el cual permite implementar y validar experimentalmente algoritmos y esquemas asociados al control de orientación de un satélite real con fines de validación tecnológica y experimentación académica. Los algoritmos finales del sistema de control de orientación de un satélite real estarán basados en especificaciones precisas de una misión, tales como especificaciones y requerimientos de la carga útil, definición de las tareas que desarrollará el satélite en órbita, requisitos de apuntamiento, posición y altura orbital del satélite, etc. Sin embargo, al no definir requerimientos iniciales de una misión en particular, la arquitectura del sistema mecatrónico que se propone en esta tesis doctoral y los resultados esperados, tienen un carácter abierto y enfocados más a la demostración de la capacidad de integración y desarrollo tecnológico propios. Ello permite además a esta investigación, experimentar con diversos módulos, componentes y estrategias vinculados al esquema de control de orientación satelital.

1.5.1. Computadora a bordo basada en FPGA

La MSA incluye como parte de su esquema de instrumentación, una computadora a bordo basada en FPGA, la cual automatiza las funciones del SDCO. La primera versión se basó en una tarjeta comercial de desarrollo FPGA Spartan 3E de Xilinx® y la segunda, en una plataforma de diseño

propio denominada SMIN. En esta computadora se construyen e integran arquitecturas computacionales polimórficas, las cuales realizan el procesamiento de los datos adquiridos de sensores de navegación inercial, así como la generación de señales de control para manejar las ruedas inerciales de la MSA.

La inclusión de una plataforma de procesamiento basada en FPGA aprovecha características del dispositivo [10] y [11], tales como flexibilidad, procesamiento en paralelo, escalabilidad y reducción de espacio en hardware. Los FPGAs admiten integrar núcleos de procesamiento y diversos periféricos en un sistema de cómputo a la medida de los requerimientos de la aplicación, en el cual es posible implementar de manera eficiente y ágil la lógica del esquema de determinación y control de orientación de un satélite.

Respecto a la flexibilidad de los FPGAs, permiten el diseño e integración de sistemas embebidos de cómputo con los recursos necesarios de entradas y salidas digitales para el manejo de interfaces con equipos externos, así como el procesamiento y manejo de datos; todo en una solución integral donde coexisten en una plataforma tecnológica elementos de hardware y software embebidos. En tanto que, el procesamiento paralelo es clave en la búsqueda de una funcionalidad eficiente y rápida en los núcleos de coprocesamiento para resolver operaciones aritméticas, matriciales y vectoriales asociadas con los algoritmos de determinación y control de orientación de la MSA [10].

Adicionalmente, la escalabilidad del sistema embebido en el FPGA, permite que éste pueda ir creciendo en nivel de complejidad para integrar posteriormente algoritmos asociados con la determinación y el control de orientación de la MSA que actualmente se ejecutan en MATLAB. En lo que corresponde a la reducción de espacio en hardware físico, los FPGAs permiten integrar sistemas embebidos complejos en un solo dispositivo, lo cual, tanto para efectos de pruebas experimentales de la MSA en tierra, como para su implementación futura en un satélite, representa un factor de conveniencia muy importante, más aún si se consideran las tendencias actuales de la miniaturización de sistemas en vehículos espaciales.

1.5.2. Esquema de cosimulación HIL como base del simulador del Subsistema de Determinación y Control de Orientación

La implementación de un SDCO constituye un proceso que requiere grandes tiempos de desarrollo para concluirlo en una arquitectura computacional embebida completamente en FPGA. Esto debido principalmente a la adecuación e integración de algoritmos complejos en el FPGA para que operen en tiempo real. Adicionalmente, se debe considerar el particionamiento lógico de cada uno de los

algoritmos asociados al esquema de control en hardware y software; ya sea para el desarrollo de núcleos de coprocesamiento que contendrán la lógica y operaciones aritméticas necesarias para su operación (solución en hardware) o bien, para el desarrollo de funciones en el firmware del microprocesador (solución en software). Por tales razones, para el desarrollo del SDCO se usó una técnica alternativa de cosimulación en tiempo real llamada *hardware-in-the-loop*. Con esta técnica se aceleró el desarrollo y la obtención de resultados experimentales, ya que una parte del sistema, la más compleja, se ejecuta en una computadora personal (PC) y la restante en un sistema embebido en el FPGA de la MSA.

La técnica HIL constituye un medio práctico y eficaz de evaluación de sistemas embebidos de control en tiempo real. Para ello se puede usar una planta real o el modelo de la planta a controlar. Sin embargo, la planta puede llegar a resultar bastante compleja, tanto para su implementación directa como para obtener resultados experimentales en el corto plazo [12]. En la MSA, HIL es una herramienta que distribuye los algoritmos que constituyen el esquema de control de orientación. Lo cual facilita la implantación y validación de sistemas como el SDCO, aunque posteriormente aún se tenga que trabajar intensamente para implementar toda la solución en hardware y software embebido en el FPGA. Adicionalmente, el uso de la técnica HIL en el esquema del simulador MSA redundará en beneficios didácticos y de entrenamiento, ya que la arquitectura modular del sistema permite entender de mejor forma la operación y la interacción entre el hardware y el software de estabilización y control de apuntamiento [13].

1.5.3. Operación del simulador MSA

Durante el inicio de operaciones del simulador MSA el usuario carga el software asociado a una arquitectura computacional a la tarjeta de desarrollo FPGA por medio de una Laptop, la cual tiene instalado el software de desarrollo para sistemas embebidos en FPGA. De esta forma se integra la arquitectura computacional a partir de núcleos de propiedad intelectual (IP) propietarios y personalizados. El sistema embebido se desarrolla, sintetiza y compila utilizando las herramientas de la suite ISE de Xilinx® (Project Navigator y EDK). La carga del software se realiza conectando una laptop con la tarjeta Spartan 3E por medio de un puerto USB. Una vez transferido el software al sistema FPGA Spartan 3E (a bordo de la MSA), queda en espera de que el usuario ejecute el software desarrollado en MATLAB® en una PC externa para establecer el vínculo de comunicación inalámbrica e iniciar la transferencia de datos para la ejecución de los algoritmos que integran el SDCO (determinación de la orientación, estimación, filtrado y control) en la PC externa, de acuerdo con el esquema de validación HIL adoptado [13].

Posteriormente, el usuario alimenta un flujo de aire a la base de la MSA por medio de un compresor, cuya presión se regula manualmente con una válvula de paso, mientras que de forma simultánea balancea manualmente a la plataforma móvil de la MSA por medio de contrapesos. Una vez que la plataforma del simulador se encuentra flotando y balanceada, se procede a ejecutar el software desarrollado en MATLAB® en la PC externa. De tal forma que los datos de los sensores de navegación inercial (magnetómetros, acelerómetros y giróscopos) se adquieren cada 10 milisegundos en el FPGA de la MSA. Estos datos se utilizan para alimentar en tiempo real a los algoritmos del esquema de control de orientación ejecutados en MATLAB®. En esta computadora se generan también comandos para los actuadores, los cuales se transmiten a la MSA por medio de dos módems de radiofrecuencia (RF), posteriormente el sistema embebido en FPGA los transfiere hacia los registros asociados con terminales físicas de salida hacia drivers de potencia que controlan a las ruedas inerciales.

Cabe señalar que la definición de maniobras de apuntamiento que ejecuta la MSA se establece en el software integrado en MATLAB® previo a la ejecución de la prueba, para lo cual se adquieren 200 lecturas del magnetómetro triaxial en la posición deseada y cuyos datos en cada eje se guardan en un archivo de texto. Una vez que finaliza la adquisición de estos datos, en MATLAB® se obtiene el promedio aritmético de cada una de las componentes X, Y y Z del magnetómetro, cuyo resultado por eje, se introduce de forma manual en la función que contiene el algoritmo de determinación de la orientación, el cual está basado en el método determinístico denominado TRIAD (*Three-axis Attitude Determination*, por sus siglas en inglés) [5] y [14].

Adicionalmente, y en función del desempeño de las pruebas, se admiten ajustes en los valores de las covarianzas del error en procesos y mediciones asociados con el Filtro de Kalman Extendido (EKF), el cual se usa como estimador de la orientación [5], [15] y [16]. Esta es otra de las funciones desarrolladas en MATLAB® en la PC externa, la cual permite la estimación de parámetros tales como la velocidad angular y la orientación de la plataforma mediante cuaterniones, que representa una parte fundamental del esquema de control de orientación de la MSA.

1.5.4. Desarrollo de herramientas de validación en hardware y software

Se desarrollaron diversos módulos tanto en hardware como en software, los cuales permitieron implantar algoritmos y estrategias de control de orientación satelital en tres ejes utilizando la MSA bajo el esquema HIL. Dichos módulos se encuentran ubicados en dos segmentos principales que integran al simulador MSA: la plataforma móvil y la computadora externa.

Respecto a la plataforma móvil, los módulos en hardware que se desarrollaron fueron la computadora a bordo FPGA y la tarjeta de potencia para actuadores y suministro de energía.

En lo que corresponde a la computadora a bordo FPGA a la cual denominamos SMIN, cuenta con todos los recursos necesarios para la adquisición y procesamiento de datos, utilizando para ello un FPGA de la familia Spartan 3E de Xilinx®.

Una característica relevante del SMIN, es su módulo de reconfiguración remota, basado en un microcontrolador, el cual permite que el FPGA pueda ser reconfigurado a distancia (reconfiguración total) una vez instalado sobre la MSA. El SMIN permite validar diversas arquitecturas computacionales, entre ellas las que describen algoritmos de determinación y estimación de la orientación y esquemas de control satelital.

En lo que corresponde al módulo de suministro de energía, se diseñó y fabricó una tarjeta electrónica que contiene reguladores conmutados de voltaje, los cuales permiten reducir el voltaje de la batería (14.8v) y generar el nivel de voltaje adecuado (5v) para la alimentación de energía tanto de la plataforma FPGA (Spartan 3E-SK o SMIN) como de los sensores de navegación inercial. En dicha tarjeta también están montados tres circuitos integrados (*drivers*), que interpretan los comandos de control que envía la computadora a bordo del FPGA en forma de señales PWM, además de manejar la potencia que se suministra a cada uno de los motores de DC que impulsan a las ruedas inerciales de la MSA.

Dentro del hardware de orientación, los actuadores son de gran interés para la realización de maniobras de control de orientación. El diseño de los actuadores se basó en modelos publicados en bibliografía internacional (tesis, artículos, reportes técnicos) y en bibliografía especializada en el tema de control de vehículos espaciales [17] y [18]. Su forma geométrica, integrada por un disco central y un anillo externo, permite generar torques del orden de 0.05 a 2 N-m [20], los cuales, al transferirse a la plataforma móvil, facilitan la realización de movimientos de control de orientación en los tres ejes de la MSA. El actuador está integrado por dos componentes básicos, un motor DC y una rueda de inercia, la cual está acoplada al eje del motor y sujeta mecánicamente.

1.6. Organización de la tesis

La tesis está organizada en 7 capítulos:

En el capítulo 1 se presenta una introducción que incluye las generalidades del tema, así como los objetivos y metas planteados en el tema de investigación doctoral.

En el capítulo 2 se presenta una revisión bibliográfica en materia de desarrollo tecnológico en SDCO para satélites pequeños y simuladores satelitales.

En el capítulo 3 se presenta el desarrollo del modelo matemático de la MSA y una vista general del esquema propuesto para el análisis teórico de la plataforma de simulación de vuelo orbital.

En el capítulo 4 se describe la instrumentación de la MSA en hardware y software. En términos de hardware, se describe tanto el hardware físico (sensores de navegación inercial, computadora a bordo y actuadores) como el hardware embebido desarrollado en el FPGA de la computadora de a bordo, en tanto que en el caso del software, se describe el software desarrollado en MATLAB.

En el capítulo 5 se muestra el desarrollo de un experimento de seguimiento de la plataforma de simulación satelital utilizando medios de visualización basados en modelos de realidad virtual. Éste, representa el primer paso para la implementación del esquema completo de control de orientación satelital.

En el capítulo 6 se presenta la integración y los resultados experimentales del sistema de simulación satelital desarrollado. Como parte de la validación operativa del sistema, se describe la implementación de un algoritmo de control basado en una ley proporcional derivativa, la cual obedece a una estrategia de control en tres ejes utilizando sensores de navegación inercial. Además se presentan los resultados obtenidos de las pruebas de control de orientación y una discusión al respecto.

En el capítulo 7 se presentan las conclusiones del trabajo de investigación desarrollado y recomendaciones para la conclusión de tareas pendientes.

En las secciones complementarias se presentan las referencias utilizadas durante la tesis, así como apéndices en los que se presenta el código fuente del software desarrollado para las pruebas de validación de la MSA.

Capítulo 2

Estado del arte en Sistemas de Control de Orientación satelital y simuladores

2.1. El subsistema de control de orientación en satélites

El SDCO de un satélite lo estabiliza y orienta hacia objetivos específicos ante la presencia de torques de perturbación externos, actuando sobre su estructura durante un vuelo orbital. El SDCO utiliza referencias externas para determinar la orientación angular del satélite con respecto a un sistema de referencia inercial fijo, generalmente considerando su origen en el centro de la Tierra. Algunas de las referencias externas que utilizan los satélites para su orientación son el Sol, el horizonte de la tierra, el campo magnético local y las estrellas. Adicionalmente, el satélite utiliza sensores de navegación inercial como giróscopos para cuantificar el desplazamiento angular y estimar la orientación del satélite entre mediciones de referencia inerciales fijas, o bien, mientras tales referencias inerciales no estén disponibles durante el vuelo orbital del satélite. Por ejemplo, el caso de los sensores finos de sol, los cuales no pueden proporcionar mediciones durante la fase de eclipse del satélite [5] y [7].

Es poco frecuente encontrar propuestas de clasificación de los sensores de navegación, lo cual resulta útil para determinar su aplicación en una misión espacial. Sin embargo, en [1] se propone, en función de su operación, una clasificación en sensores de referencia y sensores inerciales. Los sensores de referencia proporcionan su información considerando la medición de cuerpos celestes, tales como el sol o una estrella. En tanto, los sensores inerciales miden continuamente cambios en los movimientos angulares del cuerpo del satélite. Este tipo de sensores requiere una calibración a partir de datos de sensores de referencia, además de presentar errores acumulados debido al efecto de deriva.

La complejidad del esquema de sensores de navegación inercial a bordo del satélite dependerá en gran medida del procesamiento y algoritmos asociados que se consideren para cumplir los objetivos de la misión. Por ejemplo, el caso de los satélites pequeños SNAP-1 y RAX. El satélite SNAP-1, el primer nanosatélite estabilizado en tres ejes, construido por *Surrey Satellite Services Ltd. (SSTL)* en 2010, tuvo como principales objetivos demostrar el control de orientación triaxial y la realización

de maniobras en órbita [21]. Para tal efecto, utilizó como sensor de navegación inercial un magnetómetro triaxial, en tanto que como parte de los algoritmos para el esquema de estimación de orientación, utilizó un Filtro de Kalman Extendido. El satélite RAX es un cubesat diseñado y construido por la Universidad de Michigan, E.U. y SRI International [7] en 2010. La misión del satélite RAX es el estudio de las formaciones y la distribución de irregularidades de plasma alineadas con el campo magnético localizadas en la ionósfera baja.

La instrumentación para control de orientación de RAX integra magnetómetros y sensores finos de sol para cumplir los requisitos de determinación de la orientación angular del satélite de una forma relativamente simple y a bajo costo [50]. Adicionalmente, el satélite RAX utiliza tres giróscopos, cuyas mediciones son fusionadas en un Filtro de Kalman Extendido para la estimación de parámetros de navegación orbital.

Es claro entonces que, conforme los requerimientos de la misión son cada vez más estrictos, el nivel de disposición de información de navegación del satélite se vuelve más exigente, estableciendo arquitecturas que integran a más de un sensor de navegación y, en algunos casos, con redundancia y operación conjunta; estableciendo además complejos esquemas de procesamiento que permiten obtener parámetros de navegación del satélite para efectuar un apuntamiento de alta precisión hacia objetivos definidos. Por ejemplo, es común observar requerimientos de apuntamiento en satélites de percepción remota con una precisión de 1 metro, particularmente para misiones orientadas a la administración de recursos, oceanografía y cartografía, áreas en las cuales los requerimientos son exigentes, en el orden de 0.01 a 0.05° de error de apuntamiento [57].

Es el caso del microsatélite japonés INDEX, desarrollado por el *Institute of Space and Astronautical Science (ISAS)* en 2002, cuyos objetivos fueron la demostración en el desarrollo e integración de nuevas tecnologías de buses para satélites pequeños, integrando una CPU para procesamiento de alta velocidad, un banco de baterías tipo Li-ion para el suministro de energía eléctrica a bordo, así como la observación de auroras. El esquema de sensores estuvo integrado por un sensor de sol tipo *spin*, dos sensores de sol en dos dimensiones (NSAS), un seguidor de estrellas, un magnetómetro triaxial y tres giróscopos de fibra óptica [51]. Cuando no opera el seguidor de estrellas, el sensor de sol y los giróscopos de fibra óptica pueden determinar la orientación del satélite con una precisión de 0.5° [51].

En cuanto al algoritmo de control de INDEX, éste utiliza una ley de control que ya había sido probada en una misión anterior, relacionada con un telescopio astronómico. La ley de control está basada en un esquema de determinación del error de orientación, de tipo lineal con realimentación.

Otro ejemplo respecto a la integración de un esquema complejo de sensores de navegación se encuentra en el satélite FalconSat-5. El FalconSat-5 de 180 kg, fue el quinto satélite construido por el Departamento de Ingeniería Astronáutica de la Academia de la Fuerza Aérea de los Estados Unidos (USAFA) y parte del *Space Test Program*. La misión principal del FalconSat-5 era utilizar sus cargas útiles para coleccionar datos relativos al clima espacial. Con objeto de operar sus cargas útiles como era requerido, el satélite tuvo que alcanzar y mantener estabilidad en tres ejes, con requerimientos nominales de precisión de 0.5 a 1° usando un magnetómetro triaxial y de 0.1° utilizando sensores de sol digitales [22].

La instrumentación del FalconSat-5 estuvo integrada por 4 sensores de sol, un magnetómetro triaxial y una unidad de medición inercial (IMU) para la cuantificación de la tasa de cambios angulares en los tres ejes del satélite. Con el uso de estos sensores de navegación, el FalconSat-5 estima su orientación y realiza las tareas de control de apuntamiento.

Un punto adicional que cabe comentar dentro de la estructura del SDCO de los casos presentados, es el que corresponde a los actuadores, es decir, los elementos mecánicos que permiten la realización física de las maniobras de control de orientación. En [20] se establece una clasificación de actuadores agrupándolos en 4 bloques: 1) aquellos que utilizan el campo magnético terrestre, 2) aquellos que aprovechan las fuerzas de reacción a partir de expulsiones de gas o partículas de iones, 3) aquellos métodos que utilizan la presión por radiación solar sobre la superficie de los vehículos espaciales y 4) aquellos que se basan en el uso de dispositivos de intercambio de momento angular (e.g. ruedas inerciales).

Para el control de orientación, el SDCO produce torques de control que son transferidos al cuerpo del satélite, usando una serie de actuadores que pueden ser de tipo activo o pasivo. Por ejemplo, se emplean toberas impulsoras de todos los tamaños y tecnologías de combustible para el control de orientación del satélite, sin embargo, se requiere de una fuente de combustible a bordo, además de implicar costos elevados. También se utilizan otro tipo de actuadores, como las barras y bobinas de par magnético para producir torques que interactúan con el campo magnético terrestre local. Las ruedas inerciales y las bobinas de par magnético son los actuadores más utilizados en satélites, debido a que producen torques que no cambian el momento angular del satélite [7].

Aunque en principio se pueda identificar un amplio espectro de posibilidades en cuanto a técnicas de actuación de control, cada uno de los métodos mencionados presentan ventajas y desventajas técnicas que deben ser consideradas. En [20] y [34] se presenta un breve análisis de los métodos de actuación mencionados, enfatizando que aquellos que corresponden a técnicas que aprovechan el

flujo por presión solar, no pueden ser utilizados para efectuar maniobras efectivas de control de orientación, debido al bajo torque que presentan, en el orden de 3.28×10^{-8} [N-m] [18]. Sin embargo, el método basado en ruedas rotatorias en el interior del cuerpo del satélite, y cuyo principio de funcionamiento se basa en el intercambio de momento angular, es de las mejores opciones para sistemas de control de orientación satelital que demandan alta precisión y realización de maniobras moderadamente rápidas con el menor número de pares perturbadores parásitos [20].

En [20] y [51] se presenta el análisis dinámico de actuadores basados en ruedas inerciales. Particularmente en [51] se presentan detalles del principio de funcionamiento físico de la rueda inercial, así mismo se efectúa el modelado dinámico y su representación en diagrama de bloques, lo cual resulta útil para incluirlo dentro de un esquema de simulación numérico.

En cuanto a la computadora de a bordo, en la mayoría de los casos se utilizan microcontroladores endurecidos para radiaciones. Estas plataformas de cómputo, además de brindar confianza al desarrollador tradicional, contienen elementos básicos de un sistema embebido para la integración de esquemas de procesamiento del SDCO en una arquitectura fija.

No obstante, existen proyectos en los cuales la computadora a bordo está desarrollada en torno a novedosos dispositivos reconfigurables tales como FPGAs. En [52] se presenta el microsatélite Flying Laptop, en desarrollo por el Instituto de Sistemas Espaciales de Stuttgart, Alemania. Éste microsatélite, de 100 kg de peso y estabilizado en tres ejes, tiene integrada su computadora de a bordo en un FPGA, donde se define una arquitectura de triple redundancia para fines de tolerancia a fallas y con capacidad de reconfiguración dinámica al vuelo. En términos de hardware, integra 5 principales tipos de sensores de navegación: dos magnetómetros triaxiales, dos sensores de sol, cuatro giróscopos de fibra óptica, un seguidor de estrellas y tres receptores GPS.

La conexión entre los sensores de navegación y el FPGA se lleva a cabo en una conexión tipo estrella, incrementando con ello la confiabilidad del sistema [52]. Utiliza los protocolos RS422 e I²C, así como líneas de entrada salida digitales y el bus IBIS.

En tanto, en Japón [53], en la Universidad de Soka, está en desarrollo el *cubesat* Excelsior, cuyos objetivos son: la verificación de la tolerancia a fallas de la electrónica de la computadora a bordo basada en un FPGA en ambientes de alta radiación, adquisición y compresión de imágenes de la Tierra y experimentos de comunicación para amateurs.

En [54] se establece que los FPGAs son comúnmente utilizados en productos que están inmersos en ambientes de radiación, tales como aplicaciones en aviónica o espaciales. Así mismo, los FPGAs

son dispositivos altamente atractivos ya que contienen alta densidad de compuertas lógicas equivalentes en un solo dispositivo, lo cual permite desarrollar soluciones de tamaño, potencia y peso reducidos.

2.2. Técnicas de simulación para validación de pruebas de control de orientación satelital

A partir de la discusión anterior, en términos de los elementos que integran al SDCO, es evidente la gran inversión en tiempo y financiera que se debe realizar para adicionar a un satélite su capacidad de estabilización y control de apuntamiento. Por tal motivo, durante el proceso de diseño se deben desarrollar los medios de validación y simulación necesarios que permitan conferir certeza en cuanto al desempeño del satélite en órbita. En este sentido, el SDCO no es la excepción, sin embargo, su simulación y validación en tierra representan *per se* una serie de retos que se deben analizar detenidamente.

Si bien, las simulaciones por computadora representan un excelente punto de partida para el análisis del desempeño del SDCO, dichas simulaciones sólo tienen la capacidad de modelar algunas características tanto de la dinámica del propio vehículo espacial como de su entorno, incluidas las perturbaciones que afectan la orientación del satélite. No obstante, dichas consideraciones pueden impactar fuertemente su operación en un ambiente orbital real.

Una interesante alternativa que ayuda al proceso de definición y diseño del SDCO de un satélite en tierra, es el uso de plataformas de simulación experimental, las cuales tienen diversas arquitecturas y emulan particularmente dos condiciones del entorno espacial: microgravedad y ambiente de bajo torque.

Así mismo, existen técnicas de cosimulación en las que se involucran elementos de software y hardware, que permiten la rápida implementación de complejos esquemas de control, tal como la denominada *hardware-in-the-loop* (HIL). Los experimentos realizados utilizando esta técnica, ofrecen un procedimiento para abordar el problema de definir parámetros de operación no conocidos, probándolos o determinándolos a partir del análisis experimental, con todas las incertidumbres y errores que el hardware puede brindar.

En [58], se presenta un simulador de simulación para pruebas de control de orientación satelital, desarrollado en torno a la misión Chasqui-1 de la Universidad Nacional de Ingeniería de Lima, Perú. En esta propuesta, se describe la integración de un simulador, denominado “de bajo costo”, debido a que, basados en la técnica HIL, integran hardware real, principalmente actuadores y sensores de navegación. Mientras que, en una computadora personal convencional, se ejecutan en

un ambiente Simulink® los algoritmos que integran el núcleo de control del sistema, interactuando por medio del protocolo RS232 con sensores y actuadores.

Respecto a las pruebas con microgravedad, existen diversas técnicas para crear dicho ambiente sin la necesidad de ir precisamente al espacio exterior, actualmente existen tres principales técnicas: la primera consiste en el empleo de ambientes de flotación neutra (*neutral buoyancy*), las cuales son utilizadas para entrenamientos de los astronautas de la NASA. Esta técnica consiste en el uso de una alberca, figura 2.1, en la cual se sumerge el objeto o sistema bajo prueba. No obstante, su efectividad para pruebas de control de orientación resulta poco atractiva, ya que se debe cuidar de las propiedades fisicoquímicas del líquido vertido a la piscina para garantizar la flotabilidad del sistema bajo prueba, lo cual requeriría que dicho líquido tuviera un bajo índice de viscosidad para generar un ambiente de bajo torque. Sin embargo, los torques debidos al arrastre con el propio flujo hidráulico podrían desviar los resultados experimentales esperados, resultando por lo tanto una metodología de validación poco recomendable para pruebas de control de orientación [55].



Figura 2.1. Laboratorio de pruebas de flotación neutra.

En [59] se describe, por ejemplo, un sistema desarrollado en la Universidad de Maryland, el cual, basado en la arquitectura de un simulador *neutral bouyancy*, utiliza pequeños submarinos para monitorear los movimientos de un vehículo de prueba que se sumerge. Cada uno de los submarinos envía tramas de datos hacia una computadora principal, en la cual se capturan imágenes de cámaras instaladas en 6 submarinos instalados en el tanque. Dichas imágenes son de utilidad para efectos de control por realimentación hacia el vehículo bajo prueba, así como la verificación de la interacción de los diversos algoritmos que intervienen en el proceso.

El segundo método utiliza torres de caída (*drop towers*), figura 2.2. La cual es una estructura cerrada dentro de la cual se hace caer un objeto o sistema hasta una red inferior. Durante el descenso del sistema bajo prueba, el experimento tiene unos pocos segundos de microgravedad,

antes de llegar hasta el fondo sobre una red de protección. Durante la fase de microgravedad utilizando esta técnica, el único torque de perturbación que actúa sobre el experimento es el arrastre del aire al interior de la estructura, el cual puede ser eliminado si al interior de la misma se crea vacío. Desafortunadamente el tiempo que ofrece esta solución es tan pequeño que esta técnica se considera muy limitada para la realización de pruebas de control de orientación en un ambiente cercano al espacial [55].



Figura 2.2. Prueba en torre de caída.

Un tercer método de prueba es utilizando vuelos en avión en trayectoria de cero gravedad. El avión vuela en trayectorias parabólicas en intervalos de 24,000 a 34,000 ft de altura. En el pico de cada maniobra, existen periodos de 25 segundos de microgravedad, donde el avión y todo lo que vaya a bordo de él se encuentran en caída libre. A diferencia de la torre de caída, la microgravedad actúa sobre el aire dentro de la cabina, reduciendo con ello el efecto del arrastre sobre muchos de los experimentos que se llevan a cabo [55].

Otra de las técnicas utilizadas para el diseño y validación de esquemas de control de orientación de vehículos espaciales es aquella que hace uso de simuladores basados en cojinetes de aire. En este tipo de plataformas de simulación actúa directamente la fuerza gravitacional terrestre y su principio de operación se basa en la compensación de los torques perturbadores que actúan sobre él. Como se aborda en [24], si bien, este tipo de plataformas de simulación no pueden proporcionar la experiencia total de un ambiente de microgravedad, éstas permiten la manipulación y verificación de hardware de control de orientación en un ambiente de torque mínimo.

Las plataformas de simulación satelital desarrolladas en torno a cojinetes de aire (PSS) son ampliamente utilizadas en la industria aeroespacial para la validación de esquemas de control de orientación, así como en universidades y centros de investigación. Muchas de ellas tuvieron su origen en Estados Unidos, donde se construyó el primer simulador de control de orientación en el *Space Maneuvering Devices section* de la *Division of North American Rockwell Group* en 1967 por el *Marshall Space Flight Center* de la NASA [24].

Estas plataformas experimentales permiten a los investigadores probar en Tierra los algoritmos de un SDCO, en un medio que simula con gran aproximación un ambiente de torque reducido.

2.2.1. Tipos de plataformas de simulación para pruebas de validación de control de orientación basadas en sistemas de cojinete de aire

Se han desarrollado diversos tipos y formas de PSS a lo largo de la historia contemporánea del diseño de sistemas de simulación en tierra, con ciertas características que los han convertido en un referente en el campo de desarrollo de sistemas espaciales. Por ello, a continuación se describirán los tipos o formas más conocidas de las PSS utilizadas en el diseño de sistemas de control de orientación para vehículos espaciales, indicando además sus características constructivas más relevantes.

Los diseños o formas básicas que predominan en el campo de las PSS experimentales, se han clasificado de la siguiente forma [23] y [24]:

- a) **Plataformas tipo mesa.** En este tipo de plataformas, los componentes están distribuidos en la parte superior de su superficie móvil para simplificar el análisis de movimiento, figura 2.3. En cuanto al movimiento angular, este tipo de plataformas proporciona completa libertad de giro en el eje *yaw*, pero los ejes *pitch* y *roll* están generalmente restringidos a ángulos menores a $\pm 90^\circ$ (típicamente menores a 30°). En términos de desarrollo, por su sencillez y facilidad de integración a bordo, representan la mejor opción para sistemas cuyo esquema de instrumentación no ha sido definido por completo y representa el tipo de simulador que emula la forma más básica de un satélite en vuelo orbital.

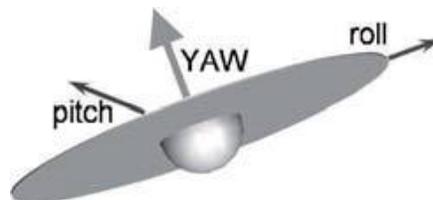


Figura 2.3. Plataforma tipo mesa.

- b) **Plataformas tipo sombrilla.** Este diseño mantiene el centro de masa del sistema muy cerca del centro de rotación de la plataforma, figura 2.4. Para ello, se suspenden los componentes debajo de la plataforma. De igual forma que el tipo mesa, para éste simulador, existen fuertes restricciones de maniobras en los ejes *pitch* y *roll*, en los cuales los movimientos son menores a $\pm 90^\circ$.



Figura 2.4. Plataforma tipo Sombrilla.

- c) **Plataformas tipo mancuerna.** Este diseño reduce la interferencia de la estructura colocando la carga útil del simulador lo más lejos posible de su centro de rotación. Como se aprecia en la figura 2.5, el diseño de la estructura para esta plataforma es más complejo que el de los casos anteriores, permitiendo movimientos sin restricción en los ejes de maniobra *roll* y *yaw*, no así en *pitch*, donde por la posición de la base y las placas laterales, su movimiento está limitado.

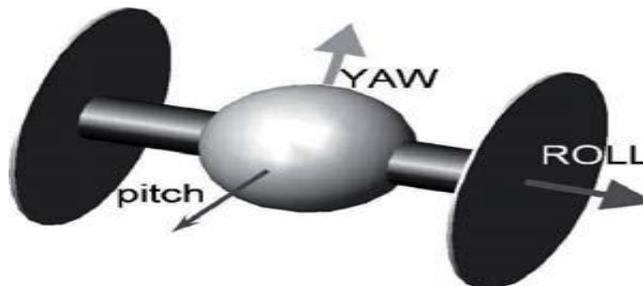


Figura 2.5. Plataforma de Mancuerna.

Con base en las características de cada una de las plataformas referidas anteriormente, los diseñadores del sistema de control eligen cuál es aquella que cumple con los requerimientos de los objetivos planteados de la misión, así como la complejidad del sistema de estabilización y control de apuntamiento.

Adicionalmente a los sistemas rotacionales, como los mencionados anteriormente, existen las PSS de tipo plano. Dichos simuladores, figura 2.6, usan una superficie plana sobre la cual los objetos colocados en su interior flotan por medio de flujos de gas comprimido.

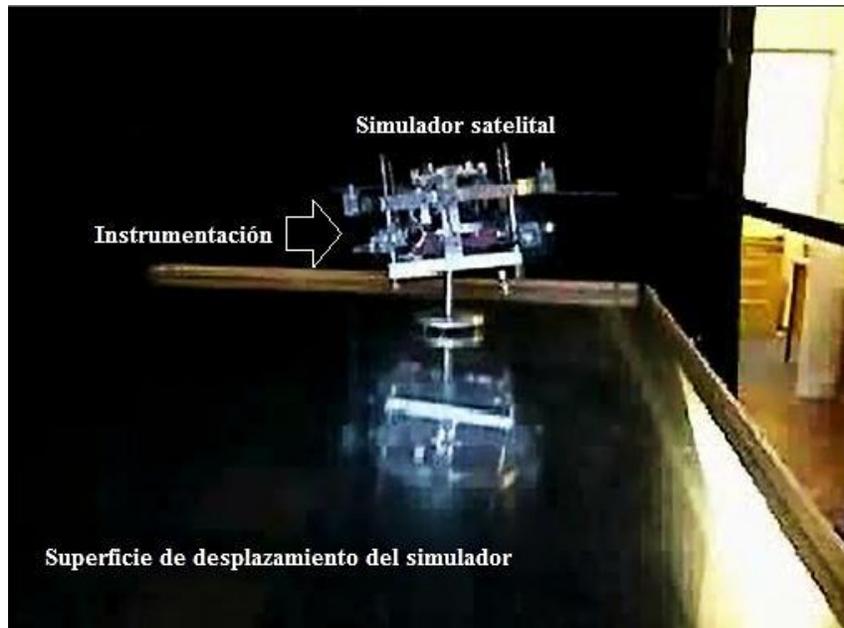


Figura 2.6. PSS tipo plano.

Los sistemas planos permiten el movimiento traslacional y –no necesariamente- movimiento de giro vertical [59]. Este tipo de sistemas, llevan a bordo su propio aire comprimido en tanques y crean su colchón de aire, el cual proporciona un movimiento deslizante con baja fricción sobre una superficie plana y lisa. El movimiento de giro vertical se alcanza usando una rueda inercial o toberas de aire comprimido. Estos simuladores son importantes particularmente para pruebas de acoplamiento entre vehículos espaciales o bien para la ejecución de maniobras [59].

La combinación de sistemas (rotacionales y planos) resulta en la generación de simuladores de 5 ó 6 grados de libertad, en donde se combinan etapas de movimiento angular y traslacional. Un ejemplo es el simulador del *Marshall Space Flight Center's Flight Robotics Laboratory* [60]. Este simulador permite la realización de movimientos con 6 grados de libertad sobre una superficie plana de 44 x 66 ft. El movimiento vertical es provisto por un ascensor cilíndrico en lugar de toberas.

En [61] se aborda el diseño de un simulador característico, el cual cuenta con 3 grados de libertad, denominado *Floatcube*, el cual tiene las dimensiones de un cubesat y tiene el objetivo de demostrar la aplicación de conceptos de *flux pinning* en maniobras de formación y acoplamiento. La técnica

flux pinning consiste en la interacción entre un superconductor de alta temperatura (HTSC) y un campo magnético. Dicha interacción crea una rigidez mecánica y amortiguamiento, lo cual influye en el movimiento respecto al campo magnético en múltiples grados de libertad, creando un mecanismo de sustentabilidad para la ejecución de pruebas de control de orientación de vehículos espaciales en tierra.

En [62] se describe el *Eropean Proximity Operations Simulator (EPOS)*, el cual está instalado en el Centro de Operaciones Espaciales de Alemania del Centro Aeroespacial Alemán. Dentro de las características de EPOS es su arquitectura, la cual está integrada por dos robots industriales, los cuales son capaces de posicionar cargas útiles con masas superiores a los 240 kg con precisión submilimétrica. Adicionalmente, tiene la capacidad para que uno de los dos robots se traslade sobre un riel que contiene un flujo de aire, a velocidades que van de los mm/s hasta los m/s, convirtiéndolo en un sistema muy eficaz para la ejecución de pruebas de acoplamiento y maniobras espaciales de alta complejidad.

Otra de las características relevantes de EPOS es que está desarrollado con base en la técnica HIL. En la cual, los robots responden a comandos de control que son generados por software desde una PC. En este esquema, el sistema HIL es, básicamente, un sistema en lazo cerrado, donde el sistema de seguimiento por medio de los robots introduce un retraso [64].

2.2.2. Torques de perturbación en las Plataformas de simulación para pruebas de validación de control de orientación

Es importante también considerar que las PSS descritas anteriormente, permiten probar en tierra algunas estrategias de control de orientación para vehículos espaciales en órbita. Sin embargo, existen factores adicionales de perturbación local que deben ser previstos durante la realización de pruebas de estabilización y control de orientación y en el planteamiento de estrategias de control con estas plataformas de simulación.

En [23] se exponen cuatro tipos fundamentales de perturbaciones:

1. **Torques derivados de la plataforma.** Incluyen desbalances estáticos, desbalances dinámicos, inestabilidad del material (estrés, temperatura, humedad, evaporación), gradiente de gravedad, etc.
2. **Torques debidos al cojinete de aire.** Dichas perturbaciones incluyen el efecto aerodinámico conocido como “turbina” y los debidos a fugas de aire de los ductos de alimentación.

3. **Torques debidos al ambiente.** En estas perturbaciones se incluyen: el efecto por corrientes de aire, campos magnéticos locales, vibración, presión por radiación y movimiento de equipo (solenoides y relevadores).
4. **Torques del sistema de pruebas.** En esta categoría se considera el efecto del cableado eléctrico a la base, desplazamiento de masas sobre el cojinete y ajustes varios realizados a la instrumentación instalada sobre la plataforma del simulador, descarga de la batería, descarga del suministro de combustible para crear flujos para pruebas con toberas impulsoras y el remplazo de componentes.

Los torques de los grupos 1 y 4 pueden ser mitigados a través de un buen diseño, tanto de la plataforma de simulación como de los elementos mecánicos que se integran a bordo. Al respecto es importante advertir la necesidad de contar con estructuras y elementos mecánicos bien diseñados y manufacturados, integrados con componentes bien seleccionados y dimensionados [23].

En cuanto a los torques del grupo 2, sus efectos recibieron más atención en los desarrollos iniciales de simuladores basados en cojinetes de aire que en los actuales. Aunque algunos efectos internos en el cojinete de aire pueden resultar importantes en el diseño y operación de sistemas basados en cojinetes de gas de tipo industrial, se presenta un efecto que puede ser despreciable [23].

La tercera clase de torques de perturbación corresponden a los presentes en el ambiente de laboratorio, los cuales representan el reto más importante aún por resolver para las plataformas de simulación de vuelo orbital en tierra [23].

Para efectos de aplicación en este trabajo de investigación doctoral, se consideró incluir en la vertiente del desarrollo de la ecuación de movimiento de la plataforma experimental, perturbaciones externas correspondientes al primer grupo, particularmente la que se debe a la gravedad.

En los siguientes párrafos se describirán los esquemas de instrumentación de ciertas PSS representativas, desarrolladas en diferentes instituciones de educación e investigación en el mundo, con particular interés en sensores de navegación inercial, computadora de vuelo y actuadores.

2.2.3. Casos destacados de Plataformas de simulación para pruebas de validación de control de orientación en el mundo

En los siguientes apartados se presentan algunos ejemplos de plataformas de simulación desarrolladas a nivel internacional. Hay que aclarar que el universo de publicaciones que las describen es muy amplio, sin embargo, para efectos de esta investigación, se tomaron casos

representativos, los cuales por la relativa sencillez en su arquitectura, permiten su descripción para establecer un marco de referencia tecnológico en relación con los objetivos de esta tesis doctoral.

2.2.3.1. Plataformas de simulación de la Universidad Yonsei, Corea del Sur

En [25] se presentan los detalles de una PSS desarrollada en la Universidad Yonsei, Corea del Sur. Dicha plataforma, cuya masa es de 50 kg, reproduce la orientación de un vehículo espacial en órbita usando una plataforma del tipo mesa, permitiendo movimientos de $\pm 180^\circ$ en el eje yaw y $\pm 45^\circ$ en los ejes roll y pitch.

Una característica importante de la instrumentación de este simulador, figura 2.7, es que está basado en el esquema HIL y en una ley de control PID. Adicionalmente, cuenta con tres ruedas inerciales y una unidad de sensores de navegación inercial (IMU) a bordo.

Utiliza un esquema de comunicaciones basado en el protocolo TCP/IP por medio de ruteadores de red inalámbricos y una tarjeta inalámbrica LAN. El suministro de energía eléctrica a bordo se basa en un banco de tres baterías selladas de plomo. Utiliza un CPU PC104 (computadora embebida para aplicaciones específicas en entornos industriales) como computadora a bordo para la ejecución de tareas como adquisición y preprocesamiento de datos y manejo de actuadores.

Adicionalmente, utiliza la herramienta xPCTarget y Real-time para establecer el esquema de cosimulación entre la plataforma móvil y el procesamiento en una computadora externa [25].



Figura 2.7. Plataforma móvil del simulador HIL de la Universidad Yonsei, Corea del Sur.

2.2.3.2. Plataforma de simulación de la Escuela de Posgraduados Navales, Monterey, California

La Escuela de Posgraduados Navales en Monterey, California, desarrolló un simulador basado en una plataforma tipo mesa para realizar pruebas experimentales al *Bifocal Relay Mirror Spacecraft* (BRMS), figura 2.8, el cual es un instrumento diseñado para redireccionar un haz láser desde una fuente determinada hacia un objetivo ya sea en la Tierra o en el espacio [26].

El simulador para pruebas de control de orientación del BRMS es capaz de sostener sobre su cojinete de aire una carga de 800 kg de masa con 70 lb-in² de aire a presión y puede realizar maniobras de rotación arriba de los 20° en los ejes pitch y roll [26] de acuerdo con una ley de control PID.

Para la estimación de su orientación, el simulador utiliza un magnetómetro triaxial, dos inclinómetros, dos sensores de sol en dos ejes y una IMU. La plataforma móvil se controla usando tres giróscopos de control de momento (CMGs), los cuales operan por medio de rotaciones de ruedas inerciales, girando a velocidad constante en torno a un eje perpendicular asociado a cada uno de los ejes de rotación [26]. Por medio de la rotación de las ruedas se crea un torque giroscópico que puede ser utilizado para el control de orientación de la plataforma móvil.

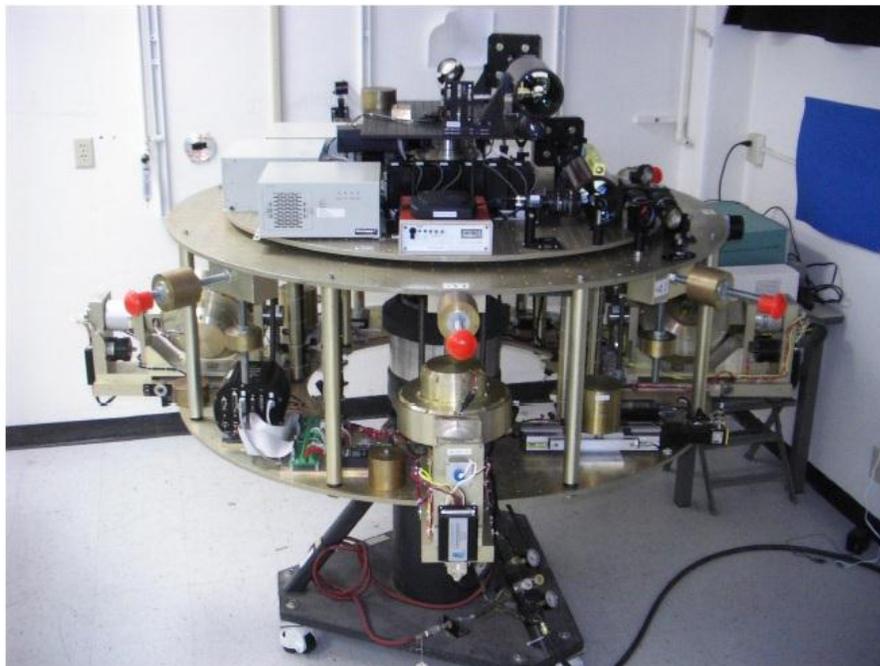


Figura 2.8. Plataforma del simulador en tres ejes de la misión BRMS.

2.2.3.3. Plataforma de simulación del Instituto de Tecnología de la Fuerza Aérea de los E.E.U.U.

El Instituto de Tecnología de la Fuerza Aérea en la base aérea Wright-Patterson en Ohio, E.E.U.U. desarrolló y actualmente opera un simulador de control de orientación denominado SimSat II, figura 2.9, el cual está basado en una plataforma tipo mesa [27].

El sistema puede rotar hasta 20° en los ejes pitch y roll y soportar hasta 136 kg. [27]. El simulador SimSat II utiliza una IMU de fibra óptica Northrop Grumman LN-200 como su único dispositivo de estimación de orientación y tres ruedas inerciales montadas de forma ortogonal.

Utiliza una Mini-Box PC como computadora a bordo, la cual, de forma inalámbrica, se comunica con otra computadora externa para realizar tareas de programación y almacenamiento de datos. SimSat II utiliza masas como contrapesos, las cuales pueden ser desplazadas de forma manual para ajustar la posición del centro de masa del sistema.

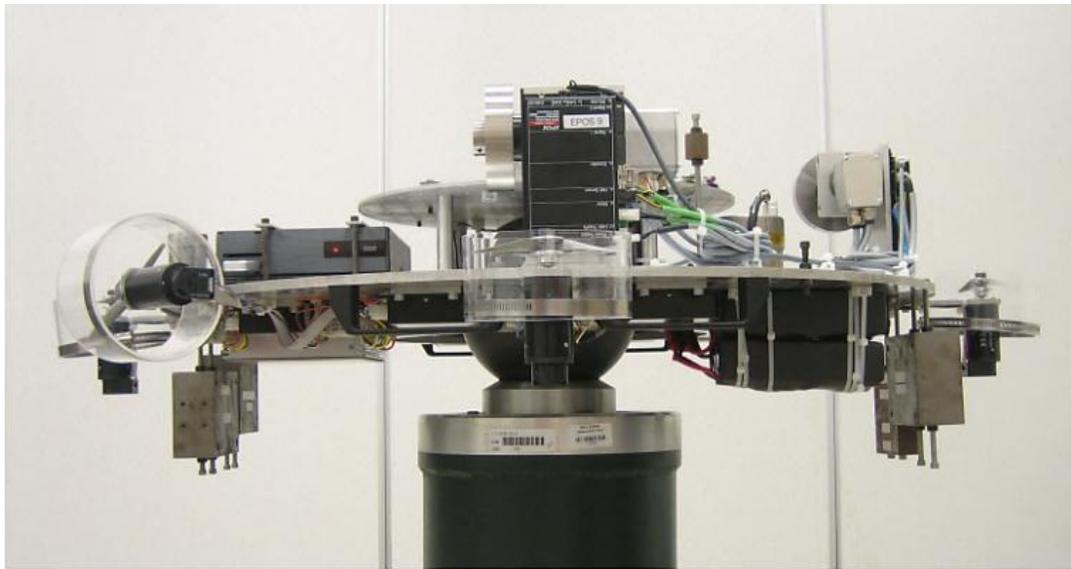


Figura 2.9. Simulador SimSat II del Instituto de Tecnología de la Fuerza Aérea de E.E.U.U.

Con base en la descripción presentada de los simuladores HILS (Yonsei), BRMS de la Escuela de Posgraduados de California y el SimSat II de la Fuerza Aérea de los Estados Unidos, es importante señalar características relevantes comunes a cada uno de ellos, principalmente en términos de instrumentación, destacando el uso de unidades de medición inercial (IMU), que integran acelerómetros, magnetómetros y giróscopos y que realizan cierto procesamiento a bordo, brindando el dato del cuaternión de orientación u otras formas de representación de la orientación de cuerpo

rígido, los cuales resultan fundamentales en el proceso de control de orientación de vehículos espaciales.

Una ventaja muy importante del uso de IMUs, es que algunos modelos comerciales ofrecen procesamiento interno de datos de cada sensor de navegación, evitando parte de la carga de procesamiento que debería realizar la computadora a bordo de la PSS, optimizando aún más el esquema de procesamiento y mejorando el desempeño del sistema en su conjunto, particularmente en aquellas plataformas que utilizan esquemas de coprocesamiento como HIL para la evaluación de hardware y depuración de software y que requieren interactuar con PCs externas.

2.2.3.4. Plataforma de simulación BIRD de Astro- und Feinwerktechnik , Alemania

La plataforma BIRD forma parte del programa FACE (*Facility for Attitude Control Experiments*) que se ha implementado en el Instituto de Sistemas Espaciales de la Agencia Espacial Alemana (DLR). El objetivo de esta plataforma es probar y validar el Sistema de Determinación y Control de Orientación de un satélite, emulando para ello algunas de las condiciones presentes en el ambiente espacial. Para ello, utiliza una estructura descrita en la figura 2.10.



Figura 2.10. Laboratorio del programa FACE.

La plataforma de FACE, BIRD [61], es un simulador para pruebas de control de vehículos espaciales, basada en un sistema de cojinete de aire. Adicionalmente, y una característica sobresaliente de BIRD es que cuenta con los recursos para emular el campo magnético y la luz del Sol. La simulación del campo magnético se produce por medio de un sistema de bobinas de Helmholtz, las cuales generan un máximo de 210,000 nT con una precisión de 1%. Las bobinas tienen un diámetro de 2.40 m y están soportadas eléctricamente por un sistema de energía basado en fuentes bipolares, así como asociadas al software de una PC por medio del bus GPIB. En cuanto a la simulación de la luz solar, éste está basado en un sistema de iluminación basado en una lámpara de luz HMI, la cual puede ser variada en un intervalo de 30 a 100% de potencia, la cual puede ser manipulada ya sea de forma automática por medio de la PC o bien de forma manual.

El medio de comunicación utilizado para interactuar de forma externa con la plataforma de BIRD es por medio de un enlace basado en un sistema WLAN. El esquema de sensores de navegación está integrado por giróscopos de fibra óptica, unidades de medición inercial y otros.

La filosofía de desarrollo de la plataforma BIRD incorpora el uso de dos técnicas de cosimulación, por un lado HIL y por otro lado *Software-in-the-loop* (SIL). La fusión de ambas técnicas de evaluación permite integrar diferentes componentes no sólo en hardware sino también en software para su evaluación operativa, convirtiendo a BIRD en una plataforma de simulación ampliamente versátil para el desarrollo e integración de esquemas de control de orientación de vehículos espaciales en tierra [61].

2.2.3.5. Plataforma de simulación del Instituto de Geografía, UNAM

En el laboratorio de Percepción Remota Alternativa y Tecnología Avanzada del Instituto de Geografía de la UNAM, se han desarrollado diferentes versiones de simuladores satelitales. Estos simuladores permiten llevar a cabo pruebas experimentales de diferentes conceptos asociados al control de orientación de vehículos espaciales, los cuales podrán posteriormente ser transferidos a plataformas satelitales reales [58].

La plataforma de simulación denominada SIMUSAT, figura 2.10, permite realizar pruebas de control de orientación para satélites pequeños, emulando la falta de fricción, una de las condiciones presentes en el ambiente espacial [58].

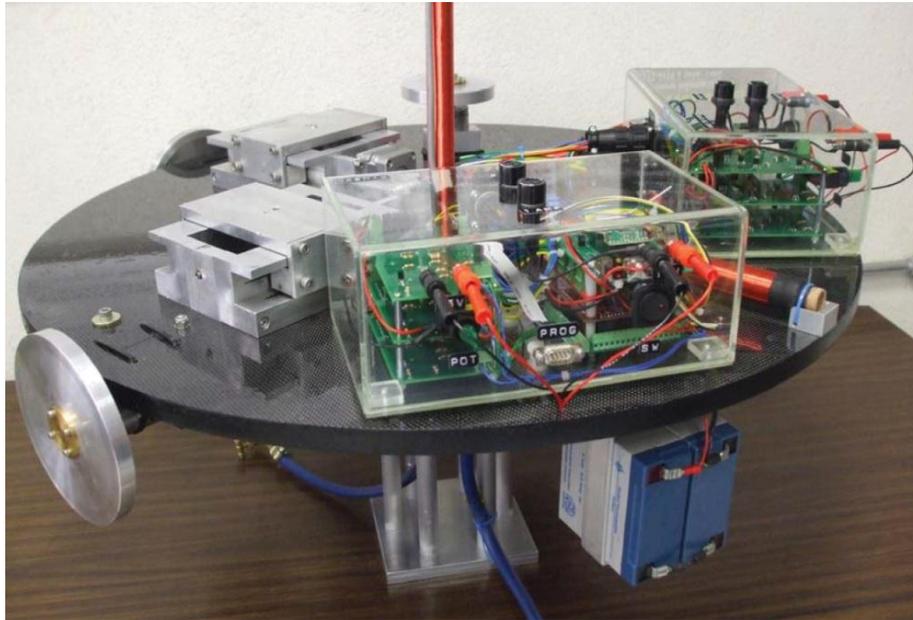


Figura 2.10. SIMUSAT del Instituto de Geografía de la UNAM.

El diseño de este simulador está basado en una plataforma móvil, donde se colocan los componentes del esquema de control. Durante la operación del simulador, la plataforma móvil flota sobre un colchón de aire permitiendo el movimiento con una fricción despreciable, coadyuvando así a la realización de pruebas experimentales de estabilización y control de apuntamiento.

Como parte de la instrumentación, la plataforma SIMUSAT integra baterías a bordo, las cuales permiten la autonomía operativa del sistema.

Adicionalmente, SIMUSAT cuenta con un sistema de balanceo automático, el cual se encarga de estabilizar mecánicamente la plataforma, lo cual reduce el tiempo de preparación para desarrollar pruebas experimentales de control, ya que de otra forma el usuario debería hacerlo de forma manual, no siempre con los resultados esperados, [55] y [56].

La arquitectura de SIMUSAT está integrada por 7 componentes esenciales:

- 1) Una plataforma circular suspendida sobre un balero de aire esférico, donde se genera un medio con fricción despreciable.
- 2) Tres ruedas inerciales que constituyen un grupo de actuadores para control de orientación de la plataforma.

- 3) Tres bobinas magnéticas, localizadas en los ejes de un sistema ortonormal asociado a la plataforma del simulador.
- 4) Dos inclinómetros y un magnetómetro.
- 5) Dos masas deslizantes para balanceo automático.
- 6) Una computadora de a bordo y tarjetas electrónicas de potencia.
- 7) Baterías de plomo-ácido para alimentación de energía eléctrica de los componentes a bordo del simulador.

Adicionalmente, SIMUSAT cuenta con una herramienta en software que permite el monitoreo inalámbrico, el cual transmite la información sobre la orientación del simulador en tres ejes durante las pruebas. Los datos obtenidos son desplegados y almacenados, permitiendo así la evaluación posterior de sensores, actuadores y algoritmos.

2.2.3.6. Plataforma de simulación del IINGEN-UNAM, México

En el caso del IINGEN-UNAM, se ha instrumentado la plataforma MSA, figura 2.11, desarrollada en la UNAM para el proyecto satelital Satex [46], [56], la cual está basada en un sistema de cojinete de aire (ABS, *Air Bearing System*) tipo mesa rotacional. La MSA opera por medio de un colchón de aire, creado por un flujo de aire constante proporcionado por una compresora que opera a niveles de 60 a 70 psi. Dicho flujo se hace pasar previamente a través de un banco de filtros (coalescente y desecante) utilizados para retirar rastros de combustible y agua, evitando con ello el taponamiento de los ductos capilares milimétricos ubicados en la base del soporte de la MSA.

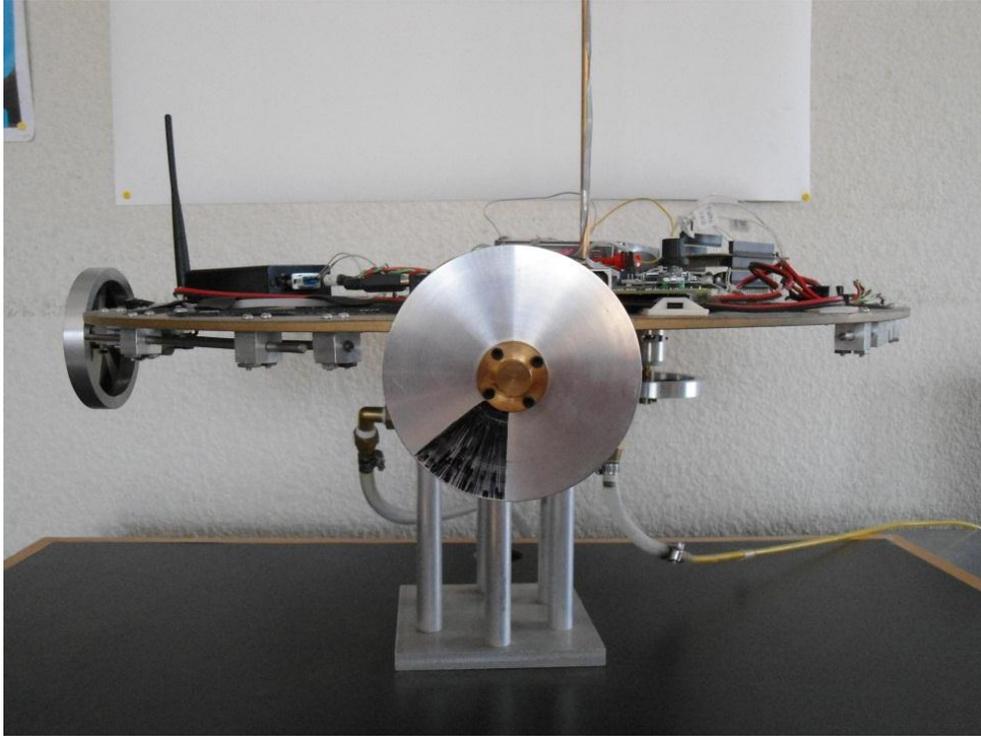


Figura 2.11. Plataforma del simulador de vuelo orbital MSA.

La plataforma móvil de la MSA está constituida por un disco de material denominado *Medium-density fibreboard* (MDF) con una densidad relativa de 0.7, con 50 cm de diámetro y 1 cm de grosor. A su vez, la plataforma está unida mecánicamente a una semiesfera que forma un colchón de aire contra una base fija. La base fija tiene 30 cm de altura y en su superficie se encuentran 6 ductos milimétricos perforados por los cuales se generan flujos de aire que sostienen a la plataforma de la MSA, creando el efecto de cero fricción.

Para naciones con un gran avance en materia aeroespacial, el desarrollo de sistemas a partir de diseños propios antes que preferir soluciones comerciales, pudiera resultar un esfuerzo irrelevante. No obstante, para naciones que incursionan en el campo de tecnología espacial, las lecciones aprendidas en cuanto al uso e implementación de arquitecturas computacionales propias, resultan productivas y enriquecedoras en términos del establecimiento de *know-how* para el diseño y generación de soluciones nativas.

La instrumentación a bordo de la MSA, figura 2.12, está integrada por cinco segmentos principales:

- 1) Sensores de navegación inercial (acelerómetros, giróscopos y magnetómetros triaxiales).
- 2) Computadora a bordo basada en FPGA.

3) Actuadores

4) Comunicaciones inalámbricas

5) Dispositivos electrónicos de soporte (batería, manejo de potencia para actuadores).

Algunos de los elementos son comerciales (sensores de navegación inercial, computadora a bordo, batería) y otros son de diseño propio (actuadores, tarjeta para manejo de potencia, computadora a bordo).

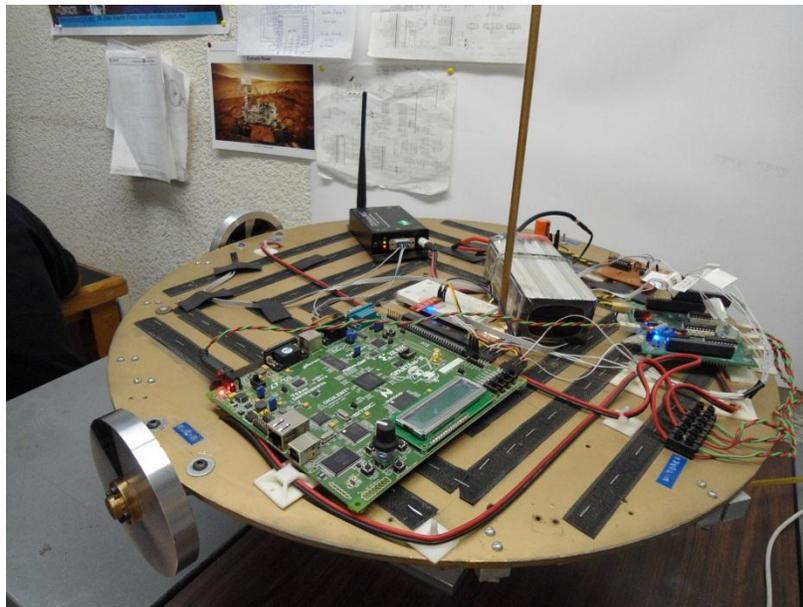


Figura 2.12. Aspectos generales del esquema de instrumentación a bordo de la MSA.

Una de las características más importantes de la MSA es su computadora a bordo, la cual está basada en una plataforma FPGA y un módulo de reconfiguración remota. El uso de tecnología reconfigurable FPGA brinda a todo el sistema la flexibilidad para el diseño e integración de una plataforma de cómputo polimórfica y a la medida de los requerimientos de la aplicación, adicionalmente coadyuva en la optimización del procesamiento de datos a bordo, mediante el desarrollo de estructuras computacionales concurrentes (paralelismo), las cuales permiten implementar experimentalmente algoritmos de control en el FPGA.

Si bien, en el caso de otros simuladores, cuyas plataformas de procesamiento están basadas en potentes computadoras dedicadas, miniaturizadas, de arquitectura fija que facilitan la integración de cualquier esquema de adquisición de datos y de control, independientemente de la complejidad de su estructura, limitan al diseñador en cuanto al desarrollo de plataformas computacionales a la

medida de los requerimientos de la aplicación, sobredimensionando recursos internos, espacio y potencia consumida.

Finalmente, se enfatiza que el sistema de simulación que se describe en esta investigación doctoral, representa un desarrollo tecnológico integral único en México en materia de diseño y validación experimental de sistemas de estabilización y control de apuntamiento satelital en tres ejes, el cual fusiona técnicas que lo hacen pionero en el país, particularmente:

- a) La MSA es un sistema de validación de sistemas espaciales que permite la coexistencia de elementos de software y hardware, particularmente hardware reconfigurable de estabilización y control de apuntamiento satelital descrito por medio de software empleando dispositivos FPGA.
- b) Su plataforma de cómputo permite el desarrollo flexible de arquitecturas de procesamiento polimórfico, así como la integración de sistemas embebidos en dispositivos FPGA que implantan en hardware algoritmos asociados a esquemas de control de orientación de satélites.

Cabe señalar que, no obstante las características innovadoras que presenta el simulador instrumentado para esta investigación doctoral, existe un trabajo de investigación previo en el área de simuladores para control de orientación satelital desarrollado también en México, [56], el cual es coincidente en los siguientes puntos de desarrollo:

- c) Integra una herramienta de seguimiento en realidad virtual, que permite observar con un modelo computacional las maniobras físicas que realiza la plataforma del simulador durante las pruebas de control de orientación experimental y que, en fases de desarrollo posteriores, tendrá la capacidad de incorporar algoritmos de control que permitan formar una plataforma portátil en software para simular y validar en tierra estrategias de control de orientación de satélites.
- d) Como elemento de análisis dinámico integra un modelo matemático, cuyo planteamiento surge de la consideración de la plataforma de simulación como un grupo de cuerpos rígidos, y que será la base para el diseño de controladores basados en modelos de descripción matemática y la validación por medio de simulaciones numéricas en computadora.

En el siguiente capítulo se presenta el desarrollo de la dinámica de la MSA. Este análisis es de suma importancia, no sólo para entender de forma cuantitativa y analítica el movimiento de la plataforma

de simulación, sino también representa el punto de partida para integrar una estrategia de implementación de un SDCO.

Capítulo 3

Modelado dinámico de la plataforma de simulación MSA

3.1. Conceptos matemáticos

La revisión de conceptos básicos de la cinemática y la dinámica en la mecánica clásica para cuerpo rígido es importante para describir y entender el movimiento de la MSA durante pruebas de control de orientación. En los siguientes apartados se presentan algunos conceptos sobre la dinámica de cuerpo rígido y otros temas asociados, los cuales coadyuvarán a entender el planteamiento y desarrollo del modelo dinámico de la MSA descrito en este capítulo.

3.1.1. Concepto de cuerpo rígido

El cuerpo rígido es un caso especial de los sistemas que están constituidos por muchas partículas, es un cuerpo en el cual las distancias entre todos sus componentes permanecen constantes aún bajo la aplicación de una o varias fuerzas o momentos, como se muestra en la figura 3.1. Por lo tanto, un cuerpo rígido conserva su forma durante todo su movimiento [29], [49].

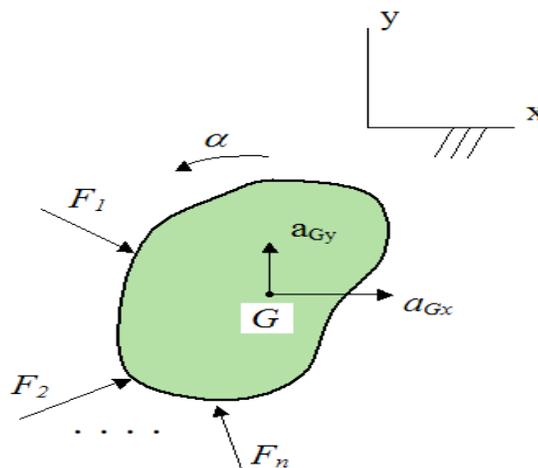


Figura 3.1 Concepto de un cuerpo rígido.

Para un cuerpo rígido se distinguen dos tipos de movimientos: **traslación y rotación**. El primero, figura 3.2 a), se produce cuando todas las partículas describen trayectorias paralelas de tal modo que las líneas que unen dos puntos cualesquiera del cuerpo permanecen siempre paralelas a su posición inicial. El segundo, figura 3.2 b), describe trayectorias circulares alrededor de una línea llamada eje de rotación.

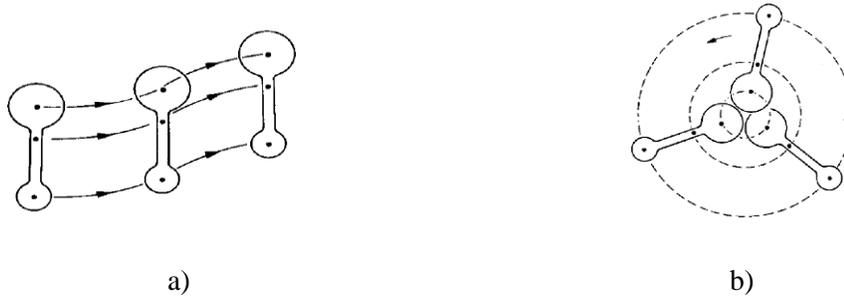


Figura 3.2 a) Movimiento de traslación de un cuerpo rígido b) Movimiento de rotación de un cuerpo rígido.

3.1.2. Representación de la orientación de un cuerpo rígido

Para conocer la orientación de un cuerpo rígido es necesario el planteamiento de sistemas de referencia: inercial y móvil. Los sistemas de referencia son definidos por un grupo de coordenadas en función del espacio en el que se considere su descripción, en el espacio de tres dimensiones se representan por una triada de coordenadas $(0, \xi, \eta, \mu)$ y definen la orientación del cuerpo.

Las coordenadas pueden algunas veces ser referidas directamente como parámetros de orientación o actitud, los cuales al estar definidos en un marco o sistema de referencia, permiten describir la posición espacial relativa a un sistema desde un punto específico del cuerpo. Para fines de descripción de movimiento de un vehículo espacial, éste puede considerarse como un conjunto de cuerpos rígidos unidos entre sí. En esta tesis se abordará el análisis descriptivo de la MSA a partir de este punto de vista.

La selección correcta de la representación del sistema de coordenadas para describir la orientación de un cuerpo rígido ayuda a comprender mejor el movimiento, evitando singularidades matemáticas y geométricas. En algunos casos, la selección del sistema de referencia puede limitar el rango de operación de un sistema controlado, requiriéndole operar dentro de un intervalo no singular de los parámetros de descripción de la orientación [30].

3.1.3. Sistema de referencia y matriz de rotación

Un vector puede expresarse dentro de una variedad de marcos de referencia, cada uno de los cuales se representa por una triada de vectores unitarios ($\bar{e}_1, \bar{e}_2, \bar{e}_3$). Cualquier número de diferentes sistemas de referencia coordenados pueden generarse por medio de la rotación de los ejes ortogonales en torno al origen O , e_1, e_2, e_3 . Por otro lado, es necesario usar un sistema coordenado con un origen diferente, O' , por consiguiente, la transformación de un sistema coordenado a otro consiste en la traslación del origen y la rotación de los ejes en torno del nuevo origen, como se muestra en la figura 3.3, [30].

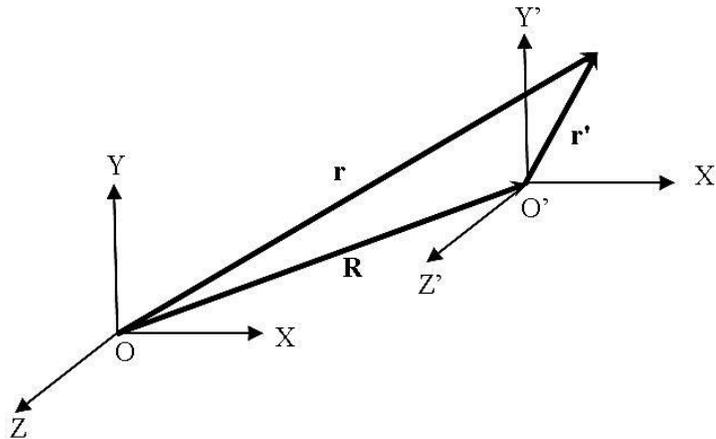


Figura 3.3. Traslación de un sistema coordenado.

La traslación de un origen a otro se aborda especificando un vector de desplazamiento \bar{R} , el cual va desde O hasta O' . Para definir el vector velocidad \bar{v} y aceleración \bar{a} respectivamente, derivamos con respecto al tiempo la ecuación (3.1), obteniendo las siguientes relaciones.

$$\bar{r} = \bar{R} + \bar{r}' \quad , \quad (3.1)$$

$$\bar{v} = \frac{d\bar{R}}{dt} + \bar{v}' \quad , \quad (3.2)$$

$$\bar{a} = \frac{d^2\bar{R}}{dt^2} + \bar{a}' \quad . \quad (3.3)$$

La representación de un vector cualquiera \bar{A} descrito en dos sistemas de referencia (O, X, Y, Z) y (O, X', Y', Z') rotados uno respecto al otro se muestra gráficamente en la figura 3.4.

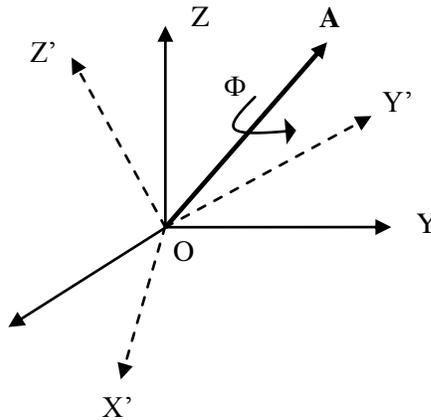


Figura 3.4. Sistemas $OXYZ$ y $OX'Y'Z'$.

Considerando el vector \bar{A} , el cual se expresa en términos de sus componentes en los sistemas coordenados de inercia y rotado, obtenemos:

$$\bar{A} = A_x \bar{i} + A_y \bar{j} + A_z \bar{k} = A'_x \bar{i}' + A'_y \bar{j}' + A'_z \bar{k}' \quad , \quad (3.4)$$

El cual puede escribirse en forma de matriz como :

$$\bar{A} = (\bar{i} \bar{j} \bar{k}) \begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} = (\bar{i}' \bar{j}' \bar{k}') \begin{bmatrix} A'_x \\ A'_y \\ A'_z \end{bmatrix} \quad . \quad (3.5)$$

Con objeto de encontrar la relación entre los dos grupos de componentes, consideramos el producto escalar $\bar{A} \cdot \bar{i}'$, $\bar{A} \cdot \bar{j}'$ y $\bar{A} \cdot \bar{k}'$, resultando en:

$$\begin{bmatrix} A'_x \\ A'_y \\ A'_z \end{bmatrix} = \begin{pmatrix} \bar{i}' \cdot \bar{i} & \bar{i}' \cdot \bar{j} & \bar{i}' \cdot \bar{k} \\ \bar{j}' \cdot \bar{i} & \bar{j}' \cdot \bar{j} & \bar{j}' \cdot \bar{k} \\ \bar{k}' \cdot \bar{i} & \bar{k}' \cdot \bar{j} & \bar{k}' \cdot \bar{k} \end{pmatrix} \begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} \quad . \quad (3.6)$$

Podemos también tomar los productos escalares de las componentes de \bar{A} en (i, j, k) y (i', j', k') para conseguir las siguientes expresiones:

$$\begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} = \begin{pmatrix} \bar{i} \cdot \bar{i}' & \bar{i} \cdot \bar{j}' & \bar{i} \cdot \bar{k}' \\ \bar{j} \cdot \bar{i}' & \bar{j} \cdot \bar{j}' & \bar{j} \cdot \bar{k}' \\ \bar{k} \cdot \bar{i}' & \bar{k} \cdot \bar{j}' & \bar{k} \cdot \bar{k}' \end{pmatrix} \begin{bmatrix} A'_x \\ A'_y \\ A'_z \end{bmatrix} . \quad (3.7)$$

Escribimos entonces la ecuación (3.5) como:

$$\begin{bmatrix} A'_x \\ A'_y \\ A'_z \end{bmatrix} = C \begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} , \quad (3.8)$$

donde C es la siguiente matriz integrada a partir de los cosenos de los ángulos entre los ejes de ambos sistemas de referencia coordenados, y por lo tanto se le llama matriz de cosenos directores, o simplemente matriz de rotación.

$$C \doteq \begin{pmatrix} \bar{i}' \cdot \bar{i} & \bar{i}' \cdot \bar{j} & \bar{i}' \cdot \bar{k} \\ \bar{j}' \cdot \bar{i} & \bar{j}' \cdot \bar{j} & \bar{j}' \cdot \bar{k} \\ \bar{k}' \cdot \bar{i} & \bar{k}' \cdot \bar{j} & \bar{k}' \cdot \bar{k} \end{pmatrix} . \quad (3.9)$$

La transformación de coordenadas, ecuación (3.6), también puede expresarse de la siguiente forma:

$$\begin{bmatrix} \bar{i}' \\ \bar{j}' \\ \bar{k}' \end{bmatrix} = C \begin{bmatrix} \bar{i} \\ \bar{j} \\ \bar{k} \end{bmatrix} . \quad (3.10)$$

3.1.4. Utilización de ángulos de Euler para la orientación de un cuerpo rígido

La descripción de la orientación o actitud de un cuerpo rígido dado un sistema de referencia con respecto a otro, resulta un concepto útil para la determinación de la orientación de vehículos de vuelo (o cualquier otro objeto), en los cuales se acostumbra establecer un sistema coordenado móvil respecto de otro sistema coordenado que generalmente se considera inercial.

El mayor número de rotaciones necesarias para especificar una orientación en un cuerpo rígido corresponde a tres, que denominamos grados de libertad. Algunas formas de rotar un cuerpo son a través de ángulos de Euler, ángulos de Krylov, cuaterniones, etc. [29] y [30]

En este apartado se mencionan solamente los ángulos de Euler y los ejes en torno a los cuales rotan, usando una notación del tipo $(\psi)_3, (\theta)_2, (\phi)_1$, que denotan una rotación $(OXYZ)$ por el ángulo ψ en torno al eje OZ , figura 3.5. Cabe señalar que durante esta rotación se presentan además la orientación intermedia $(OX'Y'Z')$, seguida de una rotación del ángulo θ en torno al eje OY' ,

resultando en $(OX''Y''Z'')$. Finalmente se tiene una rotación del ángulo ϕ respecto al eje OX'' , esto para producir una nueva rotación $(OX'''Y'''Z''')$.

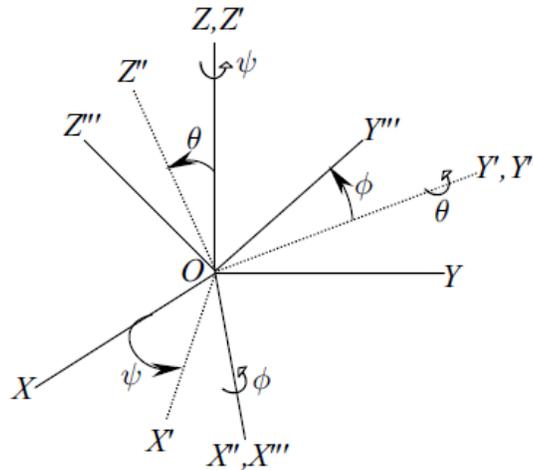


Figura 3.5. Orientación en ángulos de Euler en secuencia $(\psi)_3, (\theta)_2, (\phi)_1$.

Esta descripción de la orientación utilizando ángulos de Euler representa la actitud de una aeronave respecto al horizonte local. Se nombran además a los ángulos yaw (ψ), pitch (θ) y roll (ϕ), figura 3.5.

La matriz de rotación para la orientación $(\psi)_3, (\theta)_2, (\phi)_1$ es la siguiente:

$$C = C_1(\phi) C_2(\theta) C_3(\psi) = \quad (3.11)$$

$$\begin{pmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ (\sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi) & (\sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi) & \sin \phi \cos \theta \\ (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) & (\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) & \cos \phi \cos \theta \end{pmatrix}. \quad (3.12)$$

3.1.5. Utilización de cuaterniones para la representación de la orientación de cuerpo rígido

Los parámetros simétricos de Euler o cuaterniones son una descripción de la orientación de un cuerpo rígido mediante la representación de un eje y un ángulo principal, los cuales tienen la propiedad de estar libres de singularidades, es decir, no contienen funciones indefinidas en algunos puntos de su dominio [30].

Definición: Un cuaternión es un conjunto especial de cuatro parámetros escalares mutuamente dependientes, q_1, q_2, q_3, q_4 , donde los primeros tres elementos forman un vector, en tanto que el cuarto elemento se denomina la parte escalar.

$$q \doteq \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} . \quad (3.13)$$

El cuarto término de (3.13), q_4 representa la parte escalar. La forma de representación de orientación utilizando cuaterniones puede derivarse a partir de su descripción por medio del eje de Euler, e , y el ángulo de rotación principal Φ como se muestra en las ecuaciones (3.14) y (3.15).

$$q_i \doteq e_i \sin \frac{\Phi}{2} \quad (i=1,2,3) \quad , \quad (3.14)$$

$$q_4 \doteq \cos \frac{\Phi}{2} . \quad (3.15)$$

Adicionalmente, las ecuaciones (3.14) y (3.15) deben satisfacer la restricción siguiente:

$$q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1 . \quad (3.16)$$

3.1.6. Energía cinética rotacional

La energía cinética de un cuerpo rígido que rota en torno a un eje con una velocidad angular $\bar{\omega}$, figura 3.6, se define como se muestra en la ecuación (3.17) [31].

$$K = \frac{1}{2} \int_V (\bar{\omega} \times \bar{r})^2 \rho \, dV . \quad (3.17)$$

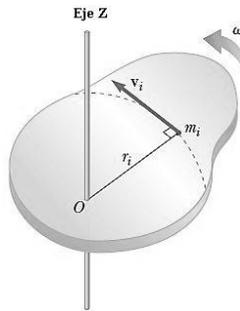


Figura 3.6. Objeto rígido rotando en torno al eje z con velocidad angular $\bar{\omega}$.

3.1.7. Momento de inercia

Para evaluar el momento de inercia de un cuerpo rígido, se considera que el objeto puede dividirse en pequeños elementos de volumen (diferenciales de volumen), cada uno con una masa Δm_i . Usando la definición de la ecuación (3.18) [31], se tiene:

$$I = \sum_i r_i^2 \Delta m_i , \quad (3.18)$$

donde se considera el límite de la suma como $\Delta m_i \rightarrow 0$

Tenemos:

$$I = \lim_{\Delta m_i \rightarrow 0} \sum_i r_i^2 \Delta m_i = \int r^2 dm . \quad (3.19)$$

3.1.8. Torque

Cuando se ejerce una fuerza sobre un cuerpo rígido en torno a un eje, el objeto tiende a rotar en torno a dicho eje. La tendencia de una fuerza a rotar un objeto en torno a algún eje se cuantifica con una cantidad vectorial llamada torque $\vec{\tau}$. Por otro lado, se considera la palanca que se presenta en el eje a través de un punto fijo llamado O, figura 3.7. La fuerza aplicada \vec{F} actúa en un ángulo ϕ respecto a la horizontal [31].

Se define entonces la magnitud del torque asociado con la fuerza \vec{F} por medio de la siguiente expresión (3.20):

$$\vec{\tau} \equiv \vec{r} \times \vec{F} . \quad (3.20)$$

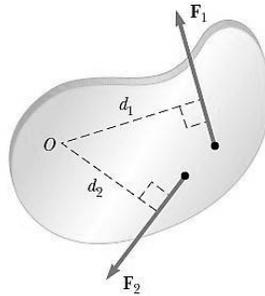


Figura 3.7. La fuerza \vec{F}_1 tiende a causar la rotación del cuerpo a la izquierda en torno a O, y \vec{F}_2 tiende a rotarlo a la derecha.

3.1.9. Relación entre torque y velocidad angular

Si consideramos una diferencial de masa dm del cuerpo rígido rotando en un círculo de radio r bajo la influencia de una fuerza tangencial \vec{F}_t y una fuerza radial \vec{F}_r , como se muestra en la figura 3.8.

La fuerza tangencial proporciona una aceleración tangencial \bar{a}_t y se define en la expresión (3.21) [31]:

$$\bar{F}_t = m \cdot \bar{a}_t \quad , \quad (3.21)$$

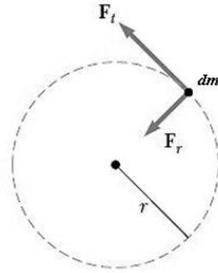


Figura 3.8. Una partícula rotando en torno a un círculo bajo la influencia de una fuerza tangencial \bar{F}_t . Una fuerza \bar{F}_r en la dirección radial debe estar presente para mantener el movimiento circular.

La magnitud del torque en torno al centro del círculo debido a \bar{F}_t es:

$$\bar{\tau} = \bar{r} \cdot \bar{F}_t \quad ,$$

considerando que $\bar{F}_t = dm \cdot \bar{a}_t$, donde \bar{a}_t es aceleración tangencial,

$$\bar{\tau} = \bar{r} \cdot (dm \cdot \bar{a}_t) \quad . \quad (3.22)$$

Debido a que la aceleración tangencial está relacionada con la aceleración angular por medio de la relación $\bar{a}_t = \bar{r} \cdot \bar{\alpha}$, entonces el torque puede expresarse como:

$$\bar{\tau} = (dm \cdot \bar{r} \cdot \bar{\alpha}) \bar{r} = (dm \cdot \bar{r}^2) \bar{\alpha} \quad . \quad (3.23)$$

Recordando que $m\bar{r}^2$ es el momento de inercia de la partícula dm en torno al eje z pasando a través del origen, se tiene:

$$\bar{\tau} = I \cdot \bar{\alpha} \quad , \quad (3.24)$$

de acuerdo con la ecuación (3.24), el torque actuando sobre la partícula es proporcional a su aceleración, y la constante de proporcionalidad es el momento de inercia. La ecuación $\bar{\tau} = I \cdot \bar{\alpha}$ es

la forma rotacional análoga a la segunda ley de movimiento de Newton ($\bar{F} = m \cdot \bar{a}$) para desplazamientos [31].

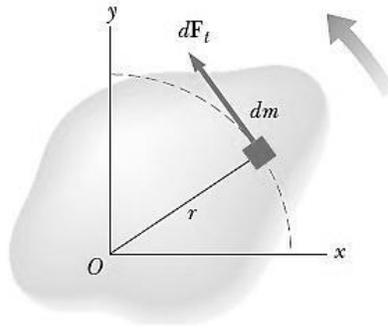


Figura 3.9. Un cuerpo rígido rotando en torno a un eje a través de O.

Si se asocia un sistema de referencia coordenado a un cuerpo rígido que rota, entonces se tiene que cada elemento de masa rota en un círculo en torno al origen, y cada una de ellas tiene una aceleración tangencial \bar{a}_t producida por una fuerza externa tangencial $d\bar{F}_t$. Para un elemento dado, se sabe de la segunda ley de Newton que:

$$d\bar{F}_t = (dm) \bar{a}_t \quad , \quad (3.25)$$

entonces, el torque asociado con la fuerza $d\bar{F}_t$ actúa en torno al origen y está dado por

$$d\bar{\tau} = \bar{r} \cdot d\bar{F}_t = \bar{a}_t \cdot \bar{r} dm \quad , \quad (3.26)$$

ya que $\bar{a}_t = \bar{r} \cdot \bar{\alpha}$, la expresión para $d\bar{\tau}$ se convierte en:

$$d\bar{\tau} = \bar{\alpha} r^2 dm \quad , \quad (3.27)$$

integrando la expresión (3.27) para obtener el torque neto $\Sigma\tau$ en torno a O debido a las fuerzas externas, resulta en (3.28):

$$\Sigma\tau = \int \alpha r^2 dm = \alpha \int r^2 dm \quad . \quad (3.28)$$

3.2. Descripción general de la plataforma de simulación MSA

La MSA es un sistema de simulación de vuelo orbital basado en una plataforma rotacional tipo mesa, la cual flota sobre un colchón de aire creado con flujos generados por una compresora, emulando con ello el efecto de ausencia de fricción, una de las características del medio espacial.

Sobre la plataforma de simulación se instala el mismo esquema de instrumentación que corresponde al SDCO de un satélite real, por lo cual integra sensores de navegación inercial, computadora de a bordo, actuadores activos basados en ruedas inerciales, medios de comunicación, potencia y diversos elementos electrónicos de soporte. En la figura 3.10 se muestra un esquema de la plataforma MSA.

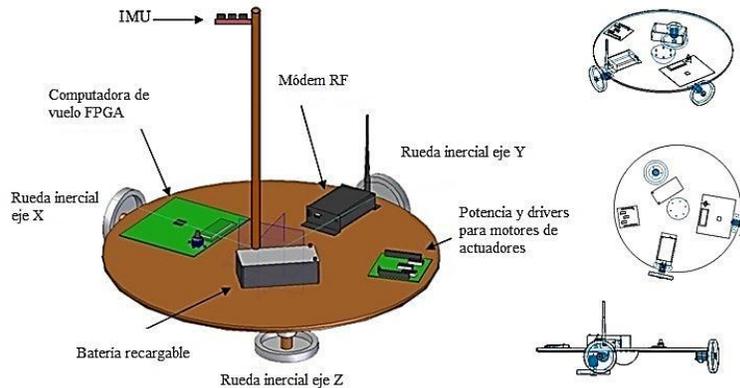


Figura 3.10. Esquema de la plataforma de simulación MSA.

La arquitectura de la MSA es abierta y flexible en términos de integración de sistemas (en software y hardware), la cual tiene capacidad para realizar maniobras rotacionales de 360° en yaw, así como de 20° en pitch y roll.

Uno de sus objetivos es servir como plataforma de I&D para experimentar estrategias de control a nivel de software y hardware. Sobre esta plataforma de desarrollo se pueden montar diversos esquemas de instrumentación del SDCO (sensores, actuadores), validar diversas plataformas de cómputo (MCU, FPGA, DSP) así como esquemas de procesamiento de datos tanto a bordo como utilizando medios externos [6].

3.3. Modelo dinámico de la MSA

El análisis de la MSA visto desde la perspectiva de un sistema físico, el cual puede ser modelado matemáticamente, implica que la descripción de dicho sistema será una representación manipulable en términos de las variables físicas asociadas, útil para una mejor planificación de estrategias de control.

El proceso de modelado o construcción del modelo dinámico de la MSA implica:

- La identificación de las entidades principales del sistema y de sus atributos característicos.

- Identificación y representación de las leyes físicas que gobiernan al sistema físico, lo cual servirá como base para el planteamiento de simulaciones numéricas y esquemas de visualización por medio de modelos en realidad virtual.
- Captación de la naturaleza de las interacciones lógicas del sistema.
- Verificación de que las reglas incorporadas al modelo son una representación válida del sistema que se modela.

3.3.1. Análisis de la cinemática y dinámica de la MSA

Se considera a la MSA como un conjunto compuesto por cuatro cuerpos rígidos: una plataforma móvil y tres ruedas inerciales. Dichos elementos se encuentran rígidamente unidos entre sí y tienen tres grados de libertad rotacionales.

Se definen para cada uno de ellos los siguientes sistemas de referencia, que además se muestran gráficamente en la figura 3.11:

1. El sistema ortonormal $\hat{\mathbf{I}}, \hat{\mathbf{J}}, \hat{\mathbf{K}}$ corresponde al sistema de referencia inercial, cuyo origen coincide con el centro geométrico de la plataforma móvil de la MSA y con sus ejes principales $OX_I Y_I Z_I$.
2. El sistema ortonormal $\hat{\mathbf{i}}_S, \hat{\mathbf{j}}_S, \hat{\mathbf{k}}_S$ corresponde al sistema de referencia móvil que se encuentra fijo a la plataforma móvil del simulador MSA.
3. El sistema ortonormal $(\hat{\mathbf{i}}_{w1}, \hat{\mathbf{j}}_{w1}, \hat{\mathbf{k}}_{w1})$ corresponde al sistema móvil que se encuentra fijo a cada una de las 3 ruedas inerciales (w_1).

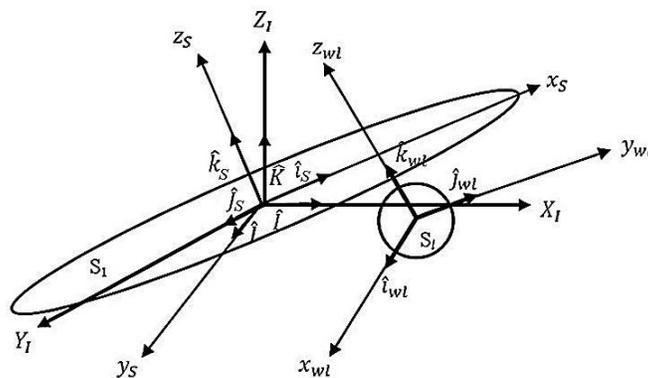


Figura 3.11. Sistemas de referencia definidos sobre la plataforma de la MSA.

Adicionalmente, en la figura 3.11, se muestran los siguientes detalles:

$Ox_I Y_I Z_I$ es el sistema de referencia inercial.

$\hat{i}, \hat{j}, \hat{k}$ son vectores ortogonales del sistema de referencia inercial.

$Ox_S y_S z_S$ es el sistema móvil.

$\hat{i}, \hat{j}, \hat{k}$ son vectores ortogonales del sistema móvil.

En la figura 3.12 se describen relaciones vectoriales de la plataforma y las ruedas inerciales respecto al sistema de referencia inercial. Se presenta el análisis para una sola rueda inercial, mientras que para el caso de las otras dos ruedas inerciales el análisis es análogo.

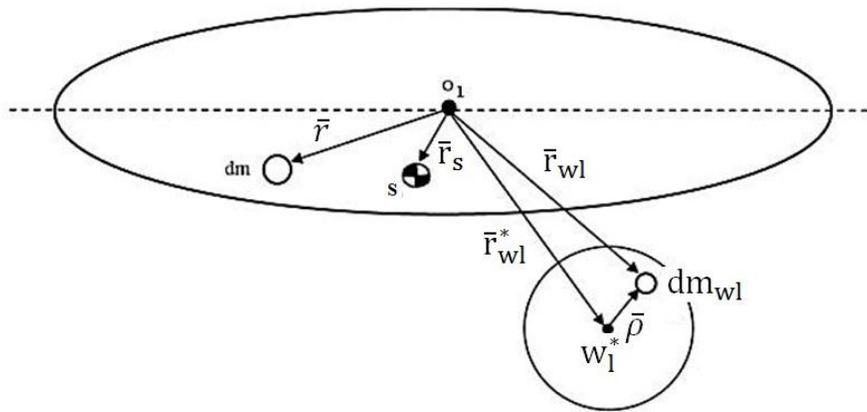


Figura 3.12. Definición de los elementos de la plataforma y la rueda inercial de la MSA.

De la figura 3.12, se establecen las siguientes definiciones:

O_1 centro geométrico de la plataforma.

S centro de masa de la plataforma.

dm diferencial de masa de la plataforma móvil de la MSA.

dm_{wl} diferencial de masa de la rueda inercial.

\bar{r}_{wl} es el vector de posición que va desde el centro geométrico O_1 de la plataforma móvil hasta la diferencial de masa dm_{wl} de la rueda inercial.

w_1^* es el centro de masa de la rueda inercial.

\bar{r}_{wl}^* es el vector centro de masa resultante que va del centro geométrico O_1 de la plataforma móvil hasta el centro de masa de la rueda inercial.

$\bar{\rho}$ es el vector de posición definido desde el centro de masa de la rueda inercial hasta una diferencial de masa dm de la propia rueda.

$\bar{\mathbf{r}}$ es un vector de posición definido desde el centro geométrico de la plataforma móvil hasta la diferencial de masa dm de la plataforma.

3.3.2. Cinemática de la MSA

Para describir el momento angular $\bar{\mathbf{H}}$ de la MSA, se considera a la plataforma S y las tres ruedas inerciales como un conjunto de cuatro cuerpos que tienen un punto en común O_k , donde $k=1,2,3,4$, O_1 corresponde al punto fijo del sistema. Se define la velocidad angular $\bar{\omega}_k$ y la velocidad absoluta $\bar{\mathbf{v}}_k$ con respecto a los puntos O_k . El planteamiento parte de la ecuación (3.29) [29], por lo que se tiene:

$$\bar{\mathbf{H}} = \int_m \bar{\mathbf{r}} \times \dot{\bar{\mathbf{r}}} \, dm \quad , \quad (3.29)$$

donde:

$\dot{\bar{\mathbf{r}}}$ es la velocidad absoluta del centro de masa de la plataforma.

Además, se definen las siguientes relaciones:

$\bar{\omega}_{S_wl}$ es la velocidad angular de la rueda respecto a la plataforma

S_1 es la plataforma móvil del simulador MSA

S_l es una de las 3 ruedas inerciales

$l = 1,2,3$ corresponde al subíndice del número de ruedas inerciales instaladas sobre la plataforma móvil de la MSA.

Si se obtiene la primera derivada de $\bar{\mathbf{r}}$, se tiene que

$$\dot{\bar{\mathbf{r}}} = \bar{\omega}_{1S} \times \bar{\mathbf{r}} \quad , \quad (3.30)$$

donde $\bar{\omega}_{1S}$ es la velocidad angular de la plataforma respecto al sistema de referencia inercial.

A partir de (3.29), se considera que el momento angular de la MSA ($\bar{\mathbf{H}}$) se determinará a partir del análisis de la plataforma S y las S_l ruedas inerciales.

Se define entonces $\bar{\mathbf{H}}_T$ como la suma del momento angular de la plataforma ($\bar{\mathbf{H}}_S$) y de las ruedas inerciales ($\bar{\mathbf{H}}_{wl}$) como:

$$\bar{\mathbf{H}}_T = \bar{\mathbf{H}}_S + \bar{\mathbf{H}}_{wl} \quad . \quad (3.31)$$

Se analizará por partes el momento angular, por lo que sustituyendo (3.30) en (3.29), se tiene:

$$H_S = \int_m \bar{r} \times (\bar{\omega}_{1S} \times \bar{r}) dm \quad , \quad (3.32)$$

integrando, se tiene que:

$$\bar{H}_S = M_S \bar{r}_{S^*} \times (\bar{\omega}_{1S} \times \bar{r}_{S^*}) \quad , \quad (3.33)$$

donde \bar{r}_{S^*} es el vector centro de masa de la plataforma S.

Por propiedades del triple producto vectorial [49], la ecuación (3.33) se puede representar de la siguiente forma:

$$\bar{H}_S = M_S (\bar{\omega}_{1S} (\bar{r}_{S^*} \cdot \bar{r}_{S^*}) - \bar{r}_{S^*} (\bar{r}_{S^*} \cdot \bar{\omega}_{1S})) \quad , \quad (3.34)$$

agrupando:

$$\bar{H}_S = M_S \bar{r}_{S^*}^2 \bar{\omega}_{1S} - M_S \bar{r}_{S^*} (\bar{r}_{S^*} \cdot \bar{\omega}_{1S}) \quad , \quad (3.35)$$

donde $J_S \cdot \bar{\omega}_{1S} = M_S \bar{r}_{S^*}^2 \bar{\omega}_{1S} - M_S \bar{r}_{S^*} (\bar{r}_{S^*} \cdot \bar{\omega}_{1S})$, entonces:

$$\bar{H}_S = J_S \bar{\omega}_{1S} \quad , \quad (3.36)$$

donde M_S corresponde a la masa total de la plataforma y J_S corresponde al tensor de inercia de la plataforma.

Para el momento angular de las ruedas inerciales \bar{H}_{wl} se tiene:

$$\bar{H}_{wl} = \int_m (\bar{r}_{wl}) \times (\dot{\bar{r}}_{wl}) dm_{wl} \quad . \quad (3.37)$$

Sustituyendo \bar{r}_{wl} y $\dot{\bar{r}}_{wl}$ en términos de sus correspondientes vectores de posición, de acuerdo a la figura 3.12, se tiene que:

$$\bar{H}_{wl} = \int_m (\bar{r}_{wl}^* + \bar{\rho}) \times (\dot{\bar{r}}_{wl}^* + (\bar{\omega}_{swl} \times \bar{\rho})) dm_{wl} \quad , \quad (3.38)$$

donde \bar{r}_{wl}^* corresponde al vector de posición que va desde el centro geométrico O_1 de la MSA al w_1^* de masa de la rueda inercial (figura 3.12).

Se define el vector \bar{r}_{wl} como:

$$\bar{r}_{wl} = \bar{r}_{wl}^* + \bar{\rho} \quad , \quad (3.39)$$

obteniendo la primera derivada de (3.39):

$$\dot{\mathbf{r}}_{wl} = \dot{\mathbf{r}}_{wl}^* + \dot{\bar{\rho}} = \dot{\mathbf{r}}_{wl}^* + (\bar{\omega}_{swl} \times \bar{\rho}) \quad ,$$

donde $\dot{\mathbf{r}}_{wl}^* = (\bar{\omega}_{IS} \times \mathbf{r}_{wl}^*)$, entonces

$$\dot{\mathbf{r}}_{wl} = (\bar{\omega}_{IS} \times \mathbf{r}_{wl}^*) + (\bar{\omega}_{swl} \times \bar{\rho}) \quad . \quad (3.40)$$

Sustituyendo (3.39) y (3.40) se define el segundo miembro de la ecuación (3.31) y desarrollando, se tiene

$$\bar{H}_{wl} = \int_m [\mathbf{r}_{wl}^* \times (\dot{\mathbf{r}}_{wl}^* + (\bar{\omega}_{swl} \times \bar{\rho})) + \bar{\rho} \times (\dot{\mathbf{r}}_{wl}^* + (\bar{\omega}_{swl} \times \bar{\rho}))] dm_{wl} \quad . \quad (3.41)$$

Resolviendo la integral de la ecuación (3.41), se obtiene:

$$\bar{H}_{wl} = \sum_{l=1}^3 M_{wl} [\mathbf{r}_{wl}^* \times (\dot{\mathbf{r}}_{wl}^* + (\bar{\omega}_{swl} \times \bar{\rho})) + \bar{\rho} \times (\dot{\mathbf{r}}_{wl}^* + (\bar{\omega}_{swl} \times \bar{\rho}))] \quad , \quad (3.42)$$

donde $l = 1,2,3$ corresponde al número de ruedas inerciales instaladas sobre la plataforma y M_{wl} corresponde a la masa de las ruedas inerciales. Desarrollando (3.42), se obtiene:

$$\bar{H}_{wl} = \sum_{l=1}^3 M_{wl} [\mathbf{r}_{wl}^* \times ((\bar{\omega}_{IS} \times \mathbf{r}_{wl}^*) + (\bar{\omega}_{swl} \times \bar{\rho})) + \bar{\rho} \times ((\bar{\omega}_{IS} \times \mathbf{r}_{wl}^*) + (\bar{\omega}_{swl} \times \bar{\rho}))] \quad , \quad (3.43)$$

agrupando:

$$\bar{H}_{wl} = \sum_{l=1}^3 M_{wl} [\mathbf{r}_{wl}^* \times (\bar{\omega}_{IS} \times \mathbf{r}_{wl}^*) + \mathbf{r}_{wl}^* \times (\bar{\omega}_{swl} \times \bar{\rho}) + \bar{\rho} \times (\bar{\omega}_{IS} \times \mathbf{r}_{wl}^*) + \bar{\rho} \times (\bar{\omega}_{swl} \times \bar{\rho})] \quad , \quad (3.44)$$

simplificando

$$\bar{H}_{wl} = \sum_{l=1}^3 [J_{wl}^* \bar{\omega}_{IS} + M_{wl} \mathbf{r}_{wl}^* \times (\bar{\omega}_{swl} \times \bar{\rho}) + M_{wl} \bar{\rho} \times (\bar{\omega}_{IS} \times \mathbf{r}_{wl}^*) + J_{wl} \bar{\omega}_{swl}] \quad . \quad (3.45)$$

El momento angular total de la MSA será entonces:

$$\bar{H}_T = J_S \bar{\omega}_{IS} + \sum_{l=1}^3 [J_{wl}^* \bar{\omega}_{IS} + M_{wl} \mathbf{r}_{wl}^* \times (\bar{\omega}_{swl} \times \bar{\rho}) + M_{wl} \bar{\rho} \times (\bar{\omega}_{IS} \times \mathbf{r}_{wl}^*) + J_{wl} \bar{\omega}_{swl}] \quad , \quad (3.46)$$

reagrupando términos de la ecuación (3.46), se tiene:

$$\bar{H}_T = J_S \bar{\omega}_{IS} + \sum_{l=1}^3 J_{wl}^* \bar{\omega}_{IS} + \sum_{l=1}^3 [M_{wl} \mathbf{r}_{wl}^* \times (\bar{\omega}_{swl} \times \bar{\rho})] + \sum_{l=1}^3 [M_{wl} \bar{\rho} \times (\bar{\omega}_{IS} \times \mathbf{r}_{wl}^*)] + \sum_{l=1}^3 J_{wl} \bar{\omega}_{swl} \quad , \quad (3.47)$$

finalmente, (3.31) la reescribimos como:

$$\bar{H}_T = (J_S + J_{wl}^*) \bar{\omega}_{IS} + \sum_{l=1}^3 [M_{wl} \mathbf{r}_{wl}^* \times (\bar{\omega}_{swl} \times \bar{\rho}) + M_{wl} \bar{\rho} \times (\bar{\omega}_{IS} \times \mathbf{r}_{wl}^*) + J_{wl} \bar{\omega}_{swl}] \quad . \quad (3.48)$$

En la ecuación (3.48), J_S corresponde al tensor de inercia de la plataforma móvil, mientras que J_{wl}^* y J_{wl} corresponden al tensor de inercia de las ruedas inerciales.

Cabe señalar que la plataforma móvil está referida al sistema inercial (I) y las ruedas inerciales están referidas al sistema móvil de la plataforma suspendida en aire.

3.3.3. Dinámica de la MSA

Utilizando el teorema $\frac{d}{dt} \bar{H} = M$, donde M es el momento de fuerzas [49], se considera que el movimiento rotacional de la plataforma de la MSA es únicamente en sus tres ejes principales referidos al marco de referencia inercial y cuyo origen coincide con el centro geométrico de la plataforma, figura 3.11.

Se define entonces la derivada con respecto al tiempo del momento angular en un punto fijo O_1 , que será igual al momento de fuerza del sistema, como sigue [29], [49]:

$$\dot{\bar{H}} = \int_m \bar{r} \times \ddot{\bar{r}} \, dm \quad . \quad (3.49)$$

Para el caso de la plataforma MSA se asume que el torque total $\dot{\bar{H}}_T$ será el resultado de la contribución del torque de la plataforma $\dot{\bar{H}}_S$ y de las ruedas inerciales $\dot{\bar{H}}_{wl}$. Por lo que, aplicando el principio de conservación, se plantea la expresión (3.50) [26]

$$\dot{\bar{H}}_T = \dot{\bar{H}}_S + \dot{\bar{H}}_{wl} \quad , \quad (3.50)$$

obteniendo para la plataforma $\dot{\bar{H}}_S$ las siguientes expresiones:

$$\dot{\bar{r}} = \bar{\omega}_{IS} \times \bar{r} \quad , \quad (3.51)$$

$$\ddot{\bar{r}} = \dot{\bar{\omega}}_{IS} \times \bar{r} + \bar{\omega}_{IS} \times \dot{\bar{r}} = \dot{\bar{\omega}}_{IS} \times \bar{r} + \bar{\omega}_{IS} \times (\bar{\omega}_{IS} \times \bar{r}) \quad , \quad (3.52)$$

sustituyendo las ecuaciones (3.51) y (3.52) en la ecuación (3.49) [49], se tiene:

$$\dot{\bar{H}}_S = \int_m \bar{r} \times (\dot{\bar{\omega}}_{IS} \times \bar{r} + \bar{\omega}_{IS} \times (\bar{\omega}_{IS} \times \bar{r})) \, dm \quad , \quad (3.53)$$

agrupando

$$\dot{\bar{H}}_S = \int_m \bar{r} \times (\dot{\bar{\omega}}_{IS} \times \bar{r}) \, dm + \int_m \bar{r} \times \bar{\omega}_{IS} \times (\bar{\omega}_{IS} \times \bar{r}) \, dm \quad . \quad (3.54)$$

De la segunda integral de la expresión (3.54), aplicando una permutación sobre el término $\bar{\omega}_{1S}$ se tiene

$$\bar{r} \times \bar{\omega}_{1S} \times (\bar{\omega}_{1S} \times \bar{r}) = \bar{\omega}_{1S} \times \bar{r} \times (\bar{\omega}_{1S} \times \bar{r}) \quad , \quad (3.55)$$

sustituyendo la permutación de la expresión (3.55) en las integrales (3.54) y resolviéndolas, se tiene:

$$\dot{\bar{H}}_S = M_S \bar{r} \times (\dot{\bar{\omega}}_{1S} \times \bar{r}) + M_S [\bar{\omega}_{1S} \times (\bar{r} \times (\bar{\omega}_{1S} \times \bar{r}))] \quad , \quad (3.56)$$

resolviendo los triples productos vectoriales indicados, se obtiene

$$\bar{r} \times (\dot{\bar{\omega}}_{1S} \times \bar{r}) = \bar{r}^2 \cdot \dot{\bar{\omega}}_{1S} - \bar{r} (\dot{\bar{\omega}}_{1S} \cdot \bar{r}) = \bar{r}^2 \cdot \dot{\bar{\omega}}_{1S} \quad , \quad (3.57)$$

donde el producto indicado $\dot{\bar{\omega}}_{1S} \cdot \bar{r} = 0$. Por otro lado, se tiene que

$$\bar{r} \times (\bar{\omega}_{1S} \times \bar{r}) = \bar{r}^2 \cdot \bar{\omega}_{1S} - \bar{r} (\bar{\omega}_{1S} \cdot \bar{r}) = \bar{r}^2 \cdot \bar{\omega}_{1S} \quad , \quad (3.58)$$

donde el producto indicado $\bar{\omega}_{1S} \cdot \bar{r} = 0$.

Sustituyendo (3.57) y (3.58) en (3.56), se tiene

$$\dot{\bar{H}}_S = M_S \bar{r}^2 \cdot \dot{\bar{\omega}}_{1S} + M_S (\bar{\omega}_{1S} \times \bar{r}^2 \cdot \bar{\omega}_{1S}) \quad , \quad (3.59)$$

$$\ddot{\bar{H}}_S = J_S \dot{\bar{\omega}}_{1S} + M_S \bar{\omega}_{1S} \times J_S \bar{\omega}_{1S} \quad . \quad (3.60)$$

Resolviendo para las ruedas inerciales $\dot{\bar{H}}_{W1}$, se tiene

$$\bar{r}_{w1} = \bar{r}_{w1}^* + \bar{\rho} \quad , \quad (3.61)$$

realizando la primera y segunda derivada de (3.61), resulta

$$\dot{\bar{r}}_{w1} = \dot{\bar{r}}_{w1}^* + \dot{\bar{\rho}} = \dot{\bar{r}}_{w1}^* + (\bar{\omega}_{sw1} \times \bar{\rho}) \quad , \quad (3.62)$$

$$\ddot{\bar{r}}_{w1} = \ddot{\bar{r}}_{w1}^* + \ddot{\bar{\rho}} = \ddot{\bar{r}}_{w1}^* + \dot{\bar{\omega}}_{sw1} \times \bar{\rho} + \bar{\omega}_{sw1} \times (\bar{\omega}_{sw1} \times \bar{\rho}) \quad . \quad (3.63)$$

Sustituyendo (3.61), (3.62) y (3.63) en la ecuación (3.49), se obtiene

$$\ddot{\bar{H}}_{W1} = \int_m (\bar{r}_{w1}^* + \bar{\rho}) \times (\ddot{\bar{r}}_{w1}^* + \dot{\bar{\omega}}_{sw1} \times \bar{\rho} + \bar{\omega}_{sw1} \times (\bar{\omega}_{sw1} \times \bar{\rho})) dm \quad , \quad (3.64)$$

$$\dot{\mathbf{H}}_{W1} = \int_m \dot{\mathbf{r}}_{w1}^* \times (\ddot{\mathbf{r}}_{w1}^* + \dot{\bar{\omega}}_{sw1} \times \bar{\rho} + \bar{\omega}_{sw1} \times (\bar{\omega}_{sw1} \times \bar{\rho})) dm + \int_m \bar{\rho} \times (\ddot{\mathbf{r}}_{w1}^* + \dot{\bar{\omega}}_{sw1} \times \bar{\rho} + \bar{\omega}_{sw1} \times (\bar{\omega}_{sw1} \times \bar{\rho})) dm , \quad (3.65)$$

$$\dot{\mathbf{H}}_{W1} = \sum_{i=1}^3 M_i [\dot{\mathbf{r}}_{w1}^* \times (\ddot{\mathbf{r}}_{w1}^* + \dot{\bar{\omega}}_{sw1} \times \bar{\rho} + \bar{\omega}_{sw1} \times (\bar{\omega}_{sw1} \times \bar{\rho}))] + M_i [\bar{\rho} \times (\ddot{\mathbf{r}}_{w1}^* + \dot{\bar{\omega}}_{sw1} \times \bar{\rho} + \bar{\omega}_{sw1} \times (\bar{\omega}_{sw1} \times \bar{\rho}))]. \quad (3.66)$$

Considerando que $\bar{\rho} = \bar{\mathbf{r}}_{cm}$, donde $\bar{\mathbf{r}}_{cm}$ es el centro de masa, permite simplificar algunos términos de (3.66), entonces:

$$\dot{\mathbf{H}}_{W1} = \sum_{i=1}^3 M_i [\dot{\mathbf{r}}_{w1}^* \times (\ddot{\mathbf{r}}_{w1}^* + \dot{\bar{\omega}}_{sw1} \times \bar{\rho} + \bar{\omega}_{sw1} \times (\bar{\omega}_{sw1} \times \bar{\rho}))] + \bar{\rho} \times \ddot{\mathbf{r}}_{w1}^* + \bar{\rho} \times \dot{\bar{\omega}}_{sw1} \times \bar{\rho} + \bar{\rho} \times \bar{\omega}_{sw1} \times (\bar{\omega}_{sw1} \times \bar{\rho})]. \quad (3.67)$$

Aplicando una permutación del producto vectorial, tenemos:

$$\bar{\rho} \times \bar{\omega}_{sw1} \times (\bar{\omega}_{sw1} \times \bar{\rho}) = \bar{\omega}_{sw1} \times \bar{\rho} \times (\bar{\omega}_{sw1} \times \bar{\rho}) , \quad (3.68)$$

sustituyendo (3.68) en la ecuación (3.67), se obtiene:

$$\dot{\mathbf{H}}_{W1} = \sum_{i=1}^3 M_i [\dot{\mathbf{r}}_{w1}^* \times (\bar{\omega}_{sw1} \times (\bar{\omega}_{sw1} \times \bar{\rho}))] + J_{w1} \dot{\bar{\omega}}_{sw1} + \bar{\omega}_{sw1} \times J_{w1} \bar{\omega}_{sw1} , \quad (3.69)$$

sustituyendo (3.69) y (3.60) en (3.50), se tiene:

$$\dot{\mathbf{H}}_T = J_S \dot{\bar{\omega}}_{IS} + M_S \bar{\omega}_{IS} \times J_S \bar{\omega}_{IS} + \sum_{i=1}^3 M_i [\dot{\mathbf{r}}_{w1}^* \times (\bar{\omega}_{sw1} \times (\bar{\omega}_{sw1} \times \bar{\rho}))] + J_{w1} \dot{\bar{\omega}}_{sw1} + \bar{\omega}_{sw1} \times J_{w1} \bar{\omega}_{sw1}] = \bar{\mathbf{M}}_T . \quad (3.70)$$

Donde $\bar{\mathbf{M}}_T$ son los momentos totales que perturban la estabilidad de la plataforma móvil de la MSA. Es importante destacar que $\bar{\mathbf{M}}_T$ resulta de la contribución de los momentos de fuerza externos ($\bar{\mathbf{M}}_{FEX}$) y los momentos de fuerza internos ($\bar{\mathbf{M}}_{IN}$), como se plantea en la ecuación (3.71)

$$\bar{\mathbf{M}}_{FEX} + \bar{\mathbf{M}}_{IN} = \bar{\mathbf{M}}_T , \quad (3.71)$$

considerando que el sistema es rígido, entonces $\bar{\mathbf{M}}_{IN} = 0$ y la única fuerza externa está dada por

$$\bar{\mathbf{M}}_{FEX} = \bar{\mathbf{r}} \times \bar{\mathbf{F}} . \quad (3.72)$$

Sustituyendo (3.72) en (3.70), se tiene

$$J_S \dot{\bar{\omega}}_{IS} + M_S \bar{\omega}_{IS} \times J_S \bar{\omega}_{IS} + \sum_{i=1}^3 M_i [\dot{\mathbf{r}}_{w1}^* \times (\bar{\omega}_{sw1} \times (\bar{\omega}_{sw1} \times \bar{\rho}))] + J_{w1} \dot{\bar{\omega}}_{sw1} + \bar{\omega}_{sw1} \times J_{w1} \bar{\omega}_{sw1}] = \bar{\mathbf{r}} \times \bar{\mathbf{F}} , \quad (3.73)$$

Se considerará que la única perturbación externa (M_{FEX}) que afecta al sistema es la fuerza gravitacional terrestre, por lo tanto

$$\bar{F} = mg\bar{\gamma} \quad , \quad (3.74)$$

donde $\bar{\gamma}$ es el vector unitario vertical al centro de la tierra (gravitacional).

Sustituyendo (3.74) en (3.73), se obtiene

$$J_S \dot{\bar{\omega}}_{IS} + M_S \bar{\omega}_{IS} \times J_S \bar{\omega}_{IS} + \sum_{i=1}^3 M_i [\bar{r}_{wi}^* \times (\bar{\omega}_{swi} \times (\bar{\omega}_{swi} \times \bar{\rho}))] + J_{wi} \dot{\bar{\omega}}_{swi} + \bar{\omega}_{swi} \times J_{wi} \bar{\omega}_{swi} = \bar{F} \times mg\bar{\gamma} \quad . \quad (3.75)$$

La ecuación (3.75) representa el **modelo de la dinámica de la MSA**, donde M_S es la masa total de la plataforma móvil y M_i es la masa total de las ruedas inerciales.

Es importante señalar que para estudios posteriores de controlabilidad y estabilidad del sistema, así como para el planteamiento del controlador *ad hoc* a la arquitectura del simulador MSA, la ecuación (3.75) es fundamental. Para tal efecto, esta expresión deberá ser proyectada a los tres ejes principales de la MSA, representada en términos de ángulos de Euler o en cuaterniones, computacionalmente como mejor convenga, y linealizarla, para después obtener sus ecuaciones de estado y realizar los estudios y planteamientos correspondientes.

3.3.4. Tensores de inercia de la MSA

Para el caso de la plataforma móvil, el tensor de inercia J_s es de la forma:

$$J_s = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}$$

Particularmente el tensor de inercia de la MSA instrumentada, calculado por medio de software CAD (Solid Works), considerado a partir del centro de masa, figura 3.12, es:

$$I_{MSA} = \begin{bmatrix} 0.96246 & 0.27118 & 0.011757 \\ -0.27131 & 0.96216 & 0.02456 \\ -0.046519 & -0.026829 & 0.99963 \end{bmatrix} [\text{kg}\cdot\text{m}^2] \quad (3.76)$$

Algunas vistas del modelo CAD desarrollado para la MSA, a partir del cual se obtuvieron los datos anteriores, se muestran en la figura 3.13.

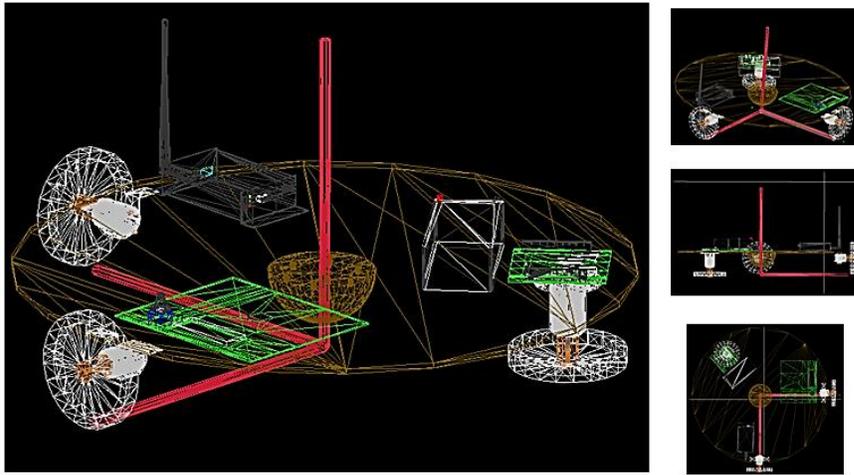


Figura 3.13. Vistas del modelo CAD para el cálculo de parámetros físicos de la MSA.

Para el caso de las ruedas inerciales los tensores de inercia J_{w1} , calculados mediante Solid Edge, son de la forma

$$\begin{aligned}
 J_{\text{rueda}_x} &= \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\
 J_{\text{rueda}_y} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & 0 \end{bmatrix} \\
 J_{\text{rueda}_z} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}
 \end{aligned} \tag{3.77}$$

Cada componente de la matriz corresponde a los momentos de inercia respecto a cada uno de los ejes principales de la MSA, cuyos valores numéricos obtenidos de un modelo CAD, son:

$$I_{xx} = I_{yy} = I_{zz} = 0.0038 \text{ kg} \cdot \text{m}^2$$

El desarrollo del modelo de la dinámica de la MSA y todos los conceptos asociados descritos en este capítulo, representan la base teórica fundamental para entender el funcionamiento del simulador durante las maniobras de control de orientación. Pero además, representa la oportunidad para detectar otras formas de validación, como por ejemplo a través de la simulación y validación por medio de modelos en realidad virtual, que serán abordados en capítulos más adelante, pero que *a priori* representan una forma viable y rápida de implementar esquemas de validación en tierra.

En el siguiente capítulo se presentarán los detalles de la instrumentación de la MSA en sus dos vertientes principales de desarrollo: hardware y software. Dicho esquema se basa en la instrumentación de un satélite y está integrado de acuerdo al esquema previsto en la técnica de cosimulación *hardware-in-the-loop*, la cual permite la interacción entre elementos reales de la instrumentación y elementos externos que complementan al sistema, operando ambos en tiempo real.

Capítulo 4

Instrumentación de la MSA

4.1. Introducción

La instrumentación de la MSA se integró tomando como base el esquema del SDCO de un satélite real, el cual está integrado por los módulos que se muestran en la figura 4.1. En dicha figura, se describen las funciones y/o componentes que integran a cada uno de los módulos (sensores de navegación inercial, computadora a bordo y actuadores), y se muestra el flujo lógico de operaciones entre los bloques y equipos [1].

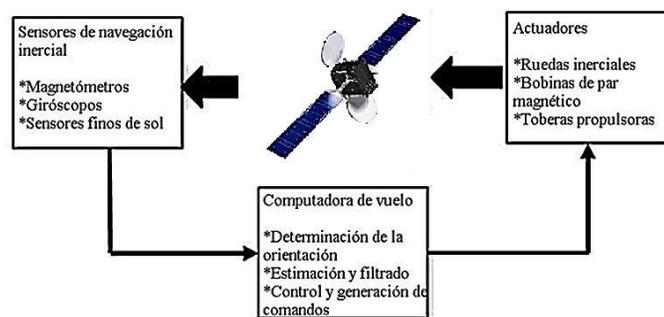


Figura 4.1. Componentes básicos y flujo de operaciones de un SDCO.

La selección de la instrumentación del SDCO es función de muchos factores, incluyendo objetivos de la misión, órbita y recursos disponibles del sistema. Los sistemas de estabilización satelital generalmente utilizan ruedas inerciales como fuentes de torques de control y se adaptan muy bien en aplicaciones de satélites pequeños debido a su probado desempeño, relativa simplicidad, versatilidad y capacidad de proporcionar control de apuntamiento de alta precisión, aunque es claro que un factor muy importante son los algoritmos asociados.

El proceso de definición del esquema de instrumentación del SDCO para un satélite, en términos de dimensionamiento y selección de componentes, es una labor complicada, la cual depende de varios parámetros. En la figura 4.2 se muestran los factores que se deben considerar en la selección de elementos de la instrumentación del SDCO [19].



Figura 4.2. Factores que influyen en la selección de la implementación de un SDCO.

Englobando los parámetros de la figura 4.2, es posible detectar el nicho de oportunidad para el desarrollo de herramientas que permitan la validación previa de un esquema de SDCO. Las cuales, además de brindar la flexibilidad al diseñador en términos de la selección de hardware y la depuración del software, deben ofrecer un mayor nivel de confianza a la misión. Dichas herramientas actualmente existen en forma de simuladores en hardware y software.

Particularmente en el caso de los simuladores en hardware, el principal reto está asociado a los algoritmos, sensores de navegación inercial y a la respuesta de los actuadores. Por tal motivo, se han instrumentado simuladores, como el que se describe en este capítulo, los cuales además de contar con los bloques fundamentales del SDCO: sensores, computadora a bordo y actuadores, utilizan un colchón de aire para reproducir el efecto de fricción cero, con el cual es posible validar operativamente en tierra el desempeño de diversas estrategias de control de orientación antes de ser incluidas dentro del SDCO de la plataforma satelital.

En este capítulo se aborda la descripción de la instrumentación integrada y desarrollada como parte del trabajo experimental de esta investigación, y que se encuentra instalada a bordo de la plataforma móvil del simulador MSA del IINGEN-UNAM. Es importante subrayar que en el planteamiento inicial de este simulador se persiguió al menos alcanzar una precisión de apuntamiento de 5° [48]. Sin embargo, debido a restricciones técnicas, el sistema desarrollado conserva un carácter de plataforma abierta.

4.2. Esquema de instrumentación de la MSA

La figura 4.3 muestra la estructura y describe las tareas de cada bloque del simulador implementado en esta investigación, el cual está basado en el esquema de cosimulación HIL [3] y que está particionado en dos segmentos principales, una PC externa y una tarjeta de desarrollo FPGA como computadora a bordo.

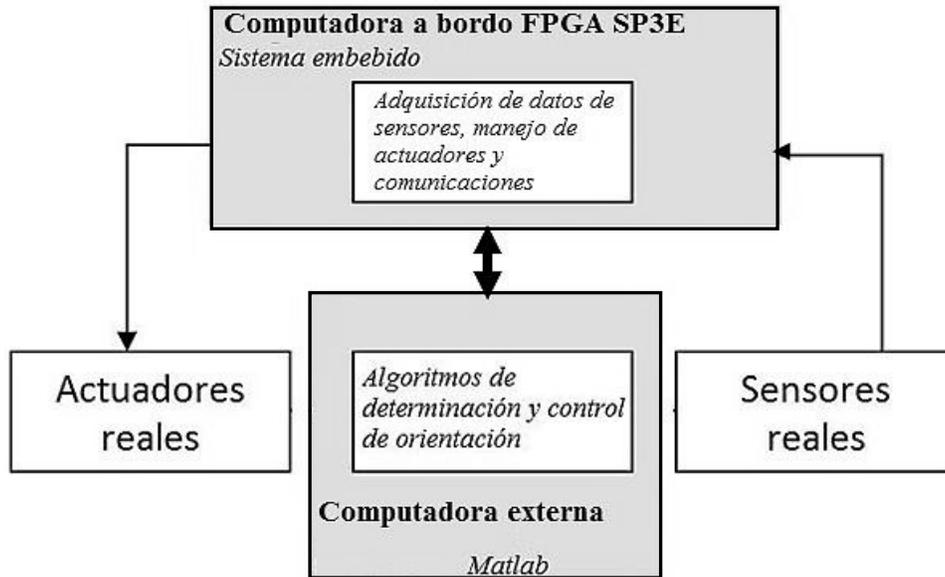


Figura 4.3. Esquema de la MSA desarrollada de acuerdo a la técnica HIL.

En la PC externa (computadora de simulación), que se utiliza como plataforma de coprocesamiento externo, se implementaron bloques con funciones en Matlab® que integran un sistema de simulación en software y que representan la parte más complicada de la lógica del SDCO. Esto permitirá la evaluación operativa de software del sistema y su interacción con hardware externo, vinculando además la evaluación con elementos de la dinámica del sistema físico bajo prueba.

En el caso de la computadora a bordo basada en FPGA, se implementó en una tarjeta de desarrollo SP3E un sistema embebido basado en núcleos de procesadores y periféricos que ejecutarán las tareas (a bordo de la MSA) de adquisición de datos, comunicaciones y control de actuadores mediante señales moduladas en ancho de pulso (PWM). Este sistema embebido tendrá la capacidad de integrar (en posteriores etapas de desarrollo del simulador) en un solo dispositivo, soluciones

particionadas en hardware y software embebido, tanto de la lógica de todo el SDCO como de algún otro subsistema de la plataforma de un satélite.

La técnica HIL considerada en el desarrollo de la MSA, desde el punto de vista de integración, es una combinación sinérgica entre un prototipo de simulación y un prototipo físico. En [25] se define un simulador HIL como un conjunto de dispositivos y equipos que emulan a un sistema completo mediante la inmersión de réplicas físicas de algunos de sus subsistemas dentro de una simulación virtual en lazo cerrado. En esta definición resalta la interacción en lazo cerrado o bidireccional entre sus componentes virtuales y físicas, característica clave del esquema de desarrollo de sistemas basado en HIL [30].

El esquema de instrumentación de la MSA basado en la estructura HIL, más allá de las ventajas técnicas que ofrece en términos de aceleración en la implementación del SDCO y en la obtención de resultados experimentales, ha permitido la interacción en tiempo real de módulos desarrollados en hardware y software.

En cuanto al hardware, facilita la evaluación de elementos de la instrumentación (sensores, plataforma de cómputo y actuadores). En cuanto al software, contribuye en el proceso de depuración y optimización de programas y códigos de firmware. Esto redundaría en la integración de un sistema de control mucho más confiable, clave en el desarrollo de sistemas espaciales, en los cuales la ejecución de rutinas de depuración o mantenimiento una vez en órbita, es prácticamente imposible o bien, en caso de ser factible, implica altos costos económicos y riesgos.

En la figura 4.4 se muestra un diagrama general del esquema de instrumentación integrado en la MSA.

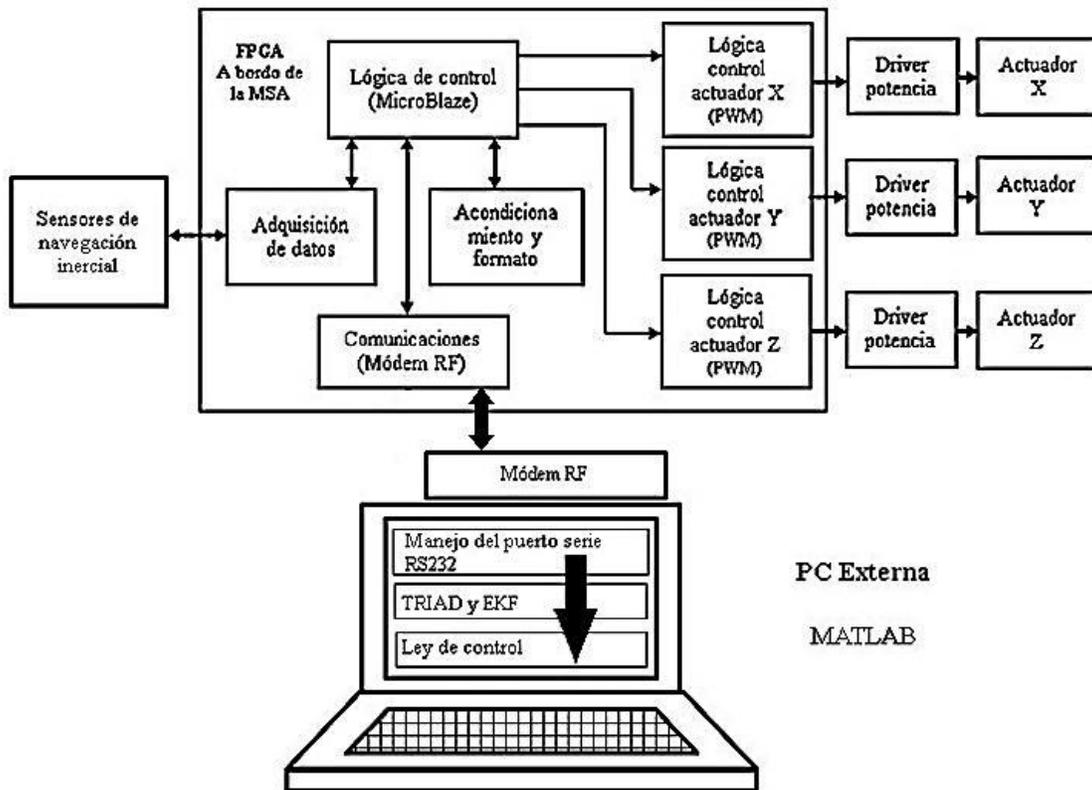


Figura 4.4. Instrumentación de la MSA.

Para la integración de la MSA se consideraron dos bloques principales:

- Segmento de datos y manejo de actuadores.** Está integrado por la plataforma móvil de la MSA y toda la instrumentación asociada (tarjeta FPGA, sensores, actuadores, comunicaciones y potencia). En este bloque se integró una plataforma FPGA, dentro de la cual se desarrolló un sistema embebido que realiza tareas de adquisición de datos temporizada (de intervalo ajustable) de los sensores de navegación inercial por medio del protocolo I2C y la generación de comandos de control para los actuadores por medio de señales PWM.
- Segmento de procesamiento.** Está integrado en una PC externa, en la cual se programan en MATLAB® funciones personalizadas que resuelven los algoritmos de determinación y control de la orientación (TRIAD, EKF y CONTROL). También se generan las tramas que contienen comandos de control para los actuadores, los cuales se transmiten al FPGA a bordo de la MSA para su decodificación y generación de señales de control para los *drivers* de potencia de cada uno de los motores de los actuadores.

Ambos segmentos están enlazados por medio de radio módems, figura 4.5.

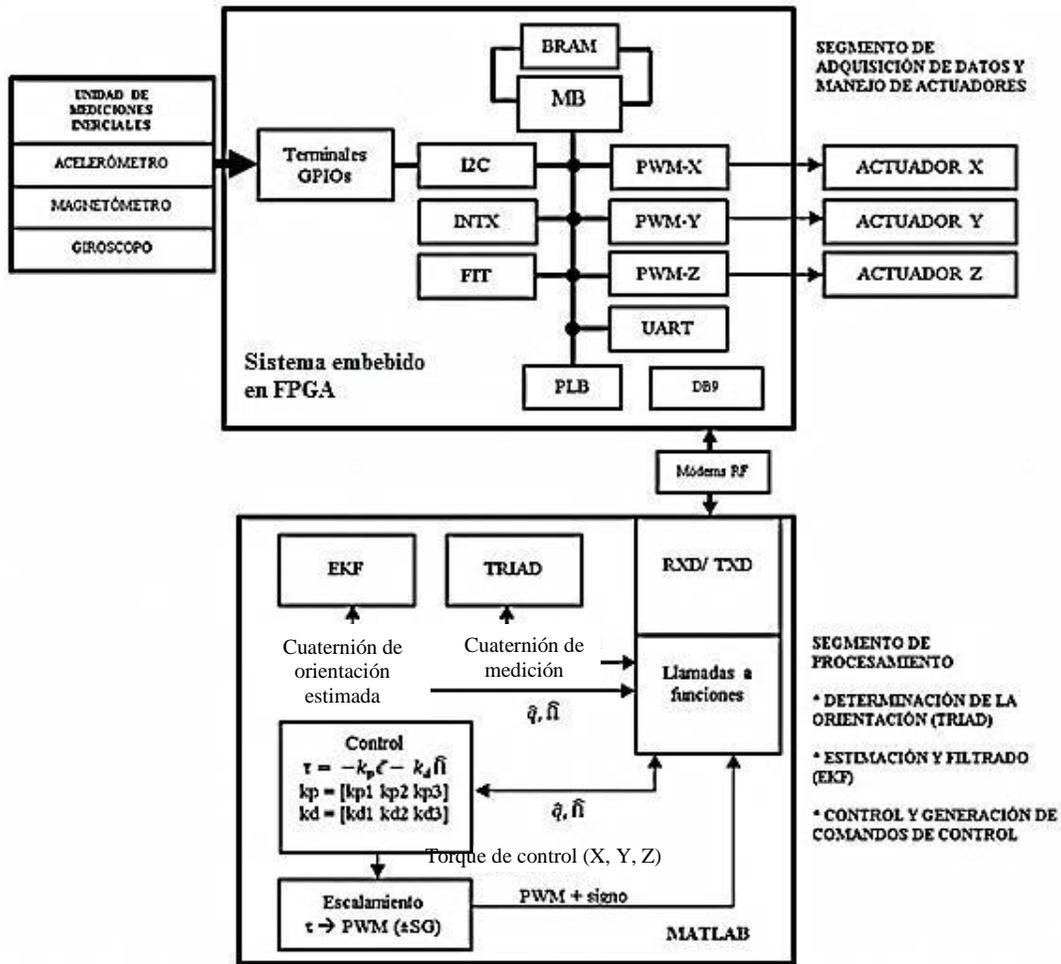


Figura 4.5. Diagrama del simulador MSA utilizando el esquema HIL.

4.3. Hardware de la MSA

El hardware que integra la instrumentación de la MSA de acuerdo con el esquema HIL, figura 4.5, está agrupado a su vez en cinco bloques principales: sensores, actuadores, plataforma de cómputo, potencia y comunicaciones. En los siguientes apartados se hace una descripción de cada uno de ellos.

4.3.1. Sensores de navegación inercial

Uno de los componentes más importantes dentro de la instrumentación de la MSA son los sensores de navegación inercial. Los datos que proporcionan son importantes para determinar la orientación de la plataforma móvil respecto a un marco de referencia inercial, así como para el cálculo de la

estimación de variables de estado del sistema. Las cuales son necesarias para la operación de algoritmos que forman parte del esquema de control de orientación.

La búsqueda, identificación y selección de los sensores de navegación inercial que fueron integrados dentro del esquema de instrumentación de la MSA, estuvo basada, en primer lugar, en esquemas presentados en fuentes como [1], [2] y [47], donde se identifican particularmente sensores tales como: magnetómetros, acelerómetros, giróscopos, sensores finos de sol, seguidores de estrellas, etc. Adicionalmente, la identificación del tipo de sensor se basó en los requerimientos de los propios algoritmos que se implementaron, particularmente determinación de la orientación (TRIAD) así como estimación y filtrado (EKF).

Como complemento a los requerimientos de los algoritmos, se consideró también para la selección de los sensores características importantes para su integración en la instrumentación de la MSA, como por ejemplo: protocolo de comunicación (preferentemente estándar), potencia de consumo reducido y dimensiones físicas mínimas.

Para ello una de las mejores opciones resultó una Unidad de Mediciones Inerciales, IMU por sus siglas en inglés. La IMU es un dispositivo electrónico que mide parámetros como velocidad, orientación y fuerza gravitacional, usando para ello una combinación de acelerómetros, giróscopos y algunas veces magnetómetros contenidos en una sola tarjeta electrónica.

En el caso de la MSA se utilizan justamente estos tres tipos de sensores de navegación inercial: magnetómetro, acelerómetro y giróscopo, cuya descripción general en el contexto de control de orientación se presenta a continuación.

- Magnetómetro

El magnetómetro es un instrumento robusto pero con una precisión que está limitada alrededor de 0.5° . Éste sensor mide la dirección y la intensidad del campo magnético local. Sin embargo, debido a anomalías en el campo magnético terrestre, hacen del magnetómetro un instrumento de uso limitado para efectos de determinación y control de orientación satelital [32] y [2].

- Giróscopo

Son sensores inerciales los cuales miden la velocidad o ángulo de rotación a partir de una referencia inicial, aún sin conocer una referencia externa absoluta. Por su construcción se pueden clasificar en tres tipos: mecánicos, ópticos y MEMs [2] y [32].

- Acelerómetro

Es un instrumento para medir la aceleración de un objeto al que va unido, para tal efecto cuantifica respecto de una masa inercial interna. Los acelerómetros son sensores que miden la segunda derivada de la posición. Un acelerómetro mide la fuerza de inercia generada cuando una masa es afectada por un cambio de velocidad [2].

Para efectos de esta tesis doctoral, se realizó un estudio de evaluación para la selección de los sensores de navegación inercial que se instalarían en la MSA. Dicho estudio tomó en consideración, en primer término, la relación costo-beneficio del sensor seleccionado.

En la MSA los tres sensores de navegación inercial antes mencionados, están integrados en un sistema comercial denominado IMU (Unidad de Mediciones Inerciales). La IMU Sensor Stick®, figura 4.6, es una pequeña tarjeta electrónica que entrega nueve mediciones. Incluye tres sensores de navegación inercial de tecnología MEM: un magnetómetro HMC5883L, un acelerómetro ADXL345 y un giróscopo ITG3200, con sistemas de referencia como se indica en la figura 4.6. El protocolo de comunicaciones común a todos los sensores es I²C.

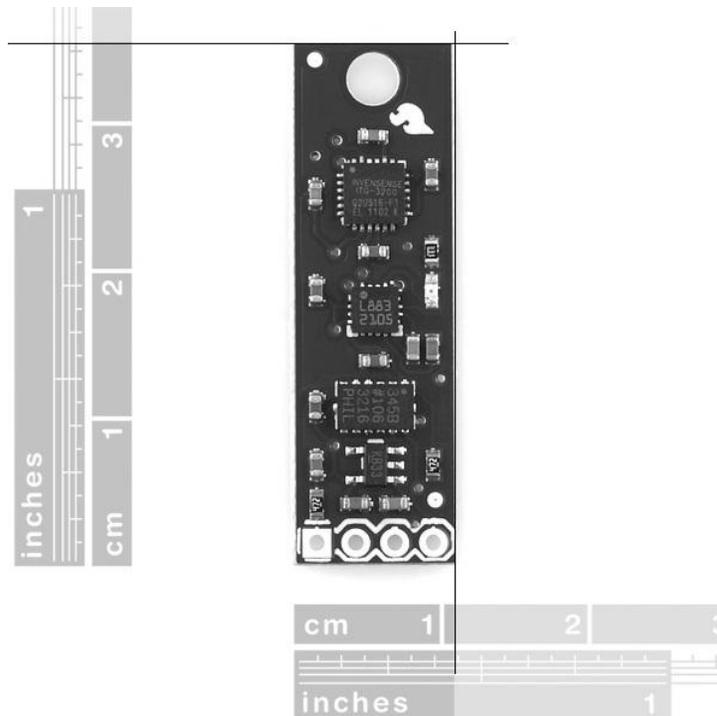


Figura 4.6. IMU 9DOF Sensor Stick®.

No obstante, cada uno de los sensores de la IMU tiene asociado su propio sistema de referencia, por lo que para su uso eficiente dentro de una aplicación es necesario considerar cada uno de ellos, como se muestra en la figura 4.7.

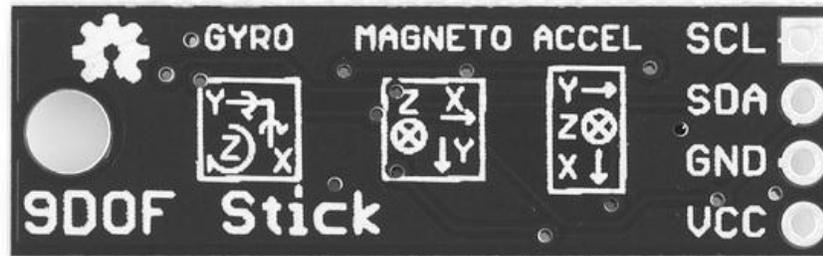


Figura 4.7. Sistemas de referencia definidos para cada sensor de la IMU.

Algunas de las características más importantes de los sensores de la IMU se describen en la tabla 4.1, destacando su resolución en bits, aspecto necesario para considerarse en el esquema de adquisición de datos digitales en la computadora de a bordo FPGA.

Tabla 4.1. Características de los sensores de navegación inercial en la IMU.

Sensor	VCC	Resolución	Intervalo de medición	Nivel de desviación (BIAS)
Magnetómetro HMC5883L	2.5 V (típico)	12 bits ADC 0.73 – 4.35 mili-Gauss	-8 a +8 Gauss	--
Giróscopo ITG3200	2.1 a 3.6 V	16 bits ADC 14.375 LSB (°/s)	+/- 2000 °/s	10° roll 11 ° pitch http://www.fact4ward.com/blog/ic-if/itg3200/
Acelerómetro ADXL345	2.5 V (típico)	10 bits ADC +/- 2g, +/-4g, +/- 8g , +/-16g	+/- 2g, +/-4g, +/- 8g , +/-16g	0g +/- 40 mg (típico)

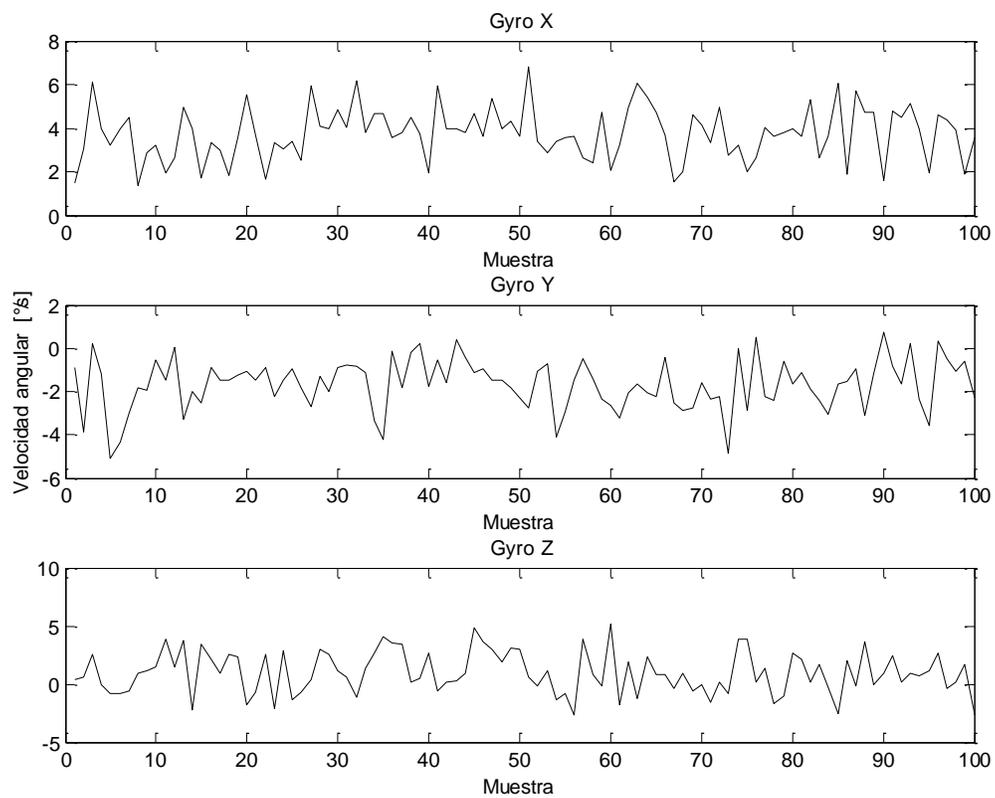
Un aspecto importante que se consideró, antes de integrar la IMU al esquema de instrumentación de la MSA, fue verificar analíticamente el comportamiento y nivel de ruido de cada uno de los sensores de navegación en operación. Para ello, se tomaron 100 muestras por medio de la

computadora FPGA y, posteriormente se enviaron a una PC, para calcular parámetros estadísticos con MATLAB (covarianza y desviación estándar). De esta manera se pudo describir su desempeño y considerar la implementación de técnicas de ajuste y compensación, tal como la habilitación de filtros digitales incluidos en algunos de los sensores de la IMU como en el giróscopo.

En las figuras 4.8, 4.9 y 4.10 se muestran las gráficas obtenidas y algunos datos estadísticos calculados a partir de los datos adquiridos de cada sensor de navegación de la IMU.

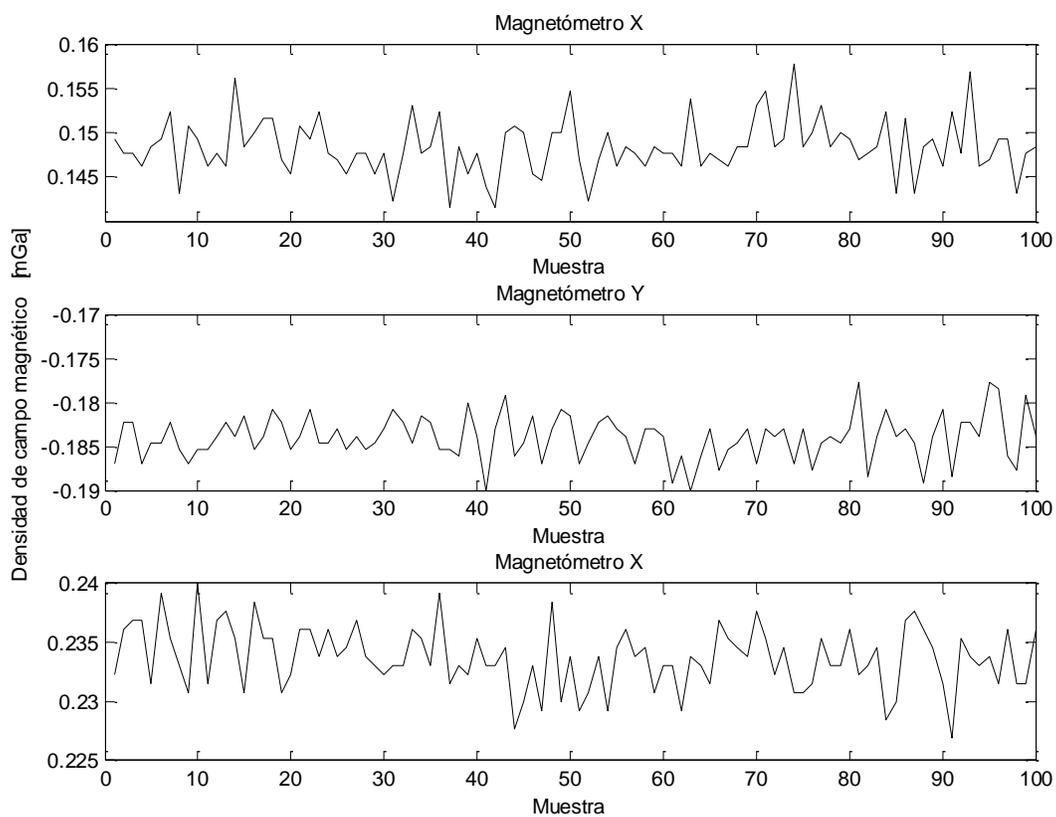
Observando los valores estadísticos asociados a cada una de las gráficas, es posible identificar que en el caso del giróscopo, figura 4.8, su gráfica de comportamiento a lo largo del número de muestras establecidas para las pruebas, tendió a ser ruidosa y poco definida, aun considerando que las pruebas se realizaron tomando una posición fija del sensor con respecto al marco de referencia inercial fijo al piso.

Resulta interesante lo que ocurre particularmente con este giróscopo, ya que como se menciona en diferentes foros sobre manejo de sensores para aplicaciones en mecatrónica, como en www.fact4ward.com/blog/ic-if/itg3200/, acerca del marcado efecto de deriva que presenta éste dentro de un lapso grande de tiempo y que es generalizado para todo giróscopo, lo que ocasiona movimientos en las lecturas por cada eje del sensor. Por tal motivo, para especificar un desplazamiento angular (ángulo) por medio del giróscopo es necesario fusionar el dato del giróscopo con otros sensores, como por ejemplo el acelerómetro, para obtener datos aún más certeros.



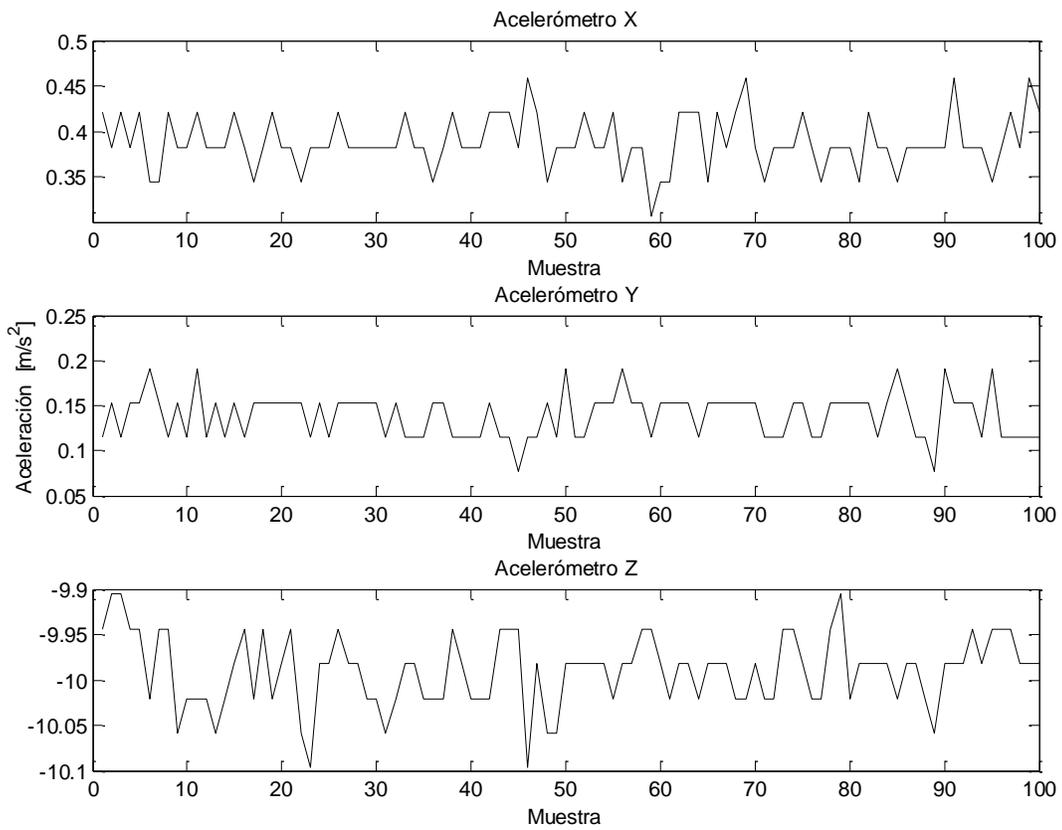
Covarianza gir6scopo eje x =	1.4965	desviaci6n est6ndar =	1.2233
Covarianza gir6scopo eje y =	1.4126	desviaci6n est6ndar =	1.1885
Covarianza gir6scopo eje z =	3.2881	desviaci6n est6ndar =	1.8133

Figura 4.8. Caracterizaci6n del gir6scopo ITG3200.



Covarianza magnetómetro eje x = 9.8750e-006	desviación estándar = 0.0031
Covarianza magnetómetro eje y = 6.3458e-006	desviación estándar = 0.0025
Covarianza magnetómetro eje z = 7.0931e-006	desviación estándar = 0.0027

Figura 4.9. Caracterización del magnetómetro HMC5883L.



Covarianza acelerómetro eje x = $8.4869e-004$	desviación estándar = 0.0291
Covarianza acelerómetro eje y = $6.1037e-004$	desviación estándar = 0.0247
Covarianza acelerómetro eje z = 0.0015	desviación estándar = 0.0392

Figura 4.10. Caracterización del acelerómetro ADXL345.

4.3.2. Actuadores

La MSA tiene una distribución de masas integrada por toda la instrumentación a bordo (sensores de navegación, computadora a bordo, actuadores y equipos electrónicos y mecánicos de soporte), la cual se consideró para el diseño y dimensionamiento de las ruedas inerciales integradas como actuadores activos del simulador.

La mejor opción para el diseño de las ruedas inerciales resultó en una geometría tipo disco interno, anillo externo, figura 4.11, por lo que la mayor cantidad de material se ubica en el anillo de la rueda para maximizar el momento de inercia que pudieran proveer y con esto garantizar maniobras de control de orientación de la plataforma de la MSA [18], [20] y [34].

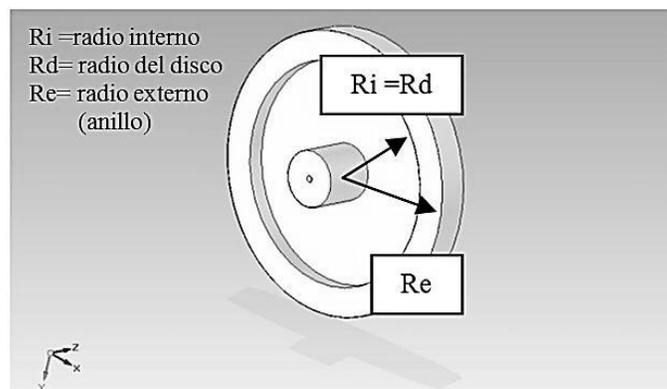


Figura 4.11. Diseño de las ruedas inerciales de la MSA.

Los momentos de inercia tanto del anillo externo, como del disco interno de la rueda inercial, con base en la geometría propuesta, son:

- Para un anillo de espesor considerable, el momento de inercia es:

$$I_a = \frac{1}{2} m(R_e^2 + R_i^2) \quad . \quad (4.1)$$

- Para un disco, su momento de inercia es:

$$I_d = \frac{1}{2} mR^2 \quad . \quad (4.2)$$

Para obtener el momento de inercia total del actuador se suman algebraicamente las ecuaciones (4.1) y (4.2) quedando las ecuaciones (4.3) y (4.4) de la siguiente forma:

$$I_{\text{actuador}} = I_a + I_d \quad , \quad (4.3)$$

$$I_{\text{actuador}} = \frac{1}{2} m_a (R_e^2 + R_i^2) + \frac{1}{2} m_d R_d^2 \quad (4.4)$$

En la ecuación (4.4) el momento de inercia del actuador está en función de los radios geométricos definidos por el anillo y disco, así como por la masa total del actuador, $m_{\text{actuador}} = m_a + m_d$ (m_a = masa del anillo, m_d = masa del disco).

Los actuadores, figuras 4.12 y 4.13, están compuestos por un motor de corriente directa de 12 V con escobillas y una rueda inercial. Las ruedas inerciales tienen un diámetro de 10.8 cm, una masa cercana a los 250 g y están instaladas ortogonalmente de acuerdo a un sistema de referencia inercial con origen en el centro geométrico de la MSA.



Figura 4.12. Construcción de las ruedas inerciales en los talleres del IINGEN-UNAM.

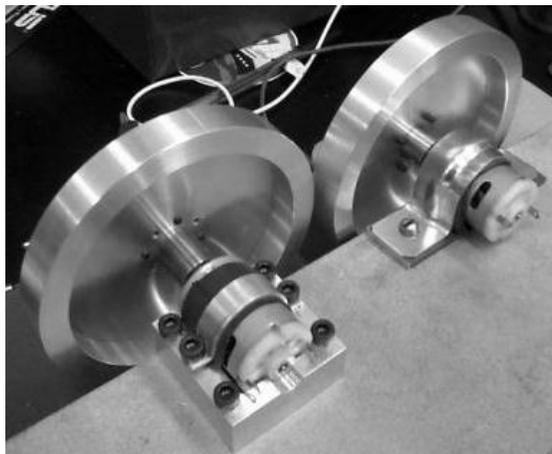


Figura 4.13. Pruebas de montaje de los actuadores integrados en el IINGEN-UNAM.

Para la caracterización de cada uno de los actuadores, se realizaron pruebas experimentales para evaluar su respuesta en velocidad angular, partiendo del reposo. Dichas pruebas se realizaron

utilizando una fuente de luz estroboscópica, como se muestra en la figura 4.14, para determinar la velocidad angular de cada una de las ruedas inerciales.

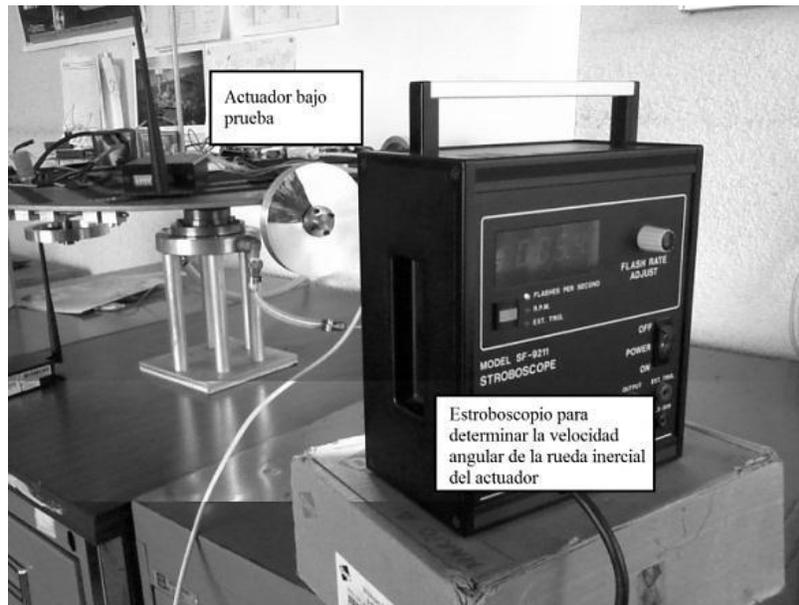


Figura 4.14. Pruebas de caracterización de los actuadores de la MSA.

Con base en la aplicación de un comando variable desde la computadora FPGA, expresado en porcentaje de ciclo de trabajo de una señal de control PWM, se obtuvieron los resultados que se muestran en las figuras 4.15, 4.16 y 4.17.

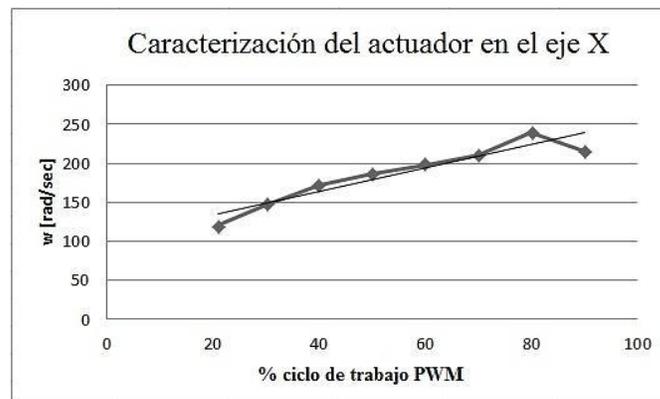


Figura 4.15. Caracterización, PWM *versus* velocidad angular actuador en el eje X.

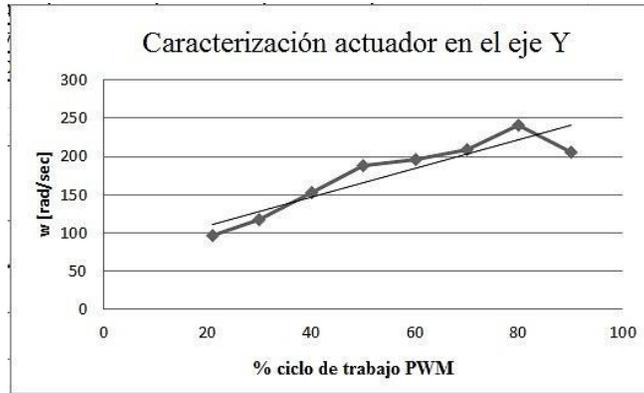


Figura 4.16. Caracterización, PWM *versus* velocidad angular actuador en el eje Y.

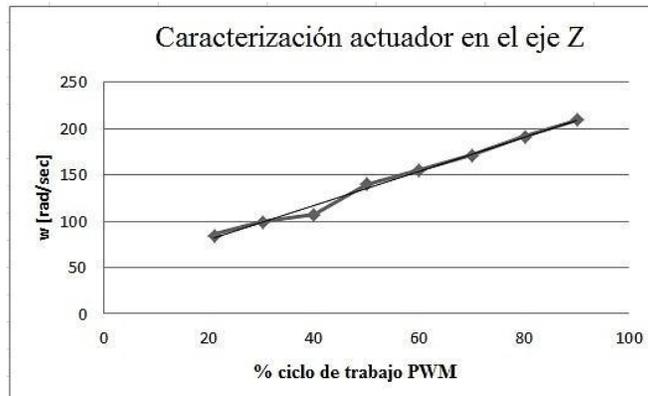


Figura 4.17. Caracterización, PWM *versus* velocidad angular actuador en el eje Z.

4.3.3. Computadora de a bordo

La computadora de a bordo se integró en una tarjeta de desarrollo comercial Spartan 3E-Starter Kit de Xilinx® (SP3E), figura 4.18. En dicha plataforma se implantan arquitecturas embebidas para formar módulos de adquisición de datos y generación de comandos de control para los actuadores, basados en señales PWM.

La SP3E cuenta con un FPGA XC3S500E de gama media de la familia Spartan 3E, con 500,000 compuertas lógicas disponibles. Otros recursos del FPGA para la construcción de arquitecturas digitales se indican en la tabla 4.2. La configuración de la SP3E se realiza por medio del protocolo JTAG, mediante una interfaz USB hacia una PC. Adicionalmente, cuenta con terminales de entrada y salida de propósito general (I/O digitales) e interfaces de comunicación RS232 (DTE y DCE), Ethernet, conector PS2, memoria FLASH, entre otras características que se muestran en la figura 4.18.

Tabla 4.2. Descripción de recursos para FPGAs de la familia Spartan 3E de Xilinx® [47].

Resumen de recursos de FPGAs de la Familia Spartan 3E

Dispositivo	CLB Rows	CLB Columns	CLB Total ⁽¹⁾	Slices	LUTs / Flip-Flops	Celdas lógicas equivalentes	RAM16 / SRL16	Distributed RAM Bits
XC3S100E	22	16	240	960	1,920	2,160	960	15,360
XC3S250E	34	26	612	2,448	4,896	5,508	2,448	39,168
XC3S500E	46	34	1,164	4,656	9,312	10,476	4,656	74,496
XC3S1200E	60	46	2,168	8,672	17,344	19,512	8,672	138,752
XC3S1600E	76	58	3,688	14,752	29,504	33,192	14,752	236,032

Device	System Gates	Equivalent Logic Cells	CLB Array (One CLB = Four Slices)				Distributed RAM bits ⁽¹⁾	Block RAM bits ⁽¹⁾	Dedicated Multipliers	DCMs	Maximum User I/O	Maximum Differential I/O Pairs
			Rows	Columns	Total CLBs	Total Slices						
XC3S500E	500K	10,476	46	34	1,164	4,656	73K	360K	20	4	232	92

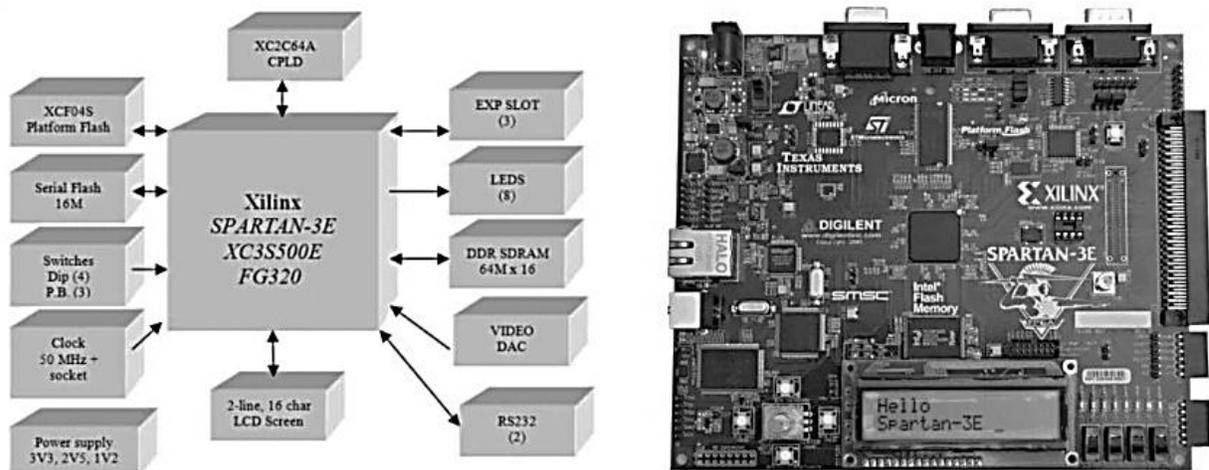


Figura 4.18. Tarjeta de desarrollo Spartan 3E-Starter Kit.

El criterio de selección de la tarjeta de desarrollo FPGA se basó principalmente en los recursos disponibles en hardware para la interconexión de equipos externos y su compatibilidad con el software de desarrollo disponible en el laboratorio del IINGEN-UNAM.

En cuanto a las interfaces, se priorizó la que contara con suficientes terminales de entrada y salida digitales tanto de propósito general como de protocolos estándar para vincularse con dispositivos periféricos como sensores de navegación inercial, actuadores y equipo de comunicaciones.

En lo que respecta a la integración de la SP3E en la instrumentación de la MSA, en el esquema de la figura 4.19 se identifica al FPGA como núcleo central que lleva el control de toda la lógica del simulador.

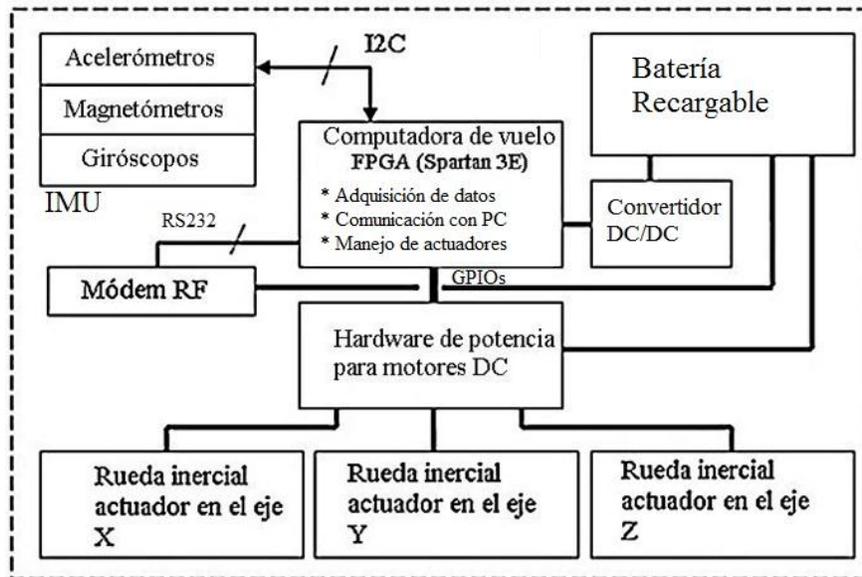


Figura 4.19. Esquema de instrumentación de la MSA con la SP3E como elemento central.

4.3.3.1 Sistema mínimo (SMIN)

En vista de que uno de los objetivos iniciales de éste trabajo de investigación doctoral fue desarrollar un simulador satelital de vuelo orbital instrumentado con tecnología mayoritariamente nacional y de diseño propio, también se diseñó, fabricó y ensambló una computadora a bordo basada en un FPGA.

La tarjeta electrónica denominada SMIN, se implementó en un circuito impreso de doble cara de 10 x 10 cm en el IINGEN-UNAM, [35], [36] y [37]. La arquitectura en hardware de esta tarjeta se basó en requisitos de interfaces, así como en entradas y salidas digitales. El sistema SMIN se usará como plataforma única de desarrollo de sistemas embebidos para implementar todos aquellos módulos periféricos y de coprocesamiento aritmético que requiera el SDCO.

El SMIN integra el mismo FPGA con el que cuenta la tarjeta Spartan 3E Starter Kit. Esto se definió así para aprovechar la experiencia en términos del desarrollo de arquitecturas digitales y la integración de sistemas embebidos. Cuenta con una serie de módulos adicionales compuestos por transceptores para comunicaciones seriales, alimentación de energía, memoria, interfaces de entrada y salida digitales y otros dispositivos electrónicos de soporte.

Adicionalmente, el diseño de la tarjeta consideró la integración de un módulo que permite la reconfiguración del FPGA así como la recarga remota de nuevas arquitecturas de cómputo utilizando para ello un microcontrolador de propósito general y una memoria EEPROM.

El diseño de sistemas basados en circuitos integrados, tales como los FPGAs, que cuentan con un gran número de terminales físicas y encapsulados de montaje superficial de dimensiones mínimas, representan retos de diseño de la tarjeta de circuito impreso (layout, ruteo), así como de ensamble y validación.

Uno de los principales retos que afrontó el desarrollo de SMIN fue lograr el correcto acoplamiento de las líneas de entre el FPGA y los módulos de comunicaciones, reconfiguración remota y memoria, especialmente por las restricciones establecidas en los niveles de impedancia, entrada de corriente y niveles de voltaje de acoplamiento.

El sistema SMIN está integrado por dos segmentos principales: **estación terrena** y **tarjeta SMIN**.

- Estación terrena

Este bloque se ejecuta en una PC y contiene el software de control de cada uno de los procesos que permiten la transmisión del *bitstream* de reconfiguración, utilizando una interfaz gráfica de usuario (GUI) como módulo principal, figura 4.20.

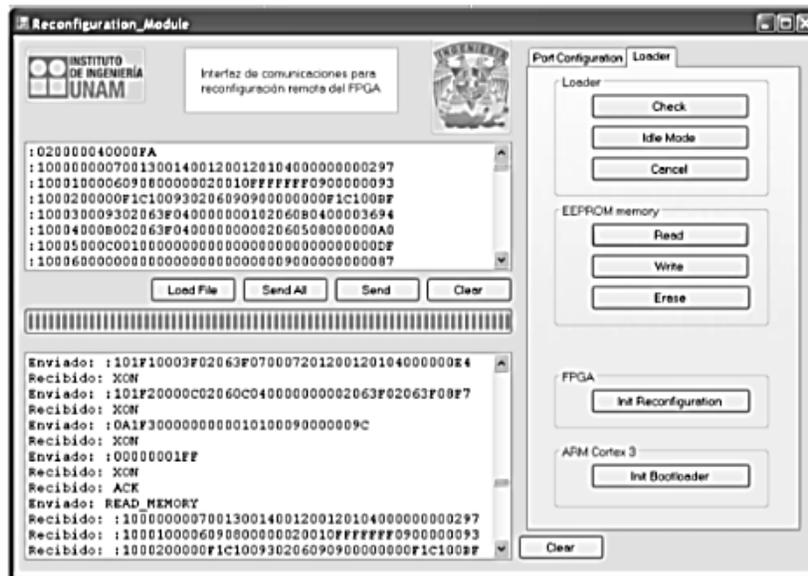


Figura 4.20. Interfaz de usuario del SMIN.

Estos procesos se dividen en distintos bloques, que se ejecutan en plataformas diferentes de software [35]. Dichos procesos son:

Diseño y síntesis de la arquitectura de cómputo o sistema embebido

Este proceso utiliza las herramientas de diseño de la suite ISE de Xilinx ® para el diseño, síntesis y compilación del hardware y software que integra al sistema embebido y las arquitecturas de cómputo que lo integran. Luego de realizar la síntesis, se genera un archivo en formato *.bit, el cual es el formato natural de configuración para FPGAs.

Creación de un archivo en formato XSVF

Proceso que genera un archivo XSVF a partir del archivo *.bit, el cual se crea al sintetizar la arquitectura digital, para ello se hace uso de iMPACT (software de ISE). Este formato de archivo es indispensable para usar las herramientas de software que proporciona el fabricante (Xilinx® en nuestro caso) para la reconfiguración de sus FPGAs.

Creación de un archivo en formato IntelHex

En este proceso se crea un archivo en formato Intel Hex a partir del archivo XSVF creado previamente, este proceso puede efectuarse utilizando cualquier compilador C. Con este formato de archivo se mantiene control sobre la transferencia de datos de forma serial, garantizando con ello la transmisión y recepción exitosa de éstos.

Interfaz gráfica de usuario

Este proceso permite tomar el control del puerto serie (USB) de la PC para transmitir el archivo de reconfiguración, así como recibir datos para verificar el estado operativo de algunos dispositivos a bordo de la tarjeta, como la memoria. También indica cada una de las tareas que deberá ser ejecutada por el microcontrolador para llevar a cabo el proceso de reconfiguración remota del FPGA.

- Tarjeta SMIN

La tarjeta SMIN está integrada por los siguientes bloques [35]:

-- Bloque de transmisión, recepción, almacenamiento en memoria y reconfiguración del FPGA (BTRAR).

- Bloque de acondicionamiento y potencia (BAP).
- Bloque de entradas y salidas digitales (BESD).

En la figura 4.21 se muestra el diseño conceptual de la tarjeta SMIN, en donde se muestran los segmentos y bloques mencionados.

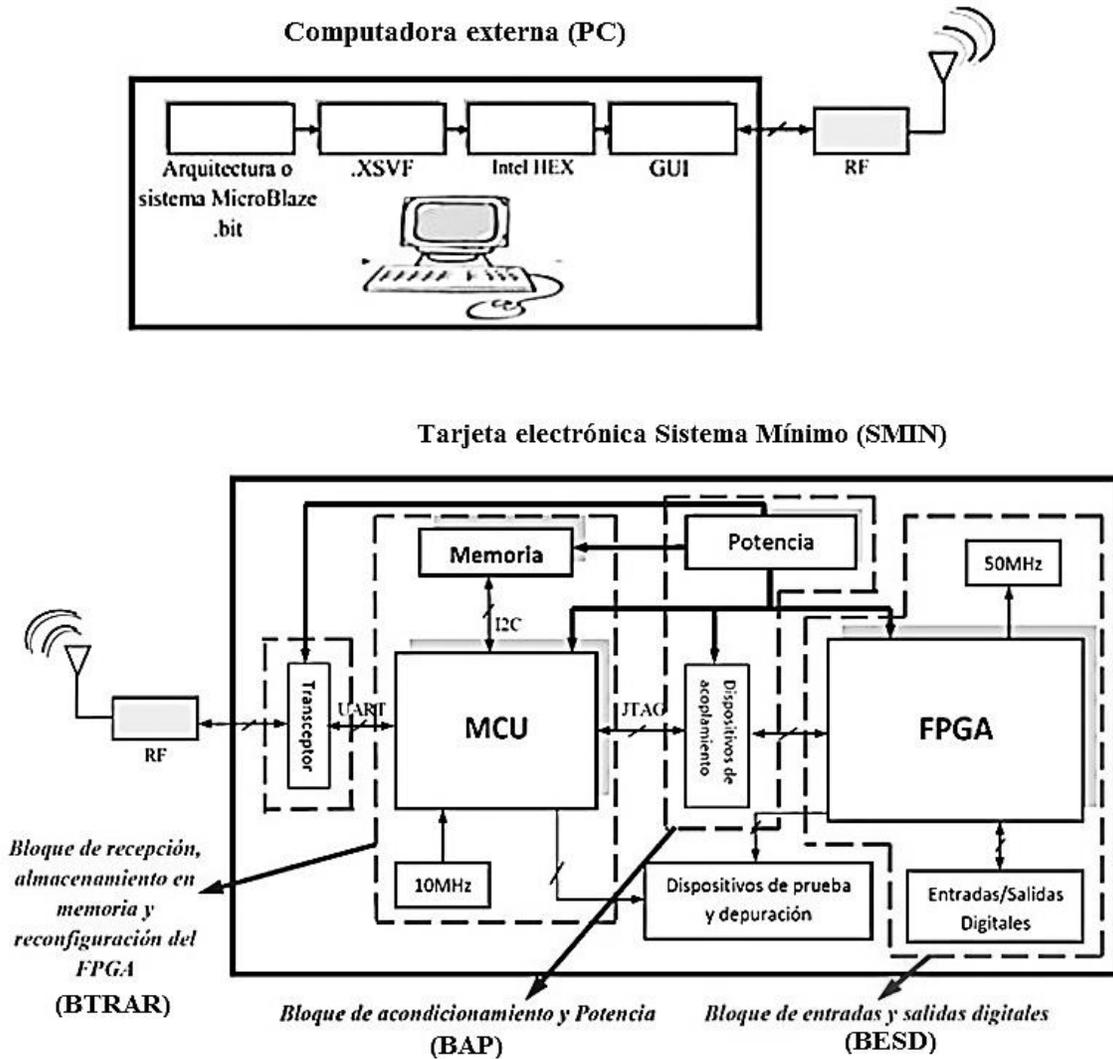


Figura 4.21. Descripción de los segmentos principales del sistema SMIN.

Como se observa en la figura 4.21, se integra un módulo de componentes de apoyo a la depuración de software, integrado fundamentalmente por LEDs e interruptores deslizables (*dip switch*). Este módulo no es parte de algún bloque específico de la tarjeta y es común al resto de los bloques operativos del sistema SMIN.

Los componentes del módulo de prueba y depuración para el BAP, indican la operación de los reguladores de voltaje a bordo de la tarjeta, responsables de suministrar los diferentes niveles de voltaje para cada uno de los componentes a bordo de la tarjeta.

Para el caso del BTRAR, indica la ejecución de los procesos de almacenamiento y lectura de memoria, así como la reconfiguración del FPGA. Mientras que la función principal del bloque BESD, es validar la carga del *bistream* de reconfiguración del FPGA, que ayuda a verificar visualmente si el FPGA ha sido configurado de forma exitosa.

Finalmente, la tarjeta desarrollada se muestra en la figuras 4.22 y 4.23, en la figura 4.23 se muestran ambos lados de la tarjeta electrónica (superior e inferior).

En [35], se pueden encontrar mayores detalles acerca del diseño del sistema SMIN.

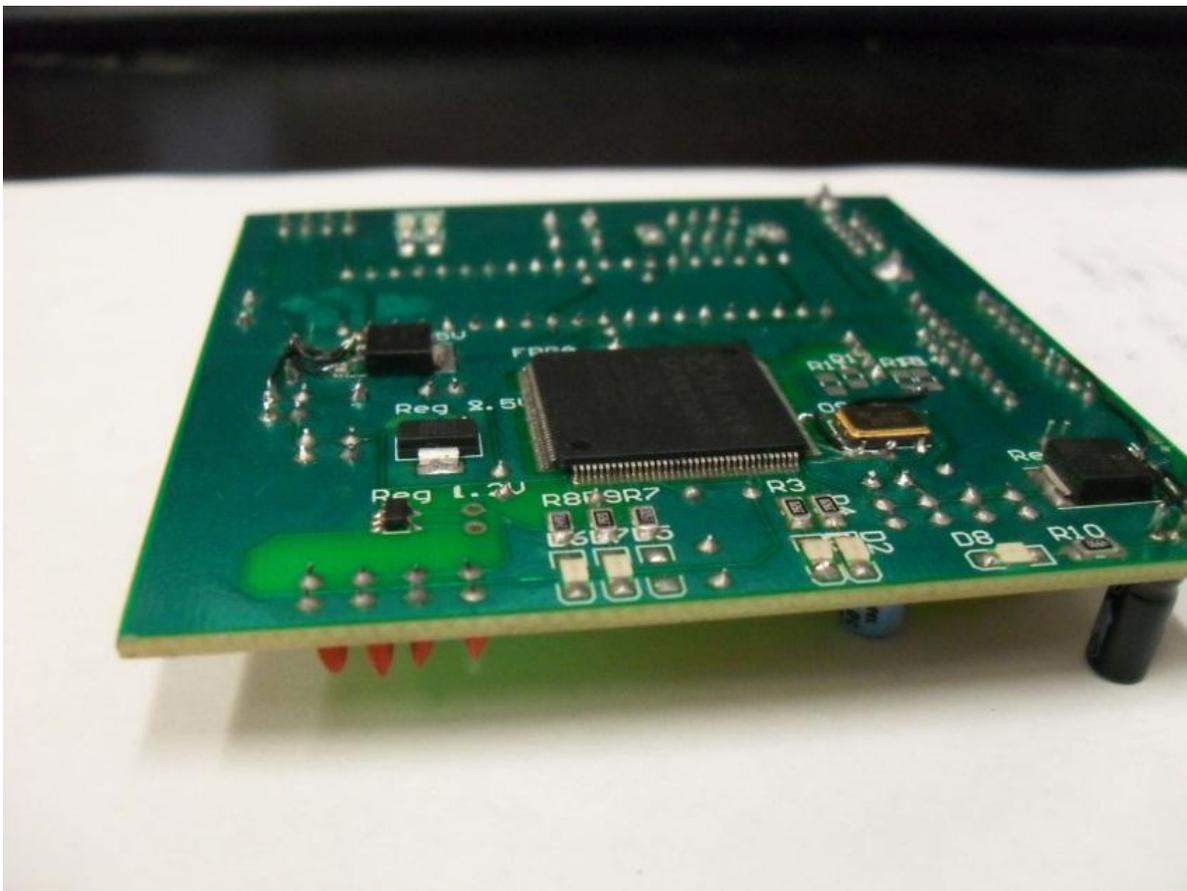


Figura 4.22. Tarjeta electrónica del sistema SMIN.

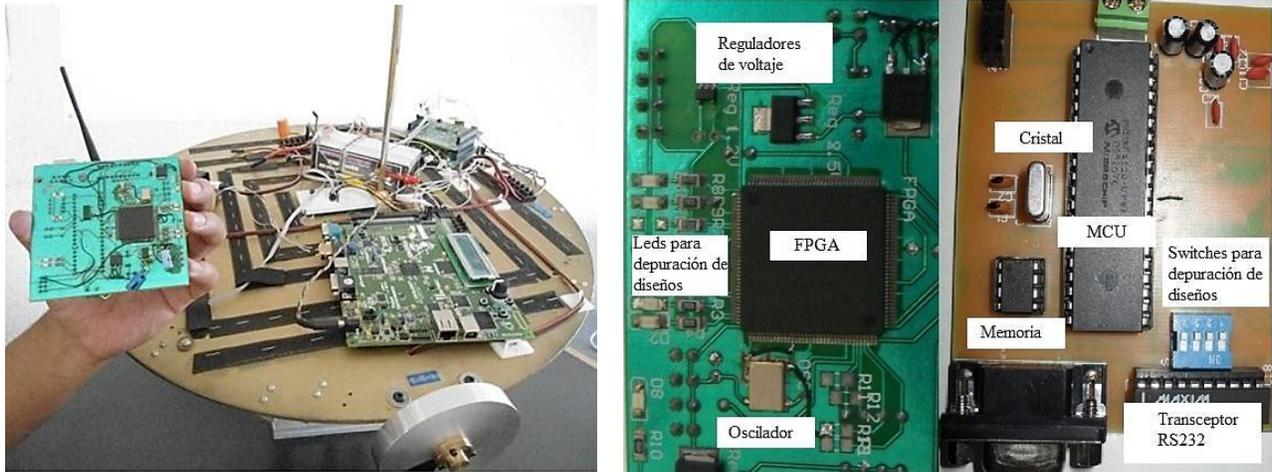


Figura 4.23. SMIN y descripción de cada uno de sus módulos ubicados en ambas caras de la tarjeta.

4.3.4. Drivers y potencia

El módulo de drivers y potencia para manejo de ruedas inerciales reside en una tarjeta de circuito impreso, diseñada y desarrollada en esta tesis, figura 4.24, en la que se encuentran instalados tres circuitos integrados L6202 para el manejo de potencia hacia los actuadores y un convertidor DC/DC para suministro de energía a la tarjeta SP3E.

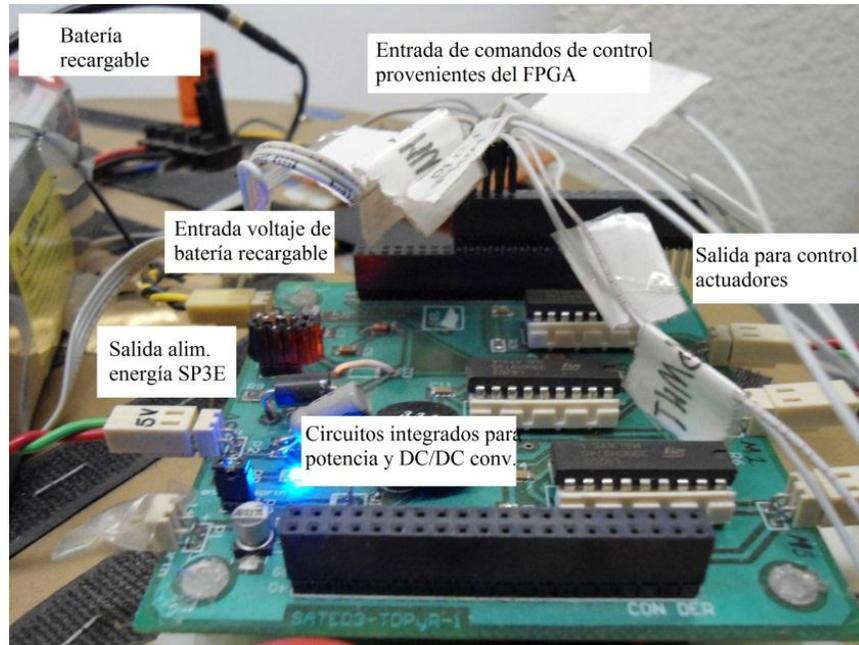


Figura 4.24. Tarjeta de drivers de potencia a bordo de la MSA.

Los drivers L6202 aplican señales de potencia hacia los tres motores DC de cada uno de los actuadores en tres ejes de la MSA (X, Y, Z), en función de comandos de control generados por la computadora de a bordo.

El L6202 es un puente H completo, que combina la tecnología de transistores de potencia DMOS con CMOS y circuitos bipolares en el mismo encapsulado. Admite una alimentación de voltaje de hasta 48V, soportando picos máximos de corriente de 5 A. La lógica de control del L6202 es compatible con niveles TTL, operando a una frecuencia máxima de 100 kHz con protección térmica y una alta eficiencia en operaciones [38].

El módulo de potencia de la MSA está compuesto por una batería recargable y un convertidor DC/DC. La batería recargable es de tecnología LiPo (Polímero de Litio) de 7700 mAh, Thunder Power RC, 4 celdas y 14.8 V, utilizada principalmente en modelos de radio control que requieren gran tiempo de autonomía, por lo que representa una buena opción en términos de eficiencia y duración de carga.

El convertidor DC/DC LM2575, es un regulador de voltaje tipo *step-down* con voltaje de salida de 5V a 1A, considerando una entrada máxima de 45V. Por su operación conmutada ofrece una eficiencia del 75% en el proceso de regulación de voltaje, con una frecuencia fija interna de 52 kHz [39]. En la figura 4.25 se muestra un esquema del sistema de potencia a bordo de la MSA.

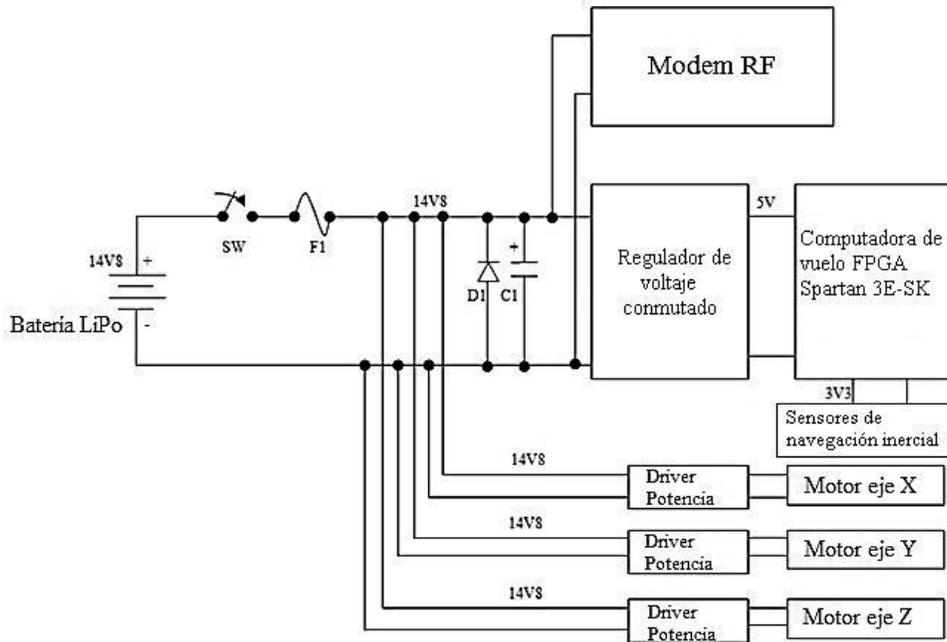


Figura 4.25. Sistema de potencia abordo de la MSA.

4.3.5. Comunicaciones

La comunicación de datos entre la plataforma de la MSA y la PC se llevó a cabo con un par de módems de radio frecuencia (RF) comerciales XStream- PKG-R, que operan en la banda de 900 MHz, con una alimentación de 9 a 18 V, 450 metros de alcance y 100 mW de potencia.

Su interfaz de comunicaciones mediante el protocolo estándar RS232 permitió la adecuada conexión con la SP3E y con la PC externa.

4.4. Software de la MSA

El software del simulador MSA está asociado con los dos segmentos principales que integran el esquema HIL: plataforma suspendida en aire y PC.

Respecto a la plataforma suspendida en aire, el software a bordo está integrado en el FPGA, desarrollado en torno al microprocesador MicroBlaze y núcleos periféricos. La estructura del sistema embebido (SE) a su vez se segmenta en: hardware del sistema embebido y firmware del MicroBlaze.

En el caso de la PC, el software está enfocado en un sistema de funciones desarrollado en Matlab®. En los siguientes apartados se describe el contenido de cada una de las plataformas antes mencionadas.

4.4.1. Software de la Plataforma Suspendida en Aire

El software asociado al FPGA de la computadora a bordo de la MSA, está desarrollado como un SE o Sistema empotrable. En este caso, el SE utiliza núcleos de coprocesamiento tipo *soft core* (microprocesadores, unidades dedicadas de procesamiento aritmético y lógico) y de aplicación específica (periféricos), es decir, hardware descrito por medio de lenguaje descriptor de hardware.

En los siguientes apartados se abordan las generalidades necesarias para entender de forma general la construcción y el concepto de SEs en el FPGA.

4.4.1.1. Sistemas Embebidos en FPGA

El diseño de un SE que permita resolver la lógica de una determinada aplicación, puede realizarse mediante dos aproximaciones diferentes: método software y método hardware [40].

Método software

- * Se utilizan circuitos procesadores estándar, ya sean de propósito general (microprocesadores) o de propósito específico (DSPs).
- * Estos circuitos presentan una serie de funciones predefinidas, asociadas a un juego de instrucciones fijo.
- * El diseñador realiza el programa (secuencia de instrucciones) que ejecutará el procesador para realizar las funciones requeridas.

Método hardware

- * El diseñador define el sistema digital necesario para realizar las funciones requeridas.
- * El sistema digital resultante se implementa mediante los dispositivos elegidos (PLD, FPGA, ASIC).

En cuanto a las limitaciones inherentes a cada uno los métodos descritos, se tiene:

Método software

- * El procesamiento secuencial de las instrucciones.
- * La arquitectura concreta del procesador utilizado.

Método hardware

- * La dificultad de diseñar sistemas digitales complejos.
- * El tiempo de diseño necesario para desarrollar sistemas con una secuencia de operaciones larga y compleja (con muchas bifurcaciones).
- * El costo del hardware de desarrollo necesario.

Por tal motivo, el desarrollo de sistemas digitales complejos exige la utilización de un SE, también conocido como SOC (*“System On Chip”*) [40].

Un SE incluye un circuito electrónico único que integra los siguientes componentes:

- Uno o varios microprocesadores
- Una o varias unidades de memoria (SRAM, FLASH-EEPROM, etc.).
- Circuitos de interfaz estándar.
- Circuitos específicos para la aplicación, realizados mediante hardware reconfigurable.
- Líneas de interconexión entre los bloques.

Las dos opciones que existen actualmente para realizar un SOC son las siguientes:

- Mediante un ASIC que incluya un microprocesador y una parte de hardware configurable.
- Mediante un FPGA de gran capacidad lógica. En este caso, los SOCs se denominan *“System On a Programmable Chip”* (SOPC).

En caso de utilizar un FPGA, como en el caso de esta tesis, para implementar un SE existen dos posibilidades:

- A partir de un bloque microprocesador hardware, ya integrado en la pastilla de Silicio, junto con el hardware configurable en el FPGA.
- A partir de un bloque microprocesador software, prediseñado para su implementación como un circuito más, en un FPGA genérico.

Junto con el desarrollo de nuevos conceptos y metodologías de diseño, los fabricantes de FPGAs también han desarrollado herramientas en software, las cuales permiten de la forma más amigable posible el desarrollo e integración de SEs. Esto ha permitido la mejora en el software de diseño de SEs, incluyendo el desarrollo de herramientas de diseño lógico integradas en un solo ambiente de desarrollo que ofrece una solución combinada de hardware y software, como se muestra en la figura 4.26, [40] y [41].

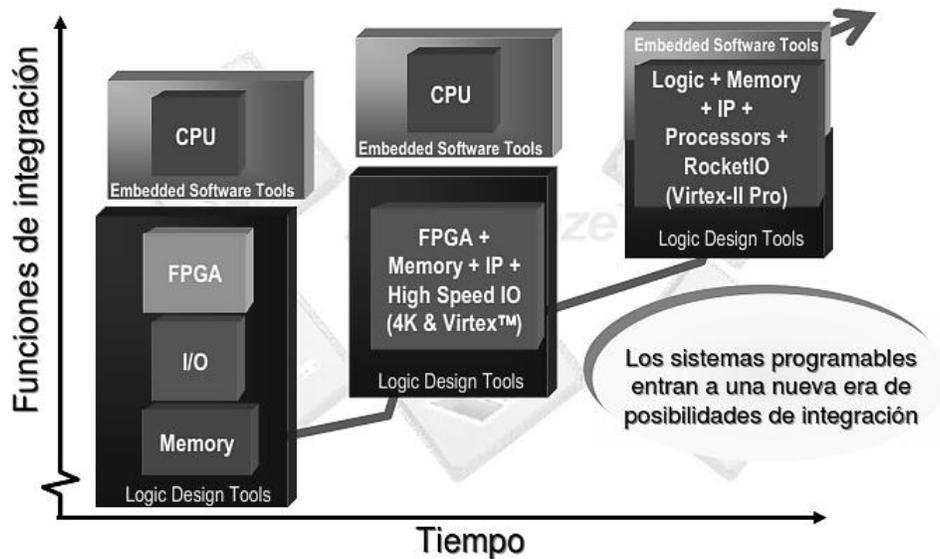


Figura 4.26. Nivel de integración de las herramientas para desarrollo de SE.

4.4.1.2. Desarrollo de Sistemas Embebidos basados en herramientas Xilinx®

En esta sección y en las subsecuentes se aborda el desarrollo de un SE a partir de herramientas en hardware y software de Xilinx®. Es necesario resaltar que el flujo lógico de algunos procesos que se describen son extrapolables a herramientas de desarrollo de otros fabricantes de FPGAs.

Al respecto, Xilinx® dispone de dos opciones para el desarrollo de SE:

- Bloque Microprocesador en hardware

Cuenta con el núcleo Power-PC (PPC), el cual es un *soft core* de la gama alta de núcleos embebidos. Tiene una arquitectura tipo Harvard RISC de 32 bits y se encuentra presente en las siguientes familias de FPGAs:

Virtex 2 Pro

Virtex 4 FX

Virtex 5 FX

- Bloque Microprocesador en software

Cuenta con dos núcleos: Picoblaze y MicroBlaze.

PicoBlaze. Es un núcleo de 8 bits, gratuito, el cual puede sintetizarse en cualquier FPGA, aún con recursos lógicos mínimos.

MicroBlaze. Es un núcleo de 32 bits, el cual puede sintetizarse en cualquier FPGA de las familias Spartan y Virtex que cuenten con suficiente capacidad lógica.

En la gráfica de la figura 4.27 se muestran los tres núcleos de microprocesadores de Xilinx® que integran las gamas baja, media y alta, junto con algunas de sus características más importantes.

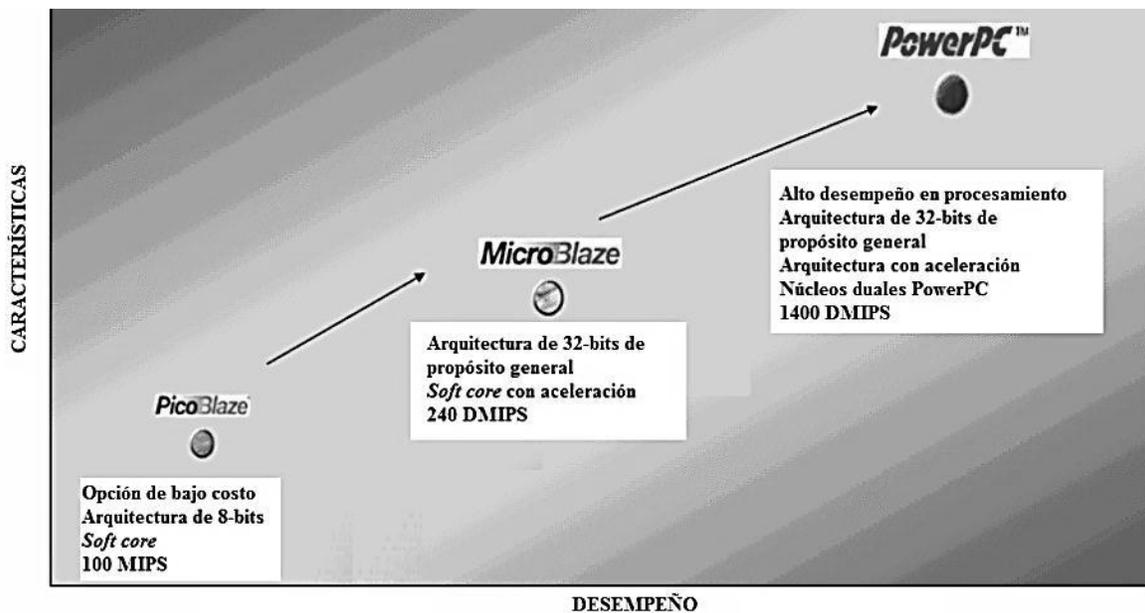


Figura 4.27. Gama de microprocesadores *soft core* de Xilinx®.

Los términos *soft core* y *hard core* son una clasificación que se hace para los núcleos de procesamiento en FPGAs. Los *soft cores* son núcleos que implementa el usuario en función de los requerimientos específicos de su aplicación, utilizando recursos lógicos del FPGA. Son una

abstracción de piezas de hardware desarrolladas en lenguaje descriptor de hardware (HDL) que se pueden integrar a un SE u operar de forma aislada [40].

Los *hard cores* son componentes dedicados dentro de la pastilla de Silicio de un circuito integrado, que integran la arquitectura de un microcontrolador y a los cuales el usuario sólo tiene acceso para su configuración, lectura y escritura de registros de datos asociados por medio de la IDE de desarrollo (software) del fabricante [40].

Para esta tesis se utilizó el *soft core* del microprocesador MicroBlaze (MB), por su desempeño, disponibilidad y compatibilidad tanto con la IDE de desarrollo como con el sistema SP3E-SK, ambos disponibles en el IINGEN-UNAM.

La integración de un SE con MB se muestra en la figura 4.28. En dicha figura se observa el desarrollo de una arquitectura computacional a partir de un núcleo central MB y varios núcleos periféricos, ya sea con funciones asociadas a interfaces de comunicaciones o de coprocesamiento aritmético o lógico.

Los núcleos o *cores*, también llamados IP (*Intellectual Property*), son programas desarrollados por el usuario en HDL, con funciones bien definidas y enfocadas a resolver algún aspecto específico de la aplicación (comunicaciones, generación de señales, temporizadores, procesamiento), los cuales posteriormente pueden ser integrados al SE para integrarse como núcleo de coprocesamiento o como periférico del MB.

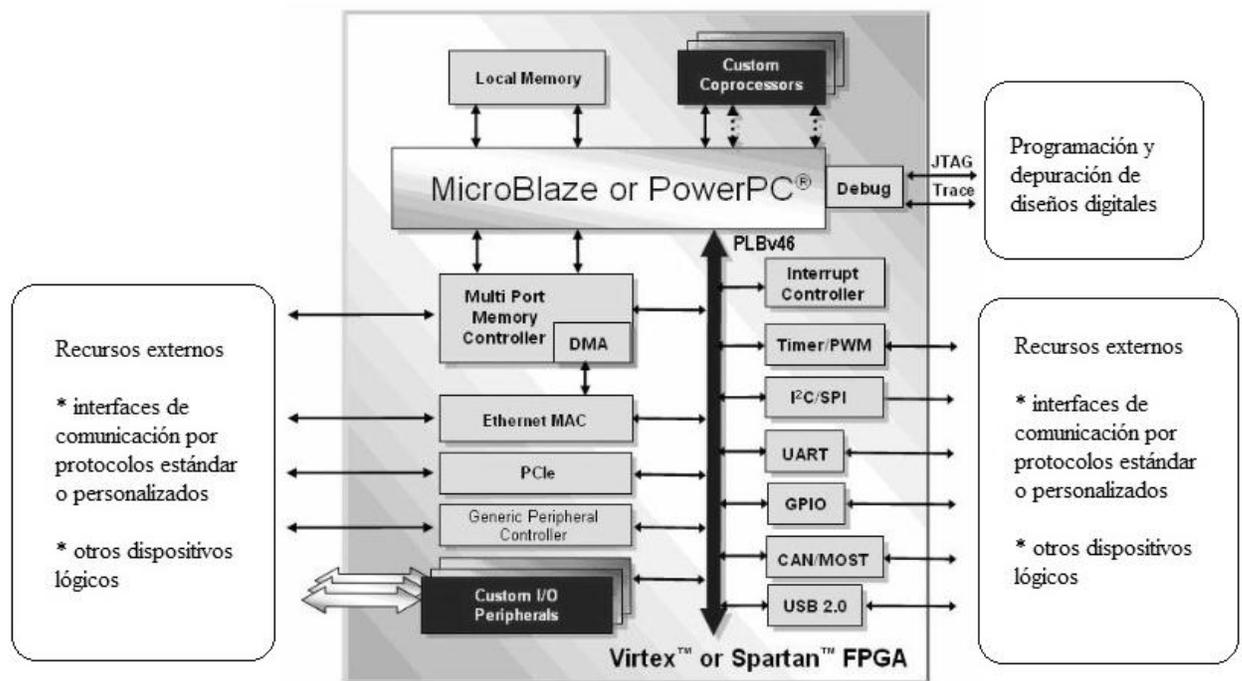


Figura 4.28. Sistema embebido para MicroBlaze y PowerPC.

4.4.1.3. Codiseño hardware/software para la integración de SE en FPGA

Una vez analizadas las bases conceptuales y presentados parte de los recursos necesarios para el desarrollo e integración de un SE en FPGA, se abordan las bases del codiseño hardware/software, el cual es un procedimiento mediante el cual la aplicación descrita en términos de un algoritmo matemático o lógico se transfiere a un sistema computacional en un FPGA.

El diseño del SE presenta una complejidad elevada, ya que además de diseñar es necesario verificar:

- El programa que va a ejecutar el microprocesador (MB, PPC).
- El hardware necesario, tanto del propio microprocesador como de sus periféricos asociados.
- El resto de los circuitos adicionales que se precisen para la aplicación.

A este proceso de diseño de un SE para una aplicación específica se denomina **codiseño hardware/software**, [40].

El codiseño se define como el diseño concurrente establecido por el equipo de desarrollo del SE, tanto de componentes en hardware como en software, usando para ello metodologías y herramientas que consideren la interacción hardware/software [40].

En términos lógicos, el proceso de codiseño se describe conforme el flujo de la figura 4.29.

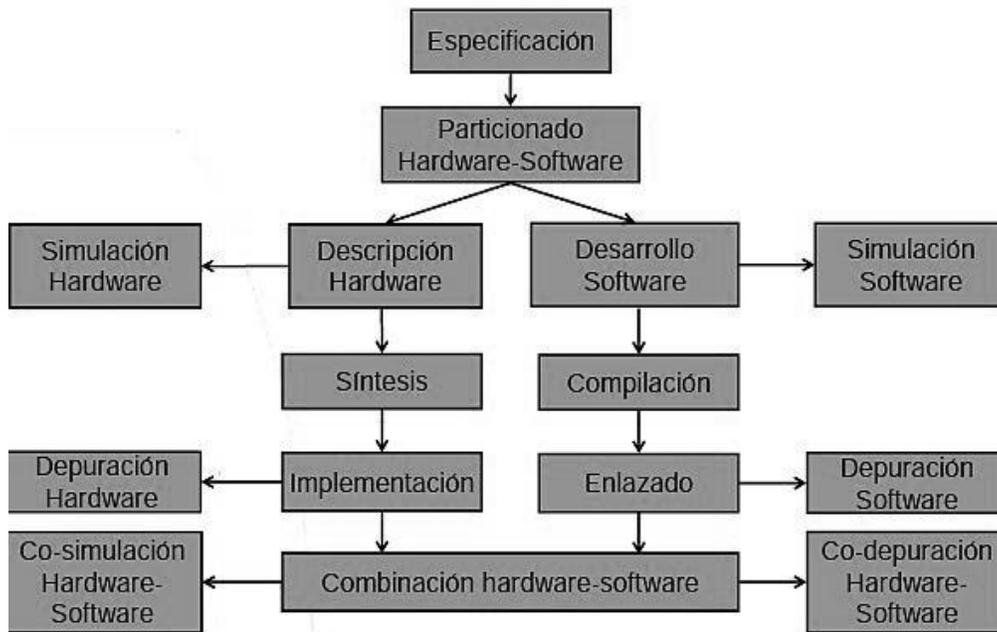


Figura 4.29. Flujo de codiseño para desarrollo de SEs en FPGAs.

Respecto a la descripción de cada uno de los procesos del codiseño de la figura 4.29, se tiene:

- * **Especificación.** Es definir la funcionalidad del sistema a desarrollar.
- * **Particionado hardware-software.** Incluye los siguientes subprocesos:
 - Decidir qué funciones se realizarán mediante rutinas en el firmware del microprocesador embebido y qué funciones se realizarán mediante circuitos en hardware (definición de núcleos IP periféricos y de coprocesamiento).
 - Implementar mediante núcleos IP las funciones u operaciones cuyas rutinas en software resulten lentas o poco eficientes.
 - Implementar mediante firmware en el microprocesador embebido las funciones u operaciones cuyos núcleos IP resulten muy complejos y/o que no requieran de una velocidad de procesamiento elevada.
- * **Descripción de hardware.** Incluye los siguientes subprocesos:
 - Elección del microprocesador y del FPGA adecuados para la aplicación.

- Selección de las herramientas de diseño, ya sean específicas del fabricante del FPGA o genéricas.
- Diseño de periféricos y otros circuitos adicionales mediante HDL u otras herramientas de diseño de alto y bajo nivel.
- * **Desarrollo software.** Incluye los siguientes subprocesos:
 - Desarrollo de aplicaciones en entornos integrados de desarrollo para cada microprocesador.
 - Desarrollo de la aplicación en lenguaje ensamblador o en lenguajes de alto nivel (C, C++, etc.).
- * **Simulación hardware.** Se realiza mediante el uso de simuladores adecuados al software de diseño utilizado. Por ejemplo, para arquitecturas en VHDL, se puede hacer uso del simulador propio del navegador de proyectos de ISE®, o bien del simulador Modelsim®. Para el caso de sistemas desarrollados con System Generator® se utiliza Simulink®.
- * **Simulación software.** Se realiza mediante el uso de programas de simulación asociados al lenguaje de programación y del microprocesador utilizado en la aplicación.
- * **Síntesis e implementación.** Se utilizan herramientas específicas del fabricante del FPGA.
- * **Compilación y enlazado.** Se utilizan herramientas específicas para el lenguaje de programación y el microprocesador que ha sido utilizado para la integración del SE.
- * **Depuración hardware.** Se realiza con instrumentación convencional (osciloscopios, analizadores de espectros, analizadores lógicos, etc.) o mediante instrumentación virtual (Chipscope® de Xilinx®).
- * **Depuración software.** Se realiza con entornos de depuración software, previo volcado del programa ejecutable al microprocesador embebido.
- * **Cosimulación hardware-software.** Se realiza mediante entornos de simulación, que combinan diversas herramientas de simulación en software para validar tanto hardware como software (firmware) del SE.
- * **Codepuración hardware-software.** Se lleva a cabo mediante el uso de entornos de depuración que combinan la depuración de ambas vertientes de desarrollo, hardware y software.

4.4.2. Sistema Embebido desarrollado para la MSA

Para el caso de la MSA, se integró un SE en el sistema SP3E-SK basado en núcleos con funciones para el manejo de interfaces de comunicaciones serie utilizando protocolos estándar (RS232, I2C),

temporizadores, interrupciones y se desarrolló además un núcleo IP para la generación de señales de control PWM para los actuadores de la plataforma MSA.

La lógica de todo el sistema se controla con un procesador MB el cual, a través del bus PLB administra todos los recursos del sistema a bordo. En la figura 4.30 se muestra el SE para la MSA.

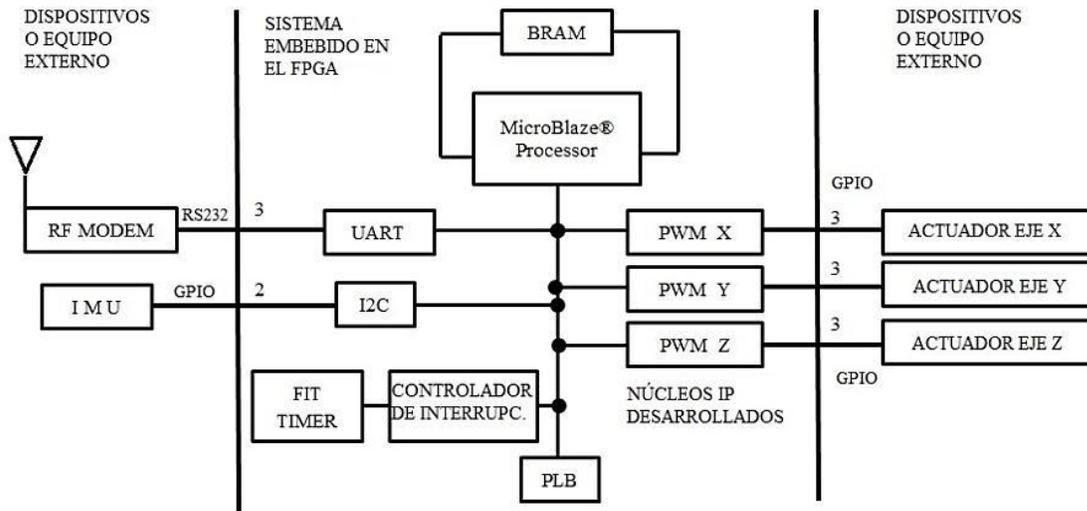


Figura 4.30. Sistema Embebido en el FPGA desarrollado para la MSA

El SE de la figura 4.30 cumple las siguientes tareas dentro de la estructura de la MSA: adquisición temporizada de datos de los sensores de navegación inercial, manejo de actuadores y comunicación de datos con la PC.

La adquisición de datos se realiza mediante un esquema de interrogación, el cual permite al MB enviar por medio de la interfaz I²C un comando que solicite los datos de cada uno de los tres sensores de navegación.

El intervalo de tiempo de adquisición puede ajustarse por medio de la IDE de desarrollo, dentro de las propiedades del núcleo FIT TIMER, figura 4.31. Actualmente dicho periodo está definido en 5 ms, después del cual se envían datos a la PC de cada uno de los tres ejes sensores de la IMU.

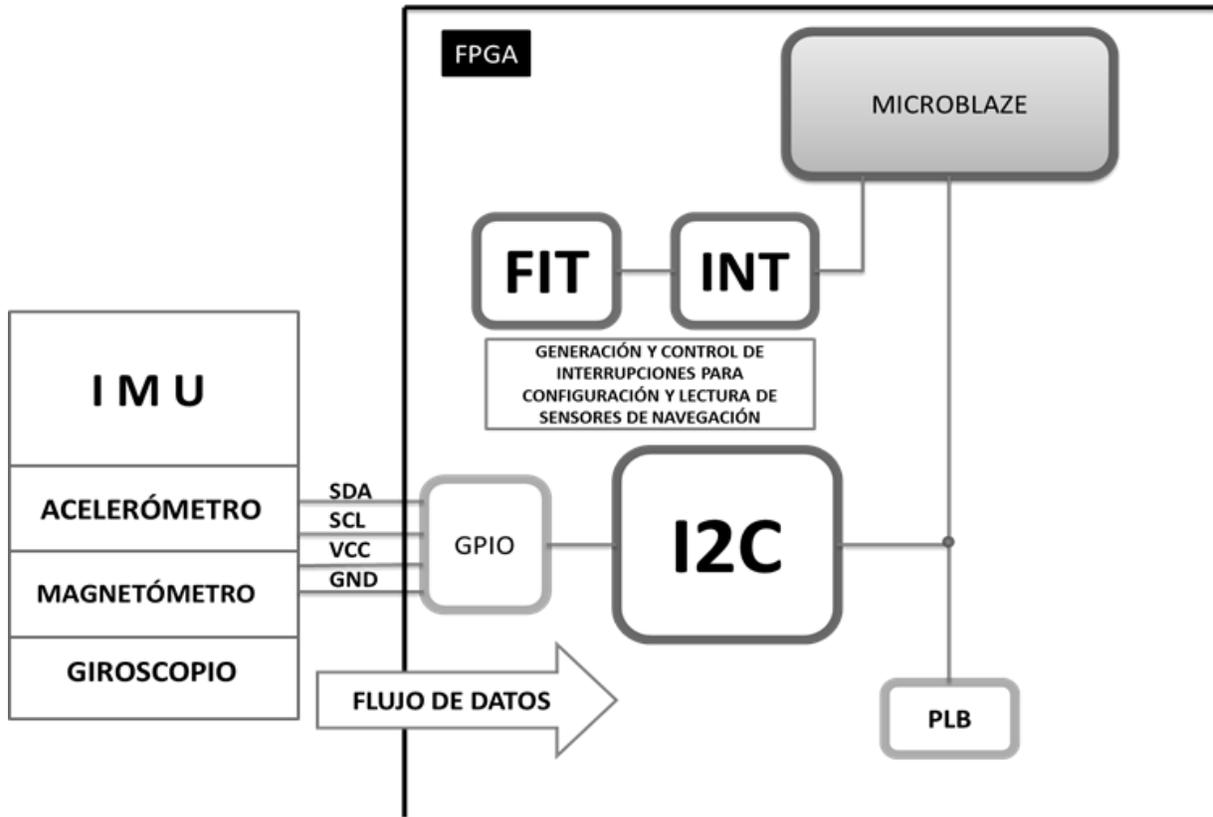


Figura 4.31. Adquisición de datos de sensores de navegación en el SE del FPGA.

Con respecto al manejo de actuadores, la lógica está basada en la recepción de los comandos de control generados en la PC mediante el núcleo UART, y en su posterior decodificación (ciclo de trabajo y sentido de giro de las ruedas inerciales). Esto se resuelve con lógica programada en el firmware del MB, donde además se especifica el orden de recepción de los bytes de datos, integrados en 8 bits para el ciclo de trabajo y un bit para el sentido de giro. Posteriormente cada uno de esos datos se envía a los registros de software asociados al núcleo PWM.

4.4.3. Diseño del núcleo IP PWM

Existen muchas aplicaciones de señales PWM en sistemas electrónicos. Una muy común es el control de velocidad de un motor DC. Las señales PWM pueden usarse para excitar la terminal *gate* de un transistor HEXFET que encienda un motor cuando la señal esté en alto “1” o lo apague cuando esté en “0”.

Si el periodo de la señal PWM es de suficiente corta duración, la velocidad del motor será proporcional al ciclo de trabajo de la señal. Si el periodo de la señal PWM es muy largo, la velocidad del motor ya no se controla con el valor promedio de la señal PWM. En lugar de ello, el

motor girará a velocidad máxima cuando la señal PWM esté en “1” y se detendrá cuando la señal esté en “0”.

En la figura 4.32 se muestra un diagrama de bloques de la lógica del núcleo IP para generación de señales PWM (GPWM) que fue desarrollado a partir del planteamiento descrito en [42] para controlar a los actuadores de la MSA.

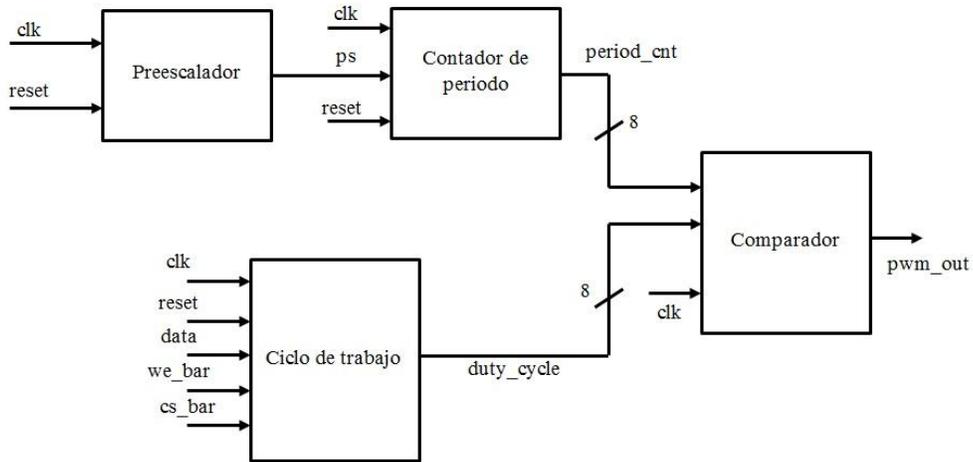


Figura 4.32. Esquema del núcleo IP PWM.

Se utiliza la señal de reloj (clk) del MB para controlar la entrada de reloj del GPWM. Adicionalmente, un preescalador divide la señal del reloj de entrada para crear una señal de frecuencia más baja (ps). Esta señal ps se utiliza para habilitar el bloque contador de periodo, que al ser habilitado, cuenta con el borde de activación del pulso de reloj de entrada.

El contador de periodo de 8 bits cuantifica el periodo de la señal PWM para que sea 256 veces el periodo de la señal preescalada ps. La salida del contador de periodo (period_cnt) es un valor de 8 bits que continuamente realiza ciclos de cuenta de 0 a 255 en binario. El tiempo que requiere la señal period_cnt para contar desde 0 hasta 255 y para volver a cero determina el periodo de la señal de salida PWM (pwm_out).

En el esquema del IP PWM se incluye un bloque que contiene un registro de 8 bits (duty_cycle), este registro es escrito por el MB para controlar el ciclo de trabajo de la señal PWM. El bus de datos de periféricos del MB (PLB) provee los datos que se escriben al registro de datos de entrada por medio de la señal “data” del bloque duty_cycle y controla la operación de escritura por medio de las señales de entrada cs_bar y we_bar.

El ciclo de trabajo de `pwm_output` es igual al valor escrito al bloque `duty_cycle` por el MB dividido entre 256. Por ejemplo, para obtener un ciclo de trabajo del 25% el MB debe escribir el valor 64 al registro `duty_cycle`.

Finalmente, el bloque comparador efectúa la comparación entre el valor en 8 bits de `period_cnt` del bloque contador de periodo con el contenido del registro `duty_cycle`, que también es de 8 bits. Si el valor de `period_cnt` es menor que `duty_cycle`, entonces `pwm_out` es “1”; en otro caso es “0”.

Una vez desarrollado el código asociado con cada uno de los bloques descritos en la figura 4.32 se validó el diseño por medio de un *test bench*, a partir del cual se obtuvieron gráficas como la que se muestra en la figura 4.33. En ella se observa en un periodo de 20 ms, datos de excitación tanto de la señal de reloj (`clk`) como de datos (`data`) de entrada, de los cuales se enmascararon 3 de los 8 bits para realizar cambios con el simulador y observar los cambios generados en la señal de salida `pwm_out`.

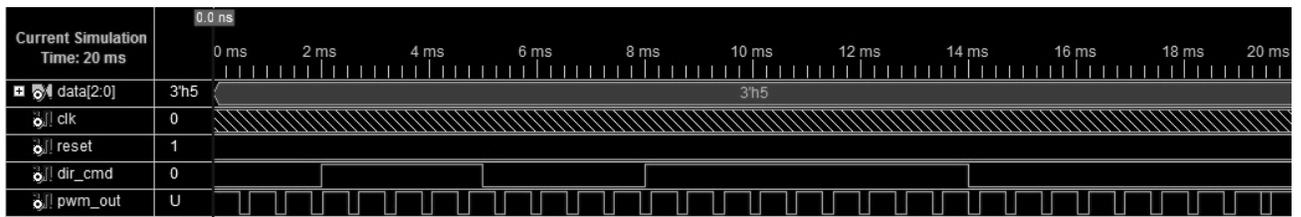


Figura 4.33. Gráfica de *test bench* para el núcleo IP PWM.

4.5. Software de la PC

Otro segmento importante dentro de la instrumentación de la MSA, es el que corresponde a la PC, en la cual se ejecutan en tiempo real en Matlab® los algoritmos de determinación, estimación y control de la orientación. La aplicación está integrada por un sistema de funciones personalizadas interconectadas entre sí, tal como se muestra en la figura 4.34.

Cada una de las funciones del sistema (descritas en código script) contiene un algoritmo ya sea de determinación (TRIAD), estimación (EKF) o control de la orientación de la MSA (CONTROL).

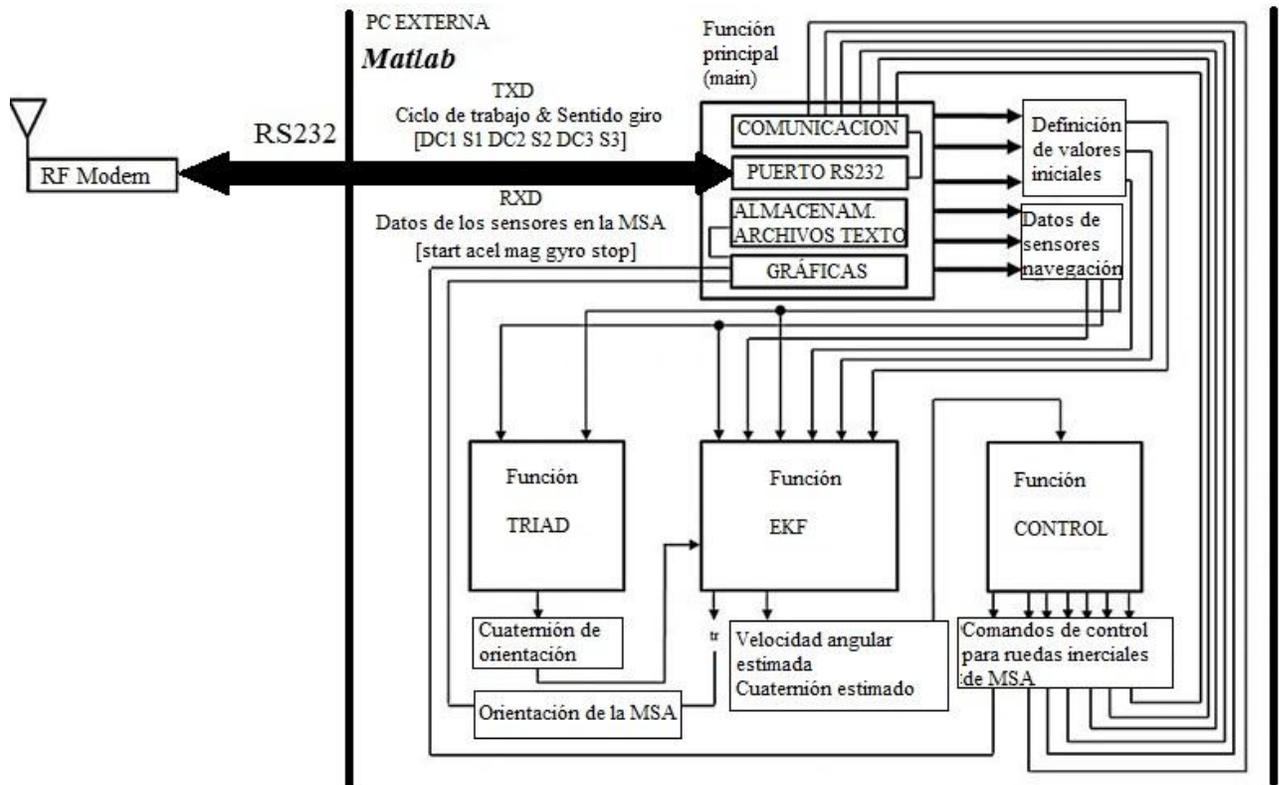


Figura 4.34. Sistema de funciones en MATLAB® para la implementación de los algoritmos de determinación y control de orientación de la MSA.

La modularidad de la estructura definida para el sistema de la figura 4.34, permite modificar los bloques de cualquiera de los algoritmos del esquema de control, por lo cual, se pueden evaluar diferentes estrategias de determinación y control de orientación.

La lógica implementada en cada uno de los bloques que integran el sistema de funciones se describe en los siguientes apartados.

4.5.1. Función Principal

Todas las funciones del esquema de la figura 4.34 se administran por medio de una función principal (main), en la cual se realizan tareas como: la definición de parámetros iniciales del algoritmo EKF, el control del puerto serie (USB) de la computadora (número de puerto, baudaje, número de bits de datos y de paro, etc.) y llamadas a las funciones correspondientes a los algoritmos del esquema de control.

Adicionalmente, los datos de cada prueba realizada se almacenan en archivos de texto plano, para posteriormente graficarlos, principalmente los que corresponden al cuaternión de orientación (de

medición y estimado), los datos de cada una de las componentes del sensor de navegación inercial (magnetómetro, acelerómetro y giróscopo) y las variables dinámicas estimadas del sistema (velocidad angular y posición).

4.5.2. Función TRIAD

El método TRIAD permite determinar la orientación de un satélite o vehículo espacial mediante la determinación de la matriz de rotación del marco de referencia fijo al propio vehículo con respecto a un marco de referencia conocido [5], [14], [15] y [43]. En términos prácticos TRIAD emplea la fusión de dos sensores de navegación inercial, para obtener la orientación del vehículo espacial respecto a un sistema de referencia inercial.

El primer paso del algoritmo considera el cálculo de tres vectores $\bar{r}_1, \bar{r}_2, \bar{r}_3$, que son ortogonales entre sí, en el sistema coordenado donde se llevaron a cabo las mediciones, de tal forma que se definen las siguientes definiciones:

$$\bar{r}_1 = \frac{\bar{u}_M}{\|\bar{u}_M\|} \quad , \quad (4.5)$$

$$\bar{r}_2 = \frac{\bar{r}_1 \times \bar{v}_M}{\|\bar{r}_1 \times \bar{v}_M\|} \quad , \quad (4.6)$$

$$\bar{r}_3 = \bar{r}_1 \times \bar{r}_2 \quad . \quad (4.7)$$

De las expresiones (4.5), (4.6) y (4.7), \bar{u}_M y \bar{v}_M corresponden a vectores de medición adquiridos por medio de dos sensores de navegación inercial distintos, los cuales en el caso de la MSA serán magnetómetro \bar{v}_M y acelerómetro \bar{u}_M . El siguiente paso del algoritmo TRIAD considera el cálculo de otros tres vectores $\bar{s}_1, \bar{s}_2, \bar{s}_3$ los cuales son ortogonales entre sí en el sistema coordenado de referencia [9] y [43], y que se definen como:

$$\bar{s}_1 = \frac{\bar{u}_R}{\|\bar{u}_R\|} \quad , \quad (4.8)$$

$$\bar{s}_2 = \frac{\bar{s}_1 \times \bar{v}_R}{\|\bar{s}_1 \times \bar{v}_R\|} \quad , \quad (4.9)$$

$$\bar{s}_3 = \bar{s}_1 \times \bar{s}_2 \quad . \quad 4.10)$$

De las definiciones (4.8), (4.9) y (4.10), $\bar{\mathbf{u}}_R$ y $\bar{\mathbf{v}}_R$ corresponden a vectores de referencia adquiridos por medio de dos sensores de navegación inercial distintos. Para el caso de la MSA serán también el magnetómetro ($\bar{\mathbf{v}}_R$) y el acelerómetro ($\bar{\mathbf{u}}_R$).

Finalmente, la matriz de transformación que relaciona ambos sistemas de referencia, el medido y el de referencia, se calcula mediante la ecuación (4.12):

$$A_m^r = \mathbf{r}_1 \cdot \mathbf{S}_1^T + \mathbf{r}_2 \cdot \mathbf{S}_2^T + \mathbf{r}_3 \cdot \mathbf{S}_3^T \quad . \quad (4.12)$$

Para efectos de optimización del procesamiento de datos, a partir de la matriz de rotación (A_m^r) obtenida de la ecuación (4.12), se realiza la transformación hacia la representación en cuaterniones unitarios.

El uso de cuaterniones evita la aparición de singularidades, las cuales resultan en indeterminaciones numéricas en los cálculos, como ocurre con los ángulos simples de Euler. Una de las técnicas más utilizadas para obtenerlos es la transformación que usa la representación intermedia del eje de Euler y ángulo principal [29] por medio de las ecuaciones (4.13), (4.14) y (4.15).

$$\cos \varphi = \frac{1}{2} (\text{tr } A_m^r - 1) \quad , \quad (4.13)$$

$$q_i \doteq e_i \text{sen } \frac{\varphi}{2} \quad , \quad i = 1, 2, 3, \quad (4.14)$$

que corresponden a la parte vectorial del cuaternión, mientras que:

$$q_4 \doteq \cos \frac{\varphi}{2} \quad , \quad (4.15)$$

corresponde a la parte escalar del cuaternión.

Se considera que q_1, q_2, q_3 y q_4 son los elementos que integran el cuaternión de orientación, el cual tiene la forma de (4.16):

$$\bar{\mathbf{q}} = [n \ \bar{\mathbf{e}}_1 \ \bar{\mathbf{e}}_2 \ \bar{\mathbf{e}}_3] \quad , \quad (4.16)$$

donde:

- n es la parte escalar del cuaternión
- $\bar{\mathbf{e}}_1 \ \bar{\mathbf{e}}_2 \ \bar{\mathbf{e}}_3$ es la parte vectorial del cuaternión

Para el simulador de control de orientación, se utilizaron las mismas ecuaciones, pero integradas en una función personalizada en *script*. En la figura 4.35 se describe la lógica programada en la función TRIAD [9] y [44].

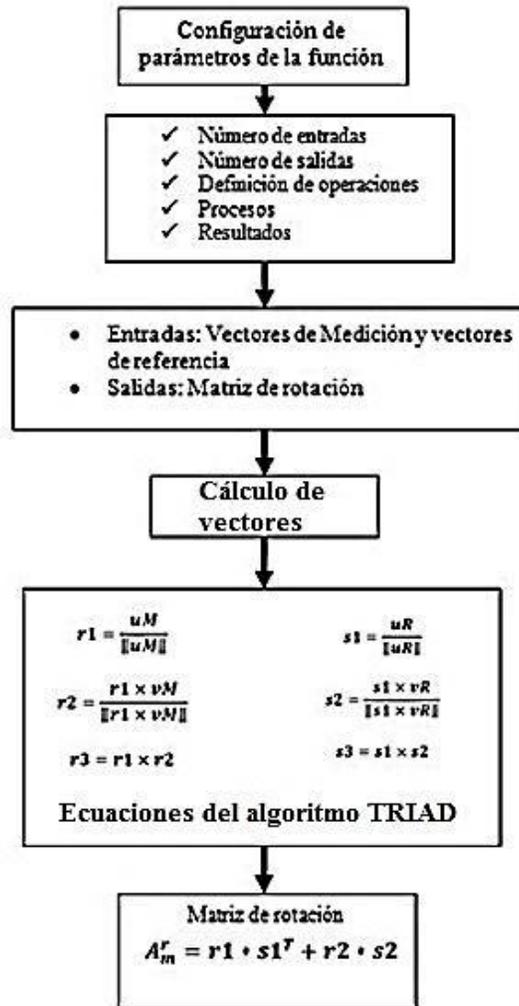


Figura 4.35. Lógica implementada en la función TRIAD.

4.5.3. Función Filtro de Kalman Extendido discreto (EKF)

El filtro de Kalman discreto estima el estado $x \in \mathbb{R}^n$ de un proceso de tiempo discreto que está gobernado por una ecuación diferencial lineal estocástica. De acuerdo a la ecuación de movimiento de la MSA (ecuación 3.75), se trata de un modelo de sistema no lineal. Por tal razón, el filtro de Kalman no se implementó directamente para estimar el valor de los estados del sistema, para ello se recurrió a una variante del filtro de Kalman, denominada Filtro de Kalman Extendido (EKF) [32].

El filtro de Kalman Extendido linealiza las estimaciones respecto a la media y a la covarianza del estado actual.

Dado un sistema de ecuaciones diferenciales estocásticas no lineales de un proceso, donde el vector de estados es $x_k \in \mathbb{R}^n$, y en donde f y h son funciones no lineales, se tiene:

$$\begin{cases} x_k = f(x_{k-1}, u_k, w_{k-1}) \\ z_k = h(x_k, v_k) \end{cases} \quad (4.17)$$

Aproximando el vector de estado y la medición sin tener en cuenta los ruidos de proceso ni el de medición, se tiene

$$\begin{cases} \tilde{x}_k = f(\hat{x}_{k-1}, u_k, 0) \\ \tilde{z}_k = h(\tilde{x}_k, v_k) \end{cases} \quad (4.18)$$

El EKF es simplemente un estimador de estados *ad hoc* que se aproxima a la optimización de las reglas de Bayes por medio de la linealización. Por lo que se plantea el sistema de ecuaciones con la linealización respectiva:

$$\begin{cases} x_k \approx \tilde{x}_k + A(x_{k-1} - \hat{x}_{k-1}) + W_{w_{k-1}} \\ z_k \approx \tilde{z}_k + H(x_k - \hat{x}_k) + V_{v_k} \end{cases} \quad (4.19)$$

Donde x_k , \tilde{x}_k , \hat{x}_k son: el valor actual, la aproximación y la estimación del vector de estado, respectivamente.

Las ecuaciones del algoritmo de filtro de Kalman son las siguientes:

Considerando condiciones iniciales: x_0, P_0

Ecuaciones de actualización de tiempo (predicción):

Proyección del estado hacia adelante: $\hat{x}_k^- = f(\hat{x}_{k-1}, u_k, 0)$

Proyección de la covarianza hacia adelante: $P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T$

Ecuaciones de actualización de mediciones (corrección)

Cálculo de la ganancia de Kalman : $K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1}$

Actualización de la estimación con la medición z_k : $\hat{x}_k = \hat{x}_k^- + K_k (z_k - h(\hat{x}_k^-, 0))$

Actualización de la covarianza del error : $P_k = (I - K_k H_k) P_k^-$

Una de las características importantes del EKF es que el Jacobiano H_k en la ecuación de la ganancia de Kalman K_k sirve para corregir la propagación o destacar solamente las componentes de datos de medición [34].

En [5] y [32] se encuentra la descripción matemática completa del algoritmo y demás procedimientos asociados. En particular, con base en los resultados obtenidos en [5], se desarrolló en Matlab®-Simulink un modelo con una versión simplificada del algoritmo EKF, cuya lógica se resume en la figura 4.36.

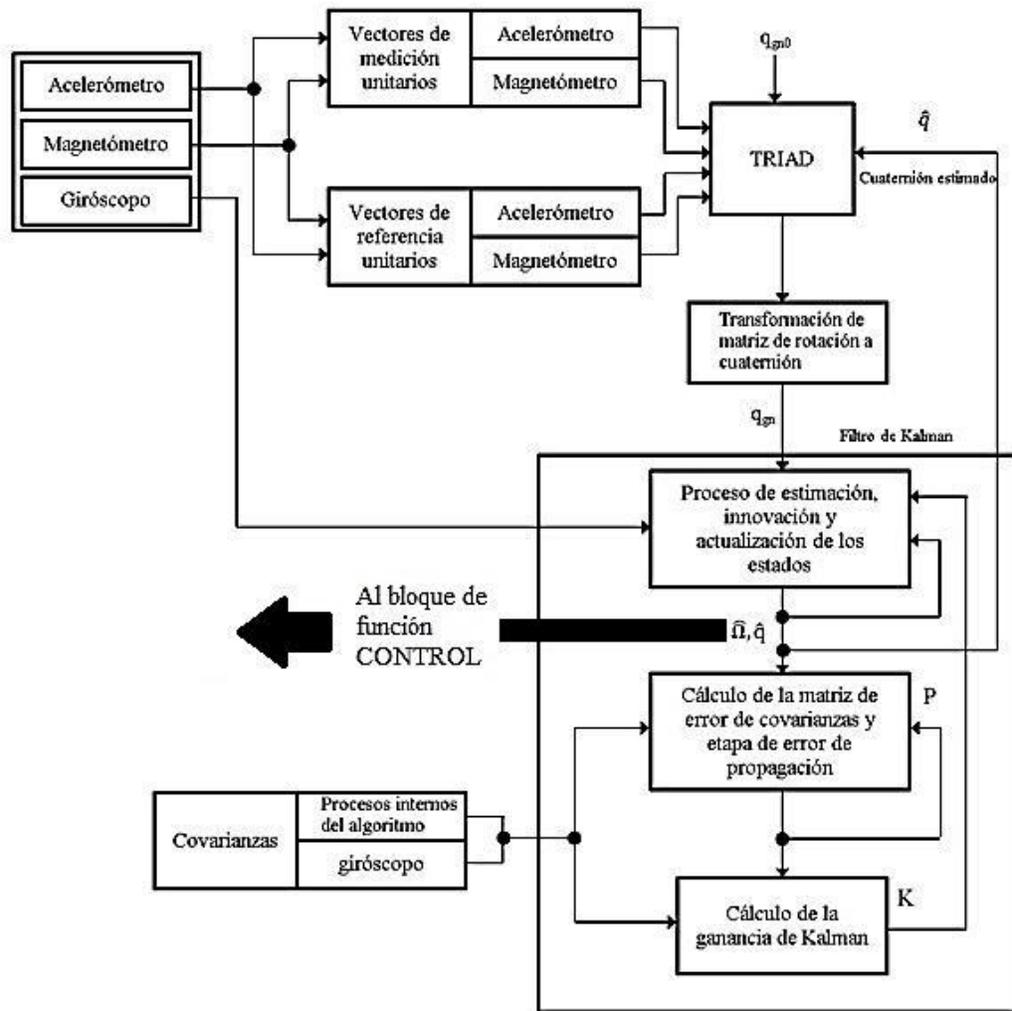


Figura 4.36. Diagrama de bloques de los algoritmos de determinación (TRIAD), estimación y filtrado (EKF) implementado en la MSA.

En la figura 4.36 q_{gn0} es el cuaternión de condiciones iniciales, K es la ganancia de Kalman y P es la matriz del error de covarianzas.

4.5.4. Función Control

La función CONTROL dentro del esquema desarrollado en la PC externa, está programada con una ley de control que corresponde a un control de orientación en tres ejes [5] y [20]. Este algoritmo permite llevar al satélite (representado por la MSA) a una orientación deseada cuando se tienen datos completos que permiten determinar la orientación del satélite (TRIAD), en nuestro caso mediante cuaterniones unitarios obtenidos a partir de un algoritmo de estimación de la orientación del satélite (EKF) con apoyo de sensores de referencia inercial como magnetómetros y acelerómetros triaxiales.

La forma del controlador corresponde a un tipo proporcional derivativo, cuya expresión se muestra en la ecuación (4.20).

$$\tau_{control} = -k_p[3 \times 3] * \hat{\epsilon}[3 \times 1] - k_d[3 \times 3] * \hat{\Omega}[3 \times 1] \quad (4.20)$$

En donde:

- el término $\tau_{control}$ representa los torques de control que se aplicarán a cada uno de los tres actuadores asociados a cada uno de los ejes del sistema de referencia móvil asociado a la MSA (X, Y, Z). El resultado final es un vector de tamaño 3x1.
- los términos k_p y k_d corresponden a las ganancias de los controladores proporcional y derivativo respectivamente, descritos en forma de arreglos matriciales de tamaño 3x3 aplicados para cada uno de los tres ejes principales del satélite (X, Y, Z), al que se consideró de una geometría cúbica.
- El componente $\hat{\epsilon}$ corresponde a la parte vectorial del cuaternión estimado, obtenido a partir de la utilización de los resultados del método de determinación de la orientación TRIAD y la aplicación del algoritmo EKF, resultando un vector de tamaño 3x1.
- $\hat{\Omega}$ representa la velocidad angular estimada, obtenida también a partir del EKF, el cual también se expresa en términos de un vector de tamaño 3x1, es decir, la velocidad angular estimada en cada uno de los ejes (X, Y, Z).

La conversión del torque de control resultante de la ecuación (4.20) debe ser convertida a porcentaje de ciclo de trabajo PWM codificado de 0 a 256 en 8 bits más 1 bit de signo, para ser escrita a los registros PWM correspondientes a cada uno de los actuadores [44].

Tal conversión considera una relación entre el torque de control calculado de la ley de control $\tau_{control}$, en magnitud y signo, el torque máximo del motor DC del actuador con respecto al ciclo de trabajo máximo (en el caso de la MSA, para reducir los efectos de saturación, se consideró un valor máximo del 65%) y la dirección en el sentido de giro del actuador, como se describe en la figura 4.37.

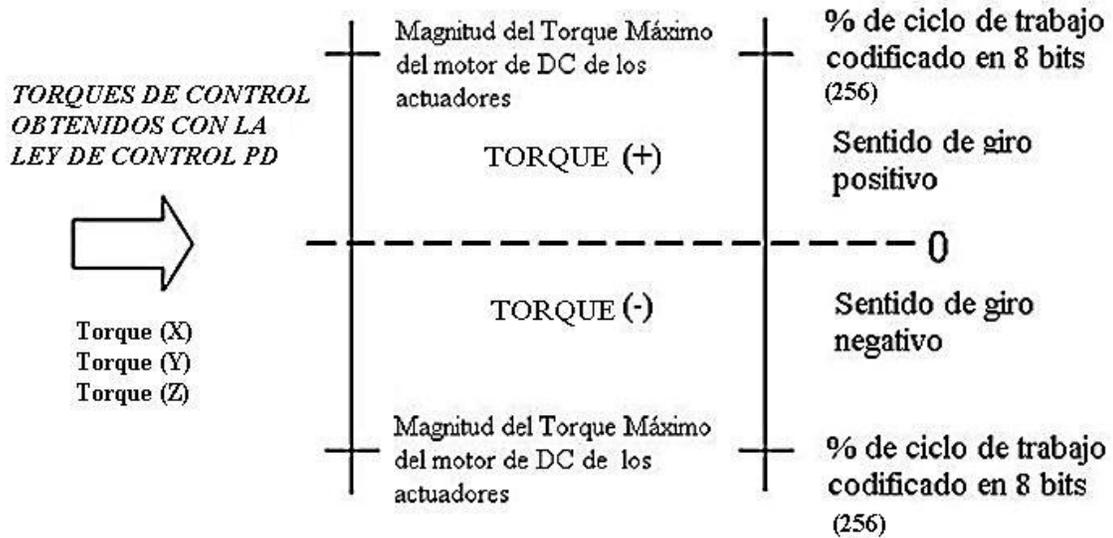


Figura 4.37. Conversión de torque de control a ciclo de trabajo y signo para cada una de las ruedas inerciales de la MSA en la función CONTROL.

En términos matemáticos, la conversión de torque de control a ciclo de trabajo y sentido de giro se muestra en la expresión (4.21).

$$DC (\%PWM) = \frac{\tau_{control} * byte_IP_PWM}{\tau_{max.DCmotor}} * 100 \quad , \quad (4.21)$$

donde:

- $\tau_{control}$ es un arreglo que contiene el torque de control calculado de la ley de control (4.20) en cada una de las tres componentes del simulador (X, Y, Z).
- $\tau_{max.DCmotor}$ es el torque máximo del motor DC usado en el actuador en su respectivo eje. Para la MSA el valor de dicho torque es de 0.07 Nm, según las especificaciones del fabricante (<http://www.compucanjes.com/manuales/rs-555sh20.pdf>).

Una vez realizada la conversión de torque de control a porcentaje de ciclo de trabajo (y sentido de giro) dicho dato se formatea para suministrarlo al núcleo PWM de cada actuador para ejecutar las tareas de control de orientación.

Considerando que el tamaño del registro de datos del núcleo PWM es de 1 byte, el byte de control se calcula conforme a la ecuación (4.22):

$$\text{byte_IP_PWM} = \left(\frac{\%PWM}{100} \right) * 256 . \quad (4.22)$$

En la figura 4.38 se muestra la lógica implementada en Matlab® para la función CONTROL. Es importante destacar que el resultado de salida del bloque son 27 bits, que contienen cada uno, 8 bits de datos (ciclo de trabajo) y 1 bit de sentido de giro. Dichos datos se reciben por medio de la UART en el SE del FPGA en la plataforma móvil de la MSA.

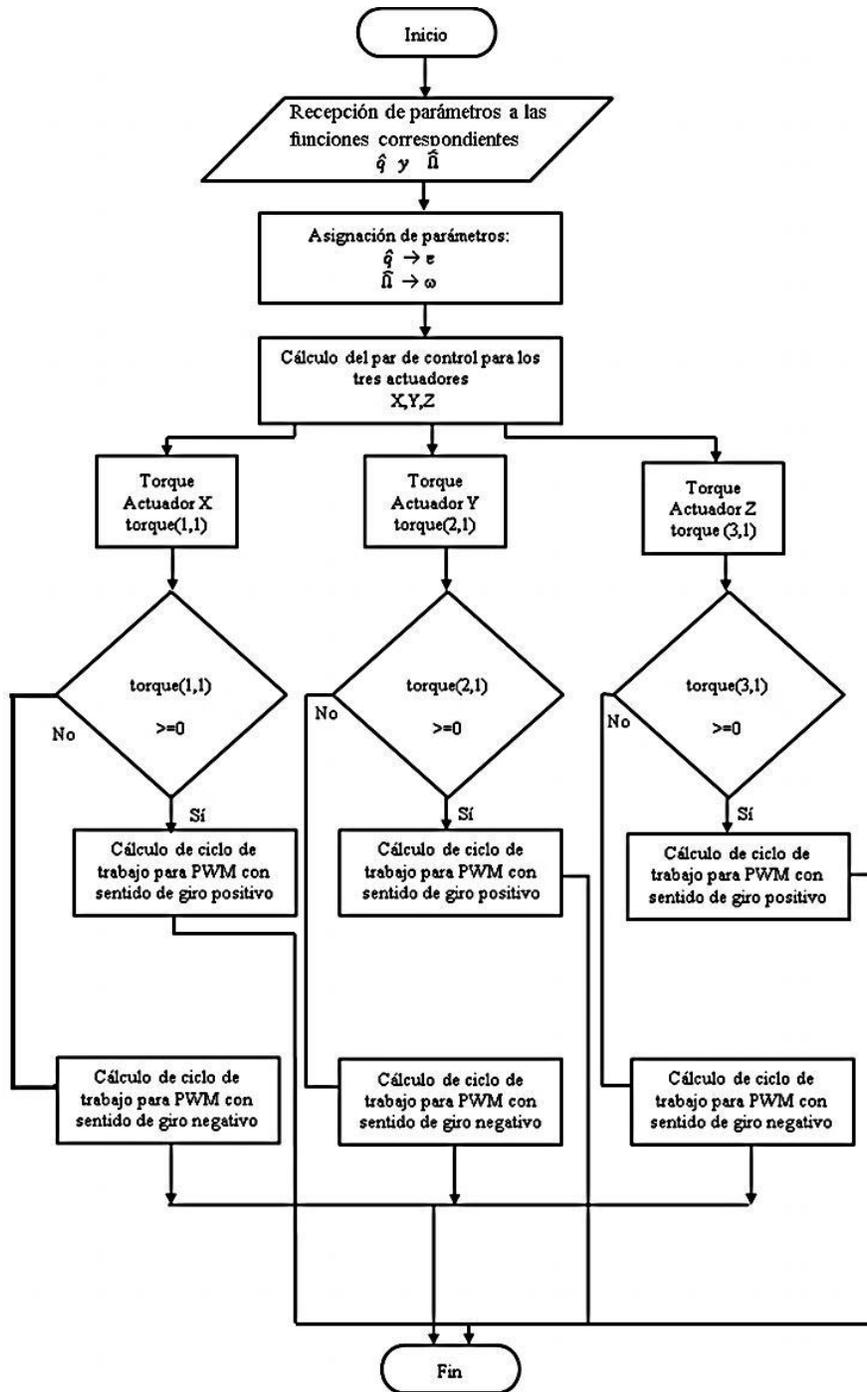


Figura 4.38. Implementación de la función CONTROL.

Es necesario señalar que para las siguientes etapas de desarrollo, particularmente en lo que corresponde a la implementación del esquema en el FPGA, será necesario tomar en consideración la carga computacional que cada uno de los algoritmos requiere. Un estudio presentado en [63] aborda de forma cuantitativa el desempeño temporal de algunos algoritmos más comunes para la

determinación y estimación de parámetros de orientación de vehículos espaciales. De dicho análisis se muestra la tabla 4.2, se observa la duración en la aplicación de cada uno de los algoritmos, particularmente TRIAD y EKF.

Tabla 4.2. Comparación entre la precisión y la ejecución de algoritmos de determinación y estimación [63]				
Algoritmo	Desviación estándar del ángulo yaw (error) (°)	Desviación estándar del ángulo pitch (error) (°)	Desviación estándar del ángulo roll (error) (°)	Tiempo promedio de ejecución (s)
TRIAD	1.2517523	0.8943686	5.7003591	0.0009
EKF	0.9241491	0.8725242	1.5811642	0.5303
Método Q (otro)	1.251716	0.8943650	5.7703597	0.0010

Los resultados de la tabla 4.2 deberán ser tomados en cuenta para la selección del siguiente esquema de instrumentación para determinación y estimación de parámetros de navegación, lo cual sin duda dependerá también de la calidad de los propios sensores y los recursos que posea la plataforma de procesamiento, estableciendo con esto un compromiso entre el desempeño del algoritmo y el costo, tanto en términos económicos como computacionales, que implique la selección e integración de uno u otro algoritmo al esquema final.

Una vez abordados los detalles de instrumentación de la MSA, en el capítulo 6 se abordará lo referente a los experimentos de validación experimental realizados con la MSA totalmente integrada de acuerdo con la técnica HIL.

También se presentan los resultados experimentales y se realiza una discusión en torno a ellos.

Capítulo 5

Experimentos de seguimiento de maniobras de la MSA

5.1. Introducción

Como paso previo a la realización de pruebas automatizadas de control de orientación con la MSA, se efectuaron experimentos utilizando medios de simulación numéricos y virtuales para el seguimiento de maniobras de la plataforma móvil. Con esto se validó y ajustó operativamente el esquema de instrumentación HIL tanto en hardware como en software. Además, se verificó la implementación del algoritmo de determinación de la orientación TRIAD, que integra el esquema completo de control de orientación por medio de un esquema de simulación en Simulink®, particularmente la determinación de la orientación de la plataforma utilizando el algoritmo TRIAD.

5.2. Determinación de la orientación de un vehículo espacial

Esencialmente todos los sistemas de control requieren tres tipos de componentes en hardware: sensores, actuadores y computadora de a bordo. Los sensores se usan para medir el estado del sistema, los actuadores se utilizan para ajustar el estado del sistema y la computadora de a bordo para realizar el procesamiento lógico y matemático de los procesos asociados a los algoritmos [43].

El SDCO de un vehículo espacial utiliza una variedad de sensores y actuadores de diferente tecnología y principio físico de operación. Adicionalmente, el control de un vehículo espacial comienza por determinar su orientación respecto a un marco de referencia inercial.

La determinación de la orientación de un satélite utiliza una combinación de sensores y modelos matemáticos para adquirir y procesar datos para integrar arreglos vectoriales que definen sistemas de referencia móviles e inerciales. Un sistema de referencia móvil generalmente se define fijo al cuerpo del vehículo espacial o respecto a cualquier otro componente de él, mientras que un sistema de referencia inercial se define habitualmente tomando el centro de la tierra como su origen.

Las componentes de los sistemas de referencia tanto inercial como móvil se usan en diferentes algoritmos para la determinación de la orientación del satélite, típicamente usando cuaterniones, ángulos de Euler o matrices de rotación como formas de representación de orientación.

Para la implementación del esquema de seguimiento de la MSA en tiempo real, se empleó el algoritmo de determinación de la orientación TRIAD cuyos detalles se abordan en los siguientes apartados.

5.3. Seguimiento de maniobras del modelo en realidad virtual de la MSA

Con la disponibilidad actual de herramientas en software de modelado y simulación, como las que integran algunas cajas de herramientas (*toolboxes*) de Matlab®-Simulink, se facilita ampliamente la tarea de implementar esquemas de modelado y simulación de sistemas físicos. En la figura 5.1 se muestra como ejemplo el diagrama de flujo que describe un proceso de modelado y simulación del movimiento de un vehículo espacial [29], en él se pueden apreciar algunos procesos que en la práctica son transparentes para el usuario.

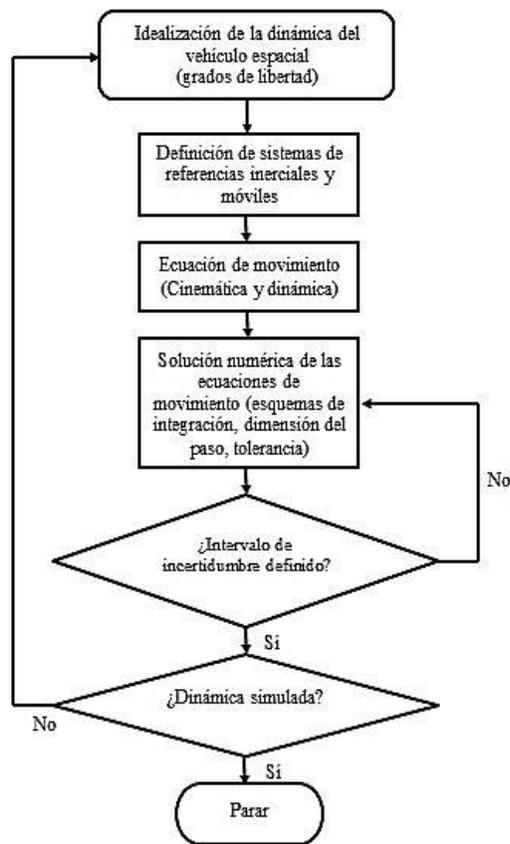


Figura 5.1. Modelado de un vehículo espacial y procedimiento de simulación.

La simulación de sistemas consiste en resolver las ecuaciones que los gobiernan, de tal manera que se logra una buena aproximación en la dinámica real del vehículo. Generalmente las ecuaciones que describen el movimiento de vehículos espaciales pueden ser del tipo no lineal o expresadas mediante ecuaciones diferenciales ordinarias, por lo cual, su solución completa eventualmente es posible, pero es necesario utilizar aproximaciones numéricas como única alternativa de solución.

Para esta primera validación de la integración de la instrumentación de la MSA, se implementó un sistema de seguimiento del modelo virtual de la MSA desarrollado a partir de un modelo en Matlab®-Simulink y visualizado por medio de un modelo en realidad virtual realizado en CAD (Solid Works). El modelo en realidad virtual contempló características físicas de la MSA como su tensor de inercia, el cual se calculó por medio de software CAD. Este modelo permitió validar funcionalmente un primer esquema de software interactuando en tiempo real con hardware de orientación.

5.3.1. Descripción de elementos del seguimiento de la MSA

La estructura operativa del seguimiento de la MSA (SMSA), figura 5.2, está particionada en dos plataformas computacionales diferentes: una tarjeta de desarrollo FPGA operando como computadora a bordo de la MSA y una computadora personal externa, ambos segmentos están enlazados por medio de dos radio módems (módems RF) para la transferencia inalámbrica de datos [9].

La computadora a bordo FPGA interactúa con la instrumentación de la plataforma de la MSA, principalmente con sensores de navegación inercial y un radio módem. Por otro lado, en la computadora personal (PC), se recibe por medio de otro radio módem datos de navegación de la MSA y se ejecuta el software de análisis y simulación matemática en Matlab®, donde está desarrollado el esquema de simulación del SMSA en Simulink® [9] por medio de bloques.

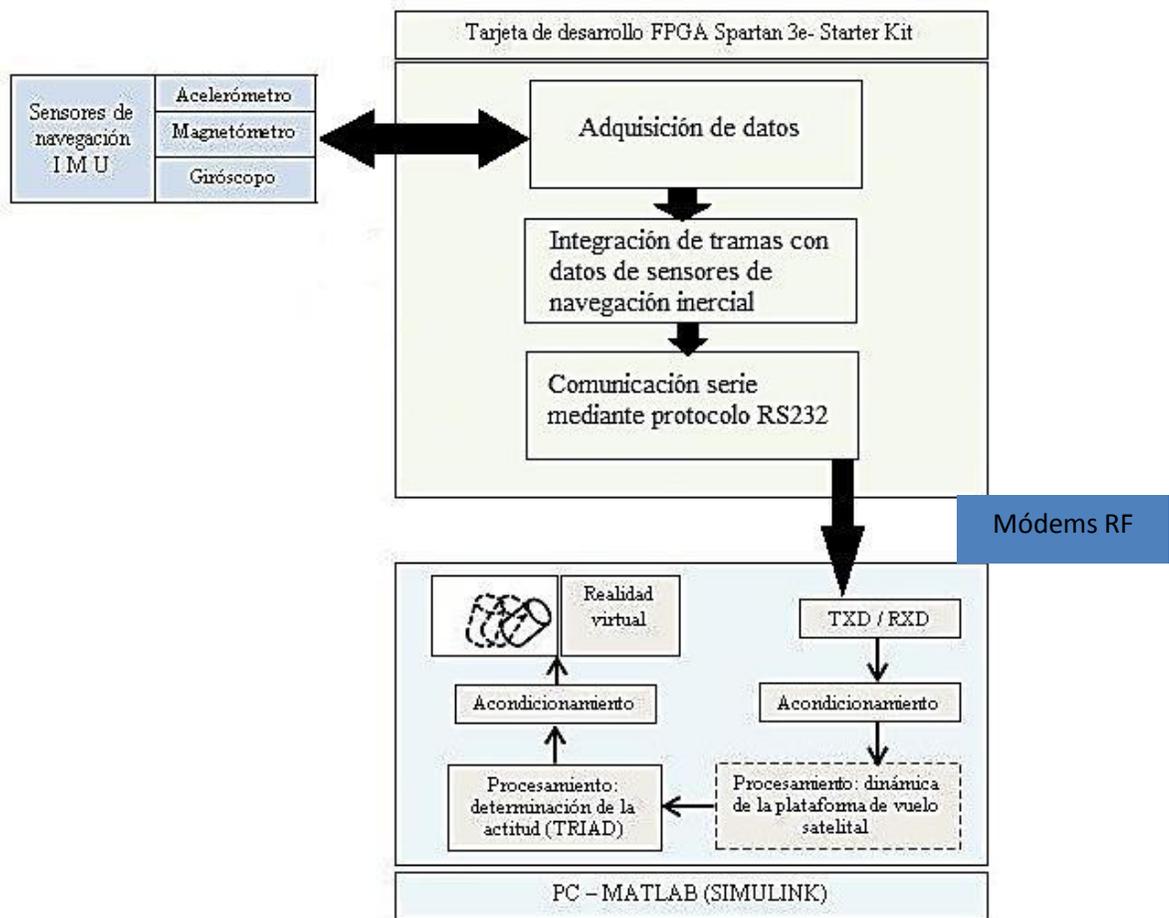


Figura 5.2. Esquema de implementación del SMSA.

La complejidad del modelo del sistema se describe en el software asociado al esquema HIL, mediante la representación matemática de los segmentos de la dinámica del sistema físico. El propósito principal de la técnica HIL es validar el hardware y software de la instrumentación por medio de un simulador antes de integrarlo al proceso real. Por lo cual, el procedimiento constituye en sí una plataforma de desarrollo y validación efectiva para pruebas de sistemas embebidos en tiempo real [9].

La asignación de tareas de los segmentos principales del simulador desarrollado con base en el modelo HIL (FPGA y PC), obedece a la necesidad de resolver de forma ágil la adquisición de datos de los sensores de navegación y el procesamiento de los mismos.

En el caso de la MSA, la adquisición de datos de la IMU (magnetómetros, acelerómetros y giróscopos) a bordo de la plataforma móvil, se realiza con un protocolo serial de alta velocidad I²C,

del cual carece una PC convencional. Mientras que la solución al procesamiento de datos para la determinación de la orientación de la plataforma por medio del algoritmo TRIAD, se realiza en una computadora personal (PC), en la cual se ejecutan los modelos de simulación completos en Matlab®-Simulink y se visualizan los movimientos por medio de un modelo en realidad virtual.

En una tarjeta de desarrollo FPGA se integra un sistema embebido basado en núcleos de procesamiento y periféricos, los cuales permiten realizar las tareas de adquisición, procesamiento y comunicación de datos, vinculando la información de los sensores de navegación y el esquema de simulación programado en la PC.

5.4. Detalles del SMSA desarrollado en Simulink

El modelo que integra los bloques del SMSA está compuesto por funciones personalizadas desarrolladas en Simulink®. Dichos bloques tratan los datos procedentes de la plataforma FPGA, desde su recepción por medio de un protocolo serie hasta su visualización en un modelo en realidad virtual como se muestra en la figura 5.3.

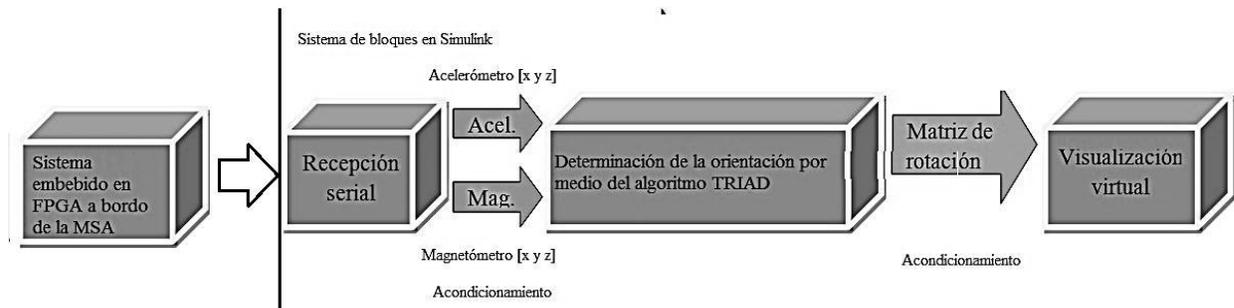


Figura 5.3. Sistema de bloques desarrollado en Simulink para el SMSA.

El procesamiento de datos que se realiza en la PC considera la recepción de datos de los sensores a través de un bloque que resuelve las comunicaciones del puerto USB de la PC. Después, pasa por otro bloque de conversión de datos (acondicionamiento) donde los datos originales (en formato ASCII) se convierten a enteros antes de ser decodificados a formatos compatibles con Simulink® para su tratamiento posterior.

Los datos acondicionados se transfieren a un bloque posterior que contiene la lógica del algoritmo TRIAD, en donde se determinará la orientación de la MSA, a partir de las mediciones de los

sensores de navegación inercial a bordo, para finalmente visualizar los movimientos por medio del modelo en realidad virtual.

En la figura 5.4 se muestra el diagrama de bloques del sistema desarrollado. Adicionalmente se tienen módulos para visualizar cada una de las componentes (X, Y, Z) tanto del acelerómetro como del magnetómetro para verificar la magnitud de los datos adquiridos y procesados.

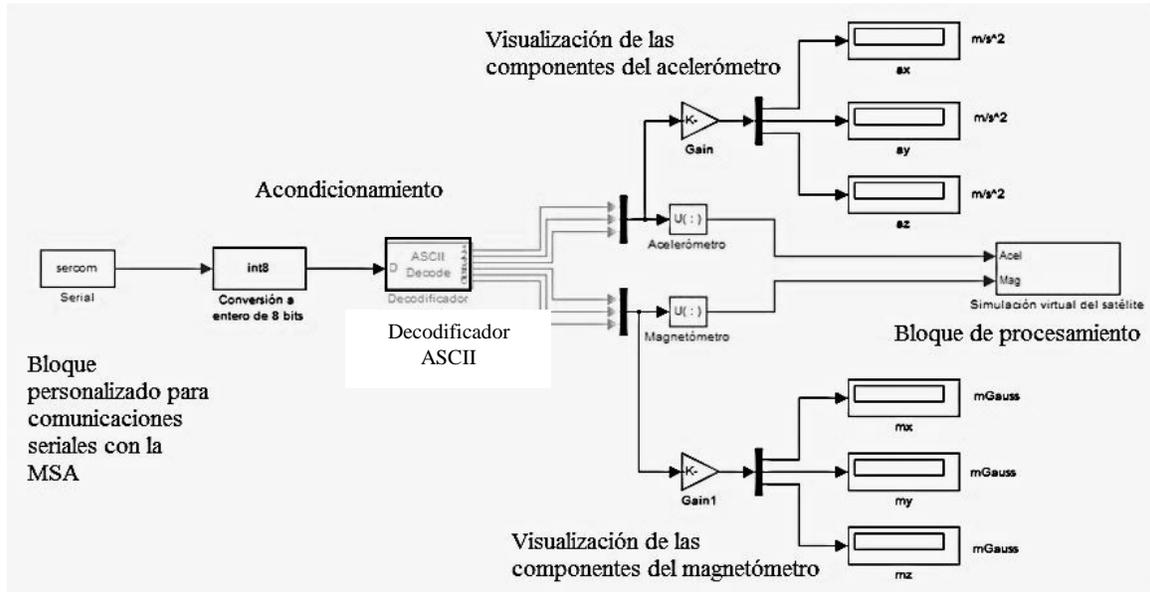


Figura 5.4. Esquema del SMSA en Simulink.

Partiendo de una clasificación de sensores de navegación, como la que se presenta en [65], donde se establecen dos tipos de sensores de orientación. Los primeros, aquellos que determinan un vector \vec{v} en los ejes cuerpo, esto es, \vec{v}^B . Además, dicho vector se supone conocido en los ejes de referencia (inerciales u orbitales), es decir, \vec{v}^N es conocido. En cuanto al segundo tipo de sensor, es aquél que mide directamente la velocidad angular $\vec{\omega}^B$ con respecto al sistema de referencia inercial. La orientación de cuerpo rígido a partir de los datos de estos sensores se determina integrando las ecuaciones diferenciales cinemáticas, sin embargo, los errores de medición se acumulan generando una cierta deriva la cual es necesario compensar con mediciones de algún sensor cuyas medidas se den en los ejes de referencia [63].

Con base en esta definición, se establece que tanto el magnetómetro como el acelerómetro son sensores de navegación cuyo vector de medición está relacionado con una medición conocida en los ejes del sistema de referencia inercial, base fundamental del planteamiento del algoritmo TRIAD para la determinación de la orientación del cuerpo rígido.

La interfaz entre la MSA y el sistema de bloques en Simulink fue el puerto USB de la PC en modo de recepción de datos. En Simulink® existen bloques dedicados para realizar la recepción y transmisión de datos por medio del puerto serie de la PC, lo cual es necesario para crear simulaciones en tiempo real.

Sin embargo, los bloques predefinidos pueden limitar fuertemente a las simulaciones durante la realización de las pruebas experimentales, tal como sucedió con el bloque de comunicaciones seriales del esquema presentado, el cual no contaba con un *buffer* de recepción de datos de longitud variable. Esto representó un problema para el SMSA debido a la variación constante de la longitud de las tramas de los sensores enviadas por el sistema embebido en el FPGA. Esto se resolvió con la creación de un bloque personalizado, el cual se describe en el siguiente apartado.

5.5. Bloque personalizado de comunicaciones seriales con *buffer* de recepción de longitud variable

Uno de los primeros retos en la implementación del esquema de SMSA estuvo enfocado en el desarrollo de un *buffer* de datos de longitud variable, en función del tamaño de las tramas de datos de medición de los sensores de navegación inercial instalados a bordo de la plataforma de simulación [9]. Aunque en Simulink® existen bloques que realizan esta tarea, sus restricciones en configuración no permitieron adaptarlos a las condiciones operativas del sistema. Por tal motivo, se desarrolló un bloque personalizado de comunicación serie a partir del bloque *Matlab S-function* del toolbox *User-defined functions*. La lógica del bloque personalizado se resume en la figura 5.5.

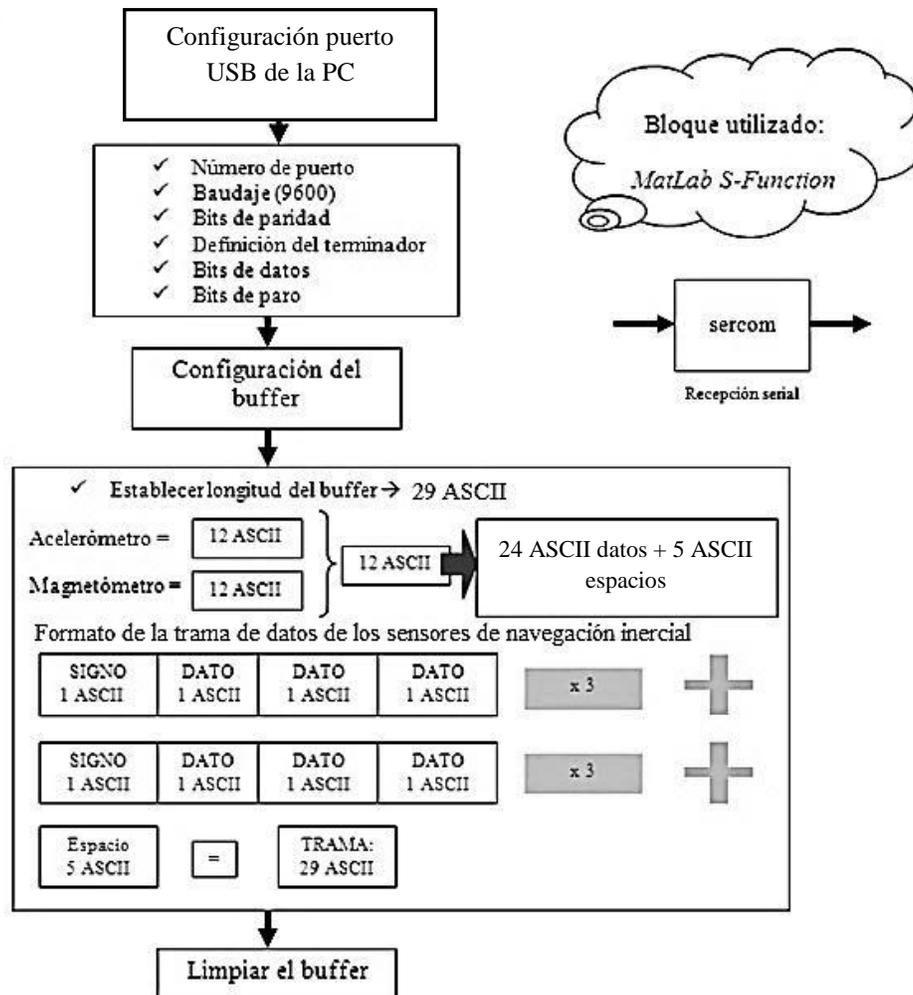


Figura 5.5. Lógica del bloque personalizado de comunicaciones serie.

Como se muestra en la figura 5.5, la lógica del bloque inicia con la definición de los parámetros de comunicaciones del puerto de la PC (número de puerto, número de bits de datos, número de bits de paro, definición del terminador).

En torno al diseño del bloque, es importante destacar la definición de la longitud del *buffer* de recepción, la cual se consideró de 29 caracteres ASCII, de los cuales 24 corresponden a los datos en tres ejes (X, Y, Z) tanto del magnetómetro como del acelerómetro y 5 caracteres extra. En estos se colocan ceros, pero sus posiciones podrán ser ocupadas por datos de navegación conservando un *buffer* de dimensión fija.

Esta estrategia resultó efectiva ante la constante variación en la longitud de los datos de cada eje de los sensores de navegación y la imposibilidad técnica de definir un *buffer* de longitud variable.

5.6. Bloque personalizado para el algoritmo de determinación de la orientación

Otro de los bloques que se requirió encapsular en un bloque de función personalizada correspondió a la lógica del algoritmo de determinación de la orientación TRIAD, que fue presentado en el capítulo 4.

5.6.1. Transferencia de la lógica del TRIAD a un bloque personalizado de Simulink®

Con base en la descripción del algoritmo TRIAD (realizada en el capítulo 4), particularmente en términos de sus ecuaciones, se implementó su lógica en un bloque personalizado de Simulink® cuya descripción se muestra en la figura 5.6.

En el diagrama de flujo de la figura 5.6, el proceso inicia con la adquisición de datos por medio del puerto serie (USB) de la PC, en donde se recibe la trama de datos provenientes de la MSA. Dichos datos están agrupados inicialmente en un bloque de 29 bits en formato ASCII, y posteriormente se separan en dos arreglos que contienen datos en tres ejes (X, Y, Z) tanto del acelerómetro como del magnetómetro.

Adicionalmente, se deben definir previamente dos vectores de referencia tanto para el magnetómetro (medición del campo magnético local) como para el acelerómetro (valor de la aceleración gravitacional). Una vez establecidos dichos arreglos de referencia, puede ejecutarse la simulación de este bloque.

Una vez que se encapsula el algoritmo dentro del bloque de función personalizada en forma de un *script*, la solución a las operaciones aritméticas asociadas a TRIAD se resolvieron con las operaciones matriciales propias del ambiente Matlab®.

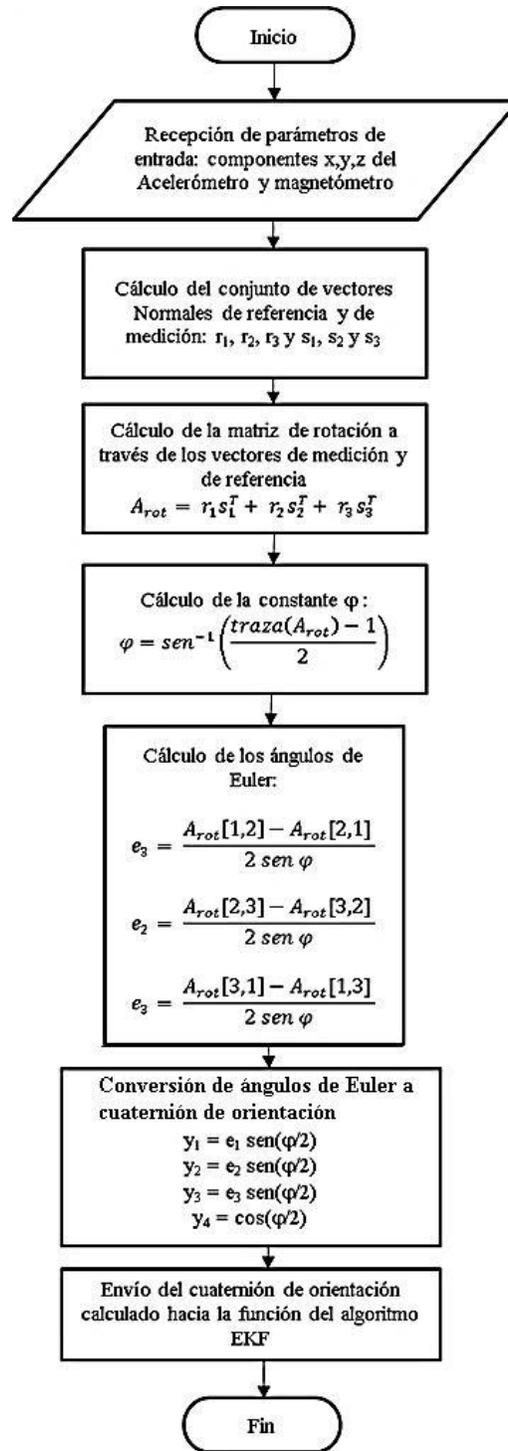


Figura 5.6. Flujo de diseño del bloque personalizado TRIAD.

En la figura 5.7 se muestra el modelo en Simulink que corresponde al cálculo de la determinación de la orientación de la MSA utilizando el método TRIAD.

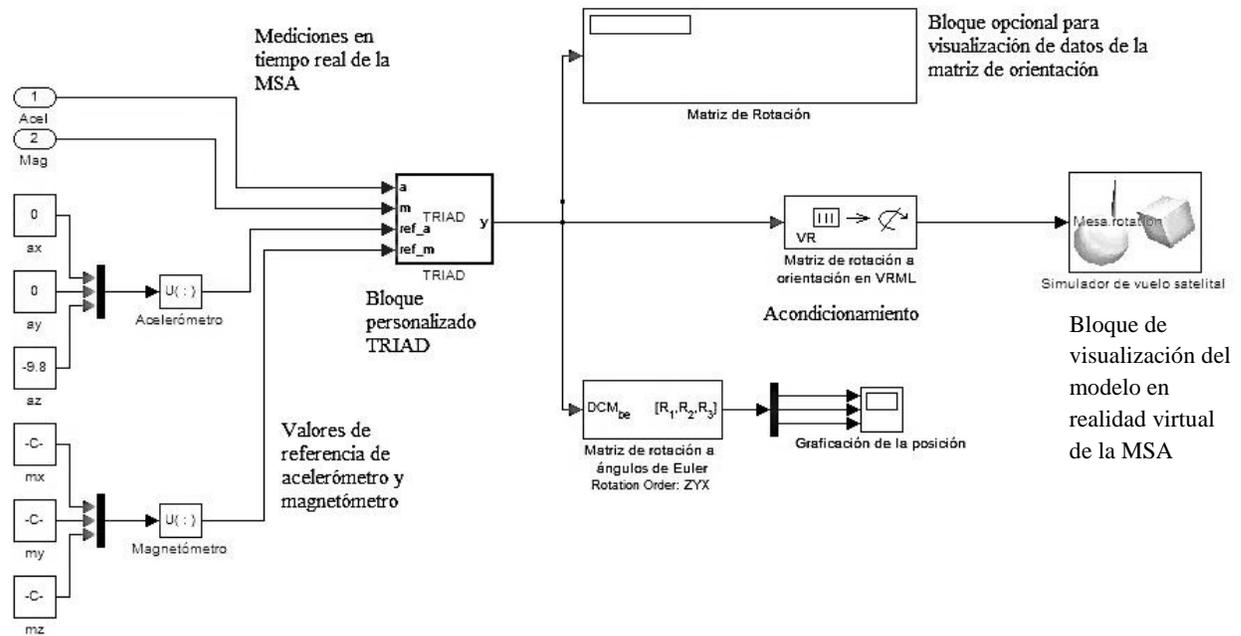


Figura 5.7. Implementación del bloque TRIAD en el SMSA.

5.6.2. Bloque de acondicionamiento: matriz de rotación a VRML

Dentro de los bloques más importantes que realizan tareas de acondicionamiento de señales de datos dentro del esquema de SMSA, se encuentra el que cambia el cuaternión unitario a representación en ángulos de Euler. Esta forma de representación de la orientación es necesaria para vincular los resultados del algoritmo TRIAD con el modelo en realidad virtual.

La transformación entre una y otra forma de representación no es directa, por lo que primero se convierte la matriz de rotación a cuaternión unitario y a partir de dicho cuaternión unitario se hace la transformación a eje principal y ángulo de Euler, como se describe en [29].

Los resultados del cálculo de la matriz de rotación son acondicionados para obtener los elementos del cuaternión unitario de orientación.

5.7. Descripción de la prueba de seguimiento virtual de la MSA

Con base en el modelo de bloques descrito en la figura 5.7, se realizaron experimentos para mover de forma manual la plataforma móvil de la MSA. Se consideraron desplazamientos angulares de 45° en torno a cada uno de los ejes X,Y,Z del sistema de referencia móvil de la plataforma, visualizándolos en la pantalla de la PC por medio del modelo en realidad virtual, como se muestra en la figura 5.8.

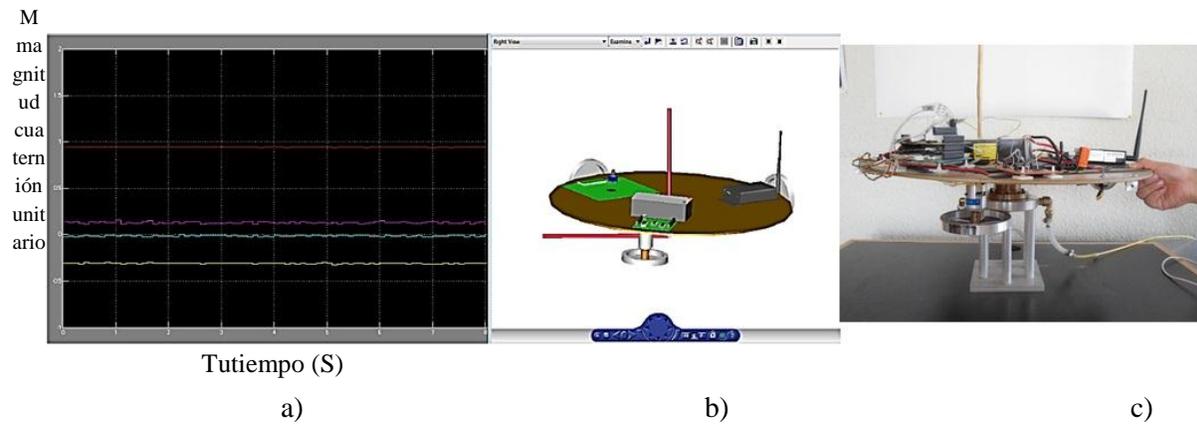


Figura 5.8. Pruebas de seguimiento del modelo en realidad virtual de la MSA.

a) Gráfica correspondiente a cada uno de los elementos del cuaternión de orientación de la MSA.

b) Modelo en realidad virtual de la MSA.

c) Plataforma móvil de la MSA.

5.7.1. Posición inicial para el inicio de pruebas de seguimiento de la MSA

Todas las pruebas partieron de la posición cero, en la que el cuaternión \bar{q} tenía un valor $\bar{q} = [1 \ 0 \ 0 \ 0]$, el cual se validó por medio de la gráfica de elementos del cuaternión mostrada en la figura 5.9. En esta figura se puede verificar, de acuerdo a la definición previa de cada uno de los cuatro elementos del cuaternión, que la parte vectorial se encuentra en cero, mientras que la parte escalar tiene un valor de 1.

5.7.2. Maniobra en torno al eje X

Se realizó un primer movimiento manual, con un valor de aproximadamente 45° en torno al eje X, relativo al sistema de referencia móvil definido sobre la plataforma de la MSA, como se muestra en la figura 5.10.

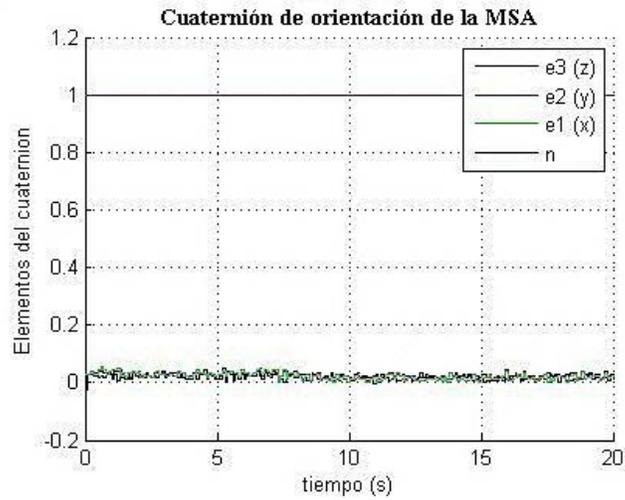


Figura 5.9. Posición cero de la MSA al inicio de cada prueba.

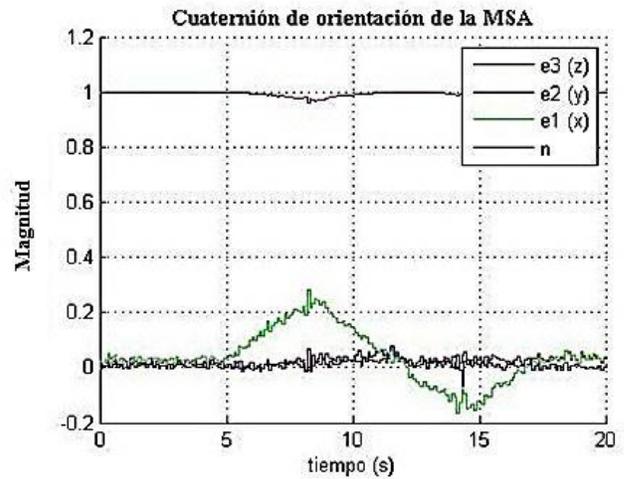


Figura 5.10. Posición de la MSA a 45° respecto al eje X.

5.7.3. Maniobra en torno al eje Y

El segundo movimiento manual con un valor de aproximadamente 45° se realizó en torno al eje Y, relativo al sistema de referencia móvil definido sobre la plataforma de la MSA, como se muestra en la figura 5.11.

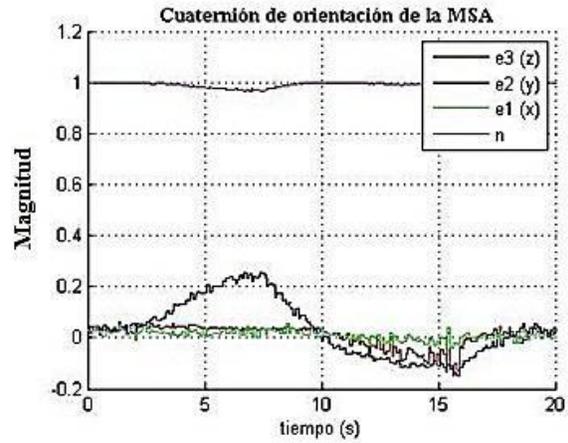
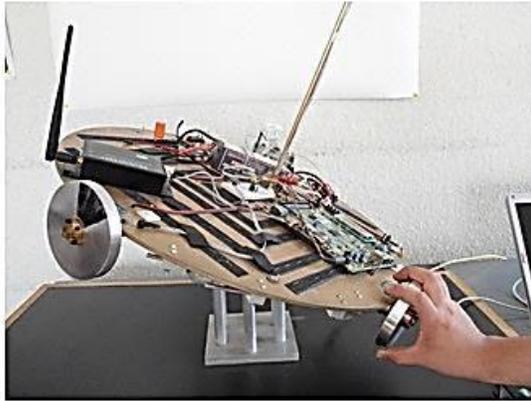


Figura 5.11. Posición de la MSA a 45° respecto al eje Y.

5.7.4. Maniobra en torno al eje Z

El tercer movimiento manual también con un valor de aproximadamente 45° se realizó en torno al eje Z, relativo al sistema de referencia móvil definido sobre la plataforma de la MSA, como se muestra en la figura 5.12.

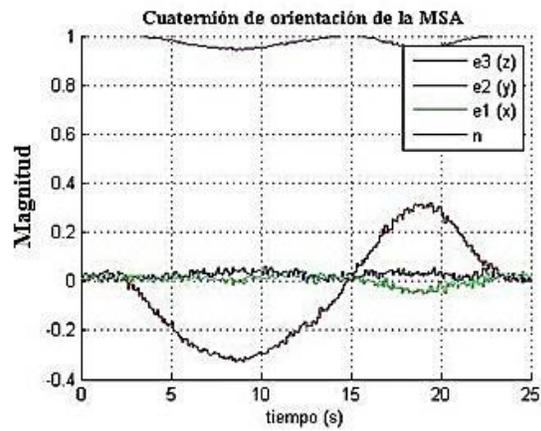
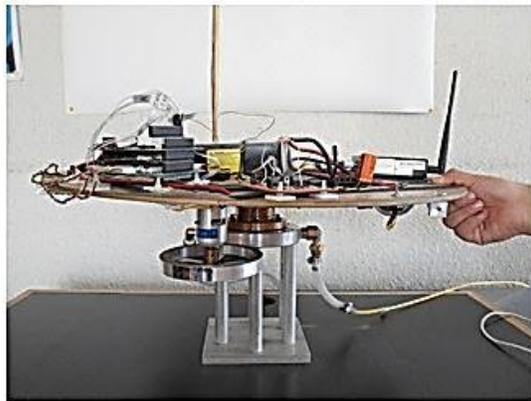


Figura 5.12. Posición de la MSA en movimiento 45° en torno al eje Z.

5.8. Discusión de resultados

A partir de los resultados de la implementación del SMSA mostrados en las secciones anteriores de este capítulo, se identifica un tema muy importante de desarrollo; la fusión del modelado y la simulación de sistemas físicos, figura 5.13.

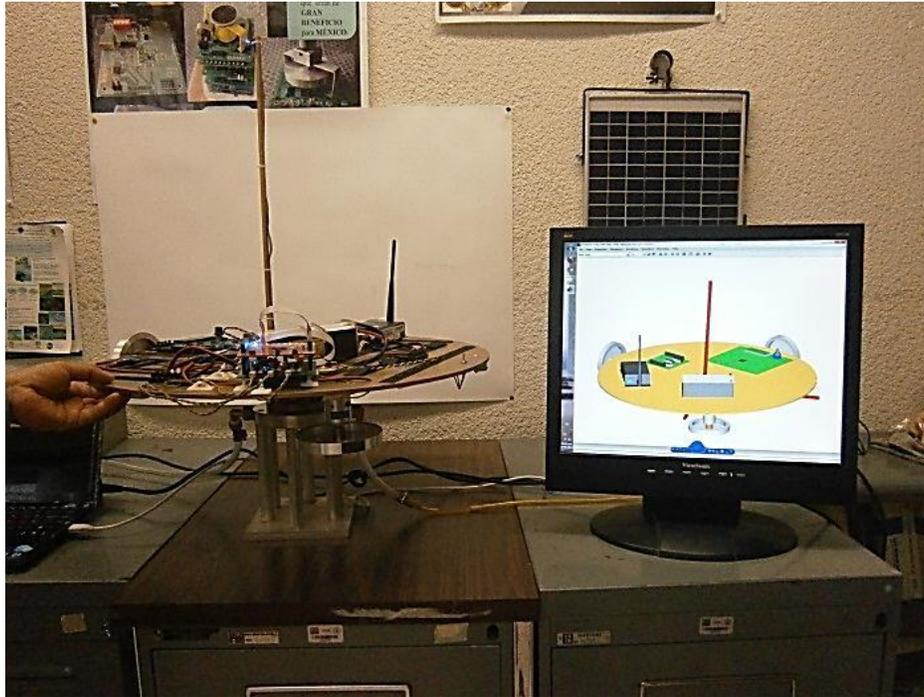


Figura 5.13. Experimentos de seguimiento virtual de la MSA desarrollados en el IINGEN-UNAM.

Esta fusión de vertientes define una opción de validación de sistemas utilizando sólo herramientas en software. Dicha opción tiene como ventaja principal un relativo menor tiempo de desarrollo y un costo bajo.

Sin embargo, utilizando solamente software existe el compromiso de contar con los modelos detallados de cada uno de los elementos del sistema que se requiera simular. Lo cual pudiera llegar a resultar una tarea complicada, casi con el mismo nivel de complejidad que la implementación en hardware. No obstante, el trabajo se compensa al generar una herramienta portable, y útil para el análisis de la dinámica del sistema y para realizar experimentos de control de orientación satelital utilizando diversas leyes (tanto lineales como no lineales), figura 5.14. Esto permite anticipar el comportamiento físico del simulador para plantear nuevas soluciones y estrategias de control.



Figura 5.14. Análisis de estrategias de control utilizando el SMSA.

Por ello, en instrumentación espacial se debe considerar el uso de herramientas en software como la que se ha presentado en este capítulo y, con base en los resultados podrá definir el hardware que lo contendrá físicamente, así como desarrollar una metodología para la transferencia de la lógica de algoritmos y procedimientos a hardware.

Es importante señalar que, tanto la simulación experimental, que forma parte del núcleo central de esta tesis doctoral, como las herramientas que forman parte de una vertiente de análisis y simulación en software, representan instrumentos muy valiosos en términos del establecimiento de medios de comprobación previos al lanzamiento de una misión espacial real.

Tanto un nivel de madurez en términos de desarrollo tecnológico como la propia confiabilidad inferida a los sistemas espaciales son elementos necesarios para minimizar una falla en una misión espacial, que de ocurrir conllevarían una serie de responsabilidades de diversa índole y alcance. No obstante, el uso de herramientas adyacentes, como las que se proponen en esta investigación, representan un cambio en el paradigma de diseño científico y tecnológico, el cual permitirá buscar satisfacer todos los requerimientos establecidos y lograr su éxito. En términos cuantitativos, como se establece en [60], existe un indicador denominado **nivel de preparación tecnológica** (*Technology Readiness Level TRL*), el cual, mientras más alto, establecerá *a priori* un alto nivel de confianza y certeza al planteamiento científico y tecnológico de la misión.

En el siguiente capítulo se describe el esquema de instrumentación de la MSA, la cual se basó en gran parte de los conceptos presentados en este capítulo, como el algoritmo de determinación de orientación TRIAD, el esquema de instrumentación validado experimentalmente y el sistema embebido en un FPGA. Todo el sistema integrado conforme a la técnica de co-simulación HIL.

Capítulo 6

Validación experimental de la MSA

6.1. Plataforma de simulación satelital de vuelo orbital MSA

La plataforma de simulación satelital MSA es un sistema que permite validar en Tierra estrategias de control de orientación para vehículos espaciales, integrada conforme a la técnica HIL, figura 6.1, consta de dos segmentos principales: una plataforma móvil suspendida sobre un cojinete de aire y una PC externa.

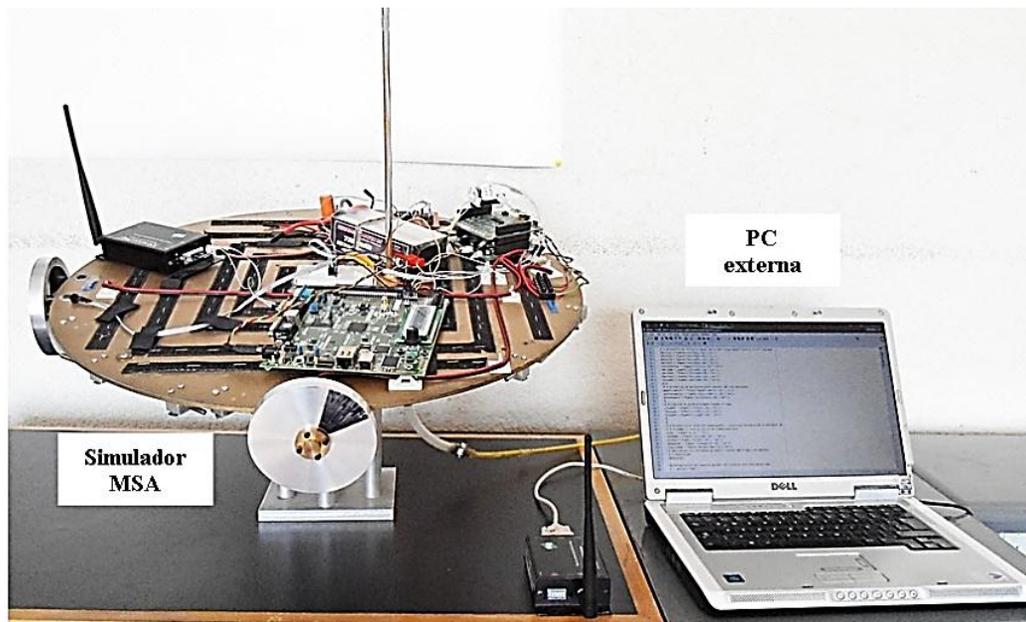


Figura 6.1. Simulador MSA.

En la plataforma móvil, donde se emula la condición bajo torque, se pueden realizar maniobras angulares de 360° en torno al eje Z y de 30° en los ejes X y Y, y se tiene instalada la instrumentación del SDCO: sensores de navegación inercial, plataforma de cómputo, actuadores, suministro de energía eléctrica a bordo y comunicaciones inalámbricas.

Los sensores de navegación inercial (acelerómetros, magnetómetros y giróscopos) están integrados en una tarjeta impresa donde comparten el protocolo I²C para comunicaciones seriales de alta velocidad.

Por su parte, en la PC, que funciona como plataforma de coprocesamiento externa de la MSA, se ejecutan en Matlab® algoritmos de procesamiento del esquema de determinación y control de orientación descritos por medio de funciones personalizadas integradas en *scripts*. Esta opción permite validar en hardware la lógica de software (algoritmos de control), al mismo tiempo que permite su optimización y reducción, coadyuvando a que en una siguiente etapa de desarrollo de la MSA, se integren totalmente como soluciones hardware o software en el sistema embebido del FPGA a bordo del simulador, permitiendo una validación inter cruzada del sistema.

6.2. Inicialización de la MSA

Considerando la arquitectura del simulador MSA, los pasos para preparar al simulador para la ejecución de pruebas de control de orientación son los siguientes:

- 1. Verificación de la carga de voltaje en la batería a bordo de la plataforma móvil de la MSA*
- 2. Revisión de las conexiones eléctricas a bordo de la plataforma móvil*
- 3. Descarga del sistema embebido a la tarjeta SP3E-SK*

La descarga de archivos para configurar al FPGA se realiza desde una PC, que tenga instalada la suite ISE de Xilinx®, particularmente el software EDK de la suite ISE 10.1®, donde se han desarrollado todos los diseños del SE. La conexión entre la SP3E-SK y la computadora externa se realiza por medio del puerto USB, figura 6.2.



Figura 6.2. Descargando software desde una PC a la tarjeta FPGA a bordo de la MSA.

4. Alimentación de aire a la base de la MSA

Una vez verificado el nivel de batería, conexiones y descargado el software de configuración al FPGA, se debe suministrar aire a presión a la base de la MSA. Al mismo tiempo que se alimenta aire a presión, se debe verificar el nivel de balanceo de la plataforma respecto a la horizontal del piso. El balanceo se ajusta por medio de contrapesos metálicos instalados en la parte baja de la plataforma, figura 6.3, o bien, reposicionando los equipos con mayor peso de la instrumentación sobre la plataforma móvil de la MSA, principalmente el radio módem y la tarjeta FPGA.

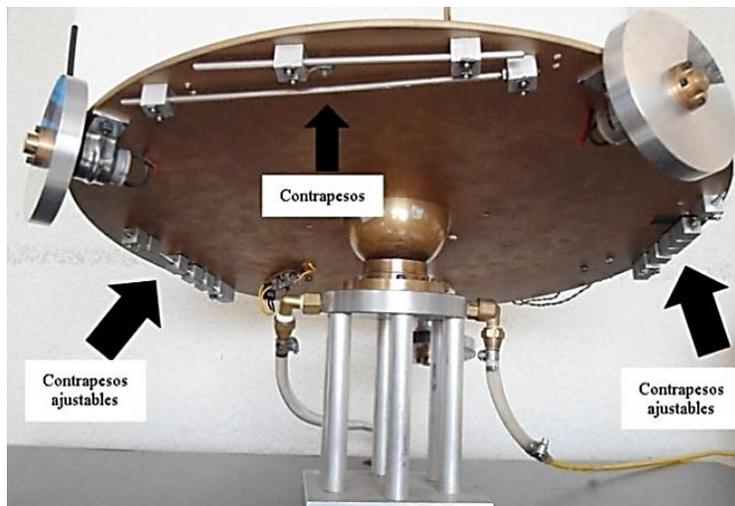


Figura 6.3. Vista de los contrapesos instalados debajo de la plataforma MSA para balanceo manual.

Una vez conseguido el balanceo horizontal de la plataforma suspendida sobre el colchón de aire, se inician las pruebas de control de orientación, considerando una posición inicial estática de la plataforma móvil.

5. Software de coprocesamiento en la PC

Una vez realizado el procedimiento de preparación en la plataforma móvil de la MSA, también se realizan algunas tareas preparativas en la PC externa. Particularmente en el software de simulación en Matlab®, en el cual residen los algoritmos que completan el esquema de procesamiento de datos de la MSA.

Adicionalmente, se debe considerar conectar un segundo radio módem al puerto serie de la PC externa, para establecer el vínculo de comunicación de datos entre los dos segmentos operativos de la MSA.

6.3. Pruebas de validación de algoritmos de control con la MSA

Como se señaló en el capítulo anterior, el esquema de control utilizado en la MSA, corresponde a una ley de control lineal basada en un controlador proporcional derivativo (PD), cuya expresión corresponde a la ecuación (6.1).

$$\tau_{control} = -k_p[3 \times 3] * \dot{\epsilon}[3 \times 1] - k_d[3 \times 3] * \ddot{\Omega}[3 \times 1] \quad (6.1)$$

Es importante recalcar que la ley de control (6.1) hace uso de elementos $(\dot{\epsilon}, \ddot{\Omega})$ calculados en los bloques TRIAD y EKF. Por tal motivo, para la descripción de la validación de la MSA se especifican las condiciones iniciales que se definen en los algoritmos de control desarrollados en Matlab®, asumiendo que durante las pruebas experimentales tales definiciones pueden modificarse para la sintonización de los algoritmos, principalmente valores de referencia en TRIAD, valores de covarianzas en EKF y valor de ganancias en el bloque CONTROL.

6.3.1. Definición de condiciones iniciales en los algoritmos del esquema de control

Se definen los siguientes valores en Matlab® para cada uno de los cuatro bloques de funciones que integran el esquema completo de control de orientación de la MSA: principal, TRIAD, EKF y CONTROL.

- Función principal

Las principales consideraciones y definiciones de valores iniciales están relacionadas principalmente con la trama de datos recibida de la MSA por medio del puerto serie (USB) de la PC y del manejo del puerto de comunicaciones para transferencia de datos.

En cuanto al manejo de la trama de datos de la MSA, que contiene datos de sensores de navegación inercial a bordo, es indispensable verificar su dimensión y formato, tanto en el FPGA como en Matlab® para evitar problemas de recepción. En este sentido, es importante considerar tanto dimensión en bits (o bytes) como orden de los datos recibidos.

Respecto al manejo del puerto serie, se deben definir los parámetros de comunicación: número de puerto, baudaje, número de bits de datos, número de bits de paro, terminador.

- Función TRIAD

En lo que corresponde a la función TRIAD, los parámetros iniciales que se definen están relacionados con la referencia de los sensores de navegación inercial, particularmente acelerómetro y magnetómetro. Esta referencia se define en términos de un arreglo vectorial [X Y Z]. En el caso del acelerómetro la condición inicial indicada es [0 0 9.81], que corresponde al valor de la aceleración actuando respecto al eje Z en el sistema de referencia definido en la MSA.

Adicionalmente, en estas mismas definiciones se considera la posición deseada para la orientación de la MSA, la cual se obtiene a partir de la ejecución de TRIAD en la posición objetivo sin la operación de las ruedas inerciales.

- Función EKF

En el caso de la función EKF, los parámetros iniciales, están relacionados con la definición de valores de covarianzas (errores) de sensores de navegación y procesos relacionados con el algoritmo para la sintonización del filtro, los valores seleccionados impactarán en la convergencia del efecto de control de la MSA, como se mostrará en apartados subsecuentes.

- Función Control

En esta función los parámetros iniciales son las ganancias correspondientes a cada uno de los controladores de la ley de control (proporcional y derivativo).

Con la selección de estos valores, se identifica experimentalmente por ejemplo, que el control derivativo apoya a frenar el giro de la plataforma luego de una perturbación inducida, externamente por el usuario o internamente por la operación del propio sistema. Mientras que el control proporcional afina con movimientos más suaves la orientación de la plataforma hacia la posición deseada.

6.4. Experimentos de validación del control de orientación de la MSA por cada eje

Para validar la instrumentación desarrollada (en torno al esquema HIL), se realizaron pruebas experimentales, tanto en la MSA como en la PC externa, para verificar el comportamiento de la plataforma, de las cuales se presentan algunos resultados.

Una de las primeras pruebas, y de las más importantes, fue la verificación del sentido de giro de cada uno de los actuadores ubicados en los tres ejes de la MSA. Para garantizar la realización de maniobras efectivas de control de orientación es importante verificar que cada una de las ruedas gire en sentido correcto, en función de la respuesta de control a los movimientos que presente la plataforma del simulador.

En vista de que las ruedas inerciales contrarrestan el movimiento que presente la plataforma, la prueba manual consistió en inclinar la plataforma aproximadamente 45° , primero respecto a X, luego respecto a Y, posteriormente respecto a Z, como se muestra en las figuras 6.4, 6.5 y 6.6. Los resultados observados experimentalmente confirmaron que la rueda inercial giraba en sentido contrario al movimiento aplicado.

Los ajustes que se realizaron en esta prueba se efectuaron tanto en hardware como en software. En hardware, revisando las conexiones de cada uno de los motores, en función de la convención de conexiones eléctricas de cada motor respecto a las terminales de la tarjeta de drivers de potencia y de las terminales físicas de salida de la tarjeta de desarrollo FPGA. En el FPGA, se verificó la coherencia en la definición de las terminales físicas establecidas en el archivo de restricciones del usuario (UCF), que vincula al sistema embebido con la plataforma del microprocesador MicroBlaze, de acuerdo con los recursos físicos disponibles en la tarjeta Spartan 3E. En la PC externa, se ajustaron los valores de los vectores de referencia definidos dentro de los procesos del bloque que contiene al algoritmo EKF, así como la definición de valores de referencia en el bloque que contiene la ley de control.

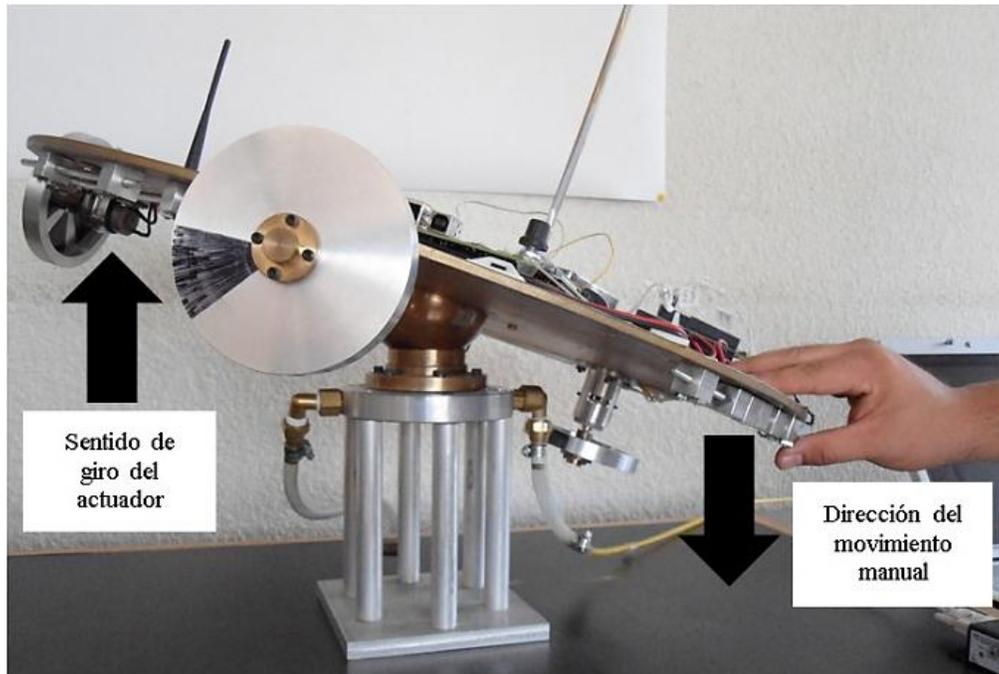


Figura 6.4. Ajuste del sentido de giro de la rueda inercial en el eje X.

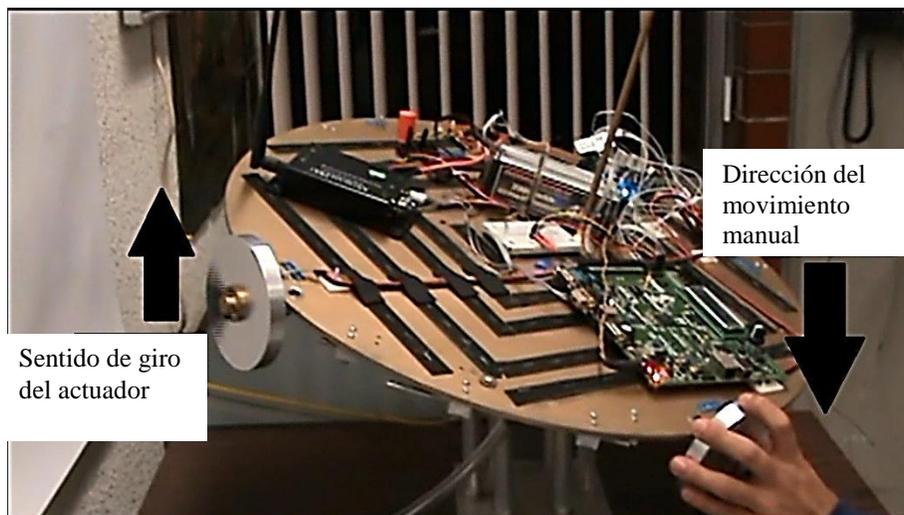


Figura 6.5. Ajuste del sentido de giro de la rueda inercial en el eje Y.

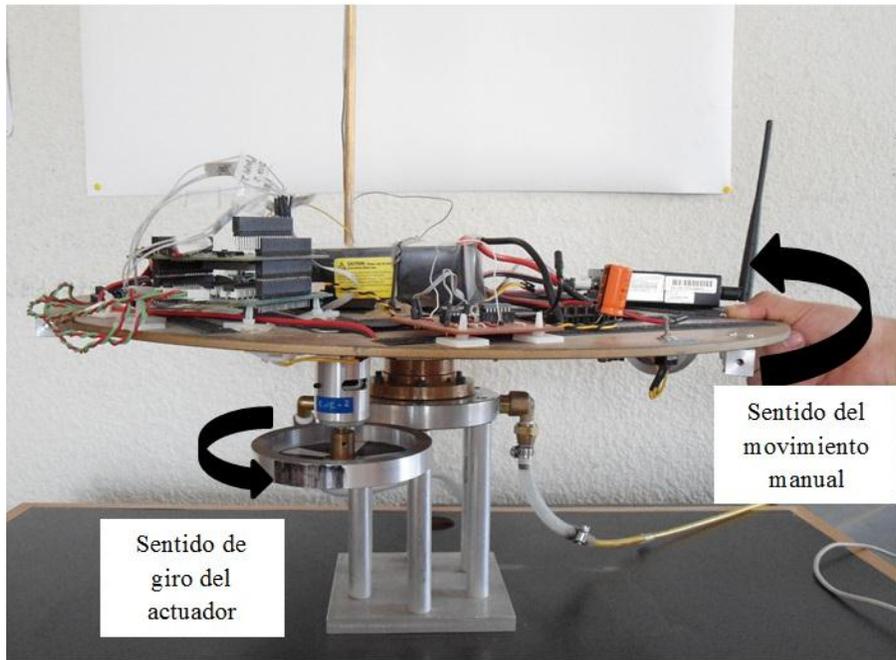


Figura 6.6. Ajuste del sentido de giro de la rueda inercial en el eje Z.

6.5. Experimentos de validación del control de orientación de la MSA en tres ejes

En la figura 6.7 se presenta el arreglo experimental de la MSA utilizado para una prueba de control de apuntamiento automatizado en tres ejes, donde la rueda inercial asociada al eje Z se toma como punto de referencia para verificar las maniobras de control de orientación de la plataforma de la MSA respecto a un punto de orientación deseado.

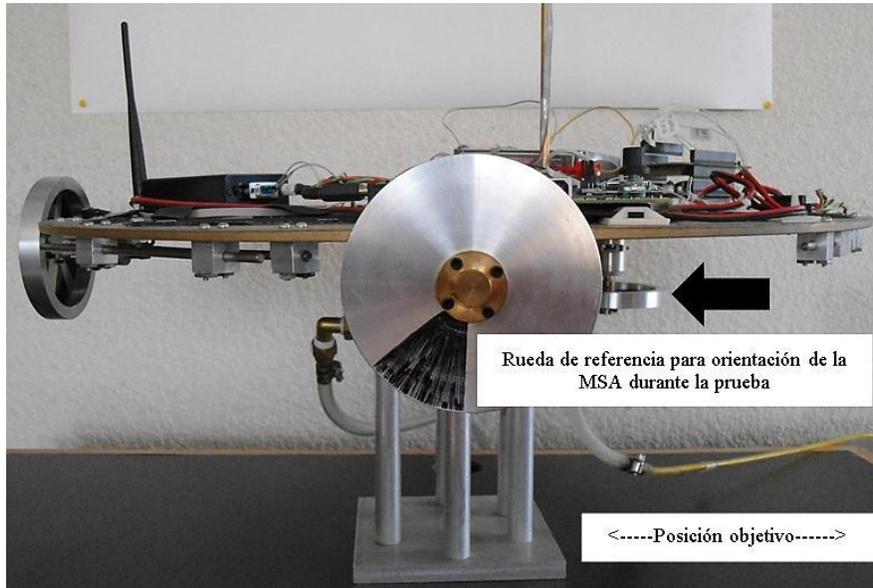


Figura 6.7. Prueba experimental de apuntamiento automatizado con la MSA.

El objetivo de la prueba fue llevar la MSA hasta una posición determinada, que para estos experimentos correspondió a la marcada en la parte inferior derecha de la figura 6.7, donde se ubicaría la rueda inercial en Z, considerando que la plataforma del simulador balanceada parte del reposo a una distancia angular de 45° .

El desempeño evaluado de las pruebas de orientación incluye condiciones que parten de una fuerte oscilación en torno al objetivo seleccionado hasta la orientación exacta en tal. En el caso de los experimentos que se describirán a continuación, el objetivo deseado está definido por la posición $-141.1, -207.9, 307.8$ (X, Y, Z), ubicado a la derecha de la posición del simulador mostrado en la figura 6.7, conforme a la medición del magnetómetro y al procesamiento en el algoritmo TRIAD.

Las gráficas se generan al final de cada prueba a partir de datos almacenados en archivos de texto. Cabe señalar que los resultados más significativos, para efectos de los objetivos de esta investigación, se observan en torno al eje Z, aún con las restricciones técnicas en la respuesta dinámica de los actuadores en los ejes X y Y con ciclos de trabajo de la señal PWM menores a 20%.

El resultado deseado para esta ronda de pruebas, se obtuvo cuando las gráficas, tanto del cuaternión de determinación como el estimado, tendieron al cero de la gráfica, lo cual indica que la plataforma de la MSA llegó al punto establecido de orientación, definido al inicio de los experimentos.

Una vez definidas tanto la referencia como el punto deseado de orientación en el algoritmo TRIAD, los ajustes se centraron en el valor de las covarianzas de procesos y sensores para sintonizar al EKF, en tanto que en el bloque CONTROL los ajustes se enfocaron en el valor de las ganancias proporcional y derivativa en los tres ejes.

- **Primera prueba de control de orientación**

En la figura 6.8 se muestran las componentes tanto del cuaternión de medición (TRIAD, línea azul) como del cuaternión estimado (EKF, línea roja).

Los valores de las ganancias utilizados para esta prueba fueron 0.33, 0.25, 0.40 y 0.2, 0.2, 0.4 aproximadamente, tanto para el controlador proporcional como para el derivativo, respectivamente. De lo anterior, se observó un efecto marcado de la parte proporcional del controlador, ya que a partir de $t=13$ s, se observa una oscilación suave en las señales del cuaternión unitario asociado al eje Z, que asumimos como referencia de apuntamiento para todas las pruebas descritas en esta sección.

Así mismo, a lo largo de la prueba, se identificó una ligera oscilación en torno a las componentes asociadas de los ejes X y Y, aunque por su magnitud, mucho menor a 1, los efectos físicos en estos dos ejes fueron prácticamente imperceptibles en los movimientos de la plataforma.

En el caso del eje Z, se observa un comportamiento inicial que tendió al cero de la gráfica (objetivo), pero que posteriormente divergió de este punto hasta ubicarse en una zona de oscilación hasta el final de la prueba. Ello se reflejó en el comportamiento de la plataforma de la MSA, que osciló fuertemente, muy alejada respecto del punto deseado de orientación.

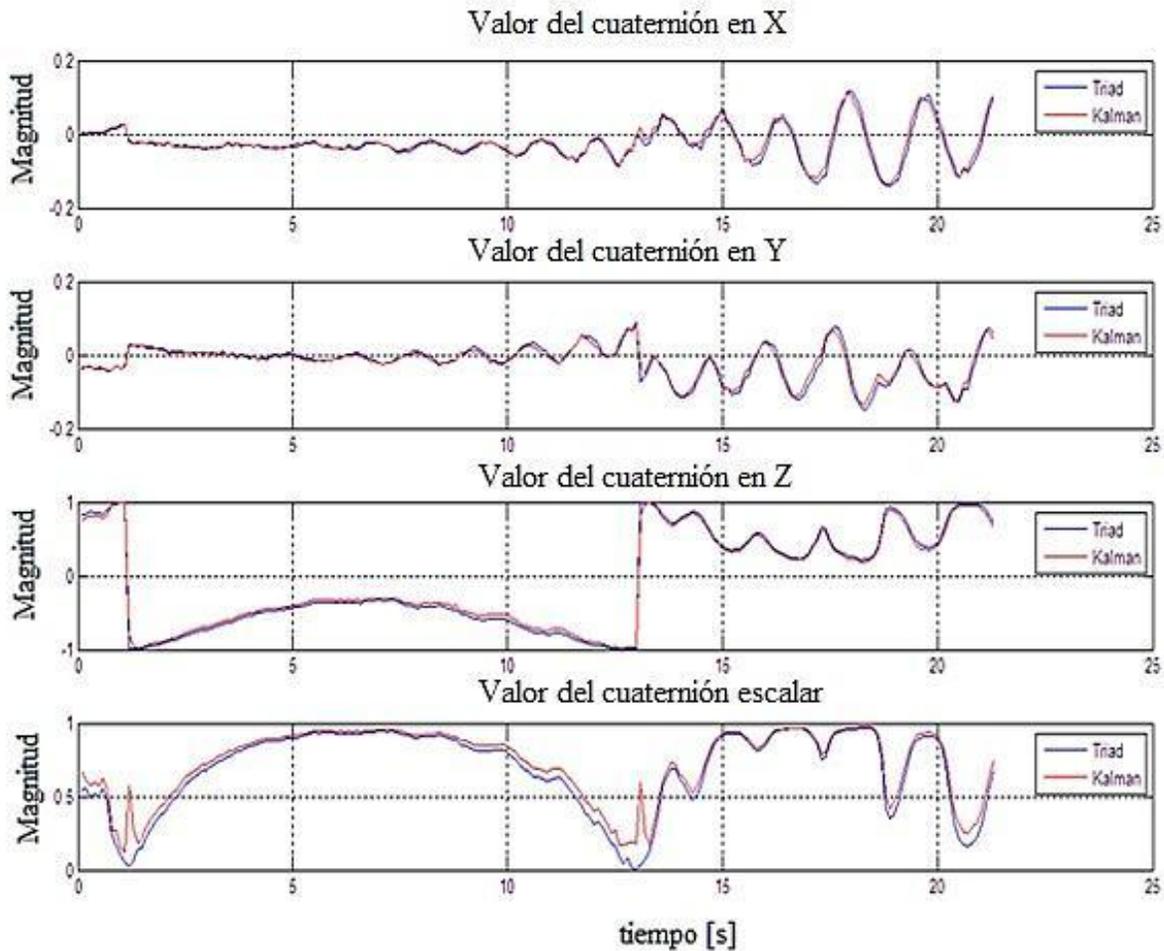


Figura 6.8. Gráficas de la primera prueba de control de orientación con la MSA.

- **Segunda prueba de control de orientación**

Se realizó una segunda prueba, figura 6.9, en la cual se realizaron ajustes en ambas partes del controlador, tanto proporcional como derivativo, definiendo para ello los siguientes valores de ganancias: 0.3, 0.2, 0.43 y 0.21, 0.23, 0.4, por cada controlador, por eje, respectivamente.

La selección del valor de las ganancias del controlador se basó en el método de aproximaciones sucesivas, sin incluir cambios en las covarianzas del EKF de la prueba anterior.

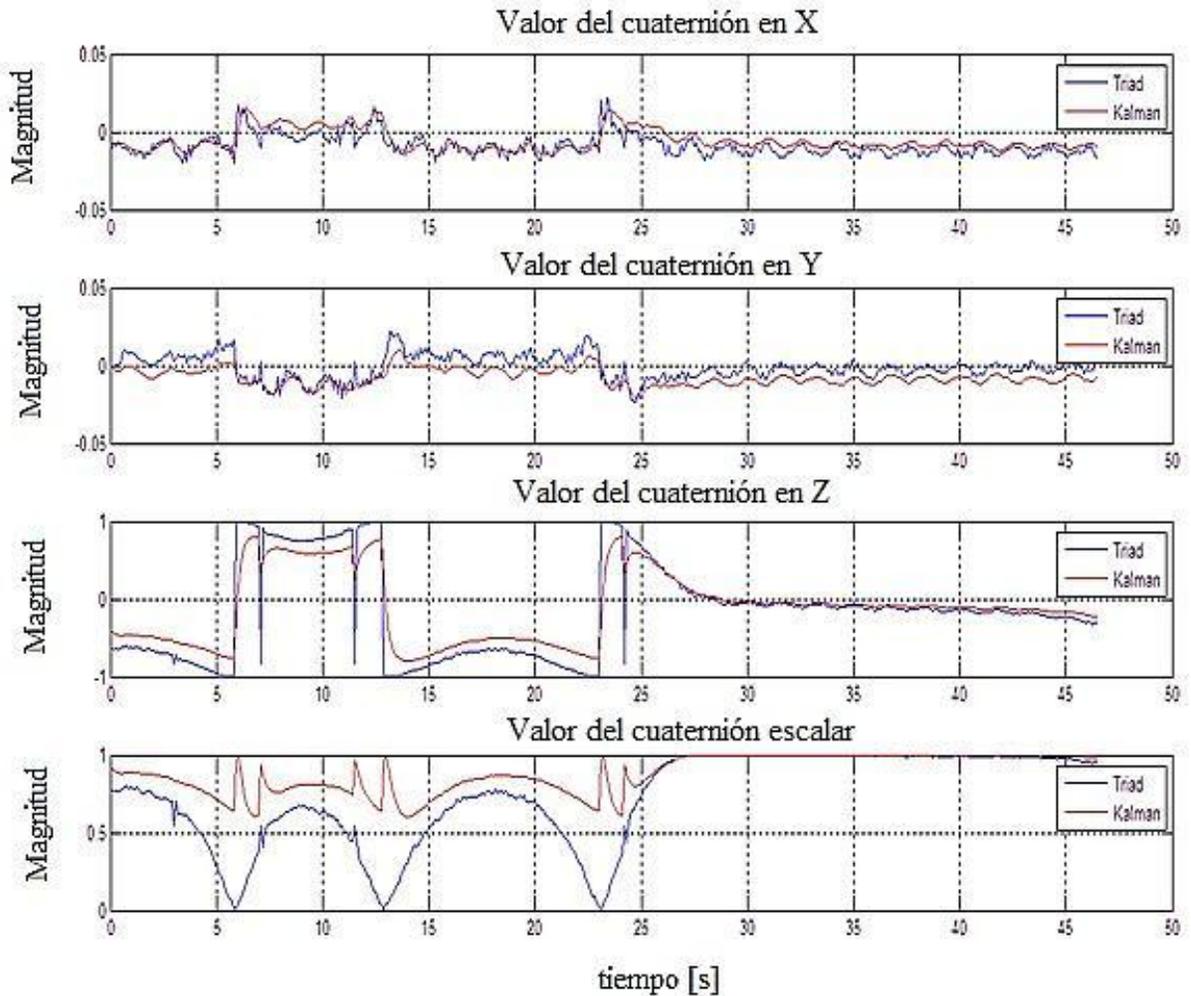


Figura 6.9. Gráficas de la segunda prueba de control de orientación con la MSA.

En la gráfica 6.9 se muestra el comportamiento de la MSA aplicando los ajustes señalados en las ganancias. De igual forma que en el caso anterior, el comportamiento en los ejes X y Y mostraron cambios mínimos, lo cual se tradujo en torques de muy baja magnitud de cada uno de los dos actuadores correspondientes. Sin embargo, en los últimos segundos de la prueba se observó un comportamiento más tendiente al cero de la gráfica, lo que se interpreta como un apuntamiento deseado al objetivo.

En el caso del componente en Z, el comportamiento en los primeros segundos de operación tiende a converger aún más en torno al punto definido de orientación, observándose un comportamiento cada vez más cercano al apuntamiento exacto al objetivo a partir del segundo 27

(aproximadamente), a partir del cual las gráficas tienden a cero, alcanzando exitosamente el punto deseado de orientación establecido para estos experimentos.

- **Tercera prueba de control de orientación**

En el último experimento, se realizaron ligeros ajustes en los valores de cada una de las ganancias de los controladores. Para esta prueba, figura 6.10, se definieron los valores de 0.28, 0.25, 0.336 y 0.15, 0.1, 0.3, siendo los tres primeros asignados al control proporcional y los tres últimos al control derivativo, respectivamente.

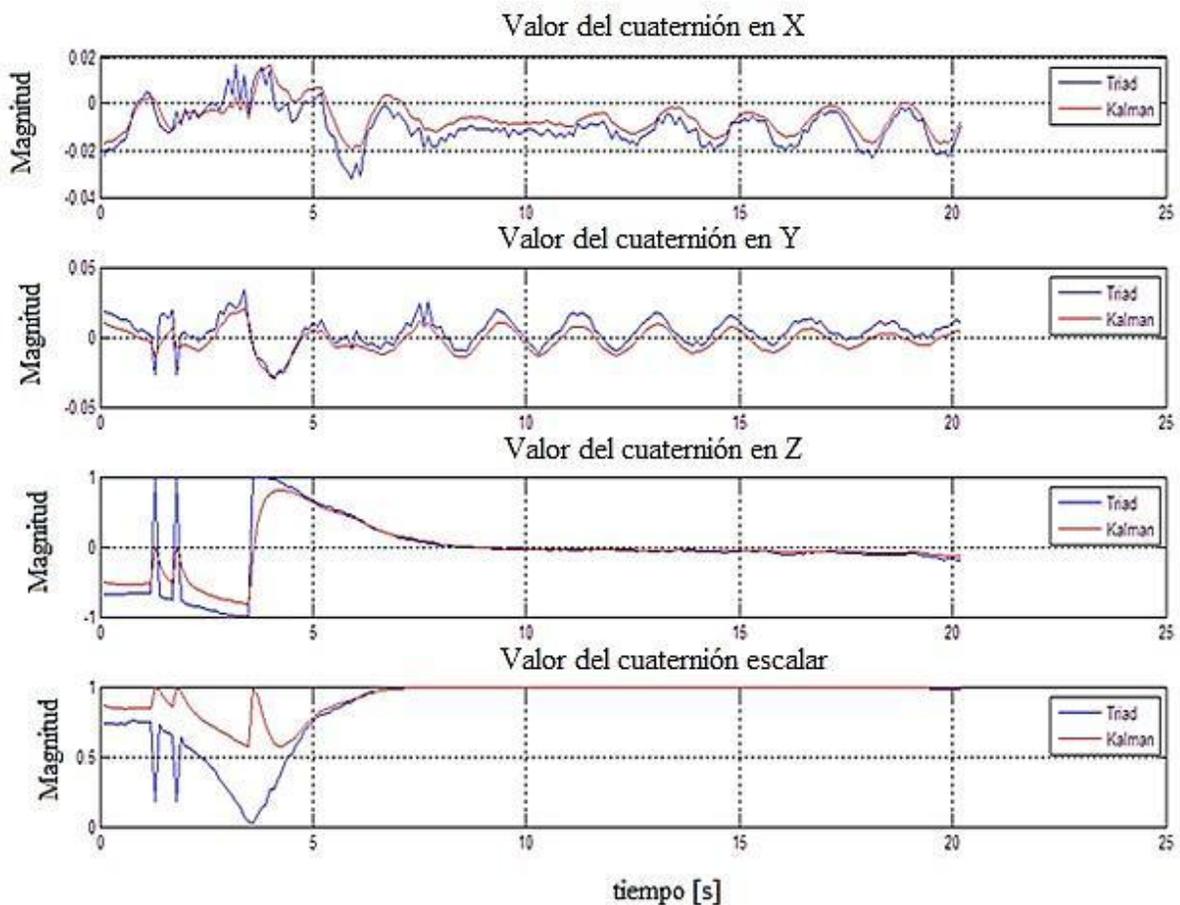


Figura 6.10. Gráficas de la tercera prueba de control de orientación con la MSA.

En la gráfica 6.10 se observa que el comportamiento de las componentes asociadas a los ejes X y Y, permanecen con una magnitud prácticamente de cero, por lo que su injerencia en las maniobras de control de orientación son despreciables.

En el caso de la componente en Z, se observa una mayor aportación en los resultados de orientación de la MSA, presentando un periodo transitorio de aproximadamente 8 segundos. Después de este tiempo, entre el segundo 10 y hasta el final de la prueba, el valor de las componentes permaneció en cero, reflejando que la plataforma de la MSA llegó al punto de orientación deseado de forma exitosa.

6.6. Discusión de resultados de los experimentos de control de orientación con la MSA

Luego de una serie de pruebas experimentales, de las cuales destacan por su progreso hacia la convergencia las descritas en las figuras 6.8, 6.9 y 6.10, particularmente en las dos últimas, se han mostrado resultados satisfactorios de las pruebas de control de orientación con la MSA, aún con restricciones técnicas señaladas en el diseño e integración del esquema de instrumentación, particularmente en lo que se refiere al desempeño de los actuadores y calidad de los sensores de navegación inercial.

Se observaron gradualmente mejoras significativas en la convergencia del controlador, permitiendo conseguir en la última prueba experimental una maniobra de control en el eje Z en aproximadamente 7 segundos, 20 segundos antes que en el caso de la segunda prueba efectuada. De esta forma se validó satisfactoriamente y dentro de nuestros estándares de éxito en la primera fase experimental de implementación de algoritmos de determinación y control de la orientación en tres ejes en laboratorio, utilizando la técnica de cosimulación HIL.

El empleo de HIL como técnica de implementación permitió, además de una rápida instrumentación e integración del simulador, una potente herramienta por medio de la cual fue posible comenzar a optimizar los algoritmos que integran el esquema de control, los cuales, si bien han sido validados previamente por medio de simulaciones numéricas, ejecutándose en una computadora personal, para la MSA fue necesario implementar ajustes y adaptaciones de código para poderlos programar en el FPGA.

Una de las principales ventajas operativas que se observaron en la MSA durante las pruebas de control de orientación, estuvo relacionada con la flexibilidad del sistema para permitir cambiar valores numéricos en los principales algoritmos del esquema, por ejemplo, valores de referencia en TRIAD, valores de covarianzas asociadas a procesos y sensores en EKF y valores de ganancias en controladores. Esto permitió observar físicamente un mejoramiento gradual en el resultado experimental dentro de una misma ronda de pruebas, particularmente en términos de la convergencia del control.

Con la flexibilidad para realizar cambios en valores numéricos importantes para cada algoritmo del esquema de control, se identificó una estrategia de sintonización de algoritmos, particularmente EKF y CONTROL, la cual consistió en definir sólo una vez los valores de las covarianzas del EKF e ir variando únicamente las ganancias de los controladores proporcional y derivativo respectivamente.

En cuanto al efecto más claro en la limitación de maniobras experimentales de control de orientación de la plataforma de la MSA, puede apreciarse la falta de movimiento en los ejes X y Y, aun cuando el esquema de control es en tres ejes y se generan los comandos de control necesarios en la PC de coprocesamiento externo.

La deficiencia de un torque de mayor magnitud en la flecha de los motores DC de las ruedas inerciales que compensen su inercia almacenada debido al giro durante los experimentos de control de orientación, y que puedan romper la inercia estática del disco al inicio de cada prueba experimental, reducen drásticamente la efectividad de los actuadores para responder ágilmente ante cambios súbitos en el sentido de giro de cada rueda inercial indicados por la ley de control. Por ello, las maniobras de control de orientación con la MSA resultan limitadas a actuar sólo en torno al eje Z en el plano XY.

Otro de los puntos de trabajo realizado durante las pruebas experimentales, estuvo relacionado con la selección de parámetros iniciales para el algoritmo EKF. El criterio de selección de los valores de covarianzas de procesos y sensores se basó en un criterio de aproximación sucesiva, partiendo de datos sugeridos por algunos fabricantes de sensores de navegación, aunque la mayoría fueron seleccionados de forma empírica. La selección de dichos valores fue muy importante pues repercutió satisfactoria o negativamente en los resultados experimentales de control de orientación de la MSA.

Por todo lo anterior, se enfatiza que al mejorar los elementos del esquema de instrumentación del simulador, se pueden obtener mejores resultados. Entre los cambios que se sugieren en la instrumentación de la MSA se encuentran los siguientes:

- Eliminar el uso de un coprocesamiento externo para reducir retrasos en el procesamiento de datos, es decir, volcar toda la arquitectura completamente al FPGA a bordo de la MSA.
- Utilizar una IMU con procesamiento a bordo que proporcione la información necesaria de determinación y estimación de orientación (cuaterniones).

- Mejorar los actuadores en su parte mecánica: motores con codificador óptico para conocer las RPM de las ruedas inerciales y la optimización mecánica de los volantes inerciales.
- Realizar ajustes en la posición del centro de masa de la plataforma del simulador mediante la adición de contrapesos u otros medios, de tal forma que permitan una distribución de masa óptima que facilite las maniobras, particularmente las de inclinación de la plataforma.

Con estas medidas, la MSA estará en condiciones de ejecutar maniobras de control de orientación aún más complejas, partiendo de una velocidad angular inicial y no necesariamente del reposo.

Capítulo 7

Conclusiones y trabajo futuro

7.1. Conclusiones

En la presente tesis se ha presentado el desarrollo e integración de la instrumentación de una plataforma de simulación para realizar en Tierra pruebas de validación de orientación satelital en tres ejes, basada en un sistema de cojinete de aire denominada MSA. La MSA en su conjunto, constituye una herramienta interactiva poderosa por medio de la cual se puede mostrar, entender y experimentar con la dinámica compleja que tiene el proceso de control de orientación de un satélite en el espacio.

De igual forma coadyuva a entender la compleja interrelación entre el software y hardware de orientación satelital en tres ejes. En términos de desarrollo tecnológico, la MSA es un sistema que permite evaluar, depurar y validar de forma práctica en Tierra, diversos esquemas y estrategias de control de orientación satelital, incluidos algoritmos y técnicas así como componentes e instrumentación asociada a sensores de navegación, plataformas de procesamiento (PC, FPGA) y actuadores.

En la primera vertiente de este trabajo de investigación, relativo al desarrollo del modelado dinámico de la MSA, permitió complementar el sistema para describir y entender analíticamente detalles de su dinámica. Resulta importante señalar que, el modelo general que se ha propuesto representa la base para la obtención de una ley de control basada en modelo, previa proyección en los tres ejes del simulador, linealización y representación en ecuaciones de estado de la ecuación de la dinámica de la MSA.

Adicionalmente, la descripción matemática de la MSA permitió identificar tres opciones de desarrollo del simulador: virtual, experimental y la combinación de ambas. Lo cual permitirá, en trabajos posteriores a éste, contar con herramientas de desarrollo y validación totalmente en software, acelerando la implementación de esquemas de control de orientación de vehículos espaciales sin la imperiosa necesidad de contar con hardware físico, representando *per se* una

poderosa herramienta portátil útil en el diseño de sistemas espaciales y con gran alcance académico para la formación de recursos humanos en cursos universitarios.

En la segunda vertiente de desarrollo, que corresponde a los experimentos de seguimiento de la MSA, resultan de amplio interés, ya que particularmente para esta investigación, representó la primer prueba de verificación de la integración de toda la instrumentación a bordo de la MSA, en particular, la validación experimental del algoritmo de determinación de la orientación (TRIAD).

La tercera vertiente del trabajo es la relativa al diseño e integración de la instrumentación a bordo de la MSA, particularmente en el uso de dispositivos de lógica reconfigurable. Dentro de la instrumentación de la MSA, se integra una computadora de a bordo basada en tecnología FPGA, la cual permite el desarrollo de arquitecturas de cómputo embebido, altamente flexibles y que facilitan la experimentación con diversas estrategias para desarrollar en hardware y software esquemas de control a partir de sistemas embebidos y arquitecturas digitales polimórficas, las cuales resultan además acordes con las tendencias actuales de miniaturización y de desarrollo de subsistemas espaciales contenidos en un solo dispositivo. Por su parte, el desarrollo de una computadora de a bordo, de diseño propio, permitió proponer un diseño y validar un primer prototipo de computadora a bordo reconfigurable al vuelo de forma remota, el cual, en fases siguientes, contendrá toda la lógica operativa de la MSA a bordo.

Cabe señalar que, en términos generales, el uso de una técnica de cosimulación como la que se utilizó en esta investigación, *hardware-in-the-loop*, permitió la implementación rápida de un esquema de control, en una arquitectura muy versátil. No obstante el trabajo que aún queda por delante, particularmente en términos de la transferencia de la lógica de procesamiento que actualmente se ejecuta en dos plataformas diferentes (FPGA y PC) a una sola a bordo de la MSA, HIL permitió el desarrollo e implementación paso a paso en hardware de cada uno de los procesos (y algoritmos asociados) que integran el esquema completo de control. Esto permitió la modularización del sistema, la reducción de la complejidad en el entendimiento de cada uno de los subprocesos que integran al esquema global y la concepción de un sistema de simulación de sistemas espaciales para laboratorio terrestre de bajo costo.

Los resultados experimentales obtenidos en los últimos meses en pruebas de control de apuntamiento utilizando la MSA, además de ser exitosos en términos de la verificación del esquema de instrumentación y la implementación experimental de algoritmos en una estructura de hardware polimórfico basado en FPGA, representan un avance muy importante en el diseño y validación experimental en Tierra de sistemas nacionales de simulación satelital. Particularmente en la

vertiente de determinación y control de orientación de vehículos espaciales, la cual es una área de alta complejidad analítica y técnica que reúne a diversos campos de la ingeniería (modelado matemático, física, mecánica, electrónica y programación) y que demanda de una plataforma de diseño y desarrollo altamente flexible.

Adicionalmente, con el desarrollo de la automatización del simulador MSA, se incursionó no sólo en el desarrollo y validación de sistemas espaciales complejos con una propuesta de solución de amplio espectro en el uso de tecnología y plataformas de desarrollo hardware/software, sino que también se identificó un área de investigación y desarrollo tecnológico nacional altamente rentable en términos de producción académica y tecnológica. Esta línea permitirá a México en el mediano-largo plazo desarrollar sistemas de estabilización y control de orientación (ADCS), que podrían ofrecerse como soluciones comerciales a nivel internacional, integrando ideas novedosas y tecnología de vanguardia.

7.2. Trabajo futuro

Actualmente y con base en los resultados de esta investigación doctoral, en el Laboratorio de Desarrollo de Sistemas Aeroespaciales del IINGEN-UNAM, se trabaja en la siguiente fase de automatización del sistema MSA en diferentes frentes de desarrollo. Particularmente en la transferencia de los algoritmos de control de orientación, ya optimizados y reducidos en complejidad, los cuales se ejecutan actualmente en MATLAB, a un sistema totalmente embebido en FPGA basado en un codiseño en hardware y software.

También se trabaja en la implementación de un nuevo esquema de control no lineal basado en una técnica de control denominada pasividad, así como en el ajuste de parámetros de configuración de los sensores de navegación inercial y el esquema de adquisición de datos en el FPGA.

En lo referente al modelado dinámico de la MSA realizado en esta investigación, se trabaja en las siguientes fases de análisis: la proyección de las ecuaciones de movimiento de los actuadores a los ejes principales, la obtención de las ecuaciones cinemáticas del sistema, considerando como primera aproximación a los ángulos de Euler, en una secuencia 3-2-1 como forma de representación de la orientación de cuerpo rígido. Una vez proyectadas las ecuaciones de movimiento de la MSA, se realizará su representación en ecuaciones de estado para diseñar una ley de control *ad hoc* a la dinámica del simulador basada en un modelo linealizado, la cual pueda ser validada por medios tanto de realidad virtual como experimentales.

Todos estos ajustes y adiciones buscan mejorar cuantitativa y cualitativamente la respuesta, el desempeño y el análisis del simulador MSA en términos de maniobras de control de orientación, para contar con una plataforma tecnológica mejorada de simulación satelital confiable y flexible, cuyos resultados puedan ser extrapolados de forma aún más directa a un próximo satélite real mexicano.

Adicionalmente, cabe señalar que gran parte del trabajo que se presenta en esta tesis doctoral, al cual han aportado varias personas con sus trabajos de tesis de licenciatura y maestría en el IINGEN-UNAM, actualmente se emplea como plataforma para la definición de estrategias para el planteamiento del esquema de control de orientación de un satélite real, que actualmente desarrolla el IINGEN-UNAM y que participa junto con otras instituciones nacionales.

Referencias

- [1] Fortescue, P. "Spacecraft Systems Engineering, third edition". Wiley, 2004. pp. 287-319.
- [2] Larson W., Wertz, J. "Space Analysis and Design, 3d Ed.". Kluwer Academic Publishers, 1999. pp.356.
- [3] Paluszek M, et.al. "Spacecraft Attitude and Orbit Control. Second Edition". Princeton Satellite Systems, 2009.
- [4] Córdova, J.R. et. al. Nonlinear attitude control for a picosatellite. Proceedings of Mexican Association of Automatic Control (AMCA), 2008.
- [5] Córdova Alarcón, J.R. "Estimación y control de orientación para el nanosatélite HUMSAT-México". Tesis de maestría, Instrumentación, Posgrado, Facultad de Ingeniería, UNAM, 2011.
- [6] Meissner, D. A three degrees of freedom test bed for nanosatellite and cubesat attitude dynamics, determination and control. Naval Postgraduate School, Monterey, California, 2009. pp. 14-15.
- [7] Corey Withcomb Crowell. "Development and Analysis of a Small satellites attitude Determination and Control System Testbed". Master Thesis. MIT, E.E.U.U. 2011.
- [8] Kim, Do-Hee. "Design and development of Hardware-in-the-loop (HIL) Simulator for Spacecraft Attitude Control using Air-Bearing System". Master Thesis. Yonsei University. Korea del sur, 2007.
- [9] Castilla Gallardo, A. "Seguimiento virtual en tiempo real de maniobras de estabilización de un simulador satelital". Tesis de licenciatura, Ingeniero mecatrónico, Facultad de Ingeniería, UNAM, 2012.
- [10] Moctezuma Eugenio, J.C., Arias-Estrada, M. "Hardware/software FPGA Architecture for Robotics Applications", J. Becker et. al. (Eds.); ARC 2009, LNCS 5453. Springer-Verlag Berlin Heidelberg, 2009. pp. 27-38.

- [11] Saldaña G., Arias-Estrada, M.. “Compact FPGA-based systolic array architecture suitable for vision systems”. Proceedings, International Conference on Information Technology (ITNG’07). IEEE, 2007.
- [12] Hosam K. F. et.al. “Review of Hardware-in-the-loop Simulation and Its Prospects in the automotive area”. Modeling and Simulation for Military Applications. Proceedings of SPIE. Vol. 6228, 62280E. 2006.
- [13] Vicente-Vivas, E. et.al. “Instrumentation of an air bearing mechatronic simulation platform and successful validation of satellite attitude control algorithms”. Journal of Engineering, Investigation and technology. Faculty of Engineering, UNAM. 2013. En refereo.
- [14] I. Bar-Itzhack, R. Harman. “Optimized TRIAD algorithm”. Journal of Guidance, Control and Dynamics, 20: 208-211, 1997.
- [15] Domínguez Méndez, P. “Procesamiento de señales de sensores de navegación para satélites sobre una plataforma FPGA”. Tesis de maestría, Telecomunicaciones, Facultad de Ingeniería, UNAM, 2011.
- [16] Marins, J.L. et.al. “An extended kalman filter for quaternion-based orientation estimation using marg sensors”. En Proceedings of the 2001 IEEE/RSJ, International Conference on Intelligent Robots and Systems, pp. 2003–2011, 2001.
- [17] Espelen Oland and Rune Schlanbusch. “Reaction wheel design for cubesats”. Dept. of Sci. Comput. & Space Technology. Narvik, University, Coll. Narvik, Norway. Recent Advances in Space Technologies, 2009. RAST’09. 4th International Conference on, pp. 778-783. 11-13 june, 2009.
- [18] Cano Garza, A.M.”Reaction wheels for Picosatellites”. Master’s thesis. SpaceMaster: Master Course in Space Science and Engineering. Lulea University of Technology. December, 2008.
- [19] Siahpush A, et.al. “A brief survey of attitude control systems for small satellites using momentum concepts”. Globesat Inc. & Honeywell Inc. Satellite System Division. AIAA Journal. Published January 01, 1988.
- [20] Sidi, M. “Spacecraft Dynamics and Control”. Cambridge University Press, 1997. pp.178-203.

- [21] Craig I. Underwood, Guy Richardson, and Jerome Savignol. "In-orbit results from the SNAP-1 nanosatellite and its future potential". Technical report, University of Surrey, Surrey, UK, November 2002.
- [22] Gavin C. McCorry. "Three-Axis Attitude Control and FalconSat-5. Technical report", United States Air Force Academy, Colorado Springs, CO.
- [23] Schwartz J.L. "The distributed Spacecraft Attitude Control System Simulator: from design concept to decentralized control". Thesis PhD, Blacksburg, Virginia, July 7, 2004.
- [24] Schwartz J.L. "Historical Review of Air-Bearing Spacecrafts Simulators". Journal of Guidance Control and Dynamics. Vol. 26, No.4, July-August, 2003.
- [25] Kim, Do-Hee. "Design and development of Hardware-in-the-loop (HIL) Simulator for Spacecraft Attitude Control using Air-Bearing System". Tesis de maestría. Yonsei University. Korea del sur, 2007.
- [26] Jae Jun Kim and Brij N. Agrawal. "Automatic Mass Balancing of Air-Bearing-Based Three-Axis Rotational Spacecraft Simulator". Technical report, Naval Postgraduate School, Monterey, CA, May 2009.
- [27] Snider, R. "Attitude Control of a satellite simulator using reaction wheels and a PID controller". Air Force Institute of Technology. Wright-Patterson Air Force, Ohio. 2010.
- [28] Curtis, Howard." Orbital Mechanics for Engineering Students". Elsevier, 2005. pp. 399-463.
- [29] Tewari, A. "Atmospheric and Space flight dynamics Modeling and Simulation with Matlab and Simulink". Editorial Birkhäuser, 2007. pp. 42-46
- [30] Riley, W.F. "Ingeniería mecánica: Dinámica". Editorial Reverté, 1990.
- [31] Halvorsen H.P. "Hardware-in-the-loop Simulation". Telemark University College. Department of Electrical Engineering, Information Technology and Cybernetics, 2009.
- [32] Vela-Díaz, D. "Estimación de actitud en aplicaciones aeroespaciales". Tesis de ingeniería electrónica. Universidad de Buenos Aires, Facultad de Ingeniería. Buenos Aires, 2010.
- [33] Notas del curso "Navegación aérea, tema 4: Sistemas de navegación autónomo. Navegación inercial. Errores". Universidad de Sevilla. Departamento de Ingeniería Aeroespacial. España, 2012.

- [34] Mendoza-Bárceñas, M., Alva Gallegos R. “Diseño de ruedas inerciales para una plataforma satelital”. Congreso de la Sociedad Mexicana de Ciencia y Tecnología Espacial SOMECYTA. Puebla, México, 2011. pp. 55-59.
- [35] Vizcaíno Torres, E. “Actualización del hardware del subsistema de control de orientación del satélite educativo SATEDU”. Tesis de licenciatura, Ingeniero eléctrico electrónico. Facultad de Ingeniería, UNAM, 2012.
- [36] Mendoza-Bárceñas, M. et.al. “Embedded Attitude Control System for the Educative Satellite SATEDU”. 22nd International Conference on Electronics, Communications and Computing CONIELECOMP, 2012.
- [37] Mendoza-Bárceñas, M. et.al. “Development and preliminary results of one-axis FPGA based Attitude Control System”. International Conference on Reconfigurable Computing and FPGAs. Conference Proceedings by IEEE Computer Society’s Conference Publishing Services (CPS). Cancún, México, 2010.
- [38] DMOS Full bridge driver L6201, L6202, L6203 datasheet. SGS Thomson Microelectronics. July, 1997.
- [39] LM2575 1-A Simple step-down switching voltage regulator datasheet. SLVS569A. January 2005.
- [40] Álvarez Ruíz de Ojeda, L.J. et.al. “Diseño de aplicaciones empotradas de 32 bits en FPGAs con Xilinx EDK 10.1 para MicroBlaze y Power PC”. Vision Libros, 2010.
- [41] Moctezuma Eugenio, J.C. “Notas del curso de Sistemas Embebidos en FPGA”. INAOE, 2010.
- [42] Romero-Troncoso, R.J. “Electrónica digital y lógica programable”. Universidad de Guanajuato, Facultad de Ingeniería Mecánica Eléctrica y Electrónica, 2007.
- [43] Hall, C. “Class notes, Chapter 4: Attitude determination”. Virginia Tech, 2008.
- [44] Osorio Jiménez, J.F. “Diseño e implementación de un sistema de control de orientación para un simulador de vuelo satelital”. Tesis, Ingeniería en Computación, Facultad de Ingeniería, UNAM, 2013.
- [45] Xilinx® Spartan 3E® FPGA Family Data Sheet. Product Specification. DS312. July 19, 2013.

- [46] Pacheco, E. et.al. "The satex project: a Mexican effort. The development of a microsatellite platform for space technologies knowledge and human resources preparation". Aerospace Conference, 2003. Proceedings. ISSN: 1095-323X, page(s): 2_667- 2_686.
- [47] Singer, F.L. "Mecánica para ingenieros: dinámica". Editorial Harla. México, 1982.
- [48] Zepeda Castillo, L.A. "Análisis de requerimientos de control de orientación satelital para comunicaciones en banda S". Tesis de licenciatura. Ingeniero en telecomunicaciones. Facultad de ingeniería, UNAM, 2011.
- [49] Griffin, M.D., French, J.R. "Space vehicle design". AIAA Education Series, 2004.
- [50] Springmann J.C. et.al. "The attitude determination system of the RAX satellite". Acta Astronautica, Elsevier, 2012. Págs. 120-135.
- [51] Pedersen, D.R. et. al. " Attitude Control System for AAUSAT-II" Technical report. Institute of Electronic Systems. Aalborg University. June, 2004.
- [52] Grillmayer, G., et.al. "FPGA based attitude control architecture for increased performance". 20th annual AIAA/ USU, Conference of small satellites. August, 2006.
- [53] UNISEC 2005, Technical report. www.unisec.jp/library/text/events_2007_e.pdf
- [54] Schmitz, H.D. "Application examples: how to use FPGA's in satellite systems". Actel Corporation.
- [55] Escobedo Lugo, L. "Simulador para pruebas de control de orientación para nanosatélites". Tesis. Ingeniero eléctrico-electrónico. Facultad de Ingeniería, UNAM. Mayo, 2012.
- [56] Prado Molina, J. et. al. "Three-axis air-bearing based platform for small satellite attitude determination and control simulation". Journal of Applied Research and Technology. Vol. 3, December, 2005.
- [57] Venkateswaran, N. et. al "Precision pointing of imaging spacecraft using gyro-based attitude reference with horizon sensor updates". Sadanha (Elsevier), Vol.29, part 2, April, 2004, p.p. 189-203.
- [58] Miyagusuku, R. et.al. "Low cost validation test-bed for small satellite's attitude determination and control". Aerospace Conference, IEEE, 2011.

- [59] Gallardo, D. Bevilacqua, R. “Six degrees of freedom experimental platform for testing autonomous satellites operations”. 8th International ESA Conference on Guidance, Navigation and Control Systems. 5-10 June, 2011.
- [60] Kato, T. et.al. “Three axis disturbance-free Attitude Control Experiment Platform: FACE”. GNC Systems, Institute of Space Systems, German Aerospace Center (DLR), Bremen, Germany.
- [61] Roemer, S. et.al. “Experiences with the BIRD ACS test facility and the resulting design of a state of the art mini- and micro-satellite ACS test facility”. Astro- und Feinwerktechnik Adlershof GmbH, DLR-OS, Small satellites for earth observation – digest of the 6th International Symposium of the Academy of Astronautics, Berlin. Pp. 165-168. April 23-26, 2007.
- [62] Mekky, T “A comparative study of spacecraft attitude determination and estimation algorithms (a cost-benefit approach)”. Aerospace Science and Technology 26 (2013), 211-215. Elsevier
- [63] Vázquez Valenzuela, R. “Vehículos espaciales y misiles. Tema 4: Control y determinación de la actitud. Parte 1: estimación de la actitud”. Departamento de Ingeniería Aeroespacial, Escuela Superior de Ingenieros, Universidad de Sevilla, España. 14 de mayo de 2012.
- [64] Ismail, Z, Varatharajoo, R. “A study of reaction Wheel configurations for a 3-axis satellite attitude control”. Advances in Space Research, COSPAR. Elsevier. 45, 2010, pp. 750-759.
- [65] Chen, Y.H. Z.C. Lin, C.H. “Aerodynamic and gravity gradient stabilization for microsattellites”. Acta astronautica, 46 (7), pp. 491-499. 2000.
- [66] Wisniewski, R., Kulczycki, P. “Slew maneuver control for spacecraft equipped with star camera and reaction wheels”. Control Engineering Practice Journal. 13, pp. 349-356. 2005.
- [67] Wilson, R.W., Peck, M. “An air-levitated testbed for flux pinning interactions at the nanosatellite Scale”. AIAA Guidance, Navigation and Control Conference, 2-5 August, 2010. Toronto, Ontario, Canada.
- [68] Zebeay, M., et.al. “Analytical and experimental stability investigation of a hardware-in-the-loop satellite docking simulator”. Acta Astronautica. 00, pp. 1-16. 2013.

Apéndice A

Software de operaciones del FPGA

Programa principal

```

//*****
//Firmware para esquema HIL en FPGA Spartan 3E-XC3s500e
//*****
#include "xparameters.h"
#include "xutil.h"
#include "stdio.h"
#include "iiclib.h"
#include "xintc.h"
#include "my_pwmhil.h"
#include "xbasic_types.h"
#include "xstatus.h"
#include "xutil.h"
#include "xuartlite.h"
#include "my_pwmhil2.h"
#include "my_pwmhil3.h"
#include "xtmrctr.h"

#include "stdio.h"
#include "math.h"

#define REG0 0
#define REG1 0
#define REG2 0

#define PWM_ADD          XPAR_MY_PWMHIL_0_BASEADDR
#define PWM2_ADD         XPAR_MY_PWMHIL2_0_BASEADDR
#define PWM3_ADD         XPAR_MY_PWMHIL3_0_BASEADDR

#define DCE_ADD          XPAR_RS232_DCE_BASEADDR
#define IIC_BASEADDR    XPAR_IIC_0_BASEADDR
#define INTC_BASEADDR   XPAR_XPS_INTC_0_BASEADDR
#define ACC_ADD         0x53
#define MAG_ADD         0x1E
#define GYR_ADD         0x68
#define ACC_MES_REG     0x32
#define MAG_MES_REG     0x03
#define GYR_MES_REG     0x1D
//*****
//Definición de variables globales
//*****
XIic IicInstance;
XIntc intController;
```

```

//*****
//Definición de funciones
//*****
ACC_Enable_Measure()
{
    writeI2C(ACC_ADD,0x2D,1<<3);
}

Measure(u8 DISP_ADD, short *Measures)
{
    u8 Data_Read[6];
    u8 MEASURE_REG;

    if(DISP_ADD == ACC_ADD)
        MEASURE_REG = ACC_MES_REG;
    else if (DISP_ADD == MAG_ADD)
        MEASURE_REG = MAG_MES_REG;
    else
        MEASURE_REG = GYR_MES_REG;

    readI2C(DISP_ADD,MEASURE_REG,Data_Read,6);

    if(DISP_ADD == ACC_ADD)
    {
        *Measures++ = ((short)Data_Read[0])|((short)Data_Read[1])<<8;
        *Measures++ = ((short)Data_Read[2])|((short)Data_Read[3])<<8;
        *Measures  = ((short)Data_Read[4])|((short)Data_Read[5])<<8;
    }
    else
    {
        *Measures++ = ((short)Data_Read[1])|((short)Data_Read[0])<<8;
        *Measures++ = ((short)Data_Read[3])|((short)Data_Read[2])<<8;
        *Measures  = ((short)Data_Read[5])|((short)Data_Read[4])<<8;
    }
}
//*****
//Manejo de la interrupción
//*****
fit_int_handler()
{
    int i;
    short Measures[3];
    int s1, s2, s3, DC1, DC2, DC3;
    int data=0,data2=0,data3=0;
    u8 cnt;

    Measure(ACC_ADD,Measures);
    for(i=0;i<3;i++)
        xil_printf("%d ",Measures[i]);

    writeI2C(MAG_ADD,0x02,0x01);
    Measure(MAG_ADD,Measures);
    xil_printf("%d ",Measures[0]);
    xil_printf("-%d ",Measures[1]);
    xil_printf("%d ",Measures[2]);
}

```

```

        Measure(GYR_ADD,Measures);
        for(i=0;i<3;i++)
            xil_printf("%d ",Measures[i]);
            xil_printf("%d\n", cnt + 1);

// ***** Escritura en registros para los ciclos de trabajo en los PWM *****

/*Se escribe el valor del "bit de signo" y "DC" a un registro interno previamente asignado
en el CORE PWM a cuyos registros podrá acceder MicroBlaze para procesar los datos */

/*MY_PWMHIL_mWriteSlaveReg0(PWM_ADD,REG0,0);Reset del CORE PWM esta inhabilitado. Este
Reset
se puede activar enviando un "1" al Reg0 del periférico PWM.*/

data = XUartLite_RecvByte(DCE_ADD);
if(data == '@'){

    s1 = XUartLite_RecvByte(DCE_ADD);
    MY_PWMHIL_mWriteSlaveReg1(PWM_ADD,REG0,s1);
    DC1 = XUartLite_RecvByte(DCE_ADD);
    MY_PWMHIL_mWriteSlaveReg2(PWM_ADD,REG0,DC1);

    s2 = XUartLite_RecvByte(DCE_ADD);
    MY_PWMHIL2_mWriteSlaveReg1(PWM2_ADD,REG0,s2);
    DC2 = XUartLite_RecvByte(DCE_ADD);
    MY_PWMHIL2_mWriteSlaveReg2(PWM2_ADD,REG0,DC2);

    s3 = XUartLite_RecvByte(DCE_ADD);
    MY_PWMHIL3_mWriteSlaveReg1(PWM3_ADD,REG0,s3);
    DC3 = XUartLite_RecvByte(DCE_ADD);
    MY_PWMHIL3_mWriteSlaveReg2(PWM3_ADD,REG0,DC3);

    }
else    cnt = cnt + 1;

}

//*****
//Esta es la función principal
//*****
int main(void)
{
    int Status;
    u8 status[1];

    Status = XIic_Initialize(&IicInstance,IIC_BASEADDR);
    XIic_Start(&IicInstance);

    ACC_Enable_Measure();

    writeI2C(GYR_ADD,0x16,0x1F);

    XIntc_Initialize(&intController,XPAR_INTC_0_DEVICE_ID);
    XIntc_Connect(&intController,XPAR_XPS_INTC_0_FIT_TIMER_0_INTERRUPT_INTR,
(XInterruptHandler)fit_int_handler,(void*)0);

```

```

XIntc_Start(&intController,XIN_REAL_MODE);
XIntc_Enable(&intController,XPAR_XPS_INTC_0_FIT_TIMER_0_INTERRUPT_INTR );
microblaze_enable_interrupts();
    while(1)
return 0;
}

```

Núcleo personalizado de PWM

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity pwm is
    Port ( data : in  STD_LOGIC_VECTOR (7 downto 0);
          clk : in  STD_LOGIC;
          reset : in  STD_LOGIC;
          dir_cmd : in  STD_LOGIC;
          pwm_out : out  STD_LOGIC;
          chg_dir_pwm1 : out  STD_LOGIC;
          chg_dir_pwm2 : out  STD_LOGIC);
end pwm;

architecture Behavioral of pwm is

--
*****
*****

    signal ps : std_logic;
    signal period_cnt :std_logic_vector(7 downto 0);
    signal duty_cycle :std_logic_vector(7 downto 0);
    signal pwm_flag : std_logic;
    signal chg_lvl_pwm : std_logic;
    signal chg_dir_pwm_signal : std_logic;

begin

--
*****
*****

    prescalar : process (clk)
        variable pscount_v : unsigned (7 downto 0);

begin

if rising_edge (clk) then
    if reset = '1' then
        pscount_v := "10000100";

```

```

        ps_v := '0';
    else
        case pscount_v is
            when "00000001" =>
                pscount_v := "10000100";
                ps_v := '1';
            when others =>
                pscount_v := pscount_v -1;
                ps_v := '0';
            end case;
        end if;

        ps <= ps_v;

    end if;
end process;

period_cntr: process(clk)
    variable count_v : unsigned(7 downto 0);

begin

if rising_edge(clk) then
    if reset = '1' then
        count_v := (others => '0');
    elsif ps = '1' then
        count_v := count_v + 1;
    end if;

        period_cnt <= std_logic_vector(count_v); --- se da la forma al vector count_v para asignarala
a period_cnt y poder compararlo con duty_cycle

    end if;
end process;

duty : process (clk)

begin
if rising_edge(clk) then
    if reset = '1' then
        duty_cycle <= (others => '0');
    else --- solo dejaremos una variable para que tenga un seguro de acceso o funcion del
comparador.
        duty_cycle <= data;
    end if;
end if;
end process;

--
*****
*****

```

```
comparator : process (clk)
```

```
begin
```

```
  if rising_edge(clk) then
    if reset = '1' then
      pwm_out <= '0';
      pwm_flag <= '0';
    elsif period_cnt < duty_cycle then
      pwm_out <= '1';
      pwm_flag <= '1';
    else
      pwm_out <= '0';
      pwm_flag <= '0';
    end if;
  end if;
```

```
  end if;
```

```
end process;
```

```
__*****
change_level_pwm : process (clk)
```

```
variable flag1_m1 : std_logic;
```

```
begin
```

```
  if rising_edge(clk) then
    if reset = '1' then

      chg_lvl_pwm <= '0';
      flag1_m1 := '0';

    else

      if pwm_flag='0' and flag1_m1='1' then
        chg_lvl_pwm <= '1';
        flag1_m1:= '0';

      elsif pwm_flag='1' and flag1_m1='0' then
        chg_lvl_pwm <= '1';
        flag1_m1:= '1';

      else
        chg_lvl_pwm <= '0';
        flag1_m1:=pwm_flag;
      end if;
    end if;
  end if;
```

```
end process;
```

```
__*****
change_dir_cmd : process (clk)
variable flag2_m1 : std_logic;
```

```

begin
  if rising_edge(clk) then
    if reset = '1' then

      chg_dir_pwm_signal <= '0';
      flag2_m1 := '0';

    else

      if dir_cmd='0' and flag2_m1='1' then

        if chg_lvl_pwm = '1' then
          chg_dir_pwm_signal <= '1';
          flag2_m1:= '0';
        else
          chg_dir_pwm_signal <= '0';
          flag2_m1:= '1';
        end if;

      elsif dir_cmd='1' and flag2_m1='0' then

        if chg_lvl_pwm = '1' then
          chg_dir_pwm_signal <= '1';
          flag2_m1:= '1';
        else
          chg_dir_pwm_signal <= '0';
          flag2_m1:= '0';
        end if;

      else
        chg_dir_pwm_signal <= '0';
        flag2_m1:=dir_cmd;
      end if;

    end if;
  end if;
end process;

```

```

_*****

```

```

change_dir_pwm : process (clk)
variable flg1 : std_logic;
variable flg2 : std_logic;
variable flg1_temp : std_logic;
variable flg2_temp : std_logic;
variable flg_dir1 : std_logic;
variable count_td : unsigned(6 downto 0);
begin
-- chg_lvl_pwm

```

```

-- chg_dir_pwm_signal

if rising_edge(clk) then
    flg1:=chg_lvl_pwm;
    flg2:=chg_dir_pwm_signal;
    if flg1='1' then
        flg1_temp:='1';
    end if;

    if flg2='1' then
        flg2_temp:='1';
    end if;

    if reset = '1' then

        flg_dir1:='0';
        count_td:="1010000";

    else

        if flg1_temp='1' and flg2_temp='1' then
            case count_td is
                when "0000001" =>
                    count_td := "1010000";

                    if dir_cmd='0' then
                        flg_dir1:='0';
                    else
                        flg_dir1:='1';
                    end if;

                    flg1_temp:='0';
                    flg2_temp:='0';

                    when others =>
                        count_td := count_td -1;
                        flg1_temp:='1';
                        flg2_temp:='1';
                    end case;

            end if;

        end if;

        chg_dir_pwm1<= flg_dir1;
        chg_dir_pwm2<= not(flg_dir1);

    end if;
end process;

end Behavioral;

```

Apéndice B

Software de operaciones en MATLAB

Programa principal

```
% % %creacion de archivos para datos del triad modificado y filtro kalman
datostx = fopen('salidaTy.txt','w+');
datosty = fopen('salidaTp.txt','w+');
datostz = fopen('salidaTr.txt','w+');
datoste = fopen('salidaTe.txt','w+');
orientacion = fopen('orienta.txt','w+');
datoskx = fopen('salidaKy.txt','w+');
datosky = fopen('salidaKp.txt','w+');
datoskz = fopen('salidaKr.txt','w+');
datoske = fopen('salidaKe.txt','w+');
% %
% % %creacion de archivos para valores de los sensores
magnetometro = fopen('magnetometro.txt','w+');
acelerometro = fopen('acelerometro.txt','w+');
giroscopio = fopen('giroscopio.txt','w+');
%
% %creacion de archivos para torque (t tau)
torque1 = fopen('torque1.txt','w+');
torque2 = fopen('torque2.txt','w+');
torque3 = fopen('torque3.txt','w+');
%
%
% % %creacion de archivos auxiliares, conversión de muestras a variable de
% % %tiempo y tasa de erros de comunicaciones
ejex = fopen('x.txt','w+');
tasaEMS = fopen('errorms.txt', 'w+');
senseG = fopen('senseG.txt', 'w+');
wangular = fopen('velAngular.txt','w+');
cuaterang = fopen('kalmancomp.txt','w+');
% % %eliminar las entradas de puerto serial abiertas y borrarlas
d = instrfind;
delete(d);

% configuración del puerto serial con lectura continua asíncrona
s = serial('COM4');
set(s,'BaudRate',19200);
set(s,'TimeOut',0.5);
set(s,'Terminator','LF');
set(s,'InputBufferSize',480012);
set(s,'RequestToSend','on');
set(s,'ByteOrder','bigEndian');
set(s,'ReadAsyncMode','continuous');
```

```
% apertura del puerto serial
```

```

fopen(s);
%limpieza del buffer de entrada
flushinput(s);
%Ns = 200;
errorRecep = 0;
cnt = 0;
q_km1 = [0 0 0 1];
xksave = [0;0;0;1;0;0;0];
Pkvar = .7;
Pksave = [Pkvar 0 0 0 0 0 0;
          0 Pkvar 0 0 0 0 0;
          0 0 Pkvar 0 0 0 0;
          0 0 0 Pkvar 0 0 0;
          0 0 0 0 Pkvar 0 0;
          0 0 0 0 0 Pkvar 0;
          0 0 0 0 0 0 Pkvar];
%ciclo para la adquisición de datos
arranca = query(s, '*IDN?');
inicio = str2num(arranca);

% while(counter ~= Ns)
while(1)
    tic;
    angulo = fgetl(s);
    %tamaño = length(angulo)
    t = str2num(angulo);
    tam = length(t);
    if tam<10
        t = inicio;
        errorRecep = errorRecep + 1;
    end

    % distribución de datos
    acel = t(1:3);
    mag = t(4:6);
    gyro = t(7:9);
    errorTrans = t(10:10);
    fprintf(tasaEMS, '%f\t %f\n', errorRecep,errorTrans);

    %fprintf(tasaEMS, '%f\t %f\n', errorRecep,errorTrans);

    %impresion de datos de los sensores
    fprintf(accelerometro, '%f\t %f\t %f\n',acel);
    fprintf(magnetometro, '%f\t %f\t %f\n',mag);
    fprintf(giroscopio, '%f\t %f\t %f\n',gyro);

% aplicación del algoritmo de kalman y triad
[tr kal Pksave xksave, q_km1] = Kalman(mag,acel,gyro, Pksave, xksave, q_km1);
    triad_x = tr(1,1);
    triad_y = tr(1,2);
    triad_z = tr(1,3);
    triad_escalar = tr(1,4);
    ekf_x = kal(1,1);
    ekf_y = kal(1,2);
    ekf_z = kal(1,3);
    ekf_escalar = kal(1,4);

```

```

vel_angular = kal(5:7);
%impresión de datos del kalman y triad
fprintf(datostx, '%f\n', triad_x);
fprintf(datosty, '%f\n', triad_y);
fprintf(datostz, '%f\n', triad_z);
fprintf(datoste, '%f\n', triad_escalar);
fprintf(orientacion, '%f\t%f\t%f\t%f\n', tr);
%comentariada
fprintf(datoskx, '%f\n', ekf_x);
fprintf(datosky, '%f\n', ekf_y);
fprintf(datoskz, '%f\n', ekf_z);
fprintf(datoske, '%f\n', ekf_escalar);
fprintf(wangular, '%f\t%f\t%f\n', vel_angular);

% *****
% aplicación del algoritmo de control

[torque DCr1 s1 DCr2 s2 DCr3 s3] = Control(kal);
tau = torque(1:3);
dc1 = DCr1;
sen1 = s1;
dc2 = DCr2;
sen2 = s2;
dc3 = DCr3;
sen3 = s3;

fprintf(torque1, '%f\n', torque(1,1));
fprintf(torque2, '%f\n', torque(2,1));
fprintf(torque3, '%f\n', torque(3,1));
fprintf(senseG, '%f\t%f\t%f\n', sen1, sen2, sen3);
fprintf(cuaterang, '%f\t%f\t%f\t%f\t%f\t%f\t%f\t%f\n', kal);

cnt = cnt + 1;
x = cnt * 0.1;
fprintf(ejex, '%f\n', x);

%escritura de la salida del bloque de control hacia los registros de la FPGA
% si cambias la arroba ya no escribirá en las ruedas.
fwrite(s, '@'); %fwrite(s, '@');
fwrite(s, sen1);
fwrite(s, dc1);
fwrite(s, sen2);
fwrite(s, dc2);
fwrite(s, sen3);
fwrite(s, dc3);

%w(cnt) = toc;
toc;
end

%cierre del puerto serial
%fin de la rutina

fclose(s);

```

%impresión de gráficas y resultados del muestreo.

```
p1 = load('magnetometro.txt');
p2 = load('acelerometro.txt');
p3 = load('giroscopio.txt');

pgx=p3(:,1)*pi/180/14.375;
pgy=p3(:,2)*pi/180/14.375;
pgz=p3(:,3)*pi/180/14.375;

xT = load('salidaTy.txt');
yT = load('salidaTp.txt');
zT = load('salidaTr.txt');
escalarT = load('salidaTe.txt');

xK = load('salidaKy.txt');
yK = load('salidaKp.txt');
zK = load('salidaKr.txt');
escalarK = load('salidaKe.txt');

tX = load('torque1.txt');
tY = load('torque2.txt');
tZ = load('torque3.txt');

ejeX = load('x.txt');
tasaE = load('errorrms.txt');
sentidoG = load('senseG.txt');
vel_ang = load('velAngular.txt');

figure(1)
subplot(411)
plot(ejeX,xT,'b'); hold
plot(ejeX,xK,'r');hold;grid
title('Valor del cuaternion en x')
legend('Triad','Kalman');
subplot(412)
plot(ejeX,yT,'b'); hold
plot(ejeX,yK,'r');hold;grid
title('Valor del cuaternion en y')
legend('Triad','Kalman');
subplot(413)
plot(ejeX,zT,'b'); hold
plot(ejeX,zK,'r');hold;grid
title('Valor del cuaternion en z')
legend('Triad','Kalman');
subplot(414)
plot(ejeX,escalarT,'b'); hold
plot(ejeX,escalarK,'r');hold;grid
title('Valor del cuaternion en escalar')
legend('Triad','Kalman');

figure(2)
subplot(311)
plot(ejeX,tX,'b');grid
title('Torque en X')
```

```

legend('torque (1,1)')
subplot(312)
plot(ejeX,tY,'r');grid;
title('Torque en Y')
legend('torque (1,2)');
subplot(313)
plot(ejeX,tZ,'g');grid;
title('Torque en Z')
legend('torque (1,3)')

```

```

figure(3)
subplot(311)
plot(ejeX,p1);hold;grid;
title('Valores del Magnetómetro')
legend('x en azul', 'y en verde', 'z en rojo')
subplot(312)
plot(ejeX,p2);hold;grid;
title('Valores del Acelerómetro')
legend('x en azul', 'y en verde', 'z en rojo')
subplot(313)
plot(ejeX,tasaE);hold;grid;
title('Tasa de Errores en la Recepción (matlab) y la Transmisión (fpga)')
legend('recepción en azul', 'transmisión en verde')

```

```

figure(4)
subplot(311)
plot(ejeX,pgx);hold;
plot(ejeX,vel_ang(:,1),'r');hold;grid;
title('Valores del Giroscopio vs Velocidad Angular');
legend('G en azul', 'VA en verde');
subplot(312)
plot(ejeX,pgy);hold;
plot(ejeX,vel_ang(:,2),'r');hold;grid;
title('Valores del Giroscopio vs Velocidad Angular');
legend('G en azul', 'VA en verde');
subplot(313)
plot(ejeX,pgz);hold;
plot(ejeX,vel_ang(:,3),'r');hold;grid;
title('Valores del Giroscopio vs Velocidad Angular');
legend('G en azul', 'VA en verde');

```

Función TRIAD

```

% TRIAD y cuaternion a la salida
%function [ang y] = Triad(mag, acel)
function y = Triad(mag, acel)

```

```

%Bloque de referencia para sensores de navegación
ref_a = [0 0 -9.81]';
%ref_m_prev = [-138 -215 303]';
%ref_m_prev = [-184 194 288]'; apuntamiento horizontal izquierda
%ref_m_prev = [136 -230 306]'; %apuntamiento al centro inclinacion 5°

```

```

%se definirán tres posiciones para pruebas finales
%posición 1)
%ref_m_prev = [-141.1 -207.9 307.8]'; %<<-bueno

%posición 2)apuntamiento derecha
%ref_m_prev = [234.5 -58.5 297.9]';
%posición 3)apuntamiento izquierda
%ref_m_prev = [-108.7 136 316.6]';
%posición 4)
%ref_m_prev = [145.4 143.7 300.8]';
%
ref_m = ref_m_prev/norm(ref_m_prev);
%*****
%inclinaciones muy pequeñas
%*****

s1 = ref_a/norm(ref_a);
s2 = cross(s1,ref_m)/norm(cross(s1,ref_m));
s3 = cross(s1,s2);

r1 = acel'/norm(acel);
r2 = cross(r1,mag)'/ norm(cross(r1,mag));
r3 = cross(r1,r2);

%Arot = s1*r1'+s2*r2'+s3*r3';

Arot = r1*s1'+r2*s2'+r3*s3';

%A1= S*R'

% de DCM a cuaternion alternativo a través de funciones trascendentes
%el orden de las operaciones debe cambiar:

diag = (trace(Arot) - 1)/2;
phi = acos(diag);

e3 = (Arot(1,2) - Arot(2,1))/(2*sin(phi));
e1 = (Arot(2,3) - Arot(3,2))/(2*sin(phi));
e2 = (Arot(3,1) - Arot(1,3))/(2*sin(phi));

y(4) = cos(phi/2);
y(1) = e1*sin(phi/2);
y(2) = e2*sin(phi/2);
y(3) = e3*sin(phi/2);

```

Función EKF

```

function [tr kal Pksave xksave q_km1] = Kalman(mag,acel,Gyr,Pksave, xksave,q_km1)

% FILTRADO EKF discreto
% Ajustes de unidades
xm = mag(1)/norm(mag);
ym = mag(2)/norm(mag);

```

```

zm = mag(3)/norm(mag);

xg = Gyr(1)*pi/(180*14.375);
yg = Gyr(2)*pi/(180*14.375);
zg = Gyr(3)*pi/(180*14.375);
Gyro = [xg; yg; zg];

%initial value of the derivative of the quaternion
%qkdm1 = [1;0;0;0];
qkdm1 = [0;0;0;1];

% % covarianzas
%estas son las ganancias buenas
cov_gyro = 0.3;
cov_qF1 = 0.3;
cov_process2 = 0.3;
cov_process1 = 0.35;

% cov_gyro = 0.70;
% cov_process2 = 0.83;
% cov_qF1 = 0.8; %m
% cov_process1 = 0.1; %m

% Tiempo de muestreo
%T_default = 0.1;
T_default = 0.2;

% inicializacion de varibales
nq=sqrt(xksave(1)^2+xksave(2)^2+xksave(3)^2+xksave(4)^2);
q1_km1=xksave(1)/nq;
q2_km1=xksave(2)/nq;
q3_km1=xksave(3)/nq;
q4_km1=xksave(4)/nq;
Om1_km1=xksave(5);
Om2_km1=xksave(6);
Om3_km1=xksave(7);

q_km1=[q1_km1;q2_km1;q3_km1;q4_km1];
Omkm1=[Om1_km1;Om2_km1;Om3_km1];

%%%Matrix H%%%
Hk1= [1 0 0 0 0 0;
      0 1 0 0 0 0;
      0 0 1 0 0 0;
      0 0 0 1 0 0;
      0 0 0 0 1 0;
      0 0 0 0 0 1 0];

%%% Matriz de Covarianza de la medicion %%%%
Rk1=diag([cov_gyro,cov_gyro,cov_gyro,cov_qF1,cov_qF1,cov_qF1]);

%%%%%%%%%
%%% GANANCIA DE KALMAN %%%%
invHP=inv(Hk1*Pksave*Hk1'+Rk1);
Xaux1a=Hk1'/invHP;

```

```

Kgain2a=Pksave*Xaux1a;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% INNOVACION DE LA ESTIMACION %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%obtención de la orientación por TRIAD
y = Triad(mag,acel)';
tr = y';

%%% Proceso de innovacion
e2 = y(1:3) - q_km1(1:3);
e1 = Gyro - Omkm1;
va1 = [e1;e2];

%%% Actualizar variables %%%
dqud=Kgain2a(4:6,:)*va1;

dq=q_km1(1:3)+dqud;
dqn=sqrt(abs(1-dq(1)^2-dq(2)^2-dq(3)^2));
qk1=[dq;dqn];
nqk=sqrt(qk1(1)^2+qk1(2)^2+qk1(3)^2+qk1(4)^2);
qk=qk1/nqk;

if (sign(qk(4)) == sign(y(4)))
    qk(4)=qk(4);
else
    qk(4)=-qk(4);
end

Omk1=Omkm1+Kgain2a(1:3,:)*va1;
Omk=Omk1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Matriz de covarianza del error %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

MpM=diag([1,1,1,1,1,1])-Kgain2a*Hk1;
Pk=(MpM)*Pksave;%*(MpM)'+Kgain2*Rk2*Kgain2';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% PROPAGACION DEL ESTADO %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Modelo linealizado del sistema %%%
A1=Amatrix(qk(1),qk(2),qk(3),qk(4),Omk(1),Omk(2),Omk(3));
A=A1;

%%% Propagacion de la Matriz de Covarianzas del Error %%%
Q = diag(
[cov_process2,cov_process2,cov_process2,cov_process1,cov_process1,cov_process1,cov_process1] );
Pkd = A*Pk*A'+Q;
Pksave = Pkd;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Propagacion de variables %%%
Sxk=[0, -qk(3), qk(2);
qk(3) .0, -qk(1);

```



```
s2 = 1;
elseif torque(2,1) < 0
    DC2= -1*(torque(2,1)*180)/(0.07);
    DCr2 = uint8(DC2)
    s2 = 0;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if torque(3,1) >= 0
    DC3= (torque(3,1)*180)/(0.07);
    DCr3 = uint8(DC3)
    s3 = 1;
elseif torque(3,1) < 0
    DC3= -1*(torque(3,1)*180)/(0.07);
    DCr3 = uint8(DC3)
    s3 = 0;
end
end
```

