



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

**FACULTAD DE INGENIERÍA**

**Actualización de la Instrumentación de  
un Simulador Satelital para Pruebas de  
Control de Orientación**

**TESIS**

Que para obtener el título de

**Ingeniero Eléctrico - Electrónico**

**P R E S E N T A**

Moreno Tagle José Carlos

**DIRECTOR DE TESIS**

Dr. Paul Rolando Maya Ortiz



**Ciudad Universitaria, Cd. Mx., 2017**



### *Dedicatoria*

*A mi padre Carlos H. Moreno Flores y a mi madre Patricia Tagle Balderas, por estar siempre a mi lado en cada paso que doy, por su amor siempre presente, por la felicidad que es tenerlos todos los días, por todo el esfuerzo para que sus hijos pudiesen llegar a este punto y sean capaces de seguir adelante con sus sueños. Para ambos con amor  $\rightarrow \infty$ .*

*A mis hermanos Iván y Lilia por estar siempre presentes brindándome su amor y compañía, por todos los momentos diarios llenos de risas y bromas, por todo lo que hemos compartido.*

*A mi novia Carolina Castro Alarcón por brindarme siempre su amor, su sonrisa, su paciencia y compañía, por todos estos años juntos, por lo mucho que hemos crecido tomados de la mano, por todo el camino recorrido y lo mucho que nos falta por explorar juntos en este viaje llamado vida. Gracias por llenarme de felicidad instantánea al caminar juntos de la mano durante todos estos años.*

*Al resto de mi familia: mis abuelos, Lilia y Rafael; a mis tí@s, Adriana, José, Memo, Manuel, Francisco, Enrique, Ricardo y Rafael. A quienes con amor siempre tengo presentes y llevo conmigo tantos recuerdos felices.*



# Agradecimientos

---

A la Universidad Nacional Autónoma de México y a la Facultad de Ingeniería por haberme dado una educación envidiable, por fomentar en mí el pensamiento crítico, por permitirme conocer a tantas personas excepcionales y encontrarme con esa gran diversidad de ideas, por todo lo que viví y crecí en esos años de estudiante dentro y fuera de sus aulas.

Al Dr. Paul Maya Ortiz, por aceptarme como su tesista y haber llevado a buen final este trabajo, por brindarme siempre su tiempo, apoyo y consejo, por todas esas pláticas donde siempre supo escuchar y mostrarse como un amigo.

Al M.I. Emilio Jiménez Madrigal por todo lo que aprendí trabajando junto a él, por toda la paciencia y asesoría recibida durante mi proyecto y por ser un buen amigo siempre dispuesto a brindar su ayuda a todos, gracias :).

Al Dr. Mario Alberto Mendoza por recibirme siempre amablemente durante mis visitas al Centro, por todo el tiempo, paciencia e ideas invertidas que dieron forma a este proyecto, por sus esfuerzos que siempre buscaron mejorar mi trabajo.

Al Dr. Esaú Vicente Vivas por aceptarme en su grupo de trabajo del Instituto de Ingeniería donde comencé esta etapa y donde pude conocer a personas excepcionales, por mostrar siempre esa amabilidad, alegría y entusiasmo que inspiraban confianza en sus alumnos.

A todos los brillantes maestros que he tenido a lo largo de mi vida académica que día a día dan su mejor esfuerzo, quienes siempre motivaron mis intereses y de quienes recibí no sólo una formación académica sino también humana.



Abu Dabi desde la Estación Espacial Internacional. Fotografía de Scott Kelly (@StationCDRKelly) durante su # YearInSpace.

*“Consider again that dot. That’s here. That’s home. That’s us. On it everyone you love, everyone you know, everyone you ever heard of, every human being who ever was, lived out their lives. The aggregate of our joy and suffering, thousands of confident religions, ideologies, and economic doctrines, every hunter and forager, every hero and coward, every creator and destroyer of civilization, every king and peasant, every young couple in love, every mother and father, hopeful child, inventor and explorer, every teacher of morals, every corrupt politician, every “superstar”, every “supreme leader”, every saint and sinner in the history of our species lived there – on a mote of dust suspended in a sunbeam.”*

*– Carl Sagan (1994), Pale Blue Dot*

# Índice general

---

<b>Índice de figuras</b>	<b>VII</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Antecedentes . . . . .	1
1.2. Motivación . . . . .	2
1.3. Objetivos de la tesis . . . . .	3
1.4. Planteamiento del problema . . . . .	3
1.5. Metodología . . . . .	4
1.6. Estado del Arte . . . . .	8
1.6.1. Diseños de simuladores de sistemas espaciales pequeños . . . . .	8
1.7. Contribución . . . . .	16
1.8. Organización del Trabajo . . . . .	16
<b>2. Arquitectura del Simulador Satelital MSA</b>	<b>19</b>
2.1. Descripción General . . . . .	19
2.2. Elementos y características mecánicas . . . . .	20
2.2.1. Plataforma circular . . . . .	20
2.2.2. Balero . . . . .	20
2.2.3. Copa y base . . . . .	20
2.2.4. Ruedas Inerciales . . . . .	21
2.2.5. Contrapesos manuales . . . . .	22
2.2.6. Características y especificaciones . . . . .	22
2.3. Instrumentación . . . . .	23
2.3.1. Batería . . . . .	23
2.3.2. Convertidores DC-DC conmutados . . . . .	23
2.3.3. Unidad de medición inercial . . . . .	24
2.3.4. Comunicaciones . . . . .	24

## ÍNDICE GENERAL

---

2.3.5. Computadora a bordo . . . . .	25
2.3.6. Driver para Motores . . . . .	25
2.3.7. Actuadores . . . . .	26
<b>3. Actualización de la Instrumentación</b>	<b>29</b>
3.1. Tarjetas de Convertidores DC-DC conmutados . . . . .	30
3.1.1. Convertidor DC-DC tipo Buck . . . . .	31
3.1.2. Convertidor DC-DC tipo Boost . . . . .	32
3.1.3. Convertidor DC-DC tipo SEPIC . . . . .	33
3.1.4. Fuente de 3.3 [V] a 2[A] . . . . .	36
3.1.5. Fuente de 5 [V] a 2 [A] . . . . .	39
3.1.6. Fuente de 12 [V] a 2[A] . . . . .	42
3.1.7. Diseño de las Tarjetas de Potencia Eléctrica . . . . .	48
3.2. IMU: ADIS16448 de Analog Devices . . . . .	50
3.3. Computadora a bordo: Zedboard . . . . .	52
3.3.1. Sistema de Adquisición de Datos . . . . .	54
3.4. Tarjeta de Drivers para Motores . . . . .	57
3.5. Actuadores: Motores BLDC EC 22 de Maxon . . . . .	59
3.6. Carga Útil: Raspberry Pi 2B y PlayStation Eye . . . . .	62
<b>4. Simulaciones y Resultados</b>	<b>65</b>
<b>5. Conclusiones</b>	<b>79</b>
<b>A. Apéndice</b>	<b>81</b>
A.1. Cuaterniones . . . . .	81
A.2. Método TRIAD . . . . .	83
A.3. Códigos . . . . .	85
A.3.1. Código en C para interactuar con la IMU ADIS16448 . . . . .	85
A.3.2. Código en MATLAB del Sistema de Adquisición de Datos . . . . .	94
A.4. Esquemáticos y Diseños de PCBs . . . . .	108
<b>Bibliografía</b>	<b>117</b>

# Índice de figuras

---

1.1. Plataforma plana del MIT para pruebas con las SPHERES [37]. . . . .	9
1.2. (a) Plataforma suspendida en aire tipo mesa y (b) Plataforma suspendida en aire tipo sombrilla [31]. . . . .	10
1.3. Plataforma rotacional tipo mancuerna [31]. . . . .	10
1.4. Plataforma tipo mesa de la Universidad de Yonsei en Corea del Sur. . .	11
1.5. Simulador tipo mesa en el SSL del MIT [14]. . . . .	12
1.6. Simulador TACT operado por la Universidad de Michigan [38]. . . . .	13
1.7. Simuladores desarrollados en el Instituto de Geografía de la UNAM [26].	14
1.8. Plataforma de pruebas operada por la Universidad de Hawái en Manoa [34]. . . . .	15
2.1. Simulador MSA del IINGEN con la instrumentación de [1]. . . . .	19
2.2. El balero y la copa del simulador de [25], maquinados en bronce fosforado.	21
3.1. Diagrama de bloques de la nueva instrumentación del simulador satelital.	29
3.2. Diagrama de bloques del nuevo sistema de potencia eléctrica. . . . .	30
3.3. Circuito simplificado del convertidor DC-DC tipo Buck. . . . .	31
3.4. Circuito simplificado del convertidor DC-DC tipo Boost. . . . .	33
3.5. Circuito simplificado del convertidor DC-DC tipo SEPIC. . . . .	34
3.6. Esquemático de la fuente conmutada de 3.3 [V] a 2 [A]. . . . .	37
3.7. Esquemático de la fuente conmutada de 5 [V] a 2 [A]. . . . .	40
3.8. Esquemático de la fuente conmutada de 12 [V] a 2 [A]. . . . .	43
3.9. Unidad de medición inercial con su sistema de referencia [13]. . . . .	50
3.10. Diagrama de conexiones de la IMU con cualquier sistema de procesamiento [13]. . . . .	51
3.11. Interfaces de la tarjeta de desarrollo Zedboard [5]. . . . .	53
3.12. Configuración del bloque <i>AXI Quad SPI</i> . . . . .	55

## ÍNDICE DE FIGURAS

---

3.13. Arquitectura implementada sobre el SoC Zynq para efectuar la comunicación por SPI. . . . .	56
3.14. Sistema de adquisición de datos en operación. . . . .	57
3.15. Tarjeta de drivers para motores. . . . .	59
3.16. Conmutación de los drivers para hacer girar un motor BLDC [12]. . . . .	61
3.17. El control de velocidad de un motor BLDC requiere de señales PWM [12].	62
3.18. Interacción entre la carga útil y una computadora personal. . . . .	63
3.19. Puesta en marcha de la carga útil. . . . .	64
4.1. Señal del voltaje de salida de la fuente de 3.3 [V]. . . . .	66
4.2. Acercamiento de la señal del voltaje de salida. . . . .	66
4.3. Señal de corriente en el capacitor de salida y en el capacitor de entrada.	66
4.4. Señal del voltaje de salida de la fuente de 5 [V]. . . . .	67
4.5. Señal de corriente en el capacitor de salida. . . . .	67
4.6. Señal de corriente en el capacitor de entrada. . . . .	67
4.7. Señal del voltaje de salida de la fuente de 12 [V] con $V_{in} = 16$ [V]. . . . .	68
4.8. Señal del voltaje de salida de la fuente de 12 [V] con $V_{in} = 11.4$ [V]. . . . .	68
4.9. Diagrama de bloques en Simulink para la ejecución de las simulaciones.	73
4.10. Diagrama de bloques del <i>ADCS</i> . . . . .	73
4.11. Resultados de las simulaciones con la ley de control (4.1). . . . .	74
4.12. Resultados de las simulaciones con la ley de control (4.2). . . . .	75
4.13. Procesamiento de los ángulos de orientación de la IMU por el UKF. . . . .	76
4.14. Comparación del método TRIAD con el algoritmo UKF en la generación del cuaternión. . . . .	76
4.15. Velocidad angular en los ejes de la IMU. . . . .	77
A.1. PCB para la alimentación de motores BLDC con fuentes de 12 [V] / 2 [A].	108
A.2. Vista en 3D de la PCB para la alimentación de motores. . . . .	108
A.3. PCB para la alimentación de plataformas de desarrollo con fuentes de 3.3, 5 y 12 [V] a 2 [A]. . . . .	109
A.4. Vista en 3D de la PCB para la alimentación de plataformas de desarrollo.	109
A.5. PCB de drivers para motores BLDC. . . . .	110
A.6. Vista en 3D de la PCB de drivers para motores BLDC. . . . .	110

---

## Capítulo 1

# Introducción

---

En muchas misiones satelitales, el subsistema de estimación y control de orientación (ADCS, del inglés “Attitude Determination and Control System”) es una parte clave para su éxito. Ya sea para realizar tareas de percepción remota, para alinear los paneles del satélite hacia el Sol o bien para mantener un apuntamiento fijo a una antena en la Tierra con la finalidad de obtener el mayor ancho de banda, el ADCS es el responsable de lograr estos objetivos.

Para validar el funcionamiento del ADCS es necesario contar en el laboratorio con una plataforma confiable que ofrezca las condiciones de un medio sin fricción similares a las que encontrará el satélite en órbita.

En el Instituto de Ingeniería (IINGEN) de la UNAM se cuenta con una mesa suspendida en aire (MSA) que imita un ambiente sin fricción simulando las condiciones de microgravedad en las que debe desempeñarse un satélite pequeño en el espacio. Este simulador tiene como objetivo, además de imitar la dinámica espacial, servir como plataforma de validación experimental de algoritmos de estimación y control de orientación en tres ejes por lo que debe contar con la instrumentación adecuada para llevar a cabo estas tareas que permitan obtener resultados confiables sobre el desempeño de los algoritmos propuestos.

### 1.1. Antecedentes

Desde hace varios años, en el laboratorio de sistemas espaciales del Instituto de Ingeniería de la UNAM se ha trabajado en el desarrollo de tecnología para satélites pequeños, parte de estos esfuerzos se han enfocado al problema de estimación y control de orientación. Dentro de esta línea de investigación, el grupo de trabajo cuenta con varios años de experiencia y diversos trabajos, tanto de licenciatura como de posgrado, han sido realizados.

Entre estos trabajos se encuentran el diseño, el ensamble y la validación del subsistema de estabilización y sensores para un satélite educativo [2], la actualización del hardware del mismo [39], la experimentación con un nuevo sistema de procesamiento basado en FPGA [40], el modelado, la estimación y control para un nanosatélite [3], [23] hasta la instrumentación mecatrónica de la MSA para la validación de algoritmos de control de orientación utilizando un FPGA [1].

### 1.2. Motivación

La motivación de este trabajo de tesis nace a partir de los resultados obtenidos por [1] y la necesidad del grupo de continuar desarrollando la misma línea de investigación, atendiendo a las observaciones y recomendaciones emitidas por el autor para la corrección de parte de los detalles detectados que se presentaron durante la fase experimental.

En versiones anteriores de instrumentación de la MSA, debido a la falta de motores y sensores asociados de buena calidad, sólo fue posible realizar un control burdo de la MSA sobre un eje. También, había fallas mecánicas en los elementos que conformaban la MSA que provocaban vibraciones mecánicas no deseadas que dificultaban el control.

Por ello, este trabajo de tesis se enfoca en actualizar la instrumentación electrónica de la MSA con:

- Una nueva unidad de medición inercial (IMU, del inglés “inercial measurement unit”) que ofrezca una tasa de ruido más baja e incorpore un filtro digital que proporcione lecturas más estables.
- Motores sin escobillas (BLDC, del inglés “Brushless Direct Current”).
- Fuentes de alimentación basadas en convertidores DC-DC acordes a las necesidades eléctricas de la nueva instrumentación.
- Un nuevo sistema de procesamiento para la implementación de los algoritmos de estimación y control empleando un esquema de control distribuido, basado en el uso del SoC (“System on a Chip”) Zynq-7020 que ofrece las bondades de una arquitectura fija junto con una reprogramable en un mismo encapsulado y el uso de microcontroladores dsPIC para el control de bajo nivel de las ruedas inerciales.
- La incorporación de una carga útil que conste de una cámara web conectada a la tarjeta Raspberry Pi 2B que transmita la imagen a una computadora personal a través de un modem con conexión inalámbrica. La carga útil tiene el propósito de verificar que el objetivo de control ha sido alcanzado.
- Proponer un esquema de control de orientación con el que pueda validarse la nueva instrumentación de la MSA tomando como referencia libros, tesis o artículos publicados y que pueda acoplarse con relativa facilidad a los desarrollos realizados previamente por el grupo de trabajo.

Se espera que con este trabajo se pueda contar con una plataforma de simulación exitosa, escalable en términos de instrumentación que permita un proceso de mejora continua y que pueda ser de utilidad para desarrollos futuros pues las líneas de trabajo que pueden abordarse son muy diversas y van desde el diseño mecánico donde queda

mucho por hacer hasta la validación de diferentes algoritmos de control que ya han sido probados por medio de simulaciones numéricas pero ha quedado pendiente su validación experimental.

### 1.3. Objetivos de la tesis

- Instrumentar la mesa suspendida en aire (MSA) con versiones actualizadas de sensores de navegación inercial, fuentes de energía basadas en convertidores DC-DC y actuadores con motores BLDC.
- Instrumentar la MSA con una nueva plataforma computacional para realizar tareas de control de orientación satelital, empleando el chip Zynq-7020.
- Crear parte de la base de hardware necesaria para tener un sistema de control distribuido para el manejo de actuadores de la MSA, en donde se combinen arquitecturas computacionales convencionales (dsPIC para el manejo de motores) junto con “sistemas en un chip” (SoC, híbrido entre un FPGA y una arquitectura fija).
- Proponer un esquema de control de orientación para la validación de la nueva instrumentación de la MSA con base en libros, artículos arbitrados o tesis.

### 1.4. Planteamiento del problema

Como se comentó previamente, se detectó que la falta de sensores y actuadores de buena calidad afectaba el desempeño de los algoritmos a validar. Aunado a ello, la falta de un sistema de procesamiento optimizado para tareas de computo intensivo, provocaba que se tuviese que recurrir a un esquema de procesamiento externo a través de una PC ejecutando Matlab (técnica conocida como hardware-in-the-loop (HIL)) comunicando ambas plataformas por medio de radios que transmitían la información en formato serial proveniente de los sensores de la MSA hacia la PC para su procesamiento y de vuelta a la MSA.

El uso de esta técnica tiene la desventaja de presentar retardos por, el tiempo que tarda la información en viajar de la MSA hacia la PC, más el tiempo de ejecución de los algoritmos sobre la PC, más el tiempo de retorno de la información hacia la MSA, perdiendo parte de la esencia de tiempo real. Estos retardos si no se tienen en consideración en el esquema de control afecta los resultados esperados.

Con esto en mente, el problema a resolver puede plantearse como sigue:

## 1. INTRODUCCIÓN

---

- Satisfacer los requerimientos eléctricos de los nuevos actuadores, sensores y tarjetas de procesamiento a través del diseño de una nueva instrumentación electrónica.
- Diseñar una instrumentación electrónica de menores dimensiones (10 x 10 [cm]), modular y escalable que facilite resolver el problema de la distribución de masas en la MSA, que permita modificaciones y mantenimiento con relativa facilidad a largo plazo.
- Diseñar tarjetas electrónicas que permitan la implementación de esquemas de control distribuido para el control de motores BLDC.
- Seleccionar una nueva plataforma computacional que permita procesar a bordo (embeber) los algoritmos de estimación y control, evitando la necesidad de enviar la información hacia una PC.
- Investigar y proponer un algoritmo de control de orientación que pueda convivir con desarrollos previos y con la nueva instrumentación, con el cual puedan obtenerse resultados experimentales a corto plazo sobre el desempeño de dichos algoritmos y la nueva instrumentación.

### 1.5. Metodología

El primer problema a abordar es el diseño de las fuentes de alimentación. Para ello, se diseñarán dos tarjetas electrónicas, una de ellas estará dedicada exclusivamente a satisfacer los requerimientos eléctricos de los motores BLDC y la segunda estará dedicada a alimentar cualquier tarjeta de procesamiento que se quiera utilizar como computadora a bordo en la MSA.

Dentro de las primeras actividades para determinar el tipo de fuentes que se diseñarán, está tomar en cuenta las características de la batería. Se trata de una batería de 4 celdas Thunder Power RC de polímero de litio (LiPo) con un voltaje de salida de 14.8 [V] y 7,700 [mAh], estos valores son importantes pues representan el voltaje de entrada a las tarjetas que habrán de diseñarse así como la corriente que la batería puede llegar a proporcionar. También, debe tenerse en cuenta que la batería se irá descargando conforme sea utilizada por lo que el tipo de fuentes para mantener los valores de voltaje y corriente necesarios para mantener la electrónica funcionando deberán ser convertidores DC-DC conmutados. Otro factor a considerar es que la batería no puede descargarse totalmente pues quedaría atrofiada, se sabe que el valor mínimo de voltaje que puede alcanzar es 11 [V] por lo que es conveniente incorporar en el diseño de la tarjeta algún circuito que indique por medio de un LED que la batería está a punto de alcanzar dicho valor por lo que debe ser cargada de inmediato.

Además de la batería, los otros elementos que deben considerarse para el diseño de las tarjetas son los motores BLDC en particular sus características eléctricas, identificando los voltajes y corrientes nominales. Posteriormente, se tendrá que fijar un valor de voltaje y corriente de operación de acuerdo a la velocidad y par que se espera obtener del motor para mover la MSA y la instrumentación que se encontrará sobre ella. Estos valores serán utilizados para fijar el voltaje y corriente de salida de la tarjeta de alimentación con las fuentes para cada motor.

Para el caso de la tarjeta que realizará la tarea de alimentar a cualquier plataforma de procesamiento que requiera utilizarse, deberán considerarse los voltajes de alimentación más comunes con los que operan la mayoría de las plataformas de desarrollo. Con ello, deberá pensarse en diseñar una sola tarjeta que cuente con tres diferentes fuentes que puedan proveer de 3.3 [V], 5[V] y 12 [V] a 2 [A].

Una vez conociendo los requerimientos de los elementos que deberán operar gracias a las fuentes de alimentación es necesario familiarizarse e investigar sobre los tres diferentes tipos de convertidores DC-DC que existen (Buck, Boost, SEPIC) y elegir al más adecuado para cada caso en particular. Al investigar sobre fuentes conmutadas, se deberá prestar especial atención a la función que realizan sus componentes más críticos como el caso de los capacitores de entrada y salida, el inductor, los transistores MOSFET, la resistencia de sensado, etc.

Habiendo seleccionado el tipo de fuente adecuada para cada caso, se investigará alguna topología construida en torno a algún circuito integrado comercial a partir de la documentación que publican los principales fabricantes de circuitos integrados en la web. Con la topología adecuada, se consultarán las ecuaciones de dicha documentación para calcular los elementos críticos antes mencionados y obtener los valores eléctricos que se requieren para este trabajo. Es recomendable, prestar atención a las sugerencias que se resaltan en la documentación para la adquisición de los componentes críticos con ciertos fabricantes de dispositivos electrónicos pues muchas veces estas recomendaciones están basadas en diversos factores como los rangos de temperatura de operación, las dimensiones de los componentes, la eficiencia, el tipo de aplicación, etc.

Para comprobar que el diseño es capaz de proporcionar los valores esperados es necesario realizar simulaciones del comportamiento de las fuentes. Para ello, se debe dibujar el circuito sobre algún software de simulación de sistemas electrónicos especializado en electrónica analógica y de potencia. Si los resultados de las simulaciones son favorables entonces ese diseño puede ser utilizado dentro un software de diseño de placas de circuitos impresos.

Es importante tener en mente que el diseño de la placa deberá de ser a lo sumo de 10 x 10 [cm] por dos razones, la primera es que el fabricante maneja esas dimensiones como su estándar por lo que placas de mayor tamaño pueden incrementar el costo de fabricación y la segunda es que se busca que la instrumentación sea más compacta y modular en comparación con la versión anterior, esto para facilitar la distribución de masas en la MSA, factor que resulta crucial para obtener sus parámetros mecánicos.

## 1. INTRODUCCIÓN

---

El segundo problema por resolver es el diseño de la placa de drivers que consiste en un sistema mínimo de procesamiento basado en el microcontrolador dsPIC33F con la electrónica de potencia necesaria para mover a los motores BLDC.

Si no se tiene experiencia previa con motores BLDC es necesario familiarizarse por medio de la documentación que diferentes fabricantes de circuitos integrados publican en sus sitios web [10], [11], [12]. Gran parte de este material es didáctico y cubre desde la teoría necesaria, desde el punto de vista de las máquinas eléctricas, hasta una explicación detallada del tipo de electrónica necesaria para su puesta en operación. En particular, se debe prestar especial atención a la electrónica de potencia pues se necesita estudiar y entender la función que realizan cada uno de sus componentes para poder escribir y cargar un primer programa en el microcontrolador que ponga en marcha al motor.

Debe notarse que independientemente de la fuente de consulta de la cual provenga la documentación, la electrónica que se propone para manejar un motor BLDC por medio de un microcontrolador es la misma y se trata de un arreglo de transistores MOSFET acompañados de sus respectivos buffer, un arreglo mejor conocido como puente trifásico.

Esta etapa será una de las más demandantes en tiempo y esfuerzo pues se debe comprender tanto el funcionamiento del motor, su electrónica, la forma de conmutar las fases, sus efectos en el sentido de flujo de la corriente, la dirección del campo magnético del rotor, etc.

Por último con el sistema mínimo y la electrónica ya dimensionados se puede comenzar a dibujar los esquemáticos en algún software de diseño electrónico para conformar la placa de circuito impreso. Es necesario incluir en el diseño de la tarjeta tres *header* macho para permitir la comunicación por medio del protocolo serial SPI con el Zynq-7020 y las otras dos tarjetas gemelas dedicadas al control de sus respectivos motores.

Esta tarjeta tendrá que ser validada una vez fabricada y soldados todos sus componentes pues debido a la falta de soporte de algunos de los componentes importantes en las aplicaciones de simulación más comunes, no sería posible llevarla a cabo. Por lo tanto, la estrategia será terminar la placa, localizar cualquier error, y en caso de existir, corregirlo sobre la tarjeta. Con esto, quedaría concluida la primera parte de actualización de la nueva instrumentación.

La última fase consiste en la familiarización con la tarjeta de desarrollo Zedboard. Esta fase constituye otro gran reto sobre todo si no se tiene experiencia alguna en el desarrollo de sistemas embebidos con FPGA por lo que si es el caso, se recomienda consultar los ejemplos desarrollados de [20], correspondientes a circuitos digitales sobre todo para ganar experiencia con el lenguaje de descripción de hardware VHDL. En este trabajo el uso de VHDL será al mínimo sin embargo es necesario conocerlo principalmente para realizar tareas básicas para conectar los periféricos de la tarjeta al Zynq-7020.

Lo siguiente, es realizar las practicas de familiarización de [5], [6] con el ambiente de diseño de arquitecturas reconfigurables Vivado, se recomienda realizar todas las

prácticas pues con ello se tiene un conocimiento más a detalle del flujo de diseño para la creación de cualquier arquitectura que se desee. Debe recordarse que se está trabajando con un SoC, el Zynq-7020, por lo que no sólo se trata de utilizar los recursos de la lógica programable sino también en hacer uso del procesador ARM Cortex A9, capaz de realizar con eficacia tareas de procesamiento de algoritmos de control, procesamiento digital de imágenes, video, etc [21].

Resulta entonces necesario aprender a desarrollar software para el procesador ARM. Para ello se utilizará SDK, un entorno de desarrollo de Xilinx que permite la creación de aplicaciones en lenguaje C/C++ y donde se cuenta a disposición del usuario con una gran cantidad de librerías que permiten la configuración del FPGA y del mismo procesador. Para lograr esta tarea, se recomienda consultar a [4] donde se explica a detalle y se muestran diferentes ejemplos desde cero sobre como programar al procesador ARM del Zynq-7020 utilizando el lenguaje C.

Con la experiencia ganada en el uso de la tarjeta de desarrollo se puede comenzar a integrar parte de la instrumentación con lo que será la computadora a bordo. Por lo que, lo primero que debe integrarse es la tarjeta Zedboard con la IMU, para ello se debe estudiar su documentación donde se detallan sus características y modos de programación. En la documentación se encontrará que la forma de comunicarse con la IMU es por medio del protocolo de comunicación serial SPI y que ofrece una característica llamada “Burst Mode” donde la IMU transmite un paquete con 12 datos de 16 bits cada uno con la información de cada sensor. Tendrá que evaluarse qué resulta más conveniente, si hacer uso de un “core” que resida en la lógica programable para lograr la comunicación con la IMU y que esa información sea retransmitida al procesador ARM para hacer uso de ella o bien, hacer uso del módulo SPI físico que el procesador ya tiene incorporado como parte de su arquitectura.

Con éxito en esta tarea se puede comenzar a programar un sistema de adquisición de datos con la finalidad de visualizar en la computadora las señales correspondientes a cada uno de los sensores. Para esta tarea puede hacerse uso de cualquier entorno de desarrollo, en este trabajo se empleó Matlab. Sin embargo, queda a criterio y comodidad del desarrollador seleccionar el software con el que se desea trabajar.

El plan a seguir será, una vez lograda la comunicación entre la tarjeta Zedboard y la IMU por SPI, habrá que retransmitir esos paquetes de datos por medio del puerto serie hacia una PC. El programa que se desarrolle deberá abrir el puerto serie de la PC, hacer las configuraciones necesarias como la velocidad a la que se está transmitiendo e incorporar algún algoritmo para ordenar los datos de los sensores provenientes de la tarjeta, se recomienda que la salida del algoritmo sea una matriz con 12 columnas correspondientes a los puntos de datos de cada sensor. Para poder tener un conjunto de muestras significativo se deben adquirir al menos 2,000 puntos de datos por sensor, esto hace que la dimensión de la matriz de datos sea de al menos 2,000 x 12.

Si se elige utilizar Matlab y con la matriz de datos ya creada, resulta muy sencillo crear las gráficas que muestren la información de los sensores pues ese entorno de

desarrollo está enfocado a trabajar con vectores y matrices. Sin embargo, presenta una gran desventaja, mostrar la información de los sensores en tiempo real resulta muy complicado por lo que una manera práctica de abordar este problema es, realizar un muestreo durante al menos 40 [s], obtener la matriz de datos, almacenarlos en un archivo de texto (para el caso de que algún otro “script” los requiera), realizar el escalamiento de datos y finalmente graficar. Con esto, quedaría finalizado el sistema de adquisición de datos.

Como complemento a este trabajo, aprovechando que se tiene un sistema de adquisición de datos y la comunicación entre los sensores y la tarjeta Zedboard, se pueden probar mediante simulaciones numéricas desarrollos previos generados por el grupo de trabajo como el llamado Filtro de Kalman Unscented (UKF) para el procesamiento de las señales de la IMU [22]. Además, puede seleccionarse algún controlador reportado en la literatura que pueda adaptarse con facilidad al UKF con la finalidad de analizar cuál sería el posible comportamiento que debería reflejarse en la MSA.

Sin embargo, debe aclararse que el trabajo correspondiente al modelado, análisis e implementación de diferentes controladores embebidos en la tarjeta Zedboard sobre la MSA, dada su extensión y complejidad, queda fuera de los alcances de esta tesis y es apropiado para otro trabajo de tesis de licenciatura o maestría.

### 1.6. Estado del Arte

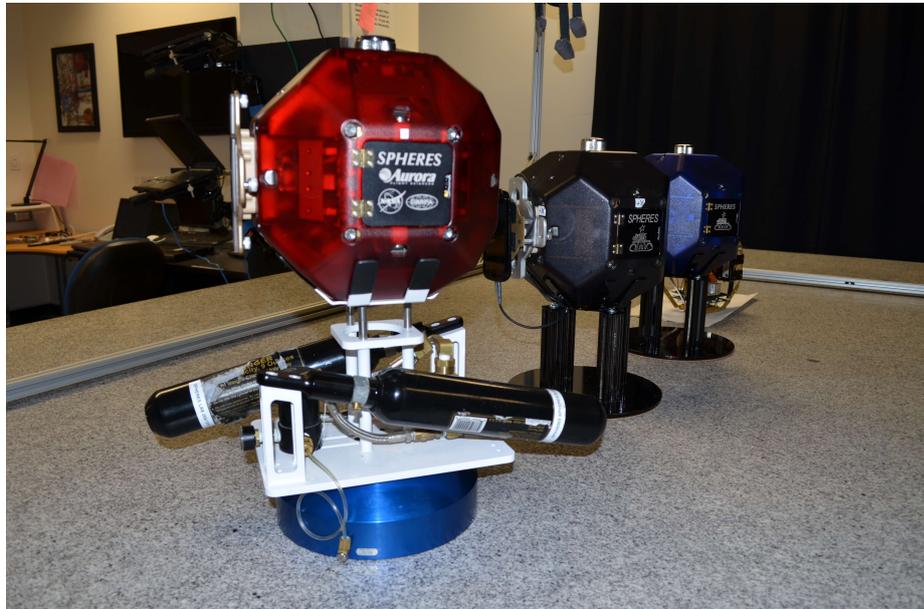
El desarrollo de simuladores totalmente integrados para la validación experimental de tecnología para sistemas espaciales es un área de interés tanto para instituciones universitarias como para entidades gubernamentales de diferentes países pues les permite probar en tierra, parte de sus desarrollos en hardware y software con la finalidad de analizar su desempeño en un ambiente controlado que emule parte de las condiciones a las que deberán enfrentarse y conocer lo que puede esperarse de ellos para incluirlos o descartarlos de los diseños finales. Permite pues ganar un nivel de certeza sobre lo que va enviarse al espacio.

En las secciones siguientes se describen los diferentes diseños para la construcción de simuladores de sistemas espaciales pequeños y algunos de los desarrollos más sobresalientes en el área.

#### 1.6.1. Diseños de simuladores de sistemas espaciales pequeños

Los diseños para simuladores de sistemas espaciales pueden clasificarse en dos grandes grupos, plataformas planas para pruebas con vehículos suspendidos en aire y plataformas suspendidas en aire rotacionales.

Las plataformas planas para pruebas con vehículos suspendidos en aire están construidas a partir de una superficie en forma de plano bien delimitada sobre la cual se colocan los vehículos espaciales que se desea probar. Cada uno de estos vehículos va montado sobre un pequeño sistema de transporte que lleva consigo dos tanques de  $CO_2$  comprimido que permiten hacer flotar al vehículo al formar un cojinete de aire entre la superficie plana al fondo del sistema de transporte y la superficie de la plataforma. Estas plataformas de prueba por lo general permiten tener tres grados de libertad, dos grados de libertad traslacionales a lo largo de la plataforma de pruebas y sólo un grado de libertad rotacional sobre el eje perpendicular a la superficie plana [14]. Además, con este tipo de simuladores se tiene la ventaja de poder probar varios vehículos a la vez tantos como las dimensiones de la plataforma lo permitan, otra de las ventajas es que se puede experimentar con algoritmos más complejos como vuelos en formación [14].



**Figura 1.1:** Plataforma plana del MIT para pruebas con las SPHERES [37].

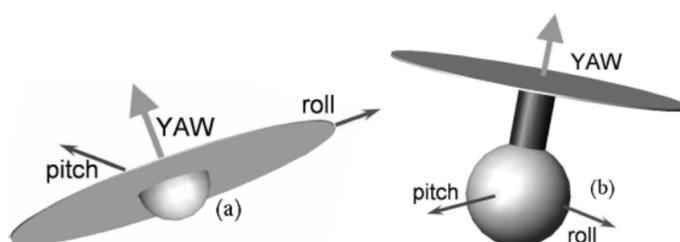
Un ejemplo representativo de lo anterior, se da en el Laboratorio de Sistemas Espaciales (SSL) del MIT donde se encuentra en marcha el proyecto SPHERES. Las SPHERES son pequeños satélites poliédricos de 18 caras, 20 [cm] de diámetro y una masa de 4.1 [kg], se espera que en el futuro sean utilizados para formar constelaciones de pequeños satélites que reemplacen las labores de los gigantes y costosos satélites actuales [32]. También, se busca lograr que las SPHERES trabajen en conjunto por ejemplo, para ensamblaje de satélites en el espacio es decir, en lugar de lanzar un satélite enorme, que se lancen varios de estos vehículos y se ensamblen solos para formar al satélite [32]. Para lograr estas tareas, en el SSL se cuenta con una plataforma de simulación plana de 1.5 [m] por 2 [m] con capacidad de hasta tres SPHERES, que ofrece las características antes descritas. Actualmente, se tienen tres de estos vehículos

## 1. INTRODUCCIÓN

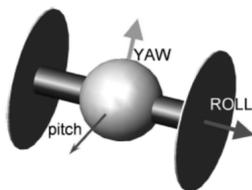
---

en la Estación Espacial Internacional, ambiente que representa la máxima plataforma de pruebas pues en ella se tienen los seis grados de libertad, tres grados de libertad rotacionales y tres grados de libertad traslacionales [14].

El otro grupo dentro del área de diseño para simuladores está conformado por las plataformas rotacionales suspendidas en aire. La gran ventaja que presentan este tipo de plataformas frente a las plataformas planas es que poseen los tres grados de libertad rotacionales lo que las vuelve ideales para realizar pruebas de estimación y control de orientación [14]. Estas plataformas operan con el uso de aire comprimido, para formar un cojinete de aire entre una estructura esférica fijada a la plataforma y una superficie cóncava que expulsa el aire. La desventaja evidente es que no poseen ningún grado de libertad traslacional.



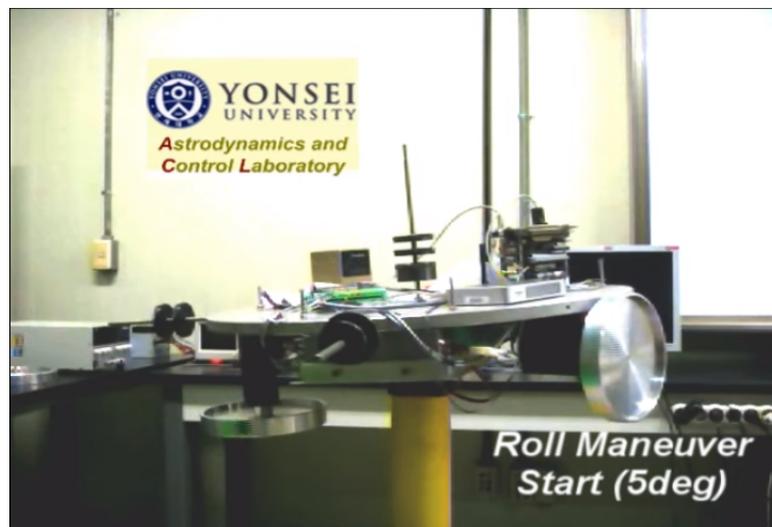
**Figura 1.2:** (a) Plataforma suspendida en aire tipo mesa y (b) Plataforma suspendida en aire tipo sombrilla [31].



**Figura 1.3:** Plataforma rotacional tipo mancuerna [31].

Dentro de este grupo se destacan tres diseños, estos son la plataforma suspendida en aire tipo mesa, la plataforma tipo sombrilla y la plataforma tipo mancuerna pues constituyen los diseños más comúnmente utilizados por los diferentes laboratorios. Los dos primeros diseños son muy similares entre sí pues ambos tienen rango completo de movimiento sobre el ángulo de guiñada, su diferencia radica en que el simulador tipo sombrilla ofrece un rango de movimiento mucho más amplio en los ángulos de rotación de alabeo y cabeceo debido a que la plataforma sobre la cual se monta la instrumentación se encuentra desplazada con respecto al elemento esférico lo que permite ese rango de movimiento extra [14]. A continuación, se describen algunos de los desarrollos más importantes realizados en diferentes instituciones que presentan alguno de los diseños anteriores.

La plataforma tipo mesa de la Universidad de Yonsei en Corea del Sur representa un buen ejemplo del desempeño y resultados que pueden lograrse con este tipo de simuladores. Se trata de una plataforma para validación de algoritmos de control de orientación en tres ejes para microsátélites, lo que obliga a que el simulador deba desempeñarse adecuadamente con una masa total de entre 50 a 100 [kg]. En [16], se reporta que el simulador desarrollado, completamente instrumentado, tiene una masa total de 50 [kg]. En cuanto a los rangos de movilidad, el simulador tiene los tres grados de libertad rotacionales y es capaz de realizar maniobras de  $\pm 180$  [°] en guiñada y  $\pm 45$  [°] en alabeo y cabeceo. Este simulador cuenta con un sistema comercial de suministro de aire comprimido con el cual es capaz de levantar cargas de hasta 110[kg].



**Figura 1.4:** Plataforma tipo mesa de la Universidad de Yonsei en Corea del Sur.

De acuerdo con [16], el simulador cuenta con la siguiente instrumentación. Un sistema de referencia de orientación y rumbo (AHRS, del inglés "Attitude and Heading Reference System") que integra giróscopos, acelerómetros y magnetómetros triaxiales; la gran diferencia entre un AHRS y una IMU es que el AHRS incorpora procesamiento a bordo por lo que se dispone directamente de la información de la orientación y el rumbo del vehículo. También, cuenta con una PC104 como computadora a bordo, capaz de procesar diferentes tipos de señales, utiliza un sistema operativo en tiempo real y es compatible con la paquetería de MATLAB que se requiere para generar el código para que la PC104 realice las tareas de control en tiempo real. Como sistema de potencia utiliza tres set de baterías de plomo-ácido (seis baterías en total) donde cada set proporciona 12 [V] y 10 [Ah]. En cuanto al sistema de comunicaciones, el simulador posee un router inalámbrico que establece la comunicación por medio del protocolo TCP/IP entre la computadora a bordo PC104 y una computadora de escritorio que opera como "host" desde donde puede recibir comandos. Finalmente, como medio de actuación, utiliza tres ruedas inerciales con motores BLDC y su electrónica asociada.

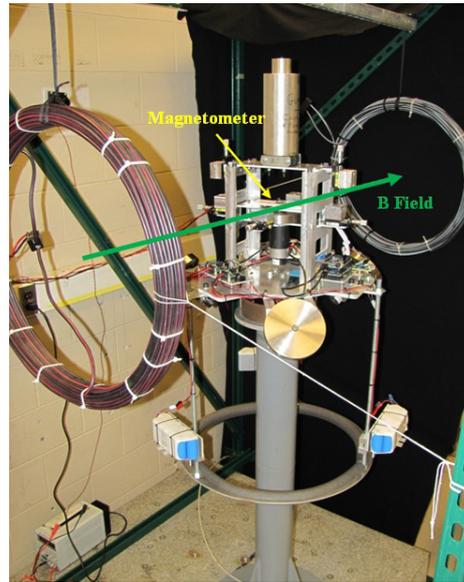
## 1. INTRODUCCIÓN

---

Otro desarrollo importante es el simulador del SSL del MIT. Según se describe en el trabajo de [14], se trata de una plataforma tipo mesa de tres grados de libertad diseñada y construida para satisfacer las necesidades del laboratorio de contar con un simulador de control y estimación de orientación para los múltiples proyectos de satélites pequeños que tienen en curso, el proyecto SPHERES y como una herramienta didáctica que permita experimentar con diferentes algoritmos a los alumnos de los cursos de teoría de control.

El simulador cuenta con toda la instrumentación necesaria para emular varios de los subsistemas en un satélite pequeño, cuenta con dos arduinos MEGA como computadora a bordo que se encargan de ejecutar los algoritmos de control y estimación, estos se enlazan por medio de un HUB USB a una de las dos tarjetas de comunicaciones USB Tru-Link, la otra se encuentra conectada a una PC externa, de esta manera se comunican el simulador y la PC que cumple el rol de estación terrena. El arduino principal hace uso de un modulo externo de reloj en tiempo real de la compañía Sparkfun, esto se debe a que el reloj interno del arduino no es lo suficientemente preciso como para llevar a cabo tareas en tiempo real. En cuanto a los sensores, posee la unidad de medición inercial triaxial ADIS 16365 de la compañía Analog Devices, un magnetómetro triaxial Micromag3 de PNI Corporation. Dentro de este rubro, pueden también considerarse los cuatro encoders correspondientes a cada motor, se trata de encoders de tres canales que entregan una señal cuadrada de 5 [V] de amplitud y 1,000 PPR o 1,000 ciclos cada revolución del motor. En cuanto al sistema de potencia, el simulador posee tres baterías de níquel-metal hidruro (NiMH) de 14 [V] donde cada una ofrece una capacidad de carga de 6.5 [Ah]. Las baterías son independientes una de otra, una de ellas es utilizada para energizar al convertidor DC-DC que se encarga de alimentar a toda la instrumentación electrónica y las otras dos están dedicadas para alimentar a su respectiva tarjeta para el control de motores. Como medio de actuación utiliza cuatro ruedas inerciales fijadas a motores de DC, cada par de motores es controlado por una tarjeta Pololu Trex que además sirve de alimentación para los encoder de cada par de motores.

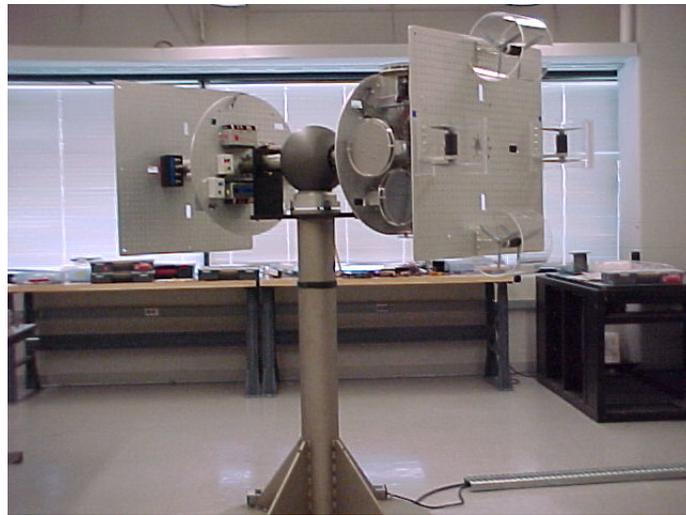
El simulador desarrollado en [14] por su diseño ofrece los siguientes rangos de maniobrabilidad,  $\pm 360$  [°] en guiñada y  $\pm 19$  [°] en alabeo y cabeceo. Además, incorpora el uso de pequeñas masas deslizantes a lo largo de los tres ejes que sirven como un medio de ajuste manual del centro de masa de la plataforma. Estas masas se manipulan por



**Figura 1.5:** Simulador tipo mesa en el SSL del MIT [14].

medio de seis perillas de ajuste fino, dos perillas en cada eje, una perilla para ajustar las masas en la parte positiva del eje y otra para la parte negativa. La única excepción es en el caso del eje Z donde se tienen las dos perillas en la parte positiva del eje lo que obliga a que deban moverse a la par. Finalmente, algo que resalta en este simulador es que posee una instalación para generar un campo magnético externo por medio de dos embobinados de 55 [cm] de radio, separados una distancia de 154 [cm] para apoyar las tareas de estimación al mejorar las lecturas del magnetómetro.

Por el lado de las plataformas tipo mancuerna, en la reseña de [14] se menciona que la Universidad de Michigan desarrolló el simulador TACT (Triaxial Attitude Control Testbed). Este diseño presenta algunas ventajas frente a las plataformas tipo mesa, principalmente en cuanto a los grados de libertad rotacionales. El TACT tiene restricción únicamente en el eje Y, pues sólo puede lograr maniobras de  $\pm 45$  [°] en cabeceo mientras que los ángulos de alabeo y guiñada se encuentran sin restricción con  $\pm 360$  [°]. En cuanto a sus características mecánicas, este simulador puede soportar cargas de hasta 163 [kg] y está diseñado para operar con diferentes tipos de actuadores que pueden ser ventiladores, ruedas inerciales y actuadores de masa. Debe aclararse que los ventiladores tienen el propósito de intentar simular al sistema de propulsión (thrusters) presente en algunos satélites mientras que los actuadores de masa tienen el propósito de simular los pares de origen gravitacional que pudiese experimentar el satélite en órbita.



**Figura 1.6:** Simulador TACT operado por la Universidad de Michigan [38].

En México, también se ha realizado investigación y desarrollo en el campo de simuladores para validación experimental de sistemas de control y estimación de orientación, como primera referencia está el IINGEN de la UNAM con los diversos trabajos y versiones del simulador MSA que anteceden a esta tesis, destacando principalmente el trabajo de [1]. El capítulo siguiente se dedica a describir a detalle el simulador MSA así como la instrumentación mínima con la que debe contar. Es importante mencionar que existen

## 1. INTRODUCCIÓN

---

otras entidades dentro la misma Universidad que han hecho aportes significativos en este campo.

En el Laboratorio de Análisis Geoespacial (LAGE) del Instituto de Geografía de la UNAM se han desarrollado simuladores satelitales e inclusive se han llegado a realizar transferencias tecnológicas exitosas para satisfacer la demanda en otras instituciones como en el caso del recién creado *Vietnam National Satellite Center* [26]. El simulador desarrollado tipo mesa descrito en [27] recibe el nombre de SIMUSAT 3.0, su plataforma circular de 58 [cm] de diámetro está fabricada de espuma de estireno comprimida por fibras de carbono-epoxy. El simulador posee una capacidad de carga de 25 [kg] y utiliza una semiesfera montada sobre una superficie cóncava llamada copa que expulsa aire comprimido a través de ductos milimétricos por su periferia para formar un colchón de aire de 0.0254 [mm] y lograr el efecto de cero fricción. El diseño del simulador permite maniobras en tres ejes de  $360$  [°] en guiñada y  $\pm 50$  [°] en alabeo y cabeceo.



**Figura 1.7:** Simuladores desarrollados en el Instituto de Geografía de la UNAM [26].

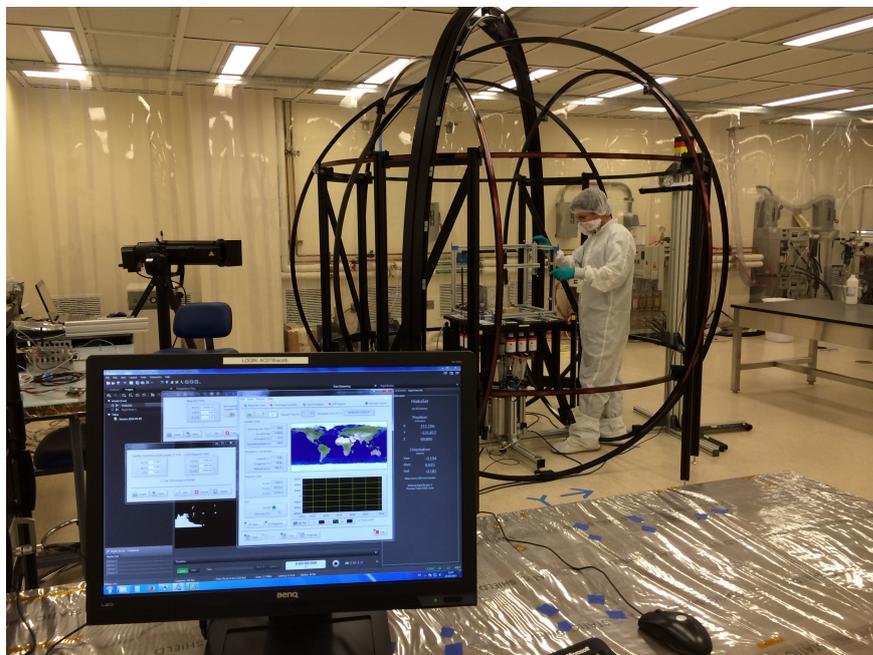
Un elemento que destaca de este simulador es que incorpora un sistema de balanceo automático por medio de dos masas deslizantes colocadas cada una en los ejes X y Y del simulador. Según se detalla en [27], estas masas se desplazan linealmente para balancear el simulador en los ejes X y Y a partir de la información recibida por una brújula electrónica hasta lograr que la plataforma se encuentre totalmente horizontal dentro de ciertos márgenes de tolerancia; este proceso es necesario para reducir los pares gravitacionales al hacer coincidir el centro de masa con el centro geométrico de la plataforma.

Finalmente, por la parte de instrumentación, este simulador utiliza como computadora a bordo una tarjeta de desarrollo LP Fox 3500 basada en el microcontrolador Rabbit 3000, como sensores utiliza una brújula electrónica EZ-COMPASS-3 que se

compone de dos inclinómetros y un magnetómetro triaxial, como sistema de comunicaciones utiliza dos radio módem que comunican al simulador con una PC que opera como estación terrena. Cabe mencionar que sobre la PC se ejecuta una interfaz gráfica que permite el envío de comandos y la captura de datos desde la plataforma para efectos de análisis. En cuanto a los actuadores, el simulador utiliza tres ruedas inerciales colocadas ortogonalmente y cuenta además con tres bobinas de torque magnético colocadas ortogonalmente entre sí y en dirección opuesta a las ruedas, las bobinas cumplen la función de desaturar las ruedas inerciales y de servir como respaldo en el control de orientación [27].

Debe resaltarse que en esta breve reseña sobre el desarrollo de simuladores satelitales sólo se describen una pequeña parte de ellos pues es un área de intenso desarrollo para cualquier institución que busca desarrollar y validar tecnología propia. Algunos otros desarrollos importantes que no se describen en este trabajo son el simulador SimSat II del Instituto de Tecnología de la Fuerza Aérea (AFIT) [29], el par de simuladores DSACSS de Virginia Tech [30], entre otros.

Por otro lado, existen empresas de tecnología aeroespacial que construyen simuladores altamente sofisticados para satisfacer la demanda de diferentes instituciones. La Universidad de Washington opera un simulador de cinco grados de libertad desarrollado por la empresa Andrews Space [33], de igual forma la Universidad de Hawái en Manoa adquirió una plataforma de pruebas de la empresa alemana Astro- und Feinwerktechnik Adlershof GmbH necesaria para el desarrollo de su satélite HiakaSat [34].



**Figura 1.8:** Plataforma de pruebas operada por la Universidad de Hawái en Manoa [34].

### 1.7. Contribución

Como resultado de este trabajo se hicieron diferentes aportaciones principalmente en el área de Instrumentación a la línea de investigación de desarrollo y validación de tecnología para satélites pequeños que lleva cultivando el Instituto de Ingeniería de la UNAM durante varios años.

Una vertiente de esta línea de trabajo busca validar experimentalmente algoritmos de estimación y control sobre diferentes sistemas embebidos en el simulador satelital MSA. Antes de poder abordar este problema es necesario diseñar, validar e integrar toda la instrumentación necesaria que permita ejecutar y monitorear el desempeño de los algoritmos propuestos sobre el simulador MSA.

La contribución de este trabajo se compone del diseño de dos nuevas tarjetas de potencia eléctrica basadas en convertidores DC-DC conmutados, el diseño y ensamble de una nueva tarjeta para el control de un motor BLDC, la integración de la nueva computadora a bordo basada en un SoC con la nueva unidad de medición inercial, la incorporación de una carga útil (cámara) para verificar el apuntamiento al objetivo de control y la validación numérica de un potencial esquema de control con desarrollos previos.

### 1.8. Organización del Trabajo

Este trabajo está compuesto de 5 capítulos y una sección de apéndices.

En el capítulo 1 se presentó una breve introducción a la importancia del sistema de determinación y control de orientación, la experiencia y parte de los desarrollos del grupo de trabajo, la motivación de continuar desarrollando sobre la validación experimental de instrumentación y algoritmos sobre un simulador satelital, los alcances y objetivos que se persiguen en el trabajo así como la problemática a la que se enfrentó para lograrlos. Al final, se presentó la metodología a modo de "receta" para llegar al punto de cierre de este trabajo y se hizo una reseña sobre el desarrollo de tecnología para simuladores de control de satélites en instituciones educativas y gubernamentales, con la finalidad de presentar el estado del arte.

En el capítulo 2 se presenta la arquitectura del simulador del Instituto de Ingeniería de la UNAM, donde se describen los elementos mecánicos que lo constituyen así como la instrumentación mínima que se requiere para operarlo, se cierra el capítulo con la contribución de este trabajo.

El capítulo 3 se centra en describir el diseño electrónico de la instrumentación desarrollada para la MSA, la arquitectura y características de la nueva computadora a bordo así como una descripción del funcionamiento y ventajas sobre la versión anterior.

También, se detallan los nuevos elementos incorporados al simulador como la nueva unidad de medición inercial, los motores sin escobillas y la carga útil.

El capítulo 4 muestra las simulaciones sobre el desempeño de la instrumentación desarrollada y del esquema de control propuesto operando junto al algoritmo de estimación desarrollado previamente en el grupo de trabajo.

El capítulo 5 presenta las conclusiones a partir del análisis de los resultados obtenidos así como el trabajo a futuro que debe atenderse como continuación a este trabajo con la finalidad de afinar y alcanzar los objetivos de la línea de investigación de este proyecto.



---

## Capítulo 2

# Arquitectura del Simulador Satelital MSA

---

### 2.1. Descripción General

En este capítulo se describe primeramente el diseño general de la MSA y posteriormente se hará énfasis en los diferentes elementos, tanto mecánicos como eléctricos, que la conforman.

La MSA es un simulador que busca imitar la dinámica de un satélite en el espacio. La primera versión del simulador que se tiene en el IINGEN se diseñó y construyó para un satélite de 80 [kg] dentro del llamado proyecto SATEX a partir de ello, la instrumentación electrónica, el desarrollo de software especializado, parte del diseño mecánico y la manufactura de varias piezas han ido evolucionando según las observaciones y necesidades del grupo de trabajo.



**Figura 2.1:** Simulador MSA del IINGEN con la instrumentación de [1].

El concepto de diseño es el de una plataforma circular tipo mesa montada sobre un balero suspendido en aire que busca imitar las condiciones de un medio sin fricción y microgravedad a las que se enfrentaría un satélite pequeño en órbita.

Para poner en operación el simulador se requiere del uso de aire comprimido por lo que es altamente recomendable contar con una instalación que provea de aire y donde la compresora se encuentre a varios metros de distancia del simulador y fuera de la zona de

trabajo. Esto tiene una doble finalidad, la primera es evitar el fuerte ruido que genera la compresora al estar funcionando y la segunda y más importante se observó que si se coloca la compresora muy cerca del simulador, las vibraciones que genera el motor al comprimir el aire, llegan hasta el simulador produciendo que la MSA vibre lo que afectará fuertemente su desempeño durante los experimentos. En cambio, si se tiene una instalación de aire comprimido donde la compresora se encuentra fuera de la zona de trabajo, las vibraciones a lo largo de su viaje por la instalación hasta la toma de aire, se atenúan lo suficiente para que al conectar la MSA se observó como logra estabilizarse por si misma.

### **2.2. Elementos y características mecánicas**

A grandes rasgos, la MSA se compone principalmente de cinco elementos mecánicos que le permiten funcionar como una plataforma de pruebas, estos son: la plataforma circular construida de MDF, el balero o semiesfera, la copa con su base, las ruedas inerciales junto a los soportes para los motores y los contrapesos.

#### **2.2.1. Plataforma circular**

En cuanto a la plataforma circular sobre la cual se monta toda la instrumentación electrónica, está construida de un material llamado MDF (Medium Density Fibreboard) de 50 [cm] de diámetro y 1 [cm] de grosor, se eligió este material dado que es muy fácil de trabajar en caso de que se requieran realizar adecuaciones y/o mantenimiento a la MSA como taladrar nuevos orificios para cables, colocar nuevos soportes para motores, pintarla, etc.

#### **2.2.2. Balero**

El otro elemento que forma parte de la estructura mecánica de la MSA es un balero o semiesfera que va fijado mecánicamente a la plataforma de MDF y permite que el simulador se deslice sobre el colchón de aire que se forma entre él y una base llamada copa por donde fluye el aire comprimido.

#### **2.2.3. Copa y base**

La copa es otra pieza clave que forma parte de la estructura mecánica del simulador. Tiene la función de dar soporte a la plataforma circular fijada al balero cuando el aire comprimido no se encuentra fluyendo y de generar los flujos adecuados para formar un



**Figura 2.2:** El balero y la copa del simulador de [25], maquinados en bronce fosforado.

colchón de aire de 0.0254 [mm] de grosor [25], entre el balero y la copa, para estabilizar y mover a la plataforma durante las pruebas. La copa está fijada mecánicamente a una base con una altura de 30 [cm], en el área superior se encuentran 6 perforaciones que forman los ductos milimétricos de 0.55 [mm] que generan el flujo de aire adecuado para lograr suspender en aire la MSA [15].

#### 2.2.4. Ruedas Inerciales

Una rueda inercial es básicamente una masa de metal de forma circular empotrada a la flecha de un motor eléctrico. En su conjunto, tienen la función de transferir momento angular al satélite para modificar su orientación. Existen diferentes esquemas de funcionamiento bajo los cuales se puede operar una rueda inercial, en [2] se puede encontrar una breve descripción de los diferentes modos de operación, en el simulador se utiliza el esquema de ruedas de reacción.

Una rueda de reacción o rueda de momento angular cero es un esquema donde las ruedas inerciales del satélite no se encuentran necesariamente en movimiento. Sólo cuando el satélite requiere corregir su orientación por algún tipo de perturbación o bien se requiere ejecutar algún modo de apuntamiento en particular, es cuando las ruedas son accionadas para alcanzar la orientación deseada.

Actualmente, el simulador MSA posee tres ruedas inerciales de aluminio diseñadas y manufacturadas acorde a la instrumentación y características de los motores empleados en su última versión, se encuentran colocadas ortogonalmente de acuerdo a un sistema de referencia inercial con origen en el centro geométrico del simulador. Estas ruedas poseen una masa de aproximadamente 250[g], un diámetro de 10.8[cm] y una geometría

circular, maquinada en forma de disco interno-anillo externo. Este diseño tiene como propósito que la mayor cantidad de masa se concentre en el anillo externo de la rueda con la finalidad de maximizar el momento angular producido [15].

Para que puedan llevarse a cabo estudios posteriores sobre estrategias de control de orientación, será necesario compaginar esa línea de trabajo con el diseño y manufactura de las nuevas ruedas inerciales pues como parte de la actualización de la MSA abordada en este trabajo de tesis, se consideró el cambio de los motores actuales por nuevos motores sin escobillas. Por lo tanto, el integrante del grupo de trabajo que aborde los problemas correspondientes a diseño mecánico deberá trabajar en conjunto con la persona encargada del análisis y control de la MSA de manera que las ruedas inerciales sean diseñadas considerando la nueva instrumentación, las características de los nuevos motores BLDC y los requerimientos de control que persigan cumplir los objetivos del grupo de trabajo.

### 2.2.5. Contrapesos manuales

El último elemento de tipo mecánico que forma parte de la MSA es un conjunto de contrapesos manuales. Se trata de tres rieles de aluminio fijados a la plataforma de MDF en su parte inferior, en el espacio comprendido entre cada motor. A cada uno de estos rieles se le puede añadir o quitar pequeñas pesas de aluminio en forma de cubo que sirven como contrapesos para estabilizar a la plataforma y contrarrestar los pares gravitacionales, si se modifica algún elemento que forme parte de la instrumentación será necesario volver a "calibrar" la plataforma.

En este tipo de simuladores es muy importante para su caracterización mecánica (por medio de su tensor de inercia), considerar la distribución de masas en el simulador es decir, como está distribuida toda la instrumentación que estará a bordo del simulador. Para ello, una vez que se cuenta con toda la instrumentación es necesario asignar a cada uno de los elementos que la componen una ubicación fija pues es necesario que antes de realizar cualquier tipo de experimento se logre que el centro de masa de la plataforma coincida con su centro geométrico, esto se logra haciendo uso de los contrapesos, ajustando las pequeñas pesas hasta lograr estabilizar a la plataforma en aire junto con toda su instrumentación a bordo. A este proceso se le conoce como balanceo manual y por lo general resulta ser lento y difícil de llevar a cabo, en [25] puede encontrarse una descripción detallada de como realizar este procedimiento.

### 2.2.6. Características y especificaciones

La MSA por su diseño mecánico puede ofrecer los siguientes rangos de maniobrabilidad,  $\pm 360$  [°] en guiñada ( $\psi$ ),  $\pm 20$  [°] en alabeo ( $\phi$ ) y  $\pm 20$  [°] en cabeceo ( $\theta$ ).

Desde su concepción, la MSA ha sido una plataforma flexible donde se busca que los desarrollos en software y hardware sean escalables, actualizables y de fácil mantenimiento. Como se comentó en la sección anterior, la calidad de los sensores, actuadores y sistemas de procesamiento tienen un impacto muy grande en las capacidades de la MSA, originalmente se pensó tener un simulador que pudiese ofrecer una precisión de apuntamiento de  $\pm 5$  [°], sin embargo esta cifra puede mejorarse con la instrumentación adecuada y experimentando con diferentes algoritmos. Queda para otro trabajo de tesis donde se explore el tema de técnicas de control de la MSA, caracterizar al simulador en términos de precisión de apuntamiento.

## **2.3. Instrumentación**

A continuación se describe los elementos que constituyen la instrumentación mínima que debe poseer un simulador satelital de este tipo.

### **2.3.1. Batería**

Dado que todos los sistemas electrónicos deben estar a bordo del simulador se necesita de una batería que provea de los voltajes y corrientes necesarios. Entre las baterías más comunes y que se han utilizado en el IINGEN son las baterías de polímero de litio (LiPo) utilizadas frecuentemente en aeromodelismo precisamente por su larga duración y vida útil.

### **2.3.2. Convertidores DC-DC conmutados**

Debido a que la batería se irá descargando con el uso a lo largo de las pruebas y si además se tiene en cuenta que al alcanzar cierto nivel de descarga debe recargarse de inmediato, surge la necesidad de tener un sistema electrónico que mantenga los niveles de alimentación adecuados para que todos los demás sistemas a bordo de la MSA continúen operando y que alerte sobre el nivel de descarga de la batería para evitar que las celdas puedan llegar a dañarse. Para satisfacer estos requerimientos, se necesita tener a bordo del simulador un sistema de fuentes conmutadas.

Debe señalarse que este sistema electrónico puede verse como una versión simplificada del subsistema de potencia eléctrica de un satélite pequeño por lo que el simulador puede utilizarse como plataforma de validación para otros subsistemas.

### 2.3.3. Unidad de medición inercial

Es indispensable que el simulador tenga a bordo una unidad de medición inercial de al menos 9 grados de libertad esto es, un giróscopo triaxial, un acelerómetro triaxial y un magnetómetro triaxial. Las lecturas de cada uno de estos sensores son necesarias para poder realizar la estimación de la orientación de la MSA, este proceso se realiza por medio de dos algoritmos, TRIAD (Triaxial Attitude Determination) y UKF.

El primero es un algoritmo de fusión de sensores que recibe las lecturas del magnetómetro y el acelerómetro junto con dos vectores de referencia asociados a estos dos sensores. Estos dos vectores de referencia son el vector de aceleración gravitacional y para fines de un simulador en tierra, se utiliza un vector de referencia de campo magnético que corresponda con cierta orientación arbitraria que se desee tomar como referencia, en el caso de un satélite real se toma el vector correspondiente al norte magnético de la tierra. La salida de este algoritmo nos entrega la orientación de la MSA expresada en la forma de un cuaternión y se le conoce como cuaternión medido.

El segundo algoritmo recibe como entrada al cuaternión medido, a las lecturas de velocidad angular provenientes del giróscopo y la información de algún otro sensor complementario como un sensor de sol en el caso de un satélite real. El UKF tiene como finalidad, descrito de forma muy escueta, filtrar la información proveniente de los sensores y reconstruir aquellas señales que lo ameriten como en el caso de un sensor de sol durante la llamada fase de eclipse. La salida del UKF es también un cuaternión, el llamado cuaternión estimado.

Si se hace uso de algún software de simulación numérica para sintonizar y evaluar el desempeño del UKF es posible obtener, a partir de la información que genera el algoritmo, la velocidad angular filtrada del simulador, los ángulos de orientación y generar un gráfico comparativo entre el cuaternión generado por el algoritmo TRIAD y el UKF. Por lo tanto, si se desea implementar un lazo de control será necesario contar con una IMU de buena calidad y con un UKF bien depurado que genere el cuaternión necesario para realimentar al controlador.

### 2.3.4. Comunicaciones

Para el análisis de la información sobre el desempeño de la MSA es necesario contar con dispositivos que puedan transmitir la información hacia una PC con la finalidad de generar gráficos que muestren las posibles señales de interés como pudieran ser los ángulos de giro de la plataforma, su velocidad angular, las señales de control, entre otras. En [1], se utilizaron un par de radio módem XStream-PKG-R que permitían la comunicación entre la computadora a bordo y una PC por medio del protocolo RS-232 lo que facilitó en gran medida su integración. Para trabajos futuros sobre estrategias de control será necesario disponer de dispositivos que transmitan las señales de interés hacia una PC para ello puede utilizarse dispositivos similares a los anteriores o bien,

actualmente se cuenta con un par de módulos de radio XBee que ofrecen diversas ventajas frente a los radio módem como, menor tamaño, menor "peso" y un menor consumo energético (3.3 [V] contra 9 a 18 [V]). Además, la XBee utiliza como interfaz de comunicación el protocolo RS-232 vía UART lo que permite la misma facilidad de integración con la computadora a bordo que los radio módem.

### 2.3.5. Computadora a bordo

La computadora a bordo (OBC, del inglés On Board Computer) es el sistema que se encarga de ejecutar los algoritmos que se buscan validar sobre el simulador. Entre los factores que deben ponderarse en la elección de la computadora están la capacidad de procesamiento, el consumo energético, los periféricos presentes en su arquitectura y el costo. Por ejemplo, actualmente existen microcontroladores de bajo costo, con un consumo de corriente del orden de 15 [mA], con una amplia gama de periféricos como diferentes módulos de comunicación (UART,  $I^2C$ , SPI, CAN, etc.) y con características que se asemejan a las de un DSP como una unidad de punto flotante.

En el caso del IINGEN, se ha experimentado con microcontroladores, procesadores de señales digitales (DSPs), FPGA y recientemente con SoCs cada uno de ellos representativo del objetivo que se buscaba validar en su respectiva época. La elección final de la computadora deberá basarse en los factores antes descritos y en evaluar si el sistema propuesto es capaz de satisfacer los requerimientos del experimento que se busca validar sobre el simulador.

### 2.3.6. Driver para Motores

Se le llama "driver" a la electrónica de potencia necesaria para accionar a los motores de corriente directa, que funciona como interfaz entre las señales de control enviadas por la computadora a bordo y los motores. Esta electrónica puede ir o no acompañada de un microcontrolador donde pueda implementarse un controlador de bajo nivel para el control de velocidad de las ruedas inerciales.

Es muy recomendable optar por un esquema de control distribuido y contar con este segundo lazo que se dedique exclusivamente al control de velocidad de las ruedas en función del par calculado por la computadora a bordo, pues de lo contrario es muy común que se presente el efecto de saturación de la rueda, donde se pierde todo dominio sobre la rueda inercial y presenta un giro incesante sin responder a los comandos de la computadora a bordo.

### 2.3.7. Actuadores

Los motores de corriente directa junto con las ruedas inerciales conforman los actuadores del simulador, tienen la función de mover y orientar al simulador hacia un determinado punto que se elija como objetivo de control. Para ello, se calcula en la computadora a bordo el par que deben generar cada uno de los motores acoplados a la rueda inercial y posteriormente, ese valor se traduce a la velocidad angular necesaria para llevar al simulador a la orientación deseada y cumplir con el objetivo de control.

Para la selección de los motores se debe considerar principalmente la masa total de la instrumentación junto con la masa del simulador que se intenta mover. Estos parámetros, junto con la masa de cada motor y el tiempo en el que se espera completar la maniobra de orientación son también críticos para el diseño de las ruedas inerciales pues se requiere obtener el tensor de inercia del simulador, parámetro que lo caracteriza mecánicamente, al ingresar los parámetros descritos en algún software de diseño mecánico.

Por otro lado, debe mencionarse que el uso de motores con ruedas inerciales no es el único sistema de actuación que se utiliza en satélite pequeños y por consiguiente en este tipo de simuladores. Existen también las llamadas Bobinas de Torque Magnético (BTM), que sirven para estabilizar al satélite y son básicamente solenoides de geometría cilíndrica sobre los que se hace pasar corriente creando un campo electromagnético que busca alinearse con el campo magnético terrestre. Las BTM son construidas a partir de alambre de cobre enrollado de forma consecutiva a lo largo de un núcleo de material ferromagnético.

En muchas misiones satelitales se ha optado por emplear un esquema donde se utilicen ambos sistemas de actuadores, donde las BTM logran estabilizar al satélite con respecto al campo magnético de la Tierra y posteriormente, entren en juego los motores con las ruedas inerciales para realizar los ajustes finos en torno a la orientación del satélite.

Por ello, diversos simuladores en tierra incluyen como parte de su instrumentación BTM colocadas sobre cada eje del sistema coordinado de referencia fijo a la plataforma. Inclusive en algunos casos, para que las BTM puedan desempeñarse adecuadamente es necesario contar con una instalación que posea un generador externo de campo magnético que conste de un arreglo de un par de embobinados concéntricos, conectados en serie entre sí y separados una distancia tal, que puedan colocarse alrededor del simulador. El generador tiene la finalidad de producir un campo magnético coherente, varias veces más intenso que el campo magnético del ambiente, que resulte útil para tareas de estimación de orientación por medio del magnetómetro además de permitir un mejor desempeño de las BTM para realizar tareas de control.

En este tipo de dispositivos, si se manipula la magnitud y dirección de la corriente que circula por los embobinados se puede cambiar el sentido y magnitud del vector de campo magnético producido. Esto tiene un fuerte impacto sobre las BTM pues el

intenso campo magnético producido en el laboratorio les permite generar un par lo suficientemente grande como para vencer los pares gravitacionales que perturban al simulador. Por el contrario, con el campo magnético presente en el ambiente, las BTM no serían capaces de producir dicho par [14].

Se espera que en versiones posteriores de la MSA se cuente con un arreglo triaxial de BTM junto con la instalación necesaria para montar un generador de campo magnético que ayude a las tareas de estimación y que permita validar algoritmos tanto para las BTM como para los motores con ruedas inerciales. De esta forma se contaría con una plataforma de simulación capaz de validar completamente los actuadores del ADCS de un satélite pequeño.

En este trabajo se realizaron aportaciones en cuanto al diseño de nuevas fuentes de alimentación, el diseño y validación de una nueva tarjeta de drivers incorporando un microcontrolador, la puesta en operación de una nueva computadora a bordo y su integración con una nueva unidad de medición inercial así como la propuesta de una estrategia de control que se acopla a los desarrollos previos y su validación por medio de simulaciones numéricas.



---

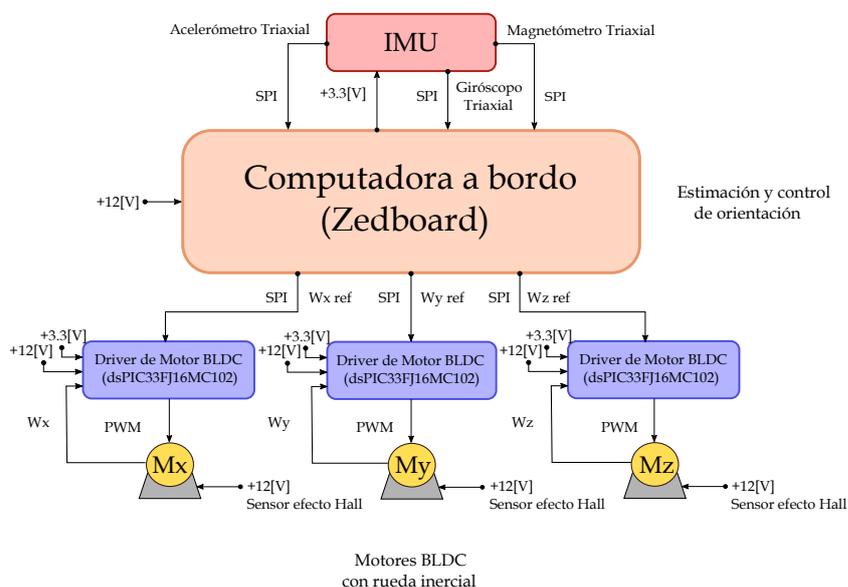
## Capítulo 3

# Actualización de la Instrumentación

---

En este capítulo se detalla el hardware y software desarrollados así como la incorporación de los nuevos elementos como parte de la actualización de la instrumentación de la MSA.

### Instrumentación a bordo



**Figura 3.1:** Diagrama de bloques de la nueva instrumentación del simulador satelital.

Las secciones que se describen están organizadas a modo que representen el diagrama de bloques funcional de un sistema, partiendo del sistema de potencia que alimenta a toda la electrónica, le sigue la unidad de medición inercial que contiene todos los sensores, la computadora a bordo que recibe la información de dichos sensores y ejecuta los algoritmos correspondientes, la información de salida llega al microcontrolador de la

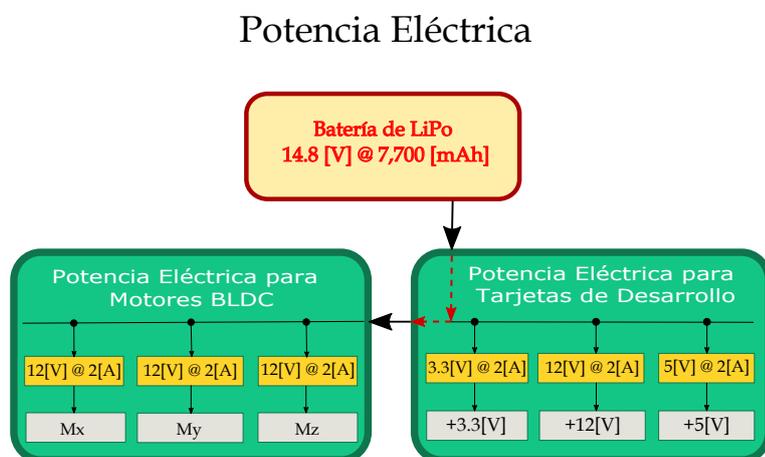
### 3. ACTUALIZACIÓN DE LA INSTRUMENTACIÓN

---

tarjeta de drivers que funciona como un sistema de procesamiento auxiliar además de controlar a la electrónica de potencia presente en la misma tarjeta, las señales enviadas desde la electrónica de potencia llegan al motor, dada su naturaleza en su sección se detalla un poco sobre su funcionamiento y operación. Finalmente, la última sección y bloque corresponde a la carga útil que para fines de este trabajo su tarea se limita a transmitir la imagen que se captura por medio de una cámara hacia una computadora personal para verificar que el *target* de control se ha cumplido.

#### 3.1. Tarjetas de Convertidores DC-DC conmutados

Para el sistema de potencia se consideró las características de la batería con la que cuenta la MSA y los requerimientos eléctricos de los elementos que estarán a bordo del simulador. Por ello, se decidió distribuir el sistema de potencia en dos tarjetas, una que cuente con convertidores DC-DC conmutados que puedan proveer de los niveles de voltaje más comunes 3.3, 5 y 12 [V] a 2 [A] y otra que provea de la alimentación adecuada a los tres motores sin escobillas. Esta segunda tarjeta cuenta con tres convertidores DC-DC donde cada uno proporciona 12 [V] a 2 [A], el diseño de estos convertidores es idéntico y es réplica de la fuente de 12 [V] presente en la primer tarjeta.



**Figura 3.2:** Diagrama de bloques del nuevo sistema de potencia eléctrica.

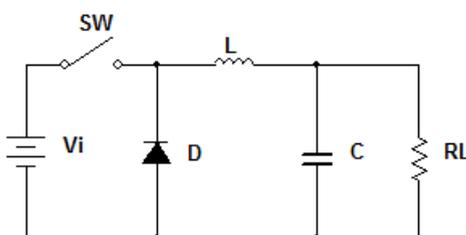
Puede determinarse el tipo de fuentes que se deben implementar a partir de los valores eléctricos de la batería (voltaje de salida de 14.8 [V], voltaje de salida mínimo por descarga de 11 [V] y 7,700 [mAh]) y de los valores de voltaje y corriente que se requiere obtener de las fuentes. Así pues para el caso de las fuentes de 3.3 y 5 [V] se

requiere implementar una fuente tipo Buck o de bajada mientras que para el caso de la fuente de 12 [V] se requiere implementar una fuente tipo SEPIC pues para voltajes de entrada provenientes de la batería mayores a 12 [V] se necesita que la fuente baje el nivel mientras que para voltajes de entrada menores a 12 [V] se requiere que la fuente suba y mantenga el nivel de voltaje en 12 [V].

Antes de describir la topología de las tres fuentes conmutadas a partir de las cuales están construidas las dos tarjetas de potencia, es conveniente revisar brevemente las topologías más elementales a partir de las cuales se comprende la operación de un convertidor DC-DC conmutado y la función que realizan los elementos que la componen.

### 3.1.1. Convertidor DC-DC tipo Buck

La topología más básica de un convertidor DC-DC tipo buck consta de cinco elementos: un switch que representa a un transistor de conmutación, un inductor, un diodo, un capacitor de filtrado a la entrada y un capacitor de filtrado a la salida. Sin embargo, también se requiere de un circuito llamado “amplificador de error y control del switch” que se encarga de sensar el voltaje a la salida, generar una señal de error y enviar una señal generalmente modulada por ancho de pulso que controla la conmutación (el encendido y apagado) del transistor. Por simplicidad en muchos diagramas eléctricos no se incluye este circuito sin embargo es necesario tener presente que se requiere para controlar la acción del transistor.



**Figura 3.3:** Circuito simplificado del convertidor DC-DC tipo Buck.

La descripción que se realiza a continuación toma como referencia la información presentada en [19]. El análisis comienza cuando el switch está abierto (apagado), en este estado no circula corriente hacia el circuito. Cuando el switch se cierra (encendido) la corriente comenzará a fluir hacia el circuito e irá incrementándose, el diodo quedará abierto (apagado) pues estaría polarizado en inversa por lo que la corriente fluirá hacia el inductor. La función principal de un inductor en un circuito es almacenar energía en forma de campo magnético producido por el flujo de corriente a través de sus espiras además de oponerse a los cambios en corriente. Al fluir la corriente por el inductor, se producirá una fuerza electromotriz (FEM) en sus terminales debida a los cambios en la corriente, la FEM se opondrá al voltaje de la batería lo que finalmente

### 3. ACTUALIZACIÓN DE LA INSTRUMENTACIÓN

---

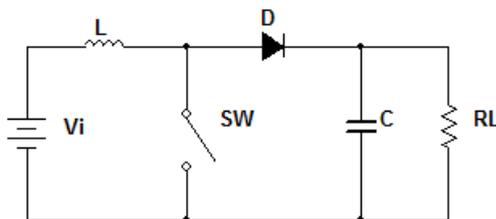
reduce el voltaje en la carga. En el tiempo  $t \rightarrow \infty$ , la variación en la corriente con respecto al tiempo se irá decrementando hasta ser prácticamente cero por lo que el flujo de corriente será constante, esto trae como consecuencia que la FEM también vaya decrementándose hasta ser prácticamente nula, lo que provocaría un incremento en el voltaje en la carga. Si el switch se abre mientras la corriente aún se encuentran cambiando, siempre estará presente la FEM del inductor que se opondrá al voltaje de la batería, resultando en un voltaje de salida menor al voltaje de la fuente. Debe recordarse que la energía almacenada por el inductor en forma de campo magnético es proporcional al flujo de corriente por lo que el inductor almacena un valor de energía constante en cuanto la corriente se vuelve constante.

En la segunda parte del análisis el switch se abre (apagado) por lo que la batería queda desconectada del circuito, esto produce nuevamente un cambio brusco en la corriente a la cual reacciona el inductor produciendo una FEM a causa del campo magnético en decremento, el inductor entonces se convierte en la nueva fuente de alimentación del circuito. La descarga de energía almacenada por el inductor en forma de campo magnético permite mantener la FEM y por ende un flujo de corriente en el circuito hacia la carga. La corriente fluye del inductor al capacitor de filtrado y finalmente se cierra el circuito gracias a la acción del diodo que es polarizado en directa. Finalmente, si el switch se cierra nuevamente (encendido) antes de que el inductor se descargue completamente, el voltaje en la carga siempre será mayor que cero. Resulta evidente la importancia del circuito de “amplificación de error y control” pues es el encargado de sensar continuamente el voltaje de salida y enviar una señal correctiva que controle el encendido y apagado del switch según se requiera, para mantener el voltaje de salida.

#### 3.1.2. Convertidor DC-DC tipo Boost

El convertidor DC-DC tipo boost se conforma por los mismos elementos que el convertidor tipo buck además, la topología que presenta es muy similar con cambios mínimos. En un convertidor tipo boost, el voltaje de salida siempre es mayor que el voltaje de entrada. Al igual que en el caso anterior, el elemento clave que permite modificar el voltaje de salida es el inductor gracias a su tendencia de oponerse a cambios en la corriente al generar un campo magnético.

La descripción del funcionamiento del circuito toma como referencia la información presentada en [18]. El análisis comienza cuando el switch se cierra (encendido) y la corriente fluye del punto de mayor potencial de la fuente hacia el inductor. La variación de la corriente provoca que el inductor genere un campo magnético variante en el tiempo y una FEM que se opone al cambio en el flujo de corriente. Con el tiempo, la corriente se vuelve constante y la FEM se vuelve prácticamente nula y como se asume que el switch es ideal, no hay caída de tensión en él, por lo que puede considerarse que se trata de un alambre. Bajo esta consideración se tendría un “corto circuito” entre esas dos terminales donde,  $R_{sw} = 0 \Rightarrow i \rightarrow \infty$ , por lo que la corriente fluiría únicamente



**Figura 3.4:** Circuito simplificado del convertidor DC-DC tipo Boost.

en la primer malla. Además, debe notarse que la polaridad del inductor es tal que se opone al flujo de corriente es decir con su terminal izquierda de signo positivo.

Cuando el switch se abre (apagado), se tiene un circuito abierto por lo que la impedancia entre sus terminales es muy alta lo que disminuye considerablemente la corriente en el circuito. El inductor para oponerse al cambio en la corriente y mantener el flujo hacia la carga, destruye su campo magnético que había logrado formar y cambia su polaridad, el inductor ahora tiene su terminal derecha positiva. Bajo esta condición, se tendrían dos fuentes de voltaje en serie (la fuente de entrada y la FEM del inductor), lo que eleva el voltaje en la carga a través del diodo D y carga al capacitor de salida.

Si el switch se conmuta lo suficientemente rápido, el inductor no se descargará completamente entre las dos etapas de carga definidas por el estado del switch y la carga siempre verá un voltaje mayor al de la fuente de entrada. Para lograr lo anterior, el capacitor de salida juega un papel fundamental, mientras el switch está abierto el capacitor se carga con el voltaje resultante del inductor y la fuente. Cuando el switch se cierra y se tiene un “corto circuito” entre sus terminales, el capacitor es el encargado de mantener el nivel de voltaje en la carga al proveer de potencia. Por su parte, el diodo realiza una función de bloqueo pues evita que el capacitor se descargue a través del switch, este último debe abrirse nuevamente lo suficientemente rápido para evitar que el capacitor se descargue demasiado.

### 3.1.3. Convertidor DC-DC tipo SEPIC

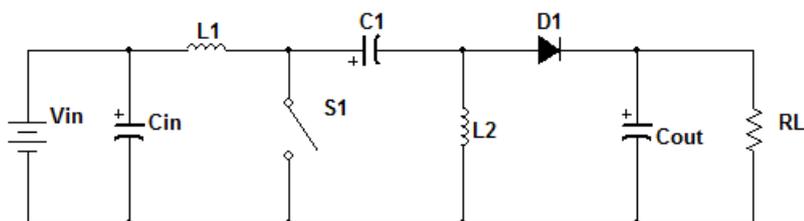
El convertidor DC-DC tipo SEPIC se construye a partir de un convertidor tipo boost junto con un convertidor tipo buck-boost. Un convertidor tipo buck-boost está construido a partir de las topologías de estos dos circuitos y tiene como finalidad reducir o elevar el voltaje de entrada hasta un nivel requerido como voltaje de salida, la principal desventaja de esta fuente es que invierte la señal de voltaje de salida con respecto a la entrada.

El convertidor SEPIC es muy similar a un convertidor tipo buck-boost pues permite tener un voltaje en su salida mayor que, menor que o igual que el voltaje de entrada, con la ventaja de que no invierte el voltaje de salida. Además, incorpora un elemento

### 3. ACTUALIZACIÓN DE LA INSTRUMENTACIÓN

---

adicional, un capacitor de acoplamiento que ayuda a transferir la energía entre los dos inductores de forma similar al núcleo de un transformador. Este capacitor de acoplamiento provee de una vía de baja impedancia para que la corriente del inductor pueda fluir del inductor de entrada (primario) al diodo y finalmente a la salida o desde el inductor de salida (secundario) de vuelta a través del switch y a tierra. La selección de este capacitor debe hacerse considerando una gran corriente RMS con respecto a la potencia de salida de la fuente, la incorporación al circuito de un capacitor que cumpla con el requerimiento de corriente RMS produce un voltaje de rizo pequeño por lo que se logra un voltaje pico cercano al voltaje de entrada de la fuente.



**Figura 3.5:** Circuito simplificado del convertidor DC-DC tipo SEPIC.

El análisis de la operación del convertidor que se presenta a continuación está basado principalmente en el artículo [17]. El convertidor SEPIC logra su propósito gracias al flujo de energía entre los capacitores e inductores. La cantidad de energía que se intercambia es controlada por un switch que en la práctica resulta ser un transistor, típicamente un MOSFET. Un convertidor SEPIC opera en “modo continuo” si la corriente que circula a través del inductor  $L_1$  nunca es cero. Si dicha corriente llega a ser cero, entonces se dice que opera en “modo discontinuo”.

Los voltajes presentes en el circuito pueden escribirse como:

$$V_{in} - V_{L1} - V_{C1} - V_{L2} = 0 \quad (3.1)$$

En estado estacionario, el voltaje en el capacitor  $C_1$  es igual que el voltaje de entrada lo que implica que:

$$V_{L1} = -V_{L2} \quad (3.2)$$

Esta última expresión nos indica la razón por la cual los dos inductores pueden enrollarse en el mismo núcleo pues si los voltajes son iguales en magnitud, los efectos de la inductancia mutua resultan ser nulos, asumiendo que la polaridad de los embobinados sea correcta. Además, dado que los voltajes son iguales en magnitud, las corrientes de rizo de los dos inductores también resultan ser iguales en magnitud.

Por el lado de las corrientes, estas pueden escribirse como sigue:

$$I_{D1} = I_{L1} - I_{L2} \quad (3.3)$$

Con los comentarios respectivos sobre el propósito del convertidor puede hablarse ahora de su análisis. El análisis comienza considerando únicamente a los voltajes de bias en el circuito y al switch  $S_1$  abierto (apagado). Cuando el switch  $S_1$  se cierra (encendido) se forma un “corto circuito” en sus terminales por lo que la corriente que circula por el inductor  $L_1$  ( $I_{L1}$ ) se incrementa al igual que la corriente en el segundo inductor  $L_2$  ( $I_{L2}$ ) con la diferencia de que esta última, fluye en sentido contrario. La potencia necesaria para incrementar la corriente en el inductor ( $I_{L1}$ ) proviene del voltaje de entrada. Mientras el switch permanezca cerrado, el voltaje en el capacitor  $C_1$  ( $V_{C1}$ ) es aproximadamente igual al voltaje de entrada y el voltaje en el segundo inductor ( $V_{L2}$ ) será igual al voltaje de entrada al circuito con polaridad opuesta ( $-V_{in}$ ). El capacitor  $C_1$  se encarga de suministrar la potencia para incrementar la corriente en el segundo inductor ( $I_{L2}$ ) y por ende la energía almacenada en su campo magnético generado por las variaciones en corriente.

Cuando el switch  $S_1$  se abre (apagado), la corriente en el capacitor  $C_1$  ( $I_{C1}$ ) pasa a ser la misma que en el inductor  $L_1$  ( $I_{L1}$ ) debido a la característica ya mencionada del inductor de no permitir cambios instantáneos en la corriente. Bajo esta condición del switch, la corriente a través del inductor  $L_1$  ( $I_{L1}$ ) y la corriente a través del inductor  $L_2$  ( $I_{L2}$ ) se sumarán para incrementar la corriente que se entrega a la carga.

Utilizando la ley de corrientes de Kirchhoff puede verse que:

$$I_{D1} = I_{C1} - I_{L2} \quad (3.4)$$

De esta última expresión, puede verse que mientras el switch  $S_1$  esté abierto (apagado), la potencia se entrega a la carga por medio de los inductores  $L_1$  y  $L_2$ . En cuanto al capacitor  $C_1$ , este se carga a partir del inductor  $L_1$  durante el periodo de tiempo en que el switch esté abierto (apagado) y se descarga en el inductor  $L_2$  durante el periodo de tiempo en que el switch esté cerrado (encendido). Debido a que la polaridad del voltaje en el capacitor  $C_1$  puede cambiar con cada ciclo de encendido/apagado del switch es recomendable emplear un capacitor no polarizado.

El capacitor de entrada  $C_{in}$  se requiere para filtrar cualquier tipo de señal ruidosa a la entrada, de hecho este capacitor no sólo tiene efecto en la entrada sino que el voltaje de rizo a la salida depende fuertemente del filtrado realizado por los capacitores de entrada y salida.

En resumen, el convertidor SEPIC está construido a partir de un convertidor boost y un convertidor buck-boost. Su capacidad para operar como un convertidor buck-boost se debe al capacitor  $C_1$  y al inductor  $L_2$ . Por su parte, el inductor  $L_1$  y el switch  $S_1$  forman un convertidor boost que produce un voltaje ( $V_{S1}$ ) que resulta más grande que el voltaje de entrada ( $V_{in}$ ) y cuya magnitud está determinada por el ciclo de trabajo

### 3. ACTUALIZACIÓN DE LA INSTRUMENTACIÓN

---

del switch  $S_1$ . Dado que el voltaje promedio en el capacitor  $C_1$  es igual al voltaje de entrada ( $V_{in}$ ), el voltaje de salida ( $V_{out}$ ) puede escribirse como:

$$V_{out} = V_{S1} - V_{in} \quad (3.5)$$

De esta última expresión puede comentarse que si el voltaje  $V_{S1}$  es más grande que el doble del voltaje de entrada ( $V_{in}$ ) entonces, el voltaje de salida será más grande que el voltaje de entrada. Por otra parte, si el voltaje  $V_{S1}$  es menor que el doble del voltaje de entrada ( $V_{in}$ ) entonces, el voltaje de salida será menor que el voltaje de entrada.

#### 3.1.4. Fuente de 3.3 [V] a 2[A]

Se requiere contar con una fuente de 3.3 [V] a 2[A] en una de las tarjetas de potencia eléctrica pues es común que diversos sistemas electrónicos requieran ese valor para su funcionamiento. A partir de las características de la batería es evidente que se requiere bajar su voltaje por medio de un convertidor DC-DC para obtener y mantener los 3.3 [V] se necesita entonces implementar un convertidor tipo buck para cumplir con dichos requerimientos.

Para la implementación de la fuente se decidió hacerlo en torno al circuito integrado LTC1624 de la compañía *Linear Technology*, se trata de un controlador/regulador que se encarga de la conmutación de un MOSFET de potencia externo de canal N y está pensado para utilizarse en los convertidores conmutados más comunes. Se seleccionó el LTC1624 debido a que opera para un amplio rango de voltajes de entrada desde 3.5 [V] hasta 36 [V], su frecuencia de operación de 200 [kHz] se encuentra ajustada internamente sin embargo, el valor de corriente que se espera obtener del circuito es configurable a través de la selección de un resistor de sentido de corriente. El LTC1624 se encuentra disponible en un pequeño encapsulado SO-8 haciéndolo ideal para aplicaciones donde el espacio es limitado como en teléfonos celulares, notebooks, cargadores de baterías, entre otras. En la hoja de especificaciones del LTC1624 [7] pueden hallarse algunas de las otras características relevantes sobre este CI; más importante aún en dicho documento pueden encontrarse los diferentes circuitos que pueden construirse en torno a este CI así como las ecuaciones de diseño para el cálculo de los elementos en cada uno de estos casos. El diseño de la fuente de 3.3 [V] que se implementó en una de las tarjetas de potencia pertenece a una de las topologías que se pueden construir en torno al CI seleccionado y se encuentra en dicho documento.

En los párrafos siguientes se introduce la topología seleccionada para la fuente y se describen los elementos que la componen así como la función que desempeñan. En la figura 3.6, se muestra el esquemático del circuito en torno al cual se construyó la fuente de 3.3 [V] la cual se encuentra en una de las tarjetas de potencia.

Como puede verse, la topología es bastante sencilla lo que permite su implementación en espacios compactos y cuenta además con todos los elementos para operar como

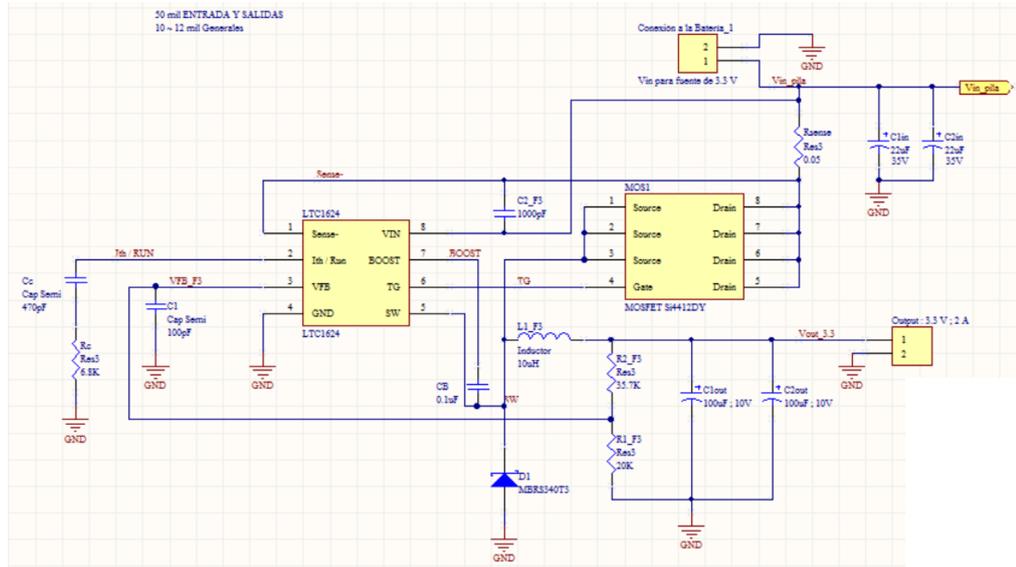


Figura 3.6: Esquemático de la fuente conmutada de 3.3 [V] a 2 [A].

un convertidor DC-DC. Partiendo desde donde ingresa la señal de voltaje de entrada, pueden observarse dos capacitores de 22 [ $\mu F$ ] a 35 [V] en paralelo, estos corresponden a los capacitores de entrada. Estos capacitores, como se comentó, filtran cualquier ruido presente en la señal de entrada y ayudan junto con los capacitores de salida a disminuir el voltaje de rizo a la salida. Esta señal filtrada viaja hacia el pin  $V_{in}$  del LTC1624 y sirve como voltaje de alimentación al CI. Dicha señal también pasa por un resistor de sentido de 0.05 [ $\Omega$ ] cuyo valor dicta la magnitud de corriente que puede entregar el circuito a una carga:

$$R_{sense} = \frac{100[mV]}{I_{max}} \quad (3.6)$$

Puede colocarse un capacitor no polarizado de 1,000 [pF] para suavizar las señales que entran al LTC1624 como se observa en el esquemático. El MOSFET cuya tarea es conmutar para regular los ciclos de carga y descarga del inductor, es controlado por el CI al conectar el pin trigger (TG) al *gate* del MOSFET. Por su parte el *drain* se conecta a la terminal del resistor de sentido que va al pin *sense*- del CI, el *source* está conectado al nodo donde convergen el cátodo del diodo, una de las terminales del inductor y el pin *switch* (SW) del CI, de hecho este nodo recibe el nombre de “conexión del nodo de conmutación al inductor”. En el circuito se emplea un diodo Schottky cuya función es conducir únicamente durante el ciclo de apagado del transistor, como se explicó a detalle anteriormente. Para la selección del diodo es necesario tener en mente ciertos parámetros tales como la corriente pico y la potencia disipada promedio, la condición más estresante para el diodo se presenta bajo un “corto circuito” a la salida ( $V_{out} = 0$ ).

### 3. ACTUALIZACIÓN DE LA INSTRUMENTACIÓN

---

Bajo condiciones normales de carga, la corriente promedio que conduce el diodo puede calcularse como:

$$I_{D(AVG)} = I_{load(AVG)} \left[ \frac{V_{in} - V_{out}}{V_{in} + V_D} \right] \quad (3.7)$$

Para el diseño de la tarjeta en el caso del diodo se recomienda en la hoja de especificaciones tener pistas cortas y verificar que el diodo se encuentre correctamente aterrizado por medio de un plano de tierra, de esta manera se evitan efectos indeseables como el efecto de *ringing* y un incremento en la disipación.

El inductor es otro de los elementos clave para la operación de la fuente, el cálculo del valor apropiado es de importancia pues el valor del inductor tiene efectos sobre la corriente de rizo y la operación del convertidor a baja corriente. La frecuencia de operación fija a 200 [kHz] del LTC1624 permite que se puedan emplear valores pequeños para el inductor. En la hoja de especificaciones del CI se sugiere utilizar valores de inductancia de entre 5 [ $\mu H$ ] a 68 [ $\mu H$ ], dependiendo del voltaje de entrada máximo y en la corriente de salida de la fuente. Para calcular el valor de inductancia debe considerarse la corriente de rizo del inductor. La corriente de rizo del inductor  $\Delta I_L$  se decrementa con valores de inductancia más altos y se incrementa con voltajes de entrada ( $V_{in}$ ) y salida ( $V_{out}$ ) mayores. Un buen punto de partida para aproximar su valor es mediante la siguiente expresión:

$$\Delta I_L = 0.4(I_{max}) \quad (3.8)$$

Debe tenerse en mente que la corriente de rizo máxima ocurre con el voltaje de entrada máximo. Con esta aproximación puede emplearse la siguiente expresión para calcular el valor más apropiado para el inductor:

$$\Delta I_L = \frac{V_{in} - V_{out}}{f \cdot L} \left( \frac{V_{out} + V_D}{V_{in} + V_D} \right) \quad (3.9)$$

En la expresión anterior,  $f$  corresponde a la frecuencia de operación y  $V_D$  corresponde al voltaje en polarización directa del diodo Schottky de salida. Una vez que se conoce el valor del inductor es necesario seleccionar el tipo material más adecuado para el núcleo. En la hoja de especificaciones del CI puede encontrarse una pequeña discusión sobre los materiales más apropiados para el núcleo de un inductor que será utilizado en un convertidor de alta eficiencia. Se discuten aquellos materiales que minimicen las pérdidas de potencia en el núcleo, su costo y algunos otros parámetros a considerar para algunos de ellos.

El voltaje de salida del convertidor se puede ajustar con el uso de dos resistores a modo de divisor de tensión. Mediante la siguiente expresión se puede programar el voltaje de salida que el diseñador desee:

$$V_{out} = 1.19[V] \left( 1 + \frac{R_2}{R_1} \right) \quad (3.10)$$

La señal de voltaje de  $R_1$  se utiliza para realimentar al LTC1624 de forma que posea la información para controlar la conmutación del MOSFET y mantener el voltaje de salida. Un capacitor no polarizado de 100 [pF] se utiliza para filtrar la señal de realimentación por lo que se conecta entre los pines  $V_{FB}$  y GND del CI.

El CI posee un pin llamado  $I_{TH}/Run$  al cual se conecta un capacitor no polarizado  $C_c$  en serie con un resistor  $R_c$ . Este pin desempeña dos funciones principales, permitir una respuesta al escalón óptima hacia la carga por medio de un capacitor de compensación  $C_c$  y dotar de una forma de apagar por completo al LTC1624. El CI posee internamente una fuente de corriente de  $2.5[\mu A]$  que carga al capacitor  $C_c$ , cuando el voltaje en el pin  $I_{TH}/Run$  alcanza los  $0.8[V]$  el CI comienza a operar, en ese momento el amplificador de error (interno) lleva al pin a su voltaje máximo de  $2.4[V]$ . También, es posible implementar un modo de arranque suave por medio de este pin lo que ayudaría a reducir la llamada corriente de arranque proveniente del voltaje de entrada  $V_{in}$ .

En su configuración de fuente de bajada se requiere conectar un capacitor no polarizado entre los pines BOOST y SW del CI, este capacitor recibe el nombre de capacitor de BOOST ( $C_B$ ). Este capacitor tiene la función de suministrar el voltaje para controlar el gate del MOSFET. El capacitor se carga a través de un diodo interno a partir del voltaje de alimentación INTVcc cuando el pin SW está en bajo. Cuando el MOSFET debe prenderse un driver interno coloca la señal de voltaje del capacitor de BOOST en el gate. En la hoja de especificaciones se garantiza que para la mayoría de aplicaciones un capacitor de BOOST con un valor de  $0.1 [\mu F]$  es suficiente.

Finalmente, se incorporan al diseño los capacitores de salida  $C_{out}$ , se trata de dos capacitores polarizados de  $100 [\mu F]$  a  $10 [V]$  cada uno. Como ya se ha comentado los capacitores de salida tienen varios propósitos, filtrar lo que será la señal de voltaje de salida de la fuente y disminuir el voltaje de rizo de salida. El capacitor de salida se elige con base en la resistencia serie efectiva (*effective series resistance*, ESR) buscando disminuir el voltaje de rizo a la salida. En la hoja del CI, se pueden encontrar diversas sugerencias sobre los principales fabricantes de capacitores dependiendo del tipo que se vaya a utilizar.

### 3.1.5. Fuente de 5 [V] a 2 [A]

Para la fuente de  $5 [V] / 2[A]$  se decidió construirla en torno al circuito integrado LTC3864 de la compañía *Linear Technology*, este circuito es un controlador DC/DC especialmente para fuentes de bajada. El circuito se encarga de controlar la conmutación de un MOSFET de potencia canal P. Este circuito es apto para un amplio rango de voltajes de entrada y salida, se eligió este CI por sobre el LTC1624 debido a que ofrece



que puede entregar la fuente. Para calcular el valor adecuado del resistor de sensado y obtener la corriente de salida máxima deseada, se utiliza la siguiente expresión:

$$R_{sense} = \frac{95[mV]}{\Delta I_L \left( I_{out(max)} + \frac{\Delta I_L}{2} \right)} \quad (3.11)$$

En la expresión anterior,  $\Delta I_L$  corresponde a la corriente de rizo del inductor y se calcula de la siguiente manera:

$$\Delta I_L = \left( \frac{V_{out}}{f \cdot L} \right) \left( 1 - \frac{V_{out}}{V_{in(max)}} \right) \quad (3.12)$$

De esta última expresión,  $f$  corresponde a la frecuencia de operación del CI y se encuentra ajustada a 350[kHz]. Por su parte,  $L$  corresponde al valor calculado para el inductor y se obtiene de la siguiente expresión:

$$L = \left( \frac{V_{out}}{f \cdot \Delta I_L(max)} \right) \left( 1 - \frac{V_{out}}{V_{in(max)}} \right) \quad (3.13)$$

Para esta última expresión, la corriente de rizo del inductor (máxima) puede estimarse, considerando el peor de los casos, como el 60 % de la corriente de salida máxima de la fuente, bajo estas consideraciones el valor del inductor de la fuente resulta ser de 10 [ $\mu H$ ].

Para la selección del MOSFET, deben tenerse en cuenta ciertos parámetros relevantes como son el voltaje de ruptura del *drain* al *source*  $V_{BR(DSS)}$ , el voltaje límite del *gate* al *source*  $V_{GS(TH)}$ , la resistencia de encendido entre el *drain* y el *source*  $R_{DS(on)}$ , la corriente máxima del *drain*  $I_{D(max)}$  junto con los parámetros de resistencia térmica del MOSFET, la resistencia térmica de la juntura al ambiente  $\theta_{ja}$  y la resistencia térmica de la juntura al encapsulado  $\theta_{jc}$ . En el diseño de la fuente se hace uso del MOSFET Fairchild FDMC5614P que posee las siguientes características:  $V_{BR(DSS)} = 60[V]$ ,  $I_{D(max)} = 5.7[A]$ ,  $R_{DS(on)} = 105[m\Omega]$ ,  $\theta_{ja} = 60[\frac{^{\circ}C}{W}]$ . Si se requiere, en la hoja de especificaciones pueden encontrarse las ecuaciones para estimar la potencia disipada y la temperatura de la juntura en el MOSFET al considerar que la fuente opera a su máxima capacidad.

Con base en los requerimientos de potencia se utiliza el diodo Schottky SBR3U100LP. Se trata de un diodo de conmutación que tiene la finalidad de conducir la corriente proveniente del inductor cuando el MOSFET se encuentra en estado de apagado. Entre los parámetros a considerar para su selección están la corriente en polarización directa, el voltaje en polarización directa y la potencia disipada. Debe tenerse presente que el peor de los casos para el diodo ocurre durante la condición de “corto circuito” donde el diodo debe manejar la corriente máxima.

### 3. ACTUALIZACIÓN DE LA INSTRUMENTACIÓN

---

Se coloca un resistor de  $100 [k\Omega]$  entre el nodo a la salida del inductor y el pin PGOOD. PGOOD es un pin de salida de “drenaje” abierto que es llevado a tierra por medio de un MOSFET interno de canal N cuando la señal de voltaje en el pin  $V_{FB}$  está fuera de una ventana de  $\pm 10\%$  de tolerancia con respecto a la referencia interna de voltaje de  $0.8 [V]$ . Cuando el voltaje en el pin  $V_{FB}$  está dentro de la ventana de tolerancia, se apaga al MOSFET y el pin PGOOD se conecta a una fuente de voltaje no mayor a  $6 [V]$  a través del resistor de  $100 [k\Omega]$ ,  $R_{PGD}$ .

El voltaje de salida queda determinado por un par de resistores, para programar el voltaje de salida de la fuente se utiliza la siguiente expresión:

$$V_{out} = 0.8[V] \left( 1 + \frac{R_{FB2}}{R_{FB1}} \right) \quad (3.14)$$

Para mejorar la respuesta transitoria del voltaje de salida se añade un capacitor de  $47 [pF]$  llamado  $C_{FF}$ , entre las líneas de  $V_{out}$  y  $V_{FB}$ .

A la salida de la fuente se añaden dos capacitores cerámicos de  $47 [\mu F]$  para el filtrado de la señal de voltaje de salida, con ello se busca disminuir el voltaje de rizo a la salida.

A la izquierda del CI se observan varios elementos en el esquemático. El primero de ellos es un capacitor de  $0.1 [\mu F]$  llamado  $C_{ss}$  colocado entre el pin SS y GND. Al colocar este capacitor en el pin SS se habilita el modo de “arranque suave” que permite que el voltaje de salida se eleve suavemente desde  $0[V]$  hasta el valor final programado.

Se observa también un pin llamado ITH. ITH es la salida del amplificador de error, dicha señal puede aprovecharse para estimar ciertos parámetros de desempeño sobre la fuente como el ancho de banda, el factor de amortiguamiento, el margen de fase, entre otros. Según la hoja de especificaciones, a este pin deben conectarse un resistor y un capacitor llamados  $R_{ITH}$  y  $C_{ITH1}$ . Estos elementos tienen un impacto en la respuesta transitoria de la fuente ante un amplio rango de cargas y capacitores de salida además su tarea es también optimizar la respuesta en lazo cerrado de la fuente, esto si se le estudia como un sistema de control. Un capacitor llamado  $C_{ITH2}$  colocado entre el pin ITH y GND se utiliza para atenuar el ruido de alta frecuencia. Para  $R_{ITH}$ ,  $C_{ITH1}$  y  $C_{ITH2}$  se utilizan directamente los valores sugeridos en la hoja de especificaciones.

#### 3.1.6. Fuente de $12 [V]$ a $2[A]$

La fuente de  $12 [V]$  está construida como un convertidor tipo SEPIC, esto debido a que la batería que alimenta la instrumentación a bordo del simulador al descargarse puede alcanzar valores inferiores a  $12[V]$ , por lo que se requiere que el convertidor eleve el voltaje en dichos casos. El diseño de la fuente de  $12 [V]$  se basa en el circuito integrado LT3757A, se trata de un controlador DC/DC con la capacidad de producir un voltaje de salida positivo o negativo. Este CI fue pensado para la construcción de



### 3. ACTUALIZACIÓN DE LA INSTRUMENTACIÓN

---

$$V_{off} = 1.22 \left( \frac{R_3 + R_4}{R_4} \right) \quad (3.15)$$

$$V_{on} = (2[\mu A] \cdot R_3) + V_{off} \quad (3.16)$$

El pin SYNC del CI es el pin de sincronización de frecuencia y puede utilizarse cuando se desea sincronizar la frecuencia de conmutación con un reloj externo. Si dicha característica no se utiliza entonces el pin debe conectarse a tierra como se observa en el esquemático.

Para ajustar la frecuencia de conmutación se tiene disponible el pin RT. La frecuencia de conmutación se ajusta mediante un resistor conectado entre el pin RT y GND, bajo ningún motivo debe dejarse este pin “abierto”. Para la selección del resistor  $R_T$ , en la hoja de especificaciones puede encontrarse una pequeña tabla donde se presentan los diferentes valores para  $R_T$  junto con la frecuencia de conmutación que se estaría seleccionando para dicho valor. Para la elección de una determinada frecuencia de conmutación se debe considerar la disipación de potencia del CI, el tamaño resultante de los componentes y la eficiencia que se desea obtener del circuito. Operar a frecuencias más bajas mejora la eficiencia sin embargo, obliga a utilizar inductores de dimensiones físicas mayores [9]. En el diseño de la fuente se emplea un resistor ( $R_T$ ) de 41.2 [k $\Omega$ ] lo que resulta en una frecuencia de conmutación de 300 [kHz].

Como en el caso de los otros CI, el LT3757A cuenta con el pin SS (*soft-start*) lo que permite programar por medio del capacitor  $C_{ss}$  un tiempo de arranque. Esta característica resuelve varios inconvenientes como los picos de corriente a causa de la conmutación y los sobrepasos en el voltaje de salida durante el arranque del CI o en la recuperación de una condición de falla [9]. El propósito principal de esta característica es evitar daño a los componentes externos al CI y a la carga [9]. En la hoja de datos se discuten con detalle las causas detrás de estos inconvenientes y como el CI a través del pin SS junto al capacitor  $C_{ss}$  logran subsanar estos efectos.

El intervalo de tiempo para el arranque suave del CI queda determinado por el valor del capacitor  $C_{ss}$ , para ello se usa la siguiente expresión:

$$T_{ss} = C_{ss} \left( \frac{1.25[V]}{10[\mu A]} \right) \quad (3.17)$$

El CI posee un pin de nombre Vc cuya función está ligada al lazo de compensación que determina la estabilidad y rendimiento durante el transitorio del convertidor. En la hoja del CI se explica que para compensar el lazo de realimentación se utiliza un resistor y un capacitor conectados en serie al pin Vc y tierra. También, se sugiere que para cualquier diseño en particular se tomen los valores del circuito más parecido (de entre los que se presentan en la hoja de especificaciones) al diseño que se pretende construir y se ajusten dichos valores hasta obtener el desempeño deseado. En el diseño

seleccionado para la fuente, el resistor tiene un valor de 10 [kΩ] y el capacitor es de 6.8 [nF].

Puede notarse un pin de nombre INTVcc, este pin corresponde a la salida de un regulador de voltaje interno de 7.2 [V] que alimenta parte de la circuitería interna del integrado. Por ende, es necesario conectar un capacitor que actúe como capacitor de *bypass* o de filtrado entre el pin del regulador INTVcc y tierra, en la hoja de especificaciones se sugiere emplear un capacitor cerámico de mínimo 4.7[μF].

Al igual que los otros integrados, el LT3757A mide la corriente del MOSFET por medio de un resistor de sensado, conectado entre el *source* del MOSFET y tierra. Para calcular el valor apropiado del resistor de sensado se utiliza la siguiente expresión:

$$R_{sense} = \frac{80[mV]}{I_{SW(pico)}} \quad (3.18)$$

En la ecuación anterior,  $I_{SW(pico)}$  corresponde a la corriente de conmutación pico. La corriente de conmutación pico puede conocerse por medio de la siguiente expresión:

$$I_{SW(pico)} = \left(1 + \frac{\chi}{2}\right) \cdot I_{out(max)} \cdot \left(\frac{1}{1 - D_{max}}\right) \quad (3.19)$$

De la expresión anterior,  $\chi$  corresponde al porcentaje de corriente de rizo pico a pico en la conmutación y  $D_{max}$  es el ciclo de trabajo máximo que ocurre cuando se tiene el voltaje de entrada mínimo. Estos dos parámetros pueden calcularse como sigue:

$$\chi = \frac{\Delta V_{sense}}{80[mV] - (0.5 \cdot \Delta V_{sense})} \quad (3.20)$$

$$D_{max} = \frac{V_{out} + V_D}{V_{in(min)} + V_{out} + V_D} \quad (3.21)$$

Es recomendable que  $\chi$  se fije en un valor de entre 0.2 a 0.4 y con ello estimar el voltaje de rizo en la resistencia de sensado ( $\Delta V_{sense}$ ). Por otra parte, en la última expresión,  $V_D$  corresponde al voltaje de polarización directa del diodo. La resistencia de sensado que se emplea en el diseño tiene un valor de 0.01 [Ω].

Para la selección del MOSFET, debe elegirse aquél con un valor de voltaje de DC ( $V_{DC}$ ) mayor que la suma del voltaje de entrada y el voltaje de salida considerando un margen de diferencia de 10 [V] entre dichos valores [9]. En el diseño del convertidor presente en la hoja de especificaciones, se sugiere el modelo y fabricante de un MOSFET en particular.

Un convertidor SEPIC exige el uso de dos inductores para cumplir con su cometido. Los llamados inductores acoplados pueden encontrarse en un mismo encapsulado tipo SMD y resultan ideales para este tipo de aplicaciones. En la hoja de especificaciones

### 3. ACTUALIZACIÓN DE LA INSTRUMENTACIÓN

---

se recomienda un modelo en particular para emplearse como el par de inductores que requiere el convertidor. Los parámetros que deben considerarse para la selección del inductor en un convertidor SEPIC son el rango de voltajes de entrada al convertidor, la frecuencia de operación y la corriente de rizo del inductor. Con estos parámetros definidos y considerando que se trata de dos inductores de igual magnitud, enrollados en el mismo núcleo, la inductancia se calcula con la siguiente expresión:

$$L = D_{max} \left[ \frac{V_{in(min)}}{\Delta I_{SW} \cdot f} \right] \quad (3.22)$$

Para seleccionar el capacitor de acoplamiento  $C_{DC}$  se debe considerar al voltaje de DC y a la corriente RMS del capacitor. El voltaje de DC del capacitor debe ser mayor que el voltaje de entrada máximo, esto es:

$$V_{CDC} > V_{in(max)} \quad (3.23)$$

Por su parte, la corriente RMS del capacitor de acoplamiento debe cumplir con:

$$I_{RMS(CDC)} > I_{out(max)} \sqrt{\frac{V_{out} + V_D}{V_{in(min)}}} \quad (3.24)$$

Para un convertidor SEPIC, en la hoja de datos del CI, se recomienda utilizar dos capacitores cerámicos de 4.7 [ $\mu F$ ] conectados en paralelo para formar al capacitor de acoplamiento. De igual manera, se puede encontrar una recomendación sobre un modelo y su fabricante para ser empleado en los capacitores de acoplamiento.

Por otra parte, se requiere de un diodo de rápida conmutación con una baja caída de voltaje en polarización directa y una baja corriente de fuga en polarización inversa para operar como el diodo de salida [9]. Cabe mencionar que la corriente promedio que circula por el diodo en polarización directa bajo condiciones de operación normales es igual a la corriente de salida, mientras que la corriente pico puede calcularse como:

$$I_{D(pico)} = \left(1 + \frac{\chi}{2}\right) \cdot I_{out(max)} \cdot \left(\frac{1}{1 - D_{max}}\right) \quad (3.25)$$

Otro criterio que debe tomarse en cuenta para elegir al diodo en un convertidor SEPIC es el voltaje inverso de pico repetitivo ( $V_{RRM}$ ). En la hoja del CI se sugiere que el voltaje inverso de pico repetitivo sea mayor que la suma del voltaje de salida y el máximo voltaje de entrada ( $V_{out} + V_{in(max)}$ ) por un margen de seguridad de 10 [V].

Para ajustar el voltaje de salida del convertidor, al igual que en los casos anteriores, se hace por medio de un divisor de tensión formado por dos resistores,  $R_1$  y  $R_2$ . La expresión para fijar el voltaje de salida es la siguiente:

$$V_{out} = 1.6[V] \left( 1 + \frac{R_2}{R_1} \right) \quad (3.26)$$

Según la hoja de datos,  $R_1$  puede tomar un valor máximo de 158 [ $k\Omega$ ]. Esto se debe a que el valor de los resistores se elige de tal forma que la señal de error que se genera a partir de la corriente que entra al pin FBX sea menor al 1 %, esta condición se logra siempre que  $R_1$  tenga un valor no mayor a 158 [ $k\Omega$ ].

Finalmente, la última etapa del convertidor se conforma por los capacitores de salida. De acuerdo con la hoja de especificaciones, los criterios de selección para los capacitores de salida de un convertidor SEPIC son los mismos que para el convertidor Boost por lo que la descripción siguiente está basada en la información presentada en la sección de este último.

Para la selección adecuada de los capacitores de salida deben considerarse las características presentes en el modelo real de un capacitor, un capacitor real puede verse como un arreglo serie compuesto por un resistor (ESR, *equivalent series resistance*), un inductor (ESL, *equivalent series inductance*) y un capacitor (C). Cada uno de estos parámetros tienen un efecto directo sobre el voltaje de rizo de salida, la selección de los capacitores adecuados debe buscar mitigar este voltaje.

El primer paso en la selección de los capacitores es fijar el voltaje de rizo máximo expresado en porcentaje. Se sugiere en la hoja de datos considerar un valor de 2 %, distribuido equitativamente entre la componente del voltaje de rizo a causa de la resistencia serie equivalente (ESR) y la componente producto de la capacitancia (C). Bajo estas consideraciones, la ESR del capacitor que aporta un 1 % al voltaje de rizo total puede estimarse con la siguiente expresión:

$$ESR_{Cout} \leq \frac{0.01(V_{out})}{I_{D(pico)}} \quad (3.27)$$

De igual manera, la capacitancia que aporta el otro 1 % puede calcularse de la siguiente forma:

$$C_{out} \geq \frac{I_{out(max)}}{0.01(V_{out} \cdot f)} \quad (3.28)$$

Otro factor a considerar es la corriente de rizo, los capacitores de salida se ven expuestos a altas corrientes de rizo (RMS) por lo que el capacitor que se seleccione debe cumplir la siguiente condición:

$$I_{RMS(Cout)} \geq I_{out(max)} \sqrt{\frac{D_{max}}{1 - D_{max}}} \quad (3.29)$$

### 3. ACTUALIZACIÓN DE LA INSTRUMENTACIÓN

---

Por lo general, para cumplir con la condición de ESR se colocan varios capacitores en paralelo a la salida; una vez que se satisface esta condición, la capacitancia resultante es la adecuada para realizar tareas de filtrado y se cumple además con el valor de corriente RMS requerido [9]. Como sucede en la topología del convertidor SEPIC, se suelen emplear capacitores cerámicos en paralelo para reducir el efecto de las inductancias parásitas en el capacitor de salida [9]. En el diseño seleccionado para el convertidor SEPIC, se emplean cuatro capacitores de salida en paralelo de  $47 [\mu F]$  a  $20 [V]$  y un capacitor cerámico de  $10 [\mu F]$  a  $25 [V]$ .

#### 3.1.7. Diseño de las Tarjetas de Potencia Eléctrica

En los párrafos siguientes se describen brevemente las especificaciones para el diseño de las tarjetas de potencia eléctrica, de igual manera se discuten algunas consideraciones que deben tenerse presente al momento de diseñar cualquier tarjeta electrónica con la finalidad de obtener el mejor desempeño del circuito montado sobre esta.

Las tarjetas de potencia eléctrica se diseñaron con las dimensiones máximas permisibles de  $10 [cm] \times 10 [cm]$ . El motivo de esta limitante está dado por dos razones principalmente: Son las dimensiones máximas para tarjetas de circuito impreso que maneja el fabricante y, se busca minimizar el tamaño con respecto a las tarjetas en la versión anterior de la instrumentación. Con respecto a este último punto, al contar con tarjetas de tamaño reducido de iguales dimensiones se busca lograr un diseño modular es decir, montar las tarjetas una sobre otra por medio de conectores que permitan el intercambio de señales donde cada una represente un módulo de alimentación eléctrica. Esto con la intención de resolver uno de los problemas que se tenían en la versión anterior de la instrumentación, el problema de la distribución de masas en la MSA. Un diseño modular permite colocar el conjunto de tarjetas apiladas en un área fija dentro del simulador evitando la tediosa tarea de volver a caracterizar mecánicamente al simulador cada vez que alguna de las tarjetas se cambia de lugar durante la puesta en marcha del mismo.

Otro de los aspectos que vale la pena comentar es el relativo al diseño electrónico de las tarjetas. Muchas de las observaciones que se comentan a continuación aplican no sólo para las tarjetas de este trabajo sino en general para cualquier tarjeta electrónica que se pretenda diseñar. Estos comentarios resultan útiles para una persona que recién comienza en el diseño electrónico pues muchas de estas consideraciones no son triviales y tienen un impacto en el buen funcionamiento del circuito. El ruteo de las pistas en la placa no es una tarea trivial, existen varios detalles a considerar, los comentarios y sugerencias que se enlistan a continuación tienen que ver principalmente con el ruteo de las pistas, sus dimensiones y la ubicación de los componentes.

- Debe procurarse que las pistas en diferentes capas no se crucen o empalmen, si esto es inevitable entonces se debe buscar que se crucen en forma de cruz (+) de

manera que se formen ángulos de 90 [°] entre ellas. Este punto es especialmente relevante cuando se trata de líneas de datos, como ocurre cuando se utilizan dentro de algún protocolo de comunicación, en donde si se empalman dos líneas en diferentes capas pueden llegar a interferirse.

- Debe procurarse que al rutear, las pistas no formen ángulos de 90 [°] pues esto produce un efecto eléctrico indeseable que puede afectar el funcionamiento del circuito en cambio las pistas deben trazarse uniendo rectas que formen entre sí ángulos menores a 90 [°], un buen ejemplo de la forma y ángulo que se espera que tengan las pistas es aquella que tienen las rectas que forman un hexágono.
- El ancho de las pistas es un punto importante a detallar. Es importante identificar aquellas pistas por las que circulará una magnitud de corriente significativa pues son esas pistas las que deben ser diseñadas con un ancho mucho mayor que las otras pistas de la placa. En las tarjetas de potencia, el tamaño estándar de las pistas es de 10 [mil] mientras que las pistas por donde circulará la mayor cantidad de corriente fueron diseñadas con un ancho de 40 [mil]. Este valor no es único, en base a la experiencia del grupo de trabajo y de las recomendaciones que pueden encontrarse en la *web* se considera como un valor apropiado un ancho de: 40 [mil] para las pistas de alta corriente, de 8 a 10 [mil] para las pistas de señales en general, de 10 a 20 [mil] para las líneas de alimentación de las tarjetas (p. ej., 5 [V], 3.3 [V]), y de 10 a 20 [mil] para las líneas de tierra. Seguir esta recomendación evita que la placa se sobrecaliente y eventualmente se dañe a causa del tamaño de sus pistas ante una gran demanda de corriente.
- Otro punto para discutir es la ubicación de los componentes dentro de la placa. En las hojas de datos de muchos CI para convertidores DC/DC se recomienda colocar por un lado todos los elementos que pudiesen manejar grandes corrientes como el MOSFET, el inductor y el diodo dentro de una misma área. Y por otro lado, en un área aparte colocar el resto de los componentes, poniendo especial atención en los capacitores pues muchos deben ser colocados lo más cerca posible al circuito integrado. Por ello, debe revisarse con cuidado las hojas de datos de cada circuito integrado pues en ellas por lo general se detalla si algún elemento requiere de consideraciones especiales además de que pueden encontrarse sugerencias para la ubicación de algunos elementos en específico.
- Finalmente, se decidió agregar en la placa un LED que indique cuando el nivel de carga de la batería es demasiado bajo. Este elemento fue algo que hacia falta en la versión anterior pues las celdas de la batería son sensibles a dañarse si se alcanza un nivel de descarga muy bajo, esto es por debajo de los 11 [V]. La implementación fue muy sencilla pues consta de un comparador de ventana con el LED conectado a su salida, por lo que al entrar la señal de voltaje de la batería dentro del umbral de “batería baja”, considerado por debajo de los 11.4 [V] a los 11 [V] (valor que nunca debería alcanzarse), el LED se enciende dando aviso

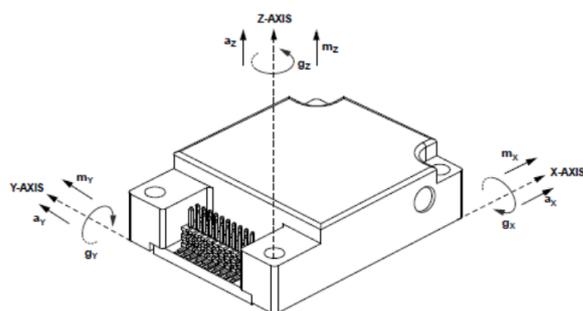
### 3. ACTUALIZACIÓN DE LA INSTRUMENTACIÓN

---

de que debe detenerse el funcionamiento del simulador y debe ponerse a cargar inmediatamente a la batería para evitar daño.

#### 3.2. IMU: ADIS16448 de Analog Devices

La nueva unidad de medición inercial del simulador es la ADIS16448 de *Analog Devices*, se trata de una IMU de 11 grados de libertad pues cuenta con un giróscopo digital triaxial, un acelerómetro digital triaxial, un magnetómetro digital triaxial, un barómetro y un termómetro interno. Está construida con tecnología MEMS lo que trae como consecuencia inmediata que se pueda contar dentro de un mismo bloque de apenas 2.4 [cm] x 3.8 [cm] x 1.1 [cm] con todos los sensores junto a la microelectrónica que le acompaña. Debe notarse que la IMU viene calibrada de fábrica lo que caracteriza cada sensor en términos de sensibilidad, alineamiento de ejes y bias. Esta característica permite pues que cada sensor posea sus propias ecuaciones de compensación lo que resulta en lecturas más precisas. La IMU cuenta con varios registros mediante los cuales se puede configurar el modo de operación de la unidad acorde a las especificaciones y necesidades de cada proyecto o acceder a la lectura de cada sensor por separado, entre otras acciones. La comunicación con la IMU se logra mediante el protocolo de comunicación SPI lo que permite establecer de forma relativamente sencilla la comunicación con cualquier microcontrolador y en general con cualquier arquitectura que cuente con un módulo para ello; se debe señalar que la IMU opera en modo esclavo mientras que el sistema de procesamiento como maestro.



**Figura 3.9:** Unidad de medición inercial con su sistema de referencia [13].

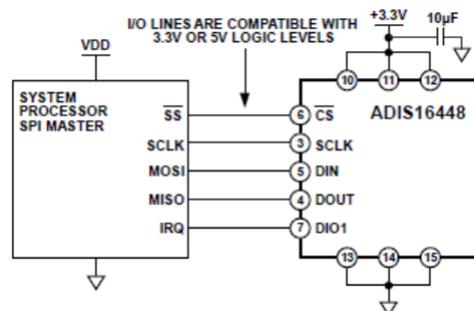
La unidad cuenta con dos modos de operación, *single register read* y *burst read*. El primero como su nombre lo indica permite leer la información del registro asociado a un sensor en particular, cada sensor posee su propio comando de 16 bits que debe transmitirse a la IMU para solicitar el contenido del registro de dicho sensor. El segundo es el modo de operación que se utilizó en este trabajo pues permite mediante un único comando solicitar la información de todos los sensores como un único tren de datos. Para utilizar la unidad en este modo se transmite el comando 0x3E00 y se reciben los datos

de la IMU en el siguiente orden: estatus del sistema, gir6scopo en X, gir6scopo en Y, gir6scopo en Z, aceler6metro en X, aceler6metro en Y, aceler6metro en Z, magnet6metro en X, magnet6metro en Y, magnet6metro en Z, bar6metro, y term6metro; se requieren de 12 ciclos de reloj para leer el flujo de datos solicitado de 16 bits cada uno.

Es importante desde el punto de vista de la instrumentaci6n hacer una descripci6n de las especificaciones relevantes de los sensores presentes en esta unidad. Cada uno de los gir6scopos ofrece un rango programable, pueden ajustarse a  $\pm 250 \left[ \frac{\circ}{s} \right]$ ,  $\pm 500 \left[ \frac{\circ}{s} \right]$  y  $\pm 1,000 \left[ \frac{\circ}{s} \right]$ ; poseen una sensibilidad de  $25 \left[ \frac{LSB}{\circ/s} \right]$ . Los aceler6metros tienen un rango m6nimo de  $\pm 18 \left[ g \right]$ ; poseen una sensibilidad de  $1,200 \left[ \frac{LSB}{g} \right]$ . Los magnet6metros tienen un rango m6nimo de  $\pm 1.9 \left[ G \right]$ ; poseen una sensibilidad de  $7 \left[ \frac{LSB}{mG} \right]$ . El bar6metro digital ofrece un rango de operaci6n calibrado de  $300 \left[ mbar \right]$  a  $1,100 \left[ mbar \right]$  sin embargo, tambi6n cuenta con un rango extendido de  $10 \left[ mbar \right]$  a  $1,200 \left[ mbar \right]$  y posee una sensibilidad de  $0.02 \left[ \frac{mbar}{LSB} \right]$ . El term6metro de la unidad posee una sensibilidad de  $0.07386 \left[ \frac{\circ C}{LSB} \right]$ . Debe sealnarse que el term6metro mide la temperatura interna de la unidad por lo que no necesariamente representa la temperatura ambiente, esto tiene la finalidad de poder monitorear los cambios de temperatura en la unidad. La IMU est6 dise~nada para operar en un rango de  $-40 \left[ \circ C \right]$  a  $85 \left[ \circ C \right]$ , el term6metro puede sensar temperaturas dentro del rango de  $-40 \left[ \circ C \right]$  a  $105 \left[ \circ C \right]$ .

Resta mencionar que la frecuencia a la cual se muestrean las se~nales generadas por los sensores MEMS es de  $819.2 \left[ SPS \right]$  y todos los sensores tienen una resoluci6n digital de 16 bits en formato de complemento a dos.

Por otra parte, una de las caracter6sticas m6s sobresalientes de esta unidad es que incorpora como parte de su microelectr6nica un filtro digital. Se trata de una etapa de acondicionamiento de se~nal que incorpora un filtro pasa-bajas junto a dos filtros promediadores en cascada con lo que se logra la respuesta de un filtro FIR [13].



**Figura 3.10:** Diagrama de conexiones de la IMU con cualquier sistema de procesamiento [13].

Sobre la alimentaci6n de la unidad, esta requiere de un voltaje de entre  $3.15 \left[ V \right]$  y  $3.45 \left[ V \right]$  para comenzar a operar. En la figura 3.10 se observa el diagrama de conexiones entre la IMU y un sistema de procesamiento cualquiera, se observa el uso de un capacitor

de 10 [ $\mu F$ ] para filtrar la señal proveniente de la fuente de alimentación así como la numeración correspondiente a cada pin de la unidad y su función dentro del protocolo de comunicación SPI.

### 3.3. Computadora a bordo: Zedboard

Se seleccionó a la tarjeta de desarrollo Zedboard como la computadora a bordo del simulador, está construida alrededor del SoC (*System-on-Chip*) Zynq XC7Z020 de *Xilinx* que ofrece las bondades de una arquitectura fija junto a una arquitectura reprogramable en un mismo encapsulado. El SoC Zynq posee un procesador ARM Cortex-A9 de doble núcleo que conforma el área llamada PS (*Processing System*), y una FPGA Artix-7 que conforma el área llamada PL (*Programable Logic*).

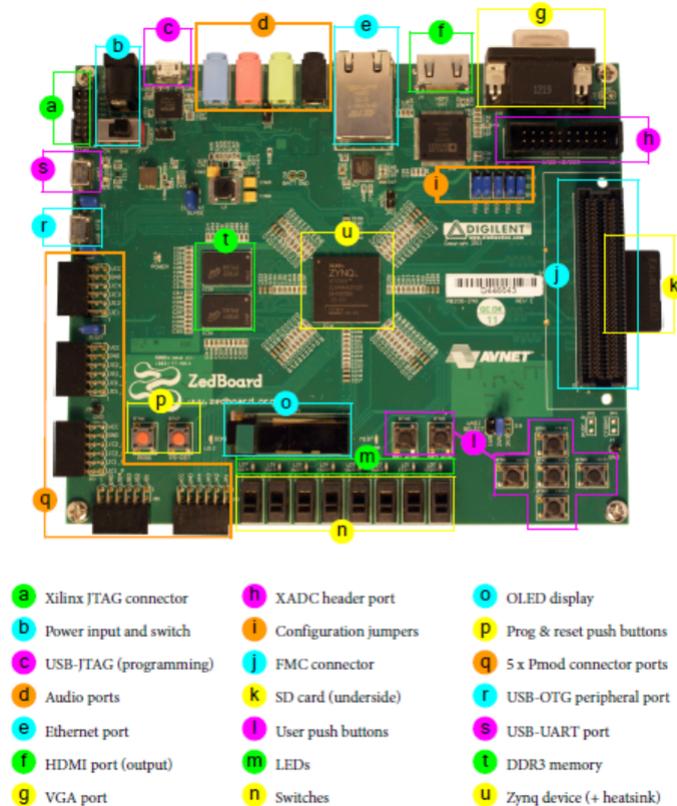
La tarjeta Zedboard cuenta con diferentes interfaces periféricas algunas de ellas son:

- Interfaz GPIO: 9 LEDs, 8 *switches*, 7 *push buttons*
- 5 interfaces Pmod
- Display OLED
- Ethernet
- Memoria flash de 256 [Mbit]
- Memoria DDR3 de 512 [MB]
- Codec de Audio (Analog Devices ADAU1761), la tarjeta cuenta con puertos para líneas de entrada y salida de audio, para micrófono y auriculares
- Video (HDMI)
- Video (VGA)
- Lector para tarjeta SD
- USB-JTAG para programación de la tarjeta
- USB-UART para comunicación

Adicionalmente, se tienen dos osciladores que proveen cada uno de una fuente de reloj, por lo que se tiene disponible un reloj con una frecuencia de 100 [MHz] y otro de 33.3333 [MHz]. La figura 3.11 muestra las interfaces y algunos integrados presentes en el área frontal de la tarjeta Zedboard.

Por sus características el SoC Zynq XC7Z020 es apropiado para aplicaciones de procesamiento digital de imágenes, procesamiento de vídeo, procesamiento de audio,

visión por computadora, robótica, control e instrumentación [5]. En particular en el campo de control e instrumentación, en el texto [5] se menciona que el SoC Zynq y las FPGAs pueden ser de gran utilidad debido a que pueden realizar procesamiento en tiempo real a gran velocidad y manejar simultáneamente múltiples entradas de sensores y salidas de actuadores.



**Figura 3.11:** Interfaces de la tarjeta de desarrollo Zedboard [5].

Una de las principales ventajas del Zynq frente a una FPGA convencional es que el primero incorpora toda una serie de periféricos como módulos de comunicación serial (UART, SPI,  $I^2C$ ), temporizadores, entre muchos otros que no están presentes en una FPGA, por lo que si se requiere utilizar alguno, obliga al desarrollador a construir el módulo con algún lenguaje de descripción de hardware.

Por otra parte, una de las principales ventajas del Zynq frente a un procesador convencional es su capacidad de reconfiguración. El Zynq permite reconfigurar su arquitectura acorde a las necesidades y evolución de un proyecto. Más aún, uno de los propósitos centrales de la lógica programable (PL) en el Zynq es expandir las capacidades del procesador (PS), esto permite por ejemplo añadir varios bloques de co-procesamiento que habiten en PL que ayuden en su carga de tareas al procesador; este tipo de arquitecturas de procesamiento en paralelo no podrían implementarse sobre

### 3. ACTUALIZACIÓN DE LA INSTRUMENTACIÓN

---

un único procesador convencional. Para conocer los detalles sobre la arquitectura del procesador ARM Cortex-A9 puede consultarse el documento [21].

La programación del SoC Zynq se hace a través de un nuevo conjunto de herramientas de software de *Xilinx* que tienen por objetivo acelerar el flujo de diseño y culminación de un proyecto; las dos principales herramientas en la nueva suite de diseño son Vivado y SDK.

En Vivado se diseña, construye y reconfigura la plataforma de hardware que esté presente en el Zynq. Este entorno de desarrollo permite seleccionar un amplio rango de dispositivos SoC de la familia Zynq-7000, algunas familias de FPGA o bien seleccionar directamente una tarjeta de desarrollo. En general se selecciona esta última pues es especialmente útil, ya que el software se “concientiza” de todos los periféricos que incorpora una tarjeta de desarrollo por lo que incluye bloques específicos que permiten interactuar con ellos. La filosofía de diseño en Vivado se basa en arrastrar e interconectar bloques de forma que es muy similar a la forma de trabajo de otras herramientas como Simulink o LabVIEW. Se permite el uso de un lenguaje de descripción de hardware (VHDL ó Verilog) para crear y añadir bloques personalizados que realicen operaciones muy específicas. También es posible aprovechar los modelos creados en Simulink por medio de una herramienta presente en MATLAB que permite la traducción del diagrama de bloques hacia el código HDL seleccionado lo que permite ejecutar sobre hardware el diseño trazado en Simulink; el resultado final es un único bloque que es exportado hacia Vivado construido a partir del modelo en Simulink.

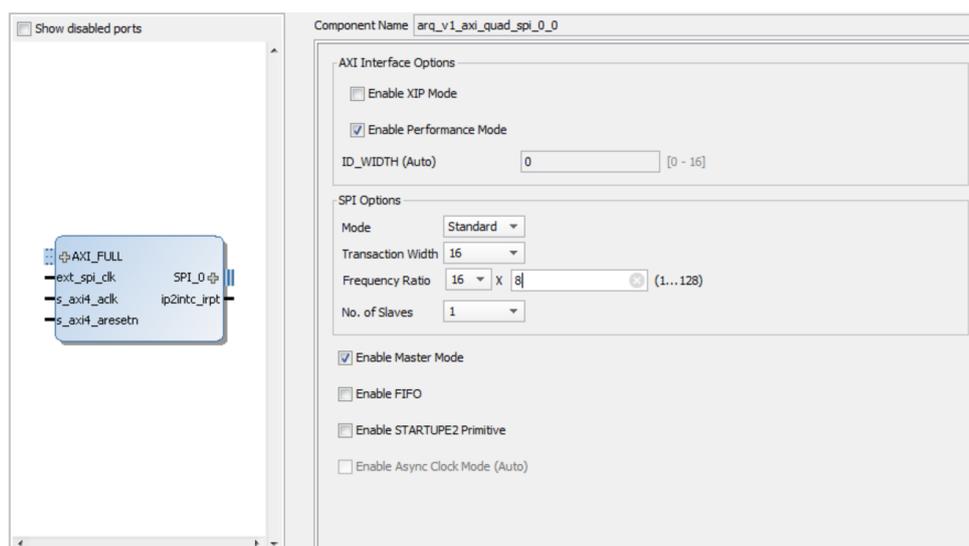
La arquitectura creada en Vivado se exporta a SDK. Al recibir SDK la arquitectura se crean las librerías necesarias para interactuar con ella, *Xilinx* incluye en SDK una herramienta que crea de forma automática una basta librería de funciones necesarias para programar tanto los periféricos propios del procesador ARM como aquellos que habitan en forma de bloque sobre la lógica programable, SDK permite desarrollar el software en lenguaje C/C++ que va a ejecutarse sobre el procesador (PS) permitiendo controlar todos los periféricos que habitan sobre la lógica programable (PL). Una vez concluida la programación, se selecciona transferir e implementar la arquitectura sobre la lógica programable de la tarjeta, el paso final es seleccionar correr el programa sobre el procesador para que todo el sistema presente en la tarjeta comience a operar.

#### 3.3.1. Sistema de Adquisición de Datos

El propósito de la computadora a bordo es procesar los algoritmos de estimación y control de orientación para ello es necesario contar con un sistema de adquisición de datos que permita la lectura y visualización de la información proveniente de los sensores a partir de la comunicación entre la unidad de medición inercial y la computadora a bordo. En este trabajo se decidió abordar este problema, al implementar y programar una arquitectura en el Zynq que hace uso tanto de PS como de PL para adquirir

y procesar los datos provenientes de los sensores de la unidad de medición inercial, dejando para trabajo a futuro la implementación de dichos algoritmos.

Para lograr la comunicación entre la IMU y la computadora a bordo se empleó el bloque IP *AXI Quad SPI*, un bloque especialmente dedicado a establecer la comunicación por SPI entre la tarjeta de desarrollo y cualquier sensor que soporte este protocolo de comunicación.



**Figura 3.12:** Configuración del bloque *AXI Quad SPI*.

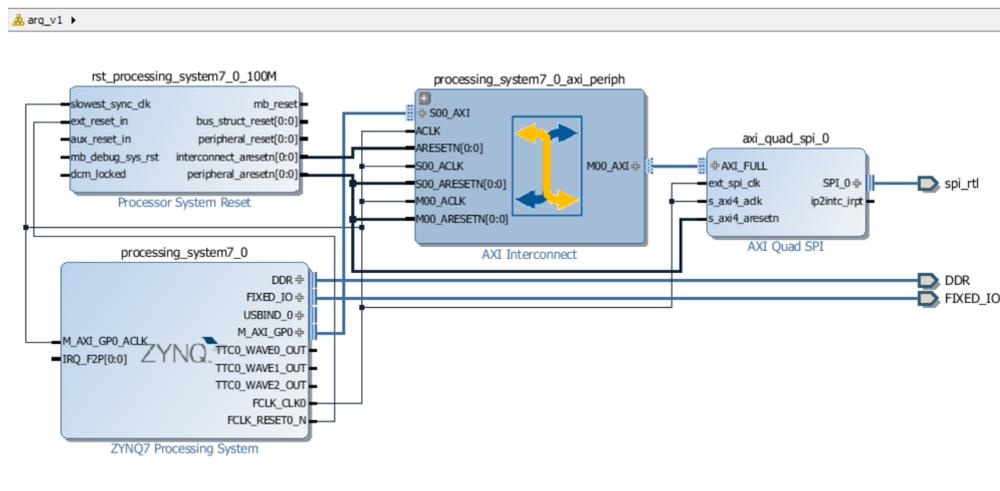
El procesador ARM incluye un módulo de comunicación SPI sin embargo se prefirió utilizar el bloque sobre la lógica programable por las siguientes razones: el módulo del procesador transmite y recibe información en formato de 8 bits mientras que la IMU transmite y recibe información en formato de 16 bits, esto puede complicar la programación que logre la comunicación entre ambos, sobre todo si se considera que la IMU se va a utilizar en modo “burst read” lo que significa un flujo continuo de datos de 16 bits hacia la tarjeta de desarrollo. Se reporta en varios foros que las librerías para interactuar con el módulo SPI del procesador presentan diferentes “bugs” en particular con la línea  $CS_0$  (*Chip Select 0*) por lo que debe tenerse especial cuidado al utilizar dicho módulo.

En contraste con el módulo del procesador, agregar el bloque reprogramable permite configurarlo para transmitir y recibir datos de 8, 16 y 32 bits además permite otro tipo de opciones como seleccionar diferentes prescalamientos para definir la frecuencia de la señal de reloj (SCLK) que va a emplearse en la comunicación. Con esto queda en evidencia algunas de las ventajas de las arquitecturas reprogramables con respecto a las arquitecturas fijas.

La arquitectura implementada sobre el Zynq se muestra a continuación en la figura 3.13. Se observa el procesador ARM conectado al bloque SPI, esta arquitectura fue

### 3. ACTUALIZACIÓN DE LA INSTRUMENTACIÓN

exportada a SDK para programar que el procesador comande al bloque en establecer la comunicación con la IMU.



**Figura 3.13:** Arquitectura implementada sobre el SoC Zynq para efectuar la comunicación por SPI.

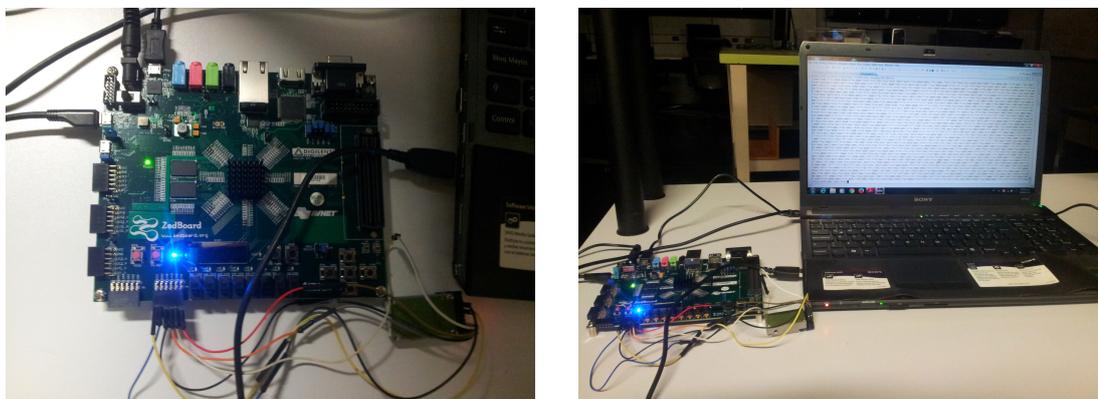
La información adquirida por la tarjeta es retransmitida a una laptop por medio de comunicación serial a través del módulo UART. Para procesar y visualizar la información de los sensores se programó un script en Matlab capaz de generar las gráficas correspondientes a cada sensor. El objetivo del script es organizar los “datos crudos” provenientes de la tarjeta dentro de una matriz de 2496 renglones y 12 columnas donde cada columna está asociada a un sensor en particular y por ende los renglones representan cada uno de los puntos de datos adquiridos por algún sensor. Posteriormente, estos datos son procesados para poder ser presentados en gráficas.

Dentro del script se incluyó una función que realiza el algoritmo de estimación TRIAD, el objetivo de este algoritmo es generar una estructura llamada cuaternión a partir de las lecturas de los sensores, el cuaternión es una manera de representar la orientación de un cuerpo. Para cada lectura de los sensores se genera un cuaternión cuyas componentes se grafican lo que permite apreciar la evolución en la orientación.

Otro de los propósitos del script es el de capturar y guardar los datos dentro de un archivo de texto llano .dat para poder ser aprovechado por cualquier otro programa. Este archivo se aprovechó para sintonizar la nueva unidad de medición inercial con el algoritmo de estimación conocido como Filtro de Kalman Unscented (UKF). Siempre que se cambia la unidad de sensores es necesario volver a sintonizar el filtro con la nueva unidad, esto se hace a través de un muestreo donde se captura una cantidad de datos estadísticamente significativa mientras la unidad está en reposo tomando lecturas, los detalles de este proceso se describen en el siguiente capítulo.

En el apéndice de este trabajo puede hallarse el código comentado en detalle del

script junto con una introducción a los cuaterniones y al algoritmo TRIAD.



**Figura 3.14:** Sistema de adquisición de datos en operación.

### 3.4. Tarjeta de Drivers para Motores

El rediseño del simulador incluye el cambio de motores de corriente directa con escobillas por motores BLDC, siendo estos últimos los que se encuentran típicamente presentes en aplicaciones aeroespaciales como en el caso de satélites pequeños.

Este cambio provoca que se requiera diseñar una tarjeta electrónica que permita utilizar a los nuevos motores. La tarjeta además se pensó para desempeñar el rol de tarjeta de co-procesamiento que auxilie en las tareas de control a la computadora a bordo, se tiene entonces un sistema de control distribuido donde se pretende que los algoritmos de estimación y control de orientación corran sobre la computadora a bordo mientras que el control de velocidad de los motores se ejecute sobre la tarjeta de drivers.

La tarjeta de drivers se compone de dos partes, por un lado el sistema de procesamiento el cual se construyó alrededor del microcontrolador dsPIC33FJ16MC102 que requiere de un sistema mínimo para poder operar y ser programado y por otra parte, la electrónica de potencia que es la responsable de servir como interfaz entre el microcontrolador y el motor.

El sistema mínimo que requiere el microcontrolador consta de muy pocos elementos principalmente resistores y capacitores de filtrado además del cristal y conectores que son necesarios. En la hoja de especificaciones del microcontrolador [35] puede hallarse el sistema mínimo así como el valor de cada uno de los componentes, el diseñador por supuesto puede añadir elementos adicionales que pudiese llegar a requerir como *headers* conectados a los pines de entrada y salida de propósito general así como LEDs

### 3. ACTUALIZACIÓN DE LA INSTRUMENTACIÓN

---

que indiquen el encendido del microcontrolador o que la transferencia de información con otro dispositivo está en marcha.

Se pretende controlar al motor por medio de señales de PWM provenientes del microcontrolador, sin embargo es evidente que las señales de PWM directamente del microcontrolador son incapaces de mover al motor, por lo que se requiere contar con la electrónica que logre ser la interfaz que una a estos dos elementos.

La electrónica de potencia presente en la tarjeta corresponde a un puente trifásico, este circuito está compuesto por un arreglo de 3 pares de MOSFETs donde cada uno está antecedido por su propio *buffer*. Los *buffer* tienen la función de elevar el nivel de voltaje de las señales de PWM provenientes del microcontrolador hasta el nivel de alimentación del motor, para la tarjeta se seleccionaron los *buffer* TC4428AVOA.

Los MOSFET son utilizados a manera de switches que abren y cierran para conmutar las fases del motor. Cada fase del motor se encuentra conectada a un par de MOSFET, como se observa en el esquemático de la tarjeta [cita]. Para arrancar un motor BLDC es necesario conmutar las fases de acuerdo a una secuencia de conmutación, dichos valores se resumen en una tabla de seis pasos, esta tabla se puede generar fácilmente conociendo los principios de funcionamiento del motor, este proceso se describe en la siguiente sección. Para seleccionar los MOSFET apropiados se debe considerar principalmente el voltaje de operación del motor para a continuación seleccionar aquellos MOSFET con un  $V_{DS}$  de poco menos del doble del voltaje de operación del motor, por ejemplo para un motor que es alimentado por 24 [V], el  $V_{DS}$  del MOSFET debe de ser de al menos 40 [V]. Otro factor a considerar es la corriente del *drain* que puede tolerar el MOSFET, pues debe ser lo suficientemente alta para soportar la corriente de arranque del motor. Finalmente, el último parámetro a considerar para la selección del MOSFET es la resistencia en estado de encendido del *drain* al *source* ( $R_{DS(on)}$ ) la cual debe ser muy pequeña, por ejemplo  $R_{DS(on)} \leq 10[m\Omega]$  para evitar que los MOSFET se sobrecalienten al manejar corrientes cercanas a 1 [A]. Con esto en mente, los MOSFET seleccionados para la tarjeta fueron el IRF9630PbF (canal P) y el IRF610PbF (canal N).

Para iniciar el proceso de arranque es necesario que el microcontrolador lea la información de los sensores de efecto Hall que indican la posición del rotor, según los valores leídos, el microcontrolador selecciona el valor de conmutación apropiado y se envía hacia los *buffer*, provocando que los MOSFET conmuten según dicho valor; este proceso de leer y enviar el valor apropiado hacia los MOSFET se repite en un *loop* y es de esta manera que se logra mantener en movimiento al motor.

La tarjeta de drivers incorpora diferentes conectores como puede verse en la imagen 3.15, a continuación se describe brevemente cada uno de ellos:

- El grupo de conectores **A** permiten comunicar a la tarjeta de drivers con la computadora a bordo o con cualquier otro dispositivo por medio del protocolo SPI.

- El conector **B** permite conectar la tarjeta de drivers con las fuentes de alimentación en las tarjetas de potencia eléctrica; la electrónica correspondiente al microcontrolador se alimenta con 3.3 [V] mientras que la electrónica de potencia con 12 [V].
- El conector **C** permite la conexión entre el motor y la tarjeta.
- El conector **D** hace posible la conexión entre el programador PICkit 3 de Microchip y la tarjeta lo que permite cargar programas nuevos al microcontrolador.

Finalmente, en el simulador habrán 3 tarjetas de drivers idénticas, cada una dedicada a manejar un motor.

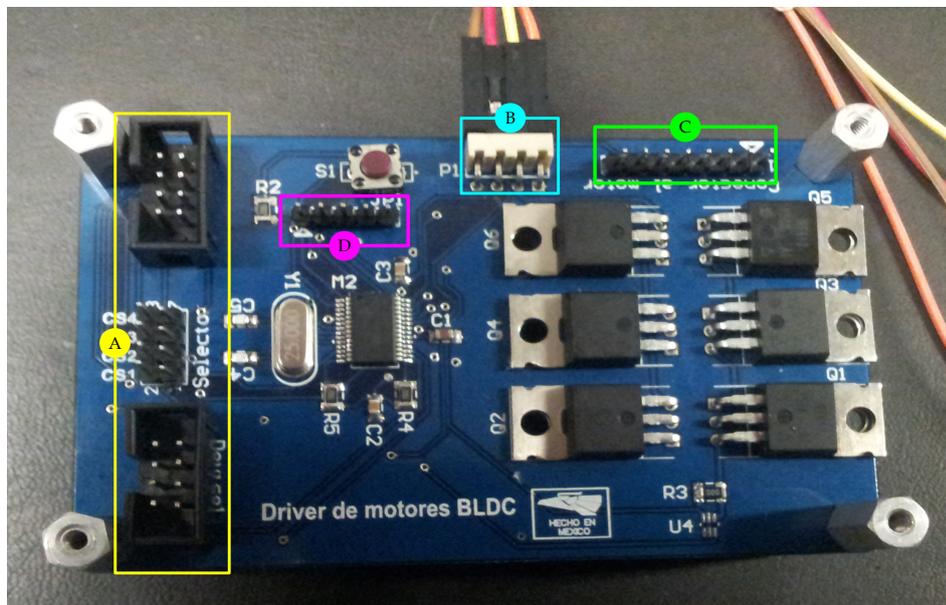


Figura 3.15: Tarjeta de drivers para motores.

### 3.5. Actuadores: Motores BLDC EC 22 de Maxon

Se seleccionaron a los motores BLDC EC 22 de Maxon como los nuevos actuadores del simulador; la descripción sobre motores BLDC que se hace a continuación se basa en la información publicada en [10], [11] y [12]. Los motores BLDC han ganado rápidamente gran popularidad pues se emplean en diferentes industrias, se pueden encontrar en aplicaciones automotrices, electrodomésticos, equipo médico y automatización industrial. En aplicaciones aeroespaciales se encuentran comúnmente en satélites pequeños y *drones*. Algunas de las ventajas de los motores BLDC sobre los motores de DC convencionales y los motores de inducción son:

### 3. ACTUALIZACIÓN DE LA INSTRUMENTACIÓN

---

- Una mejor relación velocidad vs par.
- Rangos de velocidad más altos.
- Operación silenciosa.
- Vida útil más larga.
- Poseen una alta eficiencia pues no existe una caída de voltaje en las escobillas.
- Requieren menos mantenimiento.
- La relación del par entregado con respecto al tamaño del motor es mayor, haciéndolo útil en aplicaciones donde el espacio y "peso" son factores críticos.

Dentro de las máquinas eléctricas, los motores BLDC son un tipo de máquina síncrona lo que quiere decir que el campo magnético generado por el estátor y el campo magnético del rotor rotan a la misma frecuencia. Como su nombre lo indica, los motores BLDC carecen de escobillas para realizar la conmutación por lo que ésta se realiza forzosamente de manera electrónica. La construcción de esta máquina eléctrica y sus principios físicos de operación se pueden consultar en [11].

La mayoría de los motores BLDC son trifásicos y presentan una topología en estrella, un motor con esta topología se pone en operación al energizar dos de sus fases a la vez. En este tipo de motores el par se genera debido a la interacción entre el campo magnético generado por los embobinados del estátor y los imanes permanentes del rotor. Para hacer girar al motor, los embobinados del estátor deben energizarse según una secuencia. Para poder mantener en movimiento al motor, el campo magnético generado por los embobinados debe ir cambiando su posición mientras el rotor se mueve siendo atraído por el campo magnético del estátor.

Se debe conocer la posición del rotor para poder energizar los embobinados correctos según la secuencia de conmutación, la posición del rotor se puede conocer por medio de los sensores de efecto Hall. Las señales de los sensores de efecto Hall están desfasadas 60 [°] una con respecto a otra, el desfase en las señales de los sensores de efecto Hall produce un código de 3 bits, éste código se utiliza para poder conocer la posición del rotor.

En la sección anterior se describió la electrónica típicamente empleada para conmutar este tipo de motores, sin embargo resta discutir sobre la secuencia de conmutación que permite hacer girar al motor.

Durante cada paso de la secuencia, una de las fases se polariza positivamente (la corriente entra en el embobinado) mientras que la otra se manda a tierra o bien puede polarizarse negativamente (la corriente sale del embobinado), la tercera fase se deja sin energizar. El rotor se desplaza 60 grados eléctricos cada paso de la secuencia de conmutación por lo que cada 60 grados eléctricos de rotación, uno de los sensores Hall cambia de estado. Se requieren los seis pasos de la secuencia para completar un ciclo eléctrico,

sin embargo, un ciclo eléctrico puede que no corresponda con una revolución mecánica completa del rotor. El número de ciclos eléctricos que se requieren para completar una rotación mecánica está determinado por el número de pares de polos del rotor; por cada par de polos del rotor se completa un ciclo eléctrico por lo que, el número de ciclos eléctricos equivale al número de pares de polos del rotor.

A continuación se muestra una tabla con la secuencia de conmutación para hacer girar al motor en el sentido de las manecillas del reloj a partir de las lecturas entregadas por los sensores de efecto Hall (C B A). La figura 3.16 ilustra el proceso de conmutación descrito en la tabla, se muestran los transistores que deben ser activados para polarizar adecuadamente las fases y mantener el giro del motor.

Código del sensor Hall	No. Paso	Drivers activos
1 0 1	1	Q1(PWM1) Q6(PWM6)
1 0 0	2	Q1(PWM1) Q5(PWM5)
1 1 0	3	Q3(PWM3) Q5(PWM5)
0 1 0	4	Q3(PWM3) Q4(PWM4)
0 1 1	5	Q2(PWM2) Q4(PWM4)
0 0 1	6	Q2(PWM2) Q6(PWM6)

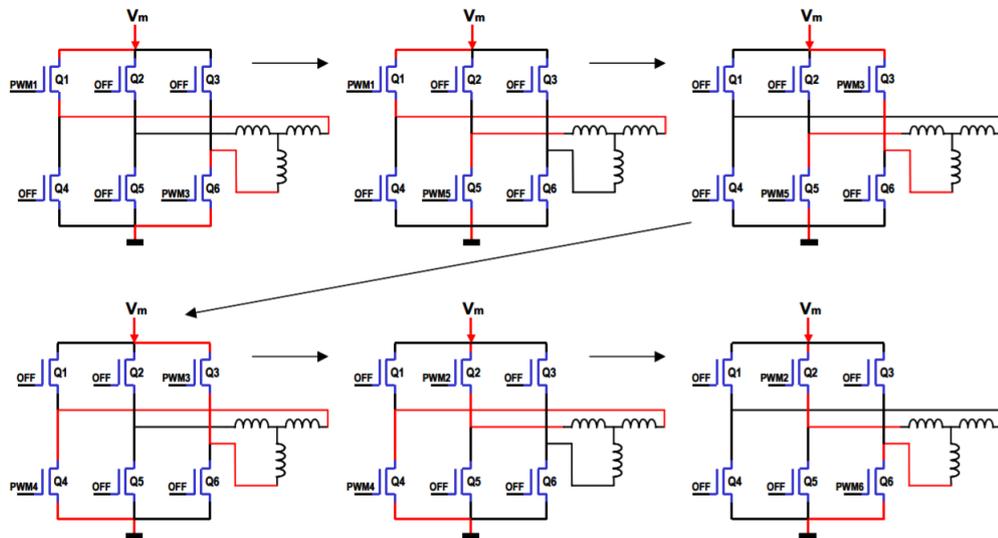


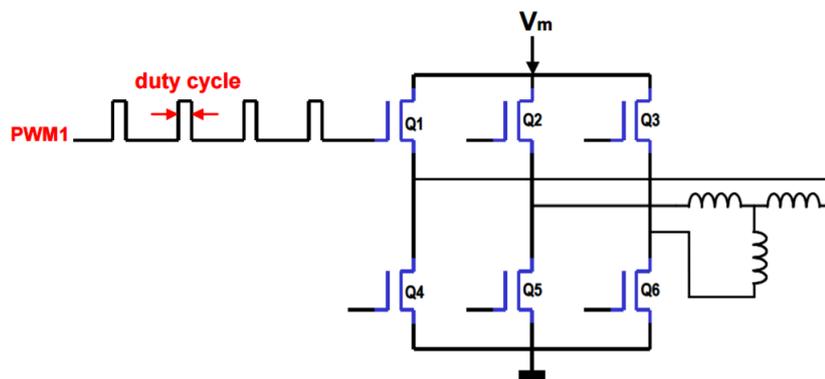
Figura 3.16: Conmutación de los drivers para hacer girar un motor BLDC [12].

Se envían señales moduladas por ancho de pulso (PWM) desde el microcontrolador hacia los transistores para variar la velocidad del motor, éstas señales deben seguir la secuencia de conmutación para mantenerlo en movimiento. Cuando se varía el ciclo de trabajo de las señales de PWM, el voltaje promedio suministrado al estátor se reduce, por lo que la velocidad del motor disminuye.

### 3. ACTUALIZACIÓN DE LA INSTRUMENTACIÓN

---

La velocidad del motor se puede controlar con un esquema de control en lazo cerrado al sensar la velocidad instantánea del motor. Esto permite calcular el error en la velocidad de referencia y la velocidad instantánea del motor. Se puede utilizar un controlador PID que logre disminuir la señal de error y ajustar continuamente el ciclo de trabajo de las señales de PWM para controlar la velocidad del motor.



**Figura 3.17:** El control de velocidad de un motor BLDC requiere de señales PWM [12].

### 3.6. Carga Útil: Raspberry Pi 2B y PlayStation Eye

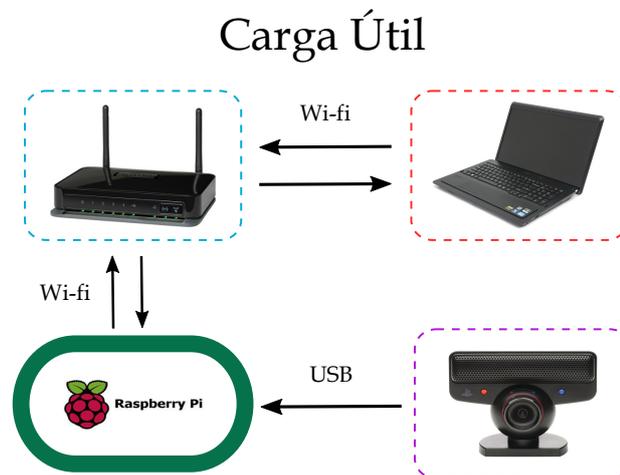
Se decidió incorporar como parte de la actualización del simulador una cámara junto a su propia tarjeta electrónica que permitan visualizar la orientación del simulador de forma remota en un computador. Para realizar este trabajo, se seleccionó a la tarjeta de desarrollo Raspberry Pi 2B junto a la PlayStation Eye como cámara.

La Raspberry Pi es una microcomputadora que posee la capacidad de procesamiento y cuenta con todos los periféricos para funcionar como una pequeña computadora de escritorio, las características técnicas de la Raspberry Pi 2B se pueden consultar en [36]. La interacción con la Raspberry se da por medio de Raspbian, un sistema operativo con interfaz gráfica que corre sobre ella. Raspbian está basado en la distribución de Linux Debian por lo que dentro de su entorno es posible correr diferentes aplicaciones como ocurre con una computadora convencional, una de esas aplicaciones es un ambiente de desarrollo llamado IDLE que nos permite programar sobre la Raspberry haciendo uso del lenguaje de programación Python.

Se justifica la selección de la Raspberry con base en que se trata de una plataforma de desarrollo amigable y sencilla de utilizar que a su vez ofrece capacidades de procesamiento sobresalientes dentro de un sistema de dimensiones pequeñas, además se tiene la ventaja de que la tarjeta cuenta con una comunidad de usuarios numerosa por lo que siempre se puede encontrar basta información y ejemplos para un proyecto.

Por otra parte, se eligió la PlayStation Eye como cámara debido a que es completamente compatible con la Raspberry en términos de alimentación eléctrica y drivers. La PlayStation Eye se alimenta de la Raspberry al conectarse a uno de los puertos USB que posee, muchas otras cámaras requieren de niveles de alimentación eléctrica diferentes a los que un puerto USB puede proveer esto provoca que se necesite comprar un HUB que provea de la alimentación eléctrica adecuada a la cámara a la vez que permita conectarse con la Raspberry. El otro punto son los drivers, al igual que ocurre con una computadora convencional donde cada vez que se conecta algún dispositivo nuevo, la computadora descarga e instala los drivers necesarios para poder interactuar con él, la Raspberry necesita contar con los drivers para interactuar con algunos dispositivos como cámaras. La comunidad de usuarios de la plataforma montó en línea un sitio de información sobre los diferentes dispositivos que pueden conectarse a la Raspberry, dentro de ese sitio puede encontrarse información sobre las diferentes cámaras incluida la PlayStation Eye así como comentarios del desempeño del dispositivo con la Raspberry, en él se asegura que la PlayStation Eye funciona apropiadamente como un dispositivo *plug and play* con solo conectarse a la Raspberry. La cámara se maneja a través de *Motion*, un programa basado en línea de comandos para sistemas operativos Linux, que monitorea las señales de vídeo provenientes de la cámara y es capaz de detectar movimiento.

El esquema para lograr transmitir imágenes hacia una computadora en forma remota se muestra a continuación:



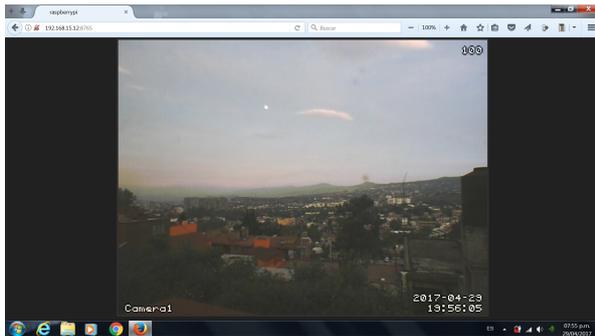
**Figura 3.18:** Interacción entre la carga útil y una computadora personal.

Como puede verse, la implementación resulta sencilla pues se trata de conectar la computadora y la Raspberry a un mismo módem para de esta manera establecer un enlace inalámbrico entre ellos. La Raspberry que se utilizó en este trabajo es parte del

### 3. ACTUALIZACIÓN DE LA INSTRUMENTACIÓN

---

kit educativo Kano en el que se incluyen diferentes accesorios, entre ellos un dongle Wi-Fi; con esta pequeña antena USB se logró la conexión entre la Raspberry y una computadora externa por medio de un módem inalámbrico. Los resultados obtenidos se muestran a continuación:



(a) Imagen captada por la cámara y transmitida por Wi-Fi.



(b) Vista desde la habitación.



(c) Carga útil para el simulador MSA.

**Figura 3.19:** Puesta en marcha de la carga útil.

---

## Capítulo 4

# Simulaciones y Resultados

---

En este capítulo se presentan las simulaciones de los diseños propuestos para las tarjetas de potencia eléctrica, de igual manera como parte de los objetivos de este trabajo, se proponen dos leyes de control de orientación y se muestra su desempeño junto al algoritmo UKF sintonizado con la nueva unidad de medición inercial, se consideran dentro de las simulaciones los parámetros mecánicos de los nuevos motores BLDC incorporados al simulador.

### Instrumentación

Los diseños electrónicos propuestos para las tarjetas de potencia se simularon en el software LTspice IV que la compañía *Linear Technology* pone a disposición del usuario, resulta conveniente emplear este software pues todos los CIs utilizados en las tarjetas de potencia son de esta compañía por lo que se puede encontrar una larga lista de diferentes circuitos integrados de esta compañía listos para añadirse a un diseño.

Los diseños de las fuentes, en especial las fuentes de 12 [V], deben ser capaces de entregar hasta 2 [A] para cumplir con la demanda máxima de corriente que se fijó para los motores. Para comprobar que son capaces de cumplir con dicha demanda se añadió en todas las simulaciones una carga que drena 2 [A] de las fuentes, además se consideró un voltaje de entrada de 16 [V] que corresponde con una batería totalmente cargada. Si las fuentes cumplen con dicho requerimiento de corriente, deben ser capaces de proveer del nivel de voltaje de salida que contempla su diseño así como para todo el rango de valores de voltaje de entrada.

Se le debe prestar atención a las señales de los capacitores de entrada, del inductor y de los capacitores de salida pues estos elementos son susceptibles de presentar elevados picos de voltaje y corriente por efecto de la conmutación del MOSFET.

Como puede verse a continuación, cada una de las fuentes conmutadas son capaces de mantener el nivel de voltaje que contempla su diseño ante una posible demanda de 2 [A] de corriente, y en los puntos de voltaje de entrada máximo (batería completamente cargada) y voltaje de entrada mínimo (valor mínimo de descarga de la batería). Finalmente, en las simulaciones se verificó que las señales de corriente en los capacitores e inductores no presentaran picos que pudiesen comprometer el resto de la electrónica por lo que los diseños sugeridos en las notas de aplicación son apropiados para construirse sobre tarjetas electrónicas.

Los resultados de las simulaciones son los siguientes:

#### 4. SIMULACIONES Y RESULTADOS

---

##### Fuente de 3.3 [V] a 2 [A]

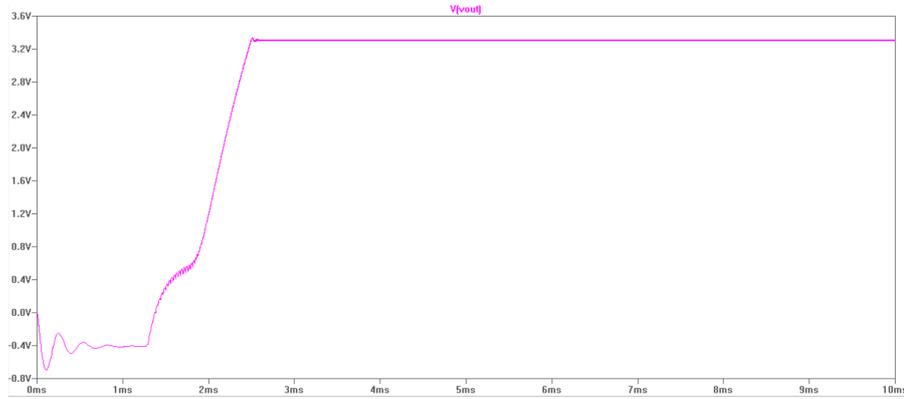


Figura 4.1: Señal del voltaje de salida de la fuente de 3.3 [V].

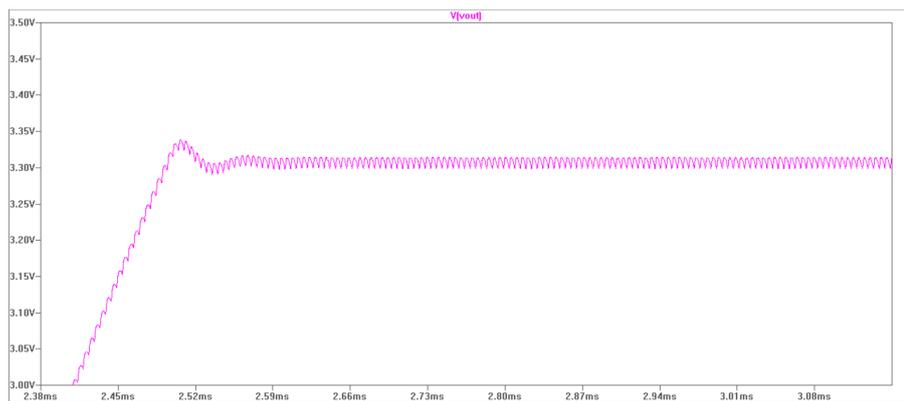


Figura 4.2: Acercamiento de la señal del voltaje de salida.

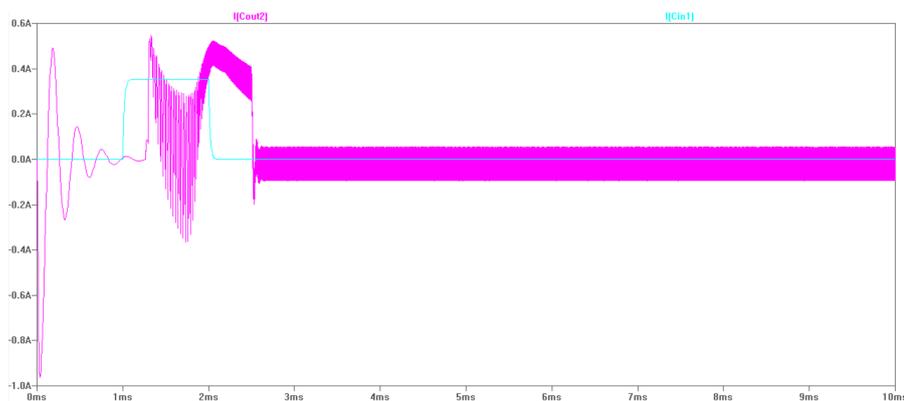
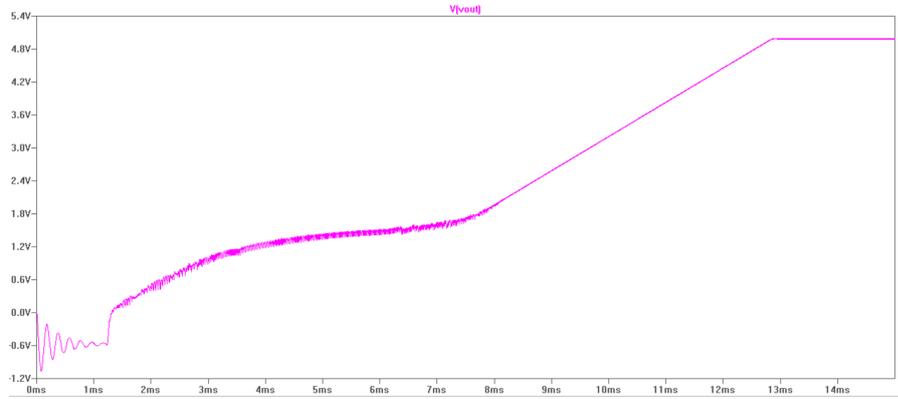


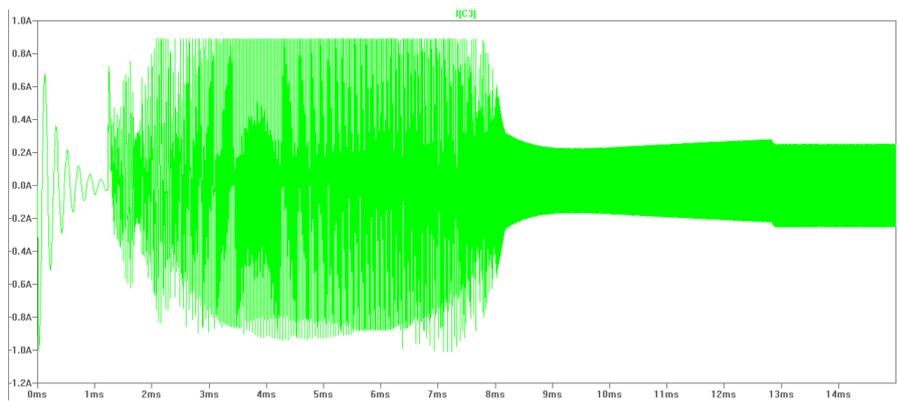
Figura 4.3: Señal de corriente en el capacitor de salida y en el capacitor de entrada.

---

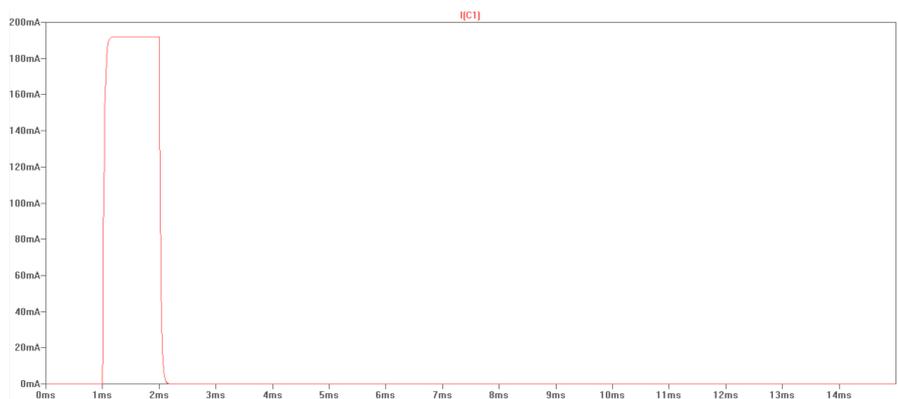
**Fuente de 5 [V] a 2 [A]**



**Figura 4.4:** Señal del voltaje de salida de la fuente de 5 [V].

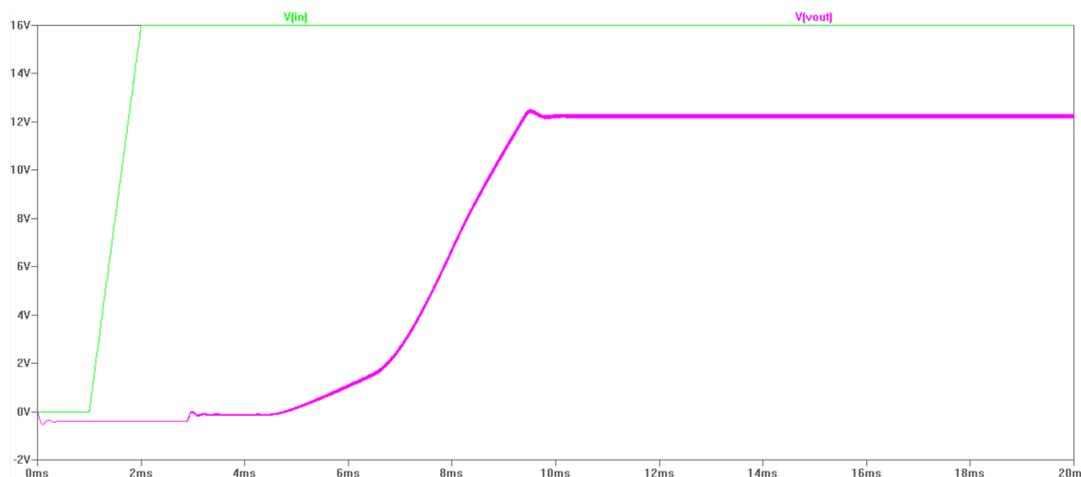


**Figura 4.5:** Señal de corriente en el capacitor de salida.

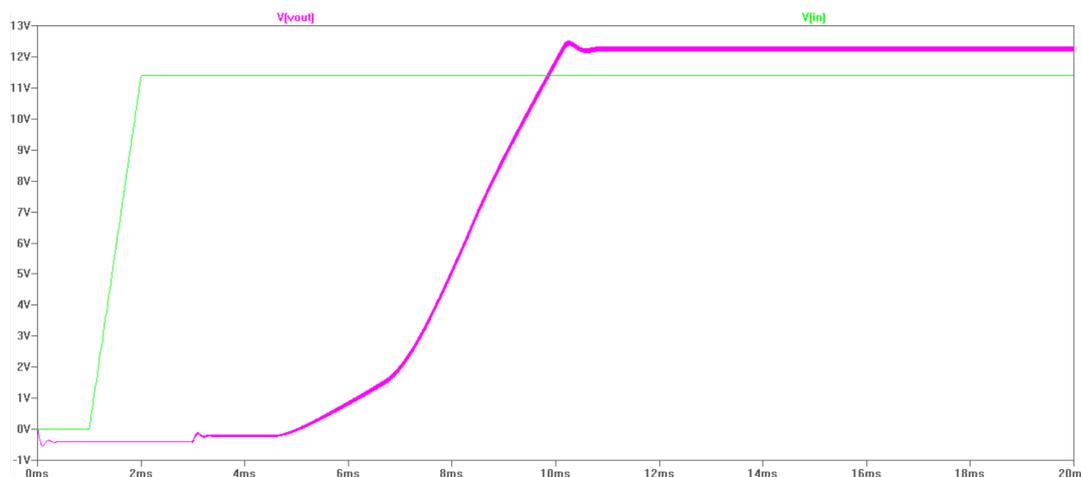


**Figura 4.6:** Señal de corriente en el capacitor de entrada.

**Fuente de 12 [V] a 2 [A]**



**Figura 4.7:** Señal del voltaje de salida de la fuente de 12 [V] con  $V_{in} = 16$  [V].



**Figura 4.8:** Señal del voltaje de salida de la fuente de 12 [V] con  $V_{in} = 11.4$  [V].

**Leyes de Control**

Se investigó en la literatura de dinámica y control de sistemas espaciales, dos leyes de control que pudiesen ser implementadas sobre un sistema embebido a bordo del simulador y que además se pudiesen llegar a desempeñar en el subsistema de control y orientación de un satélite pequeño. En este sentido, se deja una base del material recopilado durante esta tesis sobre la cual puedan partir trabajos futuros que se enfoquen específicamente en el tema de modelado, análisis y control de orientación de satélites pequeños cuyos desarrollos deberán ser validados sobre el simulador MSA.

La selección se hizo tomando en cuenta algunos criterios, se requerían leyes de

---

control que fuesen fáciles de programar sobre un sistema embebido y que no consumiesen grandes recursos computacionales, se requería que pudiesen ser útiles tanto para el simulador como para un satélite pequeño, y que los sensores de la IMU fuesen suficientes para satisfacer los requerimientos de información de dichas leyes. Por consiguiente, las leyes de control que se seleccionaron son esencialmente controladores proporcional-derivativos.

La primera de estas se reporta en el artículo [15], debe señalarse que esta ley ya fue probada sobre el simulador pero con una instrumentación totalmente diferente donde se empleó una PC como bloque de procesamiento. Se trata de un controlador proporcional-derivativo que tiene la siguiente forma:

$$\boldsymbol{\tau} = -\mathbf{K}_p \cdot \hat{\boldsymbol{\epsilon}} - \mathbf{K}_d \cdot \hat{\boldsymbol{\Omega}} \quad (4.1)$$

Donde:

- $\boldsymbol{\tau}$  es el par de control y es un vector de 3 x 1.
- $\mathbf{K}_p$  es la ganancia proporcional y es una matriz de 3 x 3.
- $\mathbf{K}_d$  es la ganancia derivativa y es una matriz de 3 x 3.
- $\hat{\boldsymbol{\epsilon}}$  es la parte vectorial del cuaternión estimado y es un vector de 3 x 1.
- $\hat{\boldsymbol{\Omega}}$  es la velocidad angular estimada del simulador y es un vector de 3 x 1.

Esta ley busca estabilizar al simulador al llevarlo hacia el cuaternión identidad,  $\mathbf{q} = [1 \ 0 \ 0 \ 0]$ .

La segunda ley es más versátil pues permite resolver tanto el problema de estabilización como de regulación de un satélite pequeño. Esta ley se tomó del libro [24] donde se explica a detalle sus por menores y se presentan simulaciones que muestran su desempeño, también se hace referencia al artículo donde apareció originalmente publicada. Como en el caso anterior, se trata de un controlador proporcional-derivativo y tiene la siguiente forma:

$$\begin{aligned} \tau_x &= 2K_{px} \cdot \epsilon_{1e} \cdot \eta_e + K_{dx} \cdot \hat{\omega}_x \\ \tau_y &= 2K_{py} \cdot \epsilon_{2e} \cdot \eta_e + K_{dy} \cdot \hat{\omega}_y \\ \tau_z &= 2K_{pz} \cdot \epsilon_{3e} \cdot \eta_e + K_{dz} \cdot \hat{\omega}_z \end{aligned} \quad (4.2)$$

$$\mathbf{q}_e = \mathbf{q}_s^{-1} \cdot \mathbf{q}_t \quad (4.3)$$

Donde:

- $\tau_x, \tau_y, \tau_z$  son los pares de control para cada uno de los actuadores en cada eje.

#### 4. SIMULACIONES Y RESULTADOS

---

- $K_{px}, K_{py}, K_{pz}$  son los valores de ganancia proporcional para su respectivo actuador.
- $K_{dx}, K_{dy}, K_{dz}$  son los valores de ganancia derivativa para su respectivo actuador.
- $\eta_e, \epsilon_{1e}, \epsilon_{2e}, \epsilon_{3e}$  son las componentes del cuaternión de error.
- $\hat{\omega}_x, \hat{\omega}_y, \hat{\omega}_z$  son las velocidades angulares estimadas del simulador.
- $\mathbf{q}_e$  es el cuaternión de error.
- $\mathbf{q}_t$  es el cuaternión de *target* que especifica la orientación que debe alcanzar el simulador o satélite con la maniobra de control.
- $\mathbf{q}_s$  es el cuaternión estimado que especifica la orientación del simulador o satélite.

Sobre lo anterior pueden hacerse algunos comentarios: como ocurre con cualquier otro controlador PD, las ganancias se ajustan de forma arbitraria buscando lograr la mejor respuesta del sistema; todos los parámetros estimados son arrojados por el Filtro de Kalman, por ejemplo, la velocidad angular estimada es generada por el Filtro de Kalman a partir de los datos crudos enviados por los giróscopos.

En [22] se simularon e implementaron diferentes algoritmos de estimación y filtrado, en particular se comparó el desempeño sobre un *DSP* del algoritmo de Inmersión e Invarianza (*I&I*) con respecto al algoritmo SR-UKF; parte del fruto de dicho trabajo se retoma en esta tesis al sintonizar la nueva unidad de medición inercial con el código para simulación del UKF. Para conocer la teoría detrás de estos algoritmos así como sus ventajas y desventajas se recomienda consultar a [22], a continuación se menciona sólo lo indispensable para sintonizar la IMU con el filtro.

Para entender el proceso de sintonización del Filtro de Kalman es necesario recurrir al modelo en el espacio de estados, en este contexto también se les conoce como el modelo del proceso y el modelo de medición. El modelo del proceso corresponde a la expresión donde se halla inmersa la dinámica del sistema a través de la matriz del sistema y el vector de estados. El modelo de medición corresponde a la expresión donde se halla la matriz de salidas junto al vector de estados. A esta representación se le añade un modelo del ruido del proceso y un modelo del ruido de la medición, por simplicidad se asume que el ruido tiene la propiedad de ser un proceso de ruido blanco gaussiano. El modelo del proceso y el modelo de medición se ilustran a continuación:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{F}\mathbf{x} + \mathbf{w} \\ \mathbf{z} &= \mathbf{H}\mathbf{x} + \mathbf{v}\end{aligned}\tag{4.4}$$

$$\mathbf{x} = [\boldsymbol{\omega} \quad \mathbf{q}]^T = [\omega_x \quad \omega_y \quad \omega_z \quad \eta \quad \epsilon_1 \quad \epsilon_2 \quad \epsilon_3]^T\tag{4.5}$$

Donde:

- $\mathbf{F}$  es la matriz del sistema.

- 
- $\mathbf{x}$  es el vector de estados cuyas componentes son las velocidades angulares en cada eje y las componentes del cuaternión.
  - $\mathbf{w}$  es el vector del ruido del proceso y representa la incertidumbre que se tiene con el modelo del sistema.
  - $\mathbf{z}$  es el vector de salidas.
  - $\mathbf{H}$  es la matriz de salida.
  - $\mathbf{v}$  es el vector del ruido de la medición y describe el ruido de los sensores.

Debido a que se trabaja con sistemas digitales, se requiere presentar el modelo en tiempo discreto [22].

$$\begin{aligned}\mathbf{x}_{k+1} &= \Phi_k \mathbf{x}_k + \mathbf{w}_k \\ \mathbf{z}_k &= \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k\end{aligned}\tag{4.6}$$

Puede verse que el modelo en tiempo discreto es esencialmente el mismo que el modelo continuo, en algunos casos tan sólo se hace un cambio en la notación. Los vectores  $\mathbf{x}_k$  y  $\mathbf{z}_k$  son el vector de estados y el vector de salidas en el tiempo  $t_k$  respectivamente,  $\Phi_k$  es la matriz de transición que describe como el sistema cambia en el tiempo  $t_{k+1}$  a partir de los estados del sistema en el tiempo  $t_k$ ,  $\mathbf{H}_k$  es la matriz de salida. Finalmente,  $\mathbf{w}_k$  y  $\mathbf{v}_k$  son los vectores del ruido del proceso y de la medición respectivamente, estos dos vectores se describen como procesos estocásticos a través de dos matrices de covarianza,  $\mathbf{Q}$  y  $\mathbf{R}$ , que indican el nivel de aleatoriedad del ruido [22].

La tarea de sintonizar el filtro consiste en encontrar los valores adecuados para la matriz de covarianza del proceso  $\mathbf{Q}$  y la matriz de covarianza de la medición  $\mathbf{R}$ . Los elementos de la matriz de covarianza de la medición  $\mathbf{R}$  se ajustan de manera estadística mientras que los elementos de la matriz de covarianza del proceso  $\mathbf{Q}$  se ajustan por “prueba y error”.

Para hallar los elementos de la matriz de covarianza de la medición se necesita hacer un muestreo con la IMU en reposo mientras el sistema de adquisición de datos captura una cantidad de datos estadísticamente significativa, para esto se requieren de al menos 2,000 puntos de datos. Esto último tiene que ver con el periodo de muestreo, se tiene un periodo de muestreo de 15.36 [ms] por lo que con 2,496 puntos de datos que fueron capturados por la tarjeta, se tiene un tiempo de simulación total de 38.32 [s]. Como se comentó anteriormente, toda esta información de los sensores se depositó en una matriz de datos y finalmente fue almacenada en un archivo `.dat` para poder ser utilizado por cualquier otro *script* que requiera de la información.

Fue así que se programó en MATLAB un pequeño *script* para calcular los elementos de la matriz de covarianza de la medición, este pequeño *script* toma la matriz de datos del archivo `.dat` y selecciona las columnas para asociarlas a una variable representativa de cada sensor; de esta manera se tiene a los datos correspondientes a cada

#### 4. SIMULACIONES Y RESULTADOS

---

sensor organizados con su respectiva variable. Finalmente, esto permite poder calcular la covarianza que existe en los datos arrojados por cada sensor y acomodar el resultado dentro de la matriz de covarianza de la medición.

La matriz de covarianza de la medición tiene la forma siguiente:

$$\mathbf{R} = \begin{bmatrix} \sigma_{gx}^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{gy}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{gz}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{\eta}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{\epsilon_1}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{\epsilon_2}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{\epsilon_3}^2 \end{bmatrix} \quad (4.7)$$

Se obtuvieron los siguientes valores para los elementos de la matriz:

- $\sigma_{gx}^2 = 0.0261$
- $\sigma_{gy}^2 = 0.0153$
- $\sigma_{gz}^2 = 0.0138$
- $\sigma_{\eta}^2 = 4.7072 \times 10^{-8}$
- $\sigma_{\epsilon_1}^2 = 2.2130 \times 10^{-5}$
- $\sigma_{\epsilon_2}^2 = 8.6335 \times 10^{-6}$
- $\sigma_{\epsilon_3}^2 = 1.4149 \times 10^{-5}$

Puede verse que cada elemento de la diagonal principal corresponde a la varianza en los datos de cada sensor y los elementos se encuentran agrupados en el mismo orden que en el vector de estados.

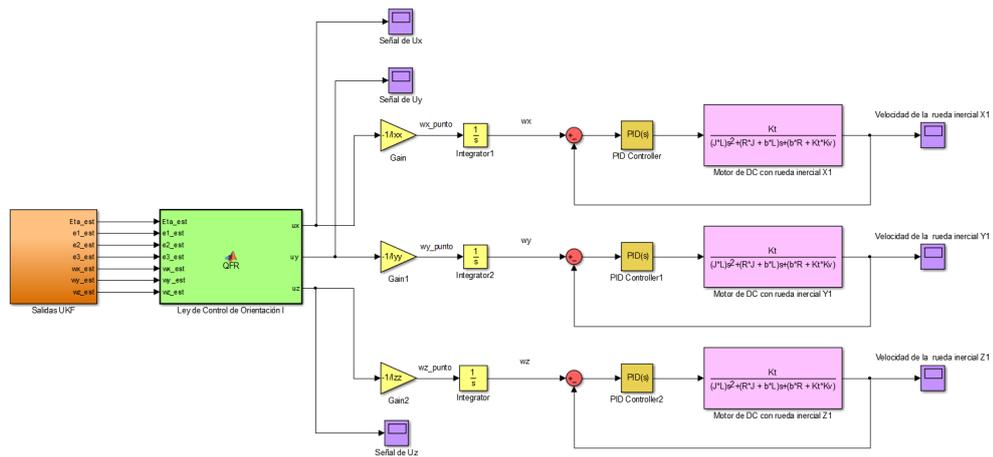
Por su parte, como se comentó, los elementos de la matriz de covarianza del proceso se ajustan por “prueba y error”. Una primera aproximación es partir de los valores encontrados para la matriz de covarianza de la medición y alejarse con pequeños incrementos o decrementos del valor original, simular y comparar el desempeño del filtro con respecto a las señales de datos crudos de los sensores. Se busca que las señales del filtro se asemejen a las señales de datos crudos pero supriman el ruido presente en ellas.  $\mathbf{Q}$  es una matriz diagonal de las mismas dimensiones que  $\mathbf{R}$ , los valores encontrados para la matriz de covarianza del proceso  $\mathbf{Q}$  fueron los siguientes:

- $q_{11} = 0.04$
- $q_{22} = 0.010$
- $q_{33} = 0.020$

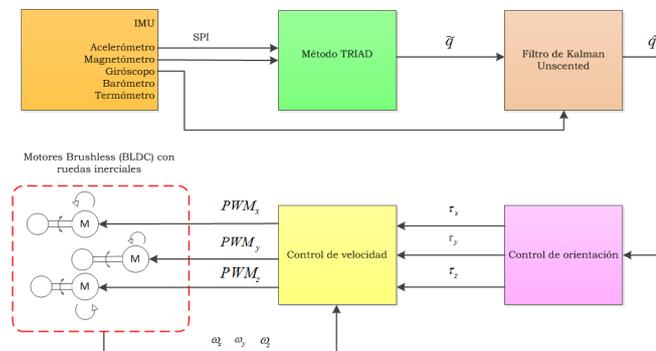
- $q_{44} = 4.85 \times 10^{-8}$
- $q_{55} = 1.4 \times 10^{-5}$
- $q_{66} = 1.9 \times 10^{-6}$
- $q_{77} = 1.3 \times 10^{-5}$

Con estos valores se encontró que el filtro mostró el mejor desempeño en estimar las señales y suprimir el ruido inherente a los sensores. Con este preámbulo concluido se pueden presentar finalmente los resultados de las simulaciones entre el algoritmo UKF, las leyes de control propuestas y la respuesta de los actuadores.

Primeramente, se presenta el diagrama general construido en Simulink para ejecutar las simulaciones el cual contiene todos los elementos presentes en el diagrama de bloques del subsistema de determinación y control de orientación (*ADCS*):



**Figura 4.9:** Diagrama de bloques en Simulink para la ejecución de las simulaciones.

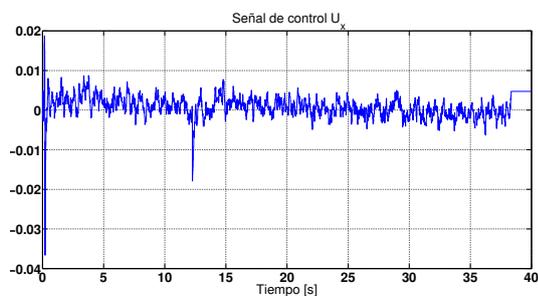


**Figura 4.10:** Diagrama de bloques del *ADCS*.

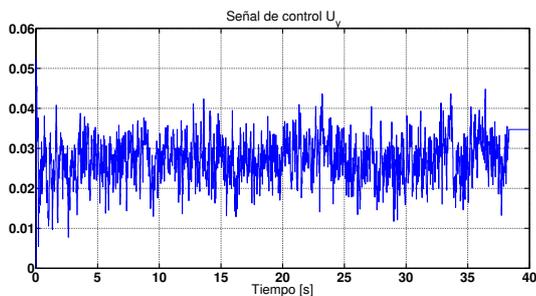
#### 4. SIMULACIONES Y RESULTADOS

---

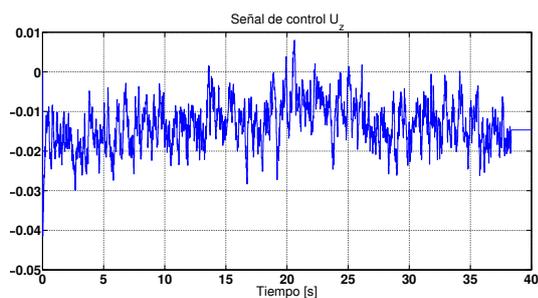
Las gráficas que se presentan a continuación muestran los resultados al emplear la ley de control (4.1):



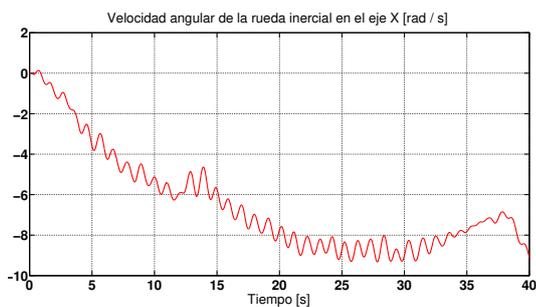
(a) Señal de control para el actuador en el eje x.



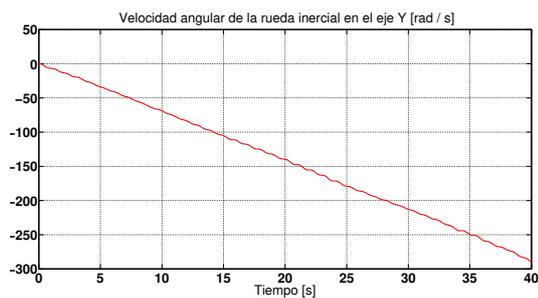
(b) Señal de control para el actuador en el eje y.



(c) Señal de control para el actuador en el eje z.



(d) Velocidad angular de la rueda inercial en x.



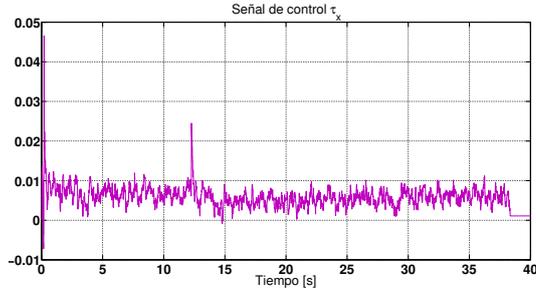
(e) Velocidad angular de la rueda inercial en y.



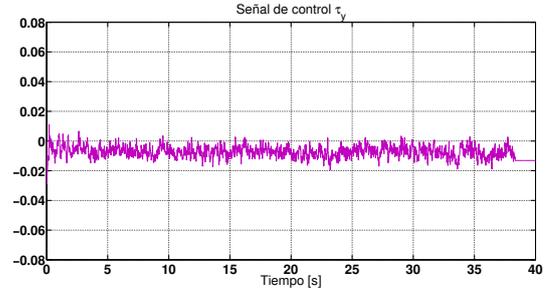
(f) Velocidad angular de la rueda inercial en z.

**Figura 4.11:** Resultados de las simulaciones con la ley de control (4.1).

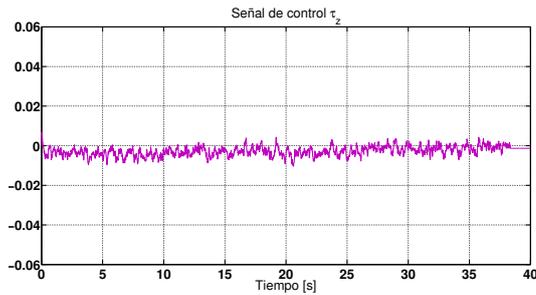
Este grupo de simulaciones corresponde a los resultados obtenidos al utilizar las leyes de control (4.2):



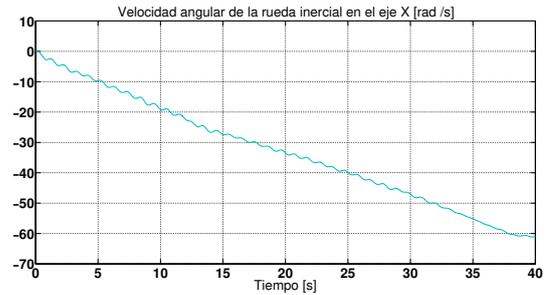
(a) Señal de control para el actuador en el eje x.



(b) Señal de control para el actuador en el eje y.



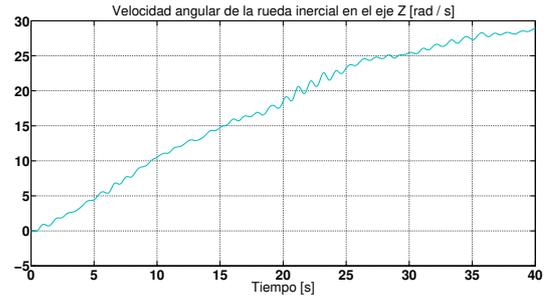
(c) Señal de control para el actuador en el eje z.



(d) Velocidad angular de la rueda inercial en x.



(e) Velocidad angular de la rueda inercial en y.

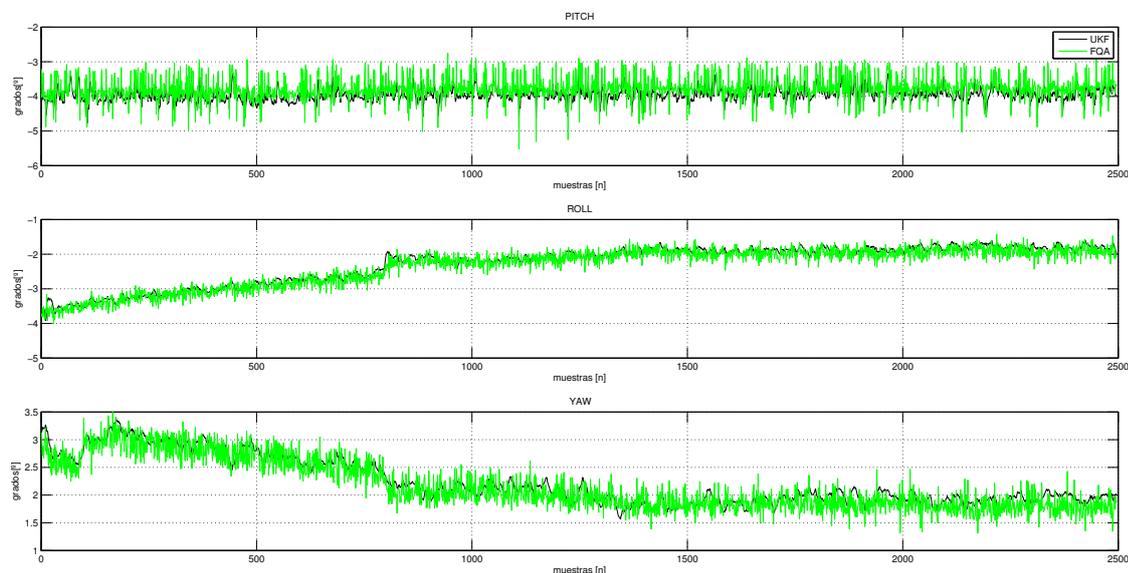


(f) Velocidad angular de la rueda inercial en z.

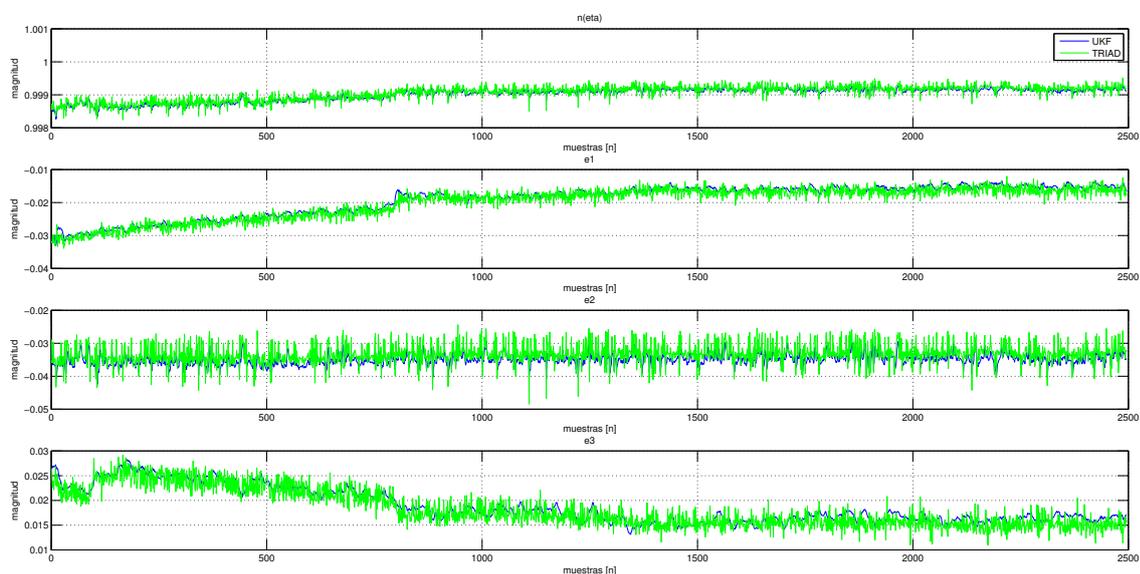
**Figura 4.12:** Resultados de las simulaciones con la ley de control (4.2).

#### 4. SIMULACIONES Y RESULTADOS

El desempeño del algoritmo UKF en su tarea de suavizar las señales “crudas” provenientes de la unidad de medición inercial se muestra a continuación:

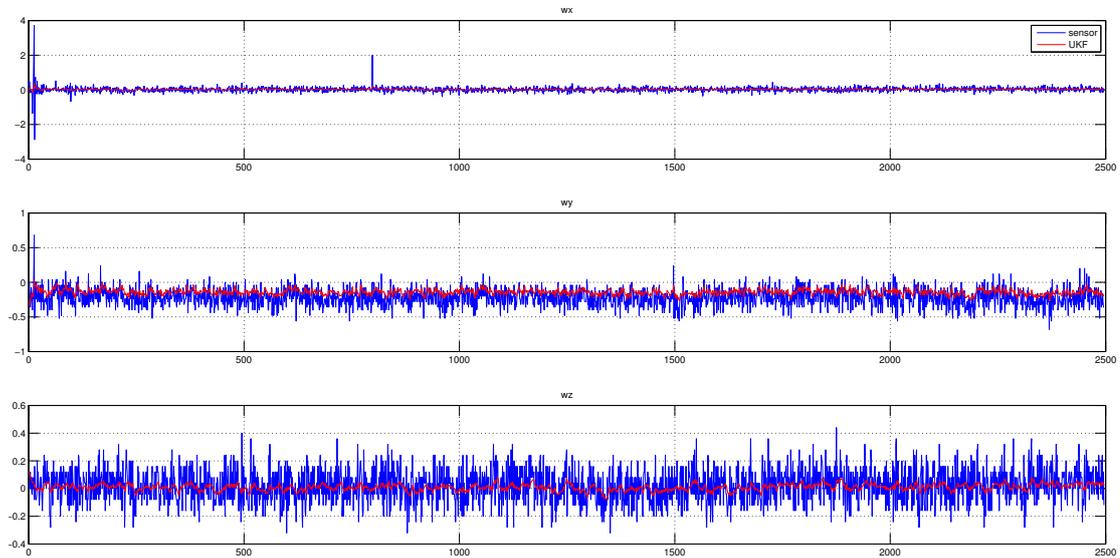


**Figura 4.13:** Procesamiento de los ángulos de orientación de la IMU por el UKF.



**Figura 4.14:** Comparación del método TRIAD con el algoritmo UKF en la generación del cuaternión.

La orientación de la unidad de medición inercial con la que fueron adquiridos los datos de los sensores hace que el sistema de referencia del simulador dado por el sistema



**Figura 4.15:** Velocidad angular en los ejes de la IMU.

de referencia fijo a la IMU coincide con el sistema de referencia inercial seleccionado lo que genera para dicha orientación al cuaternión identidad  $\mathbf{q} = [1 \ 0 \ 0 \ 0]$ . Puede verse que ambas estrategias de control se desempeñan adecuadamente: la primera intenta mantener estabilizado al simulador pues su objetivo es tratar de llevarlo a la orientación  $\mathbf{q} = [1 \ 0 \ 0 \ 0]$ , dado que la IMU que representa al simulador ya se encuentra en dicha orientación, la señal de control prácticamente no actúa. Se observa la velocidad angular de las ruedas inerciales que deben girar para mantener la orientación del simulador. La segunda estrategia es más versátil por lo que se programó como *target* de control a la orientación dada por el cuaternión identidad, esto con la finalidad de observar la respuesta del controlador al contar con la unidad de medición inercial en dicha orientación. Los resultados son similares que en la prueba anterior, la señal de control prácticamente no realiza esfuerzo y las ruedas inerciales giran para mantener la orientación.

Con esto puede decirse que ambas estrategias deben ser consideradas como candidatas para ser implementadas en las versiones futuras del simulador. Sin embargo, para un satélite pequeño sería recomendable adoptar la segunda estrategia (4.2) pues permite programar la orientación de un blanco de interés al cual se busque apuntar y el controlador con ayuda del filtro UKF, tratará de orientar el satélite al punto solicitado.



---

## Capítulo 5

# Conclusiones

---

Al final de este trabajo de tesis se generaron diferentes productos principalmente en el área de instrumentación electrónica además de haber asentado un punto de partida para comenzar estudios en el área de control.

Como parte de esta tesis se completaron los diseños de las nuevas tarjetas electrónicas del simulador satelital MSA. En particular, se construyó la nueva tarjeta de drivers para motores BLDC y fue probada con diferentes programas que ponen en marcha a un motor BLDC. Esto motivó, como parte de otro proyecto, la creación de una versión preliminar de un controlador PD para el control de velocidad de motores BLDC. Más aún, la electrónica es flexible y permite conectar un motor de corriente directa convencional.

Por su parte, los diseños de las tarjetas de potencia quedaron concluidos y listos para manufacturar además se llevaron a cabo simulaciones que muestran que los circuitos seleccionados cumplen con los requerimientos de potencia eléctrica del simulador.

Otro de los productos de esta tesis fue la creación de un sistema de adquisición de datos que logró comunicar la nueva unidad de medición inercial con la tarjeta de desarrollo Zedboard que fungirá como la nueva computadora a bordo. Esto permite, que otros trabajos, puedan comenzar a ampliar el código creado, dando inicio a la integración de algoritmos de estimación y control al tener disponible la información de los sensores.

En cuanto a la parte de investigación se seleccionaron dos estrategias de control de la literatura de esta área, estas estrategias se acoplaron con el UKF sintonizado para operar con la nueva unidad inercial. Se realizaron simulaciones que muestran que dichas estrategias pueden ser implementadas con relativa facilidad y que responden adecuadamente; en las simulaciones se tomó en cuenta los parámetros de los nuevos motores.

Como trabajo a futuro se pueden desarrollar las siguientes líneas de estudio:

- Seleccionar e implementar un sistema operativo en tiempo real que permita aprovechar de una mejor manera los recursos disponibles en el *SoC* así como la programación de tareas que mejoren el desempeño del simulador en las maniobras de control.
- Diseñar y manufacturar las nuevas ruedas inerciales considerando las dimensiones, masa y demás parámetros de los motores BLDC con los que ahora se cuenta.

## 5. CONCLUSIONES

---

- Completar el modelo dinámico del simulador y llevarlo a una forma apropiada para ser incluido en simulaciones.
- Realizar el análisis dinámico del simulador junto a los bloques de estimación y control para realizar simulaciones con diferentes técnicas de control y diferentes comandos de orientación de manera que se pueda conocer el desempeño esperado del simulador. Validar experimentalmente los resultados obtenidos en las simulaciones al implementar sobre la computadora a bordo los algoritmos de estimación y control.
- Aprovechar las capacidades de la nueva instrumentación para enviar telemetría con la información de las señales más relevantes como las señales provenientes de sensores, la orientación, el error, entre otras. De manera que puedan ser visualizadas en un monitor externo que permita evaluar el desempeño del simulador.

De manera personal, mi intención con este trabajo es dejar una guía práctica que sea de utilidad para los diferentes grupos de desarrollo de sistemas aeroespaciales que han ido apareciendo en el ecosistema de la universidad en los últimos años al mostrar un poco de la experiencia en cuanto a desarrollos experimentales que se lograron concretar en el grupo de trabajo del Instituto de Ingeniería. En particular que sea una ayuda para la persona que se encuentre trabajando en el subsistema de estimación y control de orientación.

---

## Apéndice A

# Apéndice

---

### A.1. Cuaterniones

Los cuaterniones fueron descubiertos en 1843 por Sir William Rowan Hamilton un matemático, físico y astrónomo de origen irlandés. Hamilton hizo importantes contribuciones a las matemáticas y la física, algunos de sus trabajos más conocidos son la reformulación de la mecánica newtoniana conocida como mecánica hamiltoniana y por supuesto los cuaterniones. Hamilton buscaba maneras de extender los números complejos a dimensiones superiores, finalmente al trabajar con cuatro dimensiones descubrió los cuaterniones. Existe la anécdota de que Hamilton paseaba un 16 de octubre de 1843 junto a su esposa al lado del Canal Real en Dublín cuando la ecuación que había estado buscando vino a su mente, se apresuró y la grabó con su navaja en el puente Brougham. Esta ecuación que marca el descubrimiento de los cuaterniones es la siguiente:

$$i^2 = j^2 = k^2 = i \cdot j \cdot k = -1 \quad (\text{A.1})$$

Fue hasta 1985 cuando Ken Shoemake presentó la idea de utilizar cuaterniones para computación gráfica, desde entonces los cuaterniones son herramientas comunes para realizar rotaciones y representar orientaciones en gráficos por computadora, en robótica y en el campo aeroespacial. Por ejemplo, lo más común es que el sistema de control de orientación de cualquier vehículo espacial funcione en términos de cuaterniones.

Un cuaternión es una herramienta para representar la orientación de un cuerpo y efectuar rotaciones sobre él. Presenta dos importantes ventajas frente a la matrices de rotación: la primera es que es una estructura computacionalmente más eficiente pues sólo posee cuatro términos y se representa fácilmente como un vector, la segunda es que utilizar cuaterniones evita la aparición de singularidades al realizar cierto tipo de maniobras.

Se dice que un cuaternión posee una parte escalar llamada  $\eta$  y una parte vectorial dada por  $\epsilon$  y se escribe de la siguiente manera:

$$\mathbf{q} = \eta + \epsilon_1 \cdot i + \epsilon_2 \cdot j + \epsilon_3 \cdot k \quad (\text{A.2})$$

Usualmente se escribe a los cuaterniones en forma matricial como se muestra a

continuación:

$$\mathbf{q} = \begin{bmatrix} \eta \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \\ \lambda_x \text{sen}\left(\frac{\theta}{2}\right) \\ \lambda_y \text{sen}\left(\frac{\theta}{2}\right) \\ \lambda_z \text{sen}\left(\frac{\theta}{2}\right) \end{bmatrix} \quad (\text{A.3})$$

Se observa que  $\eta$  y  $\epsilon$  se expresan en términos de las componentes de un vector unitario llamado  $\boldsymbol{\lambda}$  y un ángulo de rotación llamado  $\theta$ , con los cuales se representa la orientación de un cuerpo rígido. A esta representación también se le conoce como representación en parámetros de Euler.

Por lo general se utilizan cuaterniones unitarios, es decir aquellos que tienen norma igual a 1. Para un cuaternión unitario se cumple que:

$$\eta^2 + \epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2 = 1 \quad (\text{A.4})$$

$$\boldsymbol{\epsilon}^T \boldsymbol{\epsilon} + \eta^2 = 1 \quad (\text{A.5})$$

$$\mathbf{q}^T \mathbf{q} = 1 \quad (\text{A.6})$$

Algunas propiedades y operaciones de los cuaterniones unitarios son las siguientes:

- Norma de un cuaternión unitario

$$\|\mathbf{q}\| = 1 \quad (\text{A.7})$$

- Conjugado de un cuaternión

$$\bar{\mathbf{q}} = \begin{bmatrix} \eta \\ -\epsilon_1 \\ -\epsilon_2 \\ -\epsilon_3 \end{bmatrix} \quad (\text{A.8})$$

- Inverso de cuaternión

$$\mathbf{q}^{-1} = \frac{\bar{\mathbf{q}}}{\|\mathbf{q}\|} \quad (\text{A.9})$$

- Multiplicación de cuaterniones

- Sean los cuaterniones  $\mathbf{q}_A = \eta_a + \epsilon_{1a} \cdot i + \epsilon_{2a} \cdot j + \epsilon_{3a} \cdot k$  y  $\mathbf{q}_B = \eta_b + \epsilon_{1b} \cdot i + \epsilon_{2b} \cdot j + \epsilon_{3b} \cdot k$ , la multiplicación de estos se puede calcular como:

$$\mathbf{q} = \mathbf{q}_A \cdot \mathbf{q}_B \quad (\text{A.10})$$

$$\mathbf{q} = \begin{bmatrix} \eta_b & \epsilon_{3b} & -\epsilon_{2b} & \epsilon_{1b} \\ -\epsilon_{3b} & \eta_b & \epsilon_{1b} & \epsilon_{2b} \\ \epsilon_{2b} & -\epsilon_{1b} & \eta_b & \epsilon_{3b} \\ -\epsilon_{1b} & -\epsilon_{2b} & -\epsilon_{3b} & \eta_b \end{bmatrix} \begin{bmatrix} \epsilon_{1a} \\ \epsilon_{2a} \\ \epsilon_{3a} \\ \eta_a \end{bmatrix} \quad (\text{A.11})$$

Para conocer más sobre las operaciones y propiedades de los cuaterniones y su relación con las matrices de rotación se pueden consultar los siguientes trabajos [28], [22] y [3].

## A.2. Método TRIAD

El método TRIAD (Triaxial Attitude Determination) es un método algebraico (determinístico) desarrollado en 1964 que permite conocer la orientación de un objeto. Para ello, emplea dos pares de vectores llamados vectores de medición y vectores de referencia, por lo que emplea dos sistemas coordenados.

El primer paso del método consiste en encontrar un conjunto de vectores ortogonales entre sí llamados  $\mathbf{r}_1$ ,  $\mathbf{r}_2$  y  $\mathbf{r}_3$  en el sistema coordenado donde se realizó la medición. Estos vectores se obtienen de la siguiente manera:

$$\mathbf{r}_1 = \frac{\mathbf{a}_m}{\|\mathbf{a}_m\|} \quad (\text{A.12})$$

$$\mathbf{r}_2 = \frac{\mathbf{r}_1 \times \mathbf{B}_m}{\|\mathbf{r}_1 \times \mathbf{B}_m\|} \quad (\text{A.13})$$

$$\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2 \quad (\text{A.14})$$

Donde  $\mathbf{a}_m$  y  $\mathbf{B}_m$  son las lecturas de aceleración y campo magnético proporcionadas por el acelerómetro y magnetómetro triaxiales. El segundo paso es encontrar un conjunto de vectores ortogonales entre sí llamados  $\mathbf{s}_1$ ,  $\mathbf{s}_2$  y  $\mathbf{s}_3$  en el sistema coordenado de referencia. Estos vectores se calculan de la siguiente manera:

$$\mathbf{s}_1 = \frac{\mathbf{a}_r}{\|\mathbf{a}_r\|} \quad (\text{A.15})$$

$$\mathbf{s}_2 = \frac{\mathbf{s}_1 \times \mathbf{B}_r}{\|\mathbf{s}_1 \times \mathbf{B}_r\|} \quad (\text{A.16})$$

$$\mathbf{s}_3 = \mathbf{s}_1 \times \mathbf{s}_2 \quad (\text{A.17})$$

Donde  $\mathbf{a}_r$  y  $\mathbf{B}_r$  son los vectores de referencia de aceleración gravitacional y campo magnético. El paso final es obtener una matriz de cosenos directores que relacione ambos sistemas coordenados. Esta matriz se calcula de la siguiente forma:

$$\mathbf{A}_m^r = \mathbf{r}_1 \cdot \mathbf{s}_1^T + \mathbf{r}_2 \cdot \mathbf{s}_2^T + \mathbf{r}_3 \cdot \mathbf{s}_3^T \quad (\text{A.18})$$

Este método trabaja junto al filtro de Kalman para abordar el problema de la estimación de la orientación pues es posible obtener a partir de la matriz de cosenos directores un cuaternión llamado cuaternión de medición que es recibido a la entrada del filtro.

Para el simulador MSA se utiliza el acelerómetro y el magnetómetro para obtener los vectores de medición. De igual forma al elegir estos sensores es posible definir los vectores de referencia. Para definir el vector de referencia asociado al acelerómetro se utiliza el vector de aceleración gravitacional y para el vector de referencia asociado al magnetómetro se puede utilizar al vector de campo magnético correspondiente al norte

magnético de la Tierra o bien al vector de campo magnético de cualquier punto que sea conveniente.

Debe aclararse que en un satélite real no se emplea el acelerómetro pues evidentemente no es posible plantear al vector de referencia como el vector de aceleración gravitacional en la Tierra dado que en el espacio es mucho más débil. Esto obliga que para un satélite real se deba utilizar otro tipo de sensores como los llamados sensores de Sol o bien sensores de estrellas. De igual manera para el magnetómetro se deben tener en cuenta ciertas consideraciones pues el vector de norte magnético no es fijo en la superficie de la Tierra [22].

Finalmente, para obtener el cuaternión de medición a partir de la matriz de cosenos directores se emplea el siguiente método:

- Se obtienen las componentes del vector  $\mathbf{V}$  a partir de los elementos de la matriz de cosenos directores:

$$v_x = a_{23} - a_{32} \quad (\text{A.19})$$

$$v_y = a_{31} - a_{13} \quad (\text{A.20})$$

$$v_z = a_{12} - a_{21} \quad (\text{A.21})$$

$$\mathbf{V} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \quad (\text{A.22})$$

- Se calcula un parámetro  $\gamma$ , a partir de la traza de la matriz de cosenos directores:

$$\gamma = \frac{\text{tr}(A) - 1}{2} \quad (\text{A.23})$$

- Con  $\gamma$  se obtiene a  $\theta$ , el ángulo de rotación del cuerpo rígido:

$$\theta = \text{angcos}(\gamma) \quad (\text{A.24})$$

- Se normaliza al vector  $\mathbf{V}$  para obtener al vector  $\boldsymbol{\lambda}$  que representa al vector unitario en torno al cual se parametriza la rotación del cuerpo rígido.

$$\boldsymbol{\lambda} = \frac{\mathbf{V}}{\|\mathbf{V}\|} \quad (\text{A.25})$$

- Se calcula la parte escalar del cuaternión:

$$\eta = \cos\left(\frac{\theta}{2}\right) \quad (\text{A.26})$$

- Se calcula la parte vectorial del cuaternión:

$$\boldsymbol{\epsilon} = \boldsymbol{\lambda} \text{sen}\left(\frac{\theta}{2}\right) \quad (\text{A.27})$$

- Finalmente, se puede escribir al cuaternión de medición:

$$\mathbf{q}_m = \begin{bmatrix} \eta \\ \boldsymbol{\epsilon} \end{bmatrix} \quad (\text{A.28})$$

## A.3. Códigos

### A.3.1. Código en C para interactuar con la IMU ADIS16448

El código que se presenta a continuación corre sobre la arquitectura implementada sobre la tarjeta Zedboard y permite la comunicación con la IMU ADIS16448 para recibir la información de los sensores.

---

```
#include <stdio.h>
#include <stdlib.h>
#include "xparameters.h"
#include "platform.h"
#include "xspi.h"
#include "xgpiops.h"
#include "sleep.h"

#define SS_MASK    0x0001

#define XGYRO_OUT 0x0400
#define YGYRO_OUT 0x0600
#define ZGYRO_OUT 0x0800

#define XACCL_OUT 0x0A00
#define YACCL_OUT 0x0C00
#define ZACCL_OUT 0x0E00

#define XMAGN_OUT 0x1000
#define YMAGN_OUT 0x1200
#define ZMAGN_OUT 0x1400
```

## A. APÉNDICE

---

```
#define BARO_OUT 0x1600
#define TEMP_OUT 0x1800

#define PROD_ID 0x5600 //Debe transferirse este valor a la IMU por DIN para
    verificar la comunicacion. La IMU responde con 0x4040

#define SERIAL_NUM 0x5800
#define Burst 0x3E00

// Definiciones de Tipo
typedef u16 Datos_Buffer[13];

// Prototipos de Funciones
void comenzar_transferencia_SPI();
unsigned char esta_completa_la_transferencia();
unsigned char esta_buffer_receptor_lleno();
u16 leer_IMU();
void escribir_IMU(u16 value);

// Variables Globales
XSpi IMU_SPI;
u16 datos_de_IMU;
u16 Escribir_Buffer[13];
u16 Leer_Buffer[13];
XGpioPs my_Gpio;

// -----
// Comienza la transferencia con la IMU.
// -----
void comenzar_transferencia_SPI()
{
    //xil_printf("--I'm about to die--\n\r");
    XSpi_SetSlaveSelectReg(&IMU_SPI, ~SS_MASK);
```

```
    XSpi_WriteReg(IMU_SPI.BaseAddr, XSP_DTR_OFFSET, PROD_ID);
}

// -----
// Regresa un valor no nulo si una transferencia con la IMU esta completa.
// La funcion "comenzar_transferencia_SPI()" debe ser llamada antes de esta.
// -----
unsigned char esta_completa_la_transferencia()
{
    return (XSpi_GetStatusReg(&IMU_SPI) & XSP_SR_TX_EMPTY_MASK);
}

// -----
// Regresa un valor no nulo si el buffer de recepcion esta lleno con el dato
// proveniente de la IMU.
// La funcion "comenzar_transferencia_SPI()" debe ser llamada antes de esta.
// -----

unsigned char esta_buffer_receptor_lleno()
{
    return (XSpi_GetStatusReg(&IMU_SPI) & XSP_SR_RX_FULL_MASK);
}

// -----
// Lee los datos recibidos de la IMU despues de que una transferencia se ha
// completado.
// Las funciones "comenzar_transferencia_SPI()" y
// "esta_completa_la_transferencia()" deben ser llamadas antes de esta.
// -----
u16 leer_IMU()
{
    XSpi_SetSlaveSelectReg(&IMU_SPI, SS_MASK);
    return XSpi_ReadReg(IMU_SPI.BaseAddr, XSP_DRR_OFFSET);
}
```

## A. APÉNDICE

---

```
// -----  
// Escribe un valor a la IMU  
// -----  
void escribir_IMU(u16 value)  
{  
    XSpi_WriteReg(IMU_SPI.BaseAddr, XSP_DTR_OFFSET, value);  
}  
  
int main()  
{  
  
    int xStatus;  
    int k;  
  
    XSpi_Config *SPI_Config; //Puntero para configurar el modulo SPI  
    XGpioPs_Config *GPIO_Config; //Puntero para configurar el modulo GPIO  
  
    init_platform();  
    //xil_printf("---Entrando al main---\n\r");  
  
    /* Obteniendo los dos parametros necesarios para configurar el modulo SPI */  
    SPI_Config = XSpi_LookupConfig(XPAR_AXI_QUAD_SPI_0_DEVICE_ID);  
    // Buscar el ID del modulo en xparameters.h  
    if (SPI_Config == NULL){  
        return XST_DEVICE_NOT_FOUND;  
    }  
  
    // -----  
    // Inicializa el modulo SPI para la IMU  
    // -----  
    xStatus = XSpi_CfgInitialize(&IMU_SPI, SPI_Config, SPI_Config->BaseAddress);  
    if (xStatus == XST_SUCCESS){
```

```
    //xil_printf("Modulo SPI inicializado correctamente!\n\r");
}

else{
    //xil_printf("Error con inicializacion del modulo SPI!\n\r");
}

// Se configura el modulo SPI en SPI MODE 3: CPOL = 1 ; CPHA = 1, modo
    maestro, seleccion manual de esclavo.

xStatus = XSpi_SetOptions(&IMU_SPI, XSP_MASTER_OPTION +
XSP_MANUAL_SSELECT_OPTION + XSP_CLK_PHASE_1_OPTION +
XSP_CLK_ACTIVE_LOW_OPTION);

    if (xStatus != XST_SUCCESS){
        return XST_FAILURE;
    }

// Se selecciona el esclavo en el bus SPI de forma que pueda ser leído y
    escrito, utilizando el bus SPI.
XSpi_GetSlaveSelect(&IMU_SPI);

xStatus = XSpi_SetSlaveSelect(&IMU_SPI, SS_MASK);
    if (xStatus != XST_SUCCESS){
        return XST_FAILURE;
    }

// Se inicia al modulo SPI; las interrupciones y el modulo estan
    habilitados.

XSpi_Enable(&IMU_SPI);

/*xStatus = XSpi_Start(&IMU_SPI);
```

```
if (xStatus != XST_SUCCESS){
    return XST_FAILURE;
}*/

//Deshabilita interrupciones globales, se abordara un enfoque por poleo.
XSpi_IntrGlobalDisable(&IMU_SPI);

//Se verifica si las interrupciones globales estan deshabilitadas.
xStatus = XSpi_IsIntrGlobalEnabled(&IMU_SPI);
if (xStatus != TRUE){
    //xil_printf("Las interrupciones globales estan deshabilitadas\n\r");
}

// -----
// Modulo GPIO del Procesador (PS)
// -----

/* Obteniendo los dos parametros necesarios para configurar el modulo GPIO
*/
GPIO_Config = XGpioPs_LookupConfig(XPAR_PS7_GPIO_0_DEVICE_ID);

/* Se inicializa el modulo GPIO de PS */
xStatus = XGpioPs_CfgInitialize(&my_Gpio, GPIO_Config,
    GPIO_Config->BaseAddr);
if (xStatus == XST_SUCCESS){
    //xil_printf("Modulo GPIO inicializado correctamente!\n\r");
}

else{
    //xil_printf("Error con la inicializacion del modulo GPIO!\n\r");
}
```

```
/* Se configura el pin 7 de PS como salida (1) */
XGpioPs_SetDirectionPin(&my_Gpio, 7, 1);

//xil_printf("Hello World\n\r");

/* transfer flag = 0, indica que podemos comenzar una transferencia.
   transfer flag = 1, indica que una transferencia esta en proceso y
   estamos esperando a que termine.
*/
unsigned char transferFlag = 0;

while(1){

    if(transferFlag){

        // Delay de 50 [ms]
        usleep(50000);

        if(esta_completa_la_transferencia()){

            //xil_printf("--Transferencia completa--\n\r");

            /* Comunicacion con la IMU */

            // Delay de 50 [ms]
            usleep(50000);

            // Verificar si el buffer de recepcion esta lleno
            while(!(esta_buffer_receptor_lleno()));

            // Lectura de dato "dummy"
            datos_de_IMU = leer_IMU();
```

```
// Se imprime el dato "dummy" leído, datos_de_IMU, para
    verificar que se tiene comunicacion
//xil_printf("Dato: 0x%04x\n\r",datos_de_IMU);

// Delay de 50 [ms]
usleep(50000);

/* Se transmite un comando "dummy" a la IMU para recibir su ID
    */

XSpi_SetSlaveSelectReg(&IMU_SPI, ~SS_MASK);
escribir_IMU(0x0001);
while(!(esta_completa_la_transferencia()));
usleep(50000);
while(!(esta_buffer_receptor_lleno()));
datos_de_IMU = leer_IMU();

// Se imprime la ID de la IMU
//xil_printf("ID de la IMU: 0x%04x\n\r",datos_de_IMU);

//sleep(2); // Delay de 2 [s]

/* Se transmite comando para iniciar el BURST MODE */

datos_de_IMU = 0;
XSpi_SetSlaveSelectReg(&IMU_SPI, ~SS_MASK);
escribir_IMU(Burst);
while(!(esta_completa_la_transferencia()));
    usleep(50); // Delay de 50 [us]

//Leemos el primer dato proveniente del BURST MODE
//datos_de_IMU = XSpi_ReadReg(IMU_SPI.BaseAddr, XSP_DRR_OFFSET);
//xil_printf("Dato: 0x%04x\n\r",datos_de_IMU);
```

```
// Leemos el resto de mensajes provenientes de la IMU
for (k=0; k<13; k++){

    // Verificar si el buffer de recepcion esta lleno
    while(!(esta_buffer_receptor_lleno()));
    Leer_Buffer[k] = XSpi_ReadReg(IMU_SPI.BaseAddr,
        XSP_DRR_OFFSET);
    escribir_IMU(0x0000);
    while(!(esta_completa_la_transferencia()));
}

XSpi_SetSlaveSelectReg(&IMU_SPI, SS_MASK);
//usleep(500);

/* xStatus = XSpi_Transfer(&IMU_SPI, NULL, Leer_Buffer, 13);
if (xStatus != XST_SUCCESS){
    return XST_FAILURE;
} */

//Se imprimen los valores que se recibieron de la IMU
for (k=1; k<13; k++){
//xil_printf("Dato recibido = 0x%04x\r\n", Leer_Buffer[k]);
xil_printf("%04x!", Leer_Buffer[k]);
}

//usleep(500);

/* for (k = 0; k < longitud_resp; k++) {
XSpi_SetSlaveSelectReg(&IMU_SPI, ~SS_MASK);
escribir_IMU(0x00);
respuesta[k] = XSpi_ReadReg(IMU_SPI.BaseAddr, XSP_DRR_OFFSET);
XSpi_SetSlaveSelectReg(&IMU_SPI, SS_MASK);
} */
```

```
        } */

        transferFlag = 0;
    }
}

else{

    // Delay de 50 [ms]
    usleep(50000);
    comenzar_transferencia_SPI();
    while(!(esta_completa_la_transferencia()));
    transferFlag = 1;
}

//sleep(3); // Delay de 3 [s]
usleep(50000);

}

return 0;
}
```

---

### A.3.2. Código en MATLAB del Sistema de Adquisición de Datos

El código siguiente procesa los datos crudos transmitidos por la tarjeta Zedboard, se les aplica el método TRIAD para obtener el cuaternión de medición y se generan las gráficas para poder visualizar la información resultante.

---

```
% Primero se debe conectar la tarjeta para poder recibir el primer dato.
% Despues, arrancar el programa en Matlab

serial_file = 'Lecturas de la IMU.csv';
```

---

```
if exist(serial_file) == 0 % Si el archivo no existe, pasa a configurar el
    puerto serie para leer datos y crear el archivo.

delete(instrfindall); % Cierra todos los puertos serie abiertos.

s = serial('COM5'); % Se asigna el objeto "s" al puerto serie.

set(s, 'FlowControl', 'hardware');
set(s, 'BaudRate', 115200);
set(s, 'Parity', 'none');
set(s, 'DataBits', 8);
set(s, 'StopBit', 1);
set(s, 'Timeout',10);

disp(get(s,'Name'));
prop(1)=(get(s,'BaudRate'));
prop(2)=(get(s,'DataBits'));
prop(3)=(get(s,'StopBit'));

disp(['Configuracion del puerto, lista', ' ', num2str(prop)]);

fopen(s); % Se abre el puerto

% Se elige modo de lectura en forma asincrona
% lo que permite utilizar la ventana de comandos
% mientras se reciben datos por puerto serie

s.ReadAsyncMode = 'continuous';

readasync(s); % Se inicia una operacion de lectura asincrona en el puerto
    serie
```

## A. APÉNDICE

---

```
while(s.BytesAvailable <= 0) % Espera hasta que la IMU envíe datos

end

N = 2496;      % Numero de renglones. Aqui puede modificarse para tener mas
               muestras.

% Se tendra una matriz de datos de 2496 renglones y 12 columnas,
% por lo tanto de 29,952 elementos

matriz_datos_totales = zeros(N, 12);

% Se tendran 29,952 elementos en la matriz con N = 2,496.

eL = N*12;

% Se plantea el uso de una matriz de 312 x 96
% para almacenar el contenido del vector
% de datos seriales que se actualiza con cada
% ciclo.

datos_crudos = zeros(312,96);

for i = 1:312 % Se haran 312 ciclos con los cuales se busca actualizar el
              contenido del "vector_datos_serie_ajustado".

    datos_serie = fscanf(s); % Leer datos del puerto serie
    flushinput(s);          % Limpiar el buffer de entrada; vaciar datos

    % Se separan los datos del stream del puerto serie,
    % usando "!" para distinguir entre ellos

    vector_datos_serie = regexp(datos_serie, '!', 'split');
```

```
vector_datos_serie_ajustado = vector_datos_serie([1:96]);

% El puerto serie guarda los datos en arreglos de celdas de
    cadenas,
% para poder procesarlos deben transformarse a arreglos comunes
    para
% ello se usa "char", que nos arroja un arreglo de cadenas.

vcharacter_column_datos_serie = char(vector_datos_serie_ajustado);

% Teniendo el vector, "vcharacter_column_datos_serie",
% se puede transformar su contenido a
% decimales signados.

datos_legibles_crudos =
    typecast(uint16(hex2dec(vcharacter_column_datos_serie)), 'int16');

% Se transpone el vector, datos_legibles_crudos, para darle el
    formato que se necesita.

datos_legibles_crudos_T = datos_legibles_crudos' ;

% Se utiliza la matriz "datos_crudos" para poder almacenar en una
    sola estructura
% los diferentes valores que "datos_legibles_crudos_T" va teniendo
    con cada iteracion

datos_crudos(i,:) = datos_legibles_crudos_T;

end

% No se requiere mas adquisicion de datos, entonces se cierra el
```

## A. APÉNDICE

---

```
% puerto.

fclose(s);
delete(instrfindall); % Cerrar todos los puertos serie abiertos
% delete(s);

% Organizacion de datos adquiridos.

disp(['Ready :']);

% Se extrae cada renglon de la matriz "datos_crudos" y se unen en un
  unico vector

for i = 1:312

    d{i} = datos_crudos(i,:);

end

vector_datos_totales = cell2mat(d);

% Con el vector obtenido, vector_datos_totales, se
% acomoda su contenido en la forma de una matriz para tener
% columnas de datos correspondientes a cada sensor.

matriz_datos_totales = vec2mat(vector_datos_totales, 12);

csvwrite(serial_file, matriz_datos_totales); % Guarda los datos
  adquiridos en un archivo CSV
dlmwrite('Lecturas de la IMU.txt', matriz_datos_totales, 'delimiter',
  '\t');
```

```
else

    matriz_datos_totales = csvread(serial_file); % Si existe un archivo csv
        con lecturas previas, se carga

    if exist('Lecturas de la IMU.txt') == 0

        dlmwrite('Lecturas de la IMU.txt', matriz_datos_totales, 'delimiter',
            '\t');

    else

        disp(['El archivo .txt de la matriz de datos ya existe, puede
            utilizarlo']);

    end

end

end

%-----
% Procesamiento
%-----

status = matriz_datos_totales(:,1);

gyroX = matriz_datos_totales(:,2) / 25 ;
gyroY = matriz_datos_totales(:,3) / 25 ;
gyroZ = matriz_datos_totales(:,4) / 25 ;

accX = matriz_datos_totales(:,5) / 1200 ;
accY = matriz_datos_totales(:,6) / 1200 ;
accZ = matriz_datos_totales(:,7) / 1200 ;
```

## A. APÉNDICE

---

```
mgnetX = matriz_datos_totales(:,8) / 7 ;
mgnetY = matriz_datos_totales(:,9) / 7 ;
mgnetZ = matriz_datos_totales(:,10) / 7 ;

press = matriz_datos_totales(:,11) * 0.02 ;

temp = (matriz_datos_totales(:,12) / 13.53912808) + 31 ;

%-----
% Aplicacion del TRIAD
%-----

acc_ref = [0; 0; -9.81]; % Vector de aceleracion de referencia
mag_ref = [mgnetX(1); mgnetY(1); mgnetZ(1)]; % Vector de campo magnetico de
referencia.

i = 0;

acc = [accX, accY, accZ];

mag = [mgnetX, mgnetY, mgnetZ];

quat_med = zeros(4,2496);

for i = 1:2496

quat_med(:,i) = TRIAD(acc_ref, acc(i,:))' , mag_ref, mag(i,:))';

end

% Extrayendo los valores de cada componente del cuaternion.

Eta = quat_med(1,:);
```

```
Ex = quat_med(2,:);
Ey = quat_med(3,:);
Ez = quat_med(4,:);

tam = size(quat_med)

%-----
% Graficando datos
%-----

t = 0 : 15.36 : 38323.2 ; % Escala de tiempo en [ms]
    % N*(15.36), con N = 2495

figure

%-----
% Giroscopos
%-----

subplot(2,3,1)
plot(t, gyroX, 'r')
grid on
title('Giroscopo en X');
xlabel('Tiempo [ms]');
ylabel('Tasa de giro [grados/s]');

subplot(2,3,2)
plot(t, gyroY, 'r')
grid on
title('Giroscopo en Y');
xlabel('Tiempo [ms]');
ylabel('Tasa de giro [grados/s]');
```

```
subplot(2,3,3)
plot(t, gyroZ, 'r')
grid on
title('Giroscopo en Z');
xlabel('Tiempo [ms]');
ylabel('Tasa de giro [grados/s]');
```

```
%-----
% Acelerometros
%-----
```

```
subplot(2,3,4)
plot(t, accX, 'r')
grid on
title('Acelerometro en X');
xlabel('Tiempo [ms]');
ylabel('Aceleracion [g]');
```

```
subplot(2,3,5)
plot(t, accY, 'r')
grid on
title('Acelerometro en Y');
xlabel('Tiempo [ms]');
ylabel('Aceleracion [g]');
```

```
subplot(2,3,6)
plot(t, accZ, 'r')
grid on
title('Acelerometro en Z');
xlabel('Tiempo [ms]');
ylabel('Aceleracion [g]');
```

```
figure
```

```
%-----  
% Magnetometros  
%-----  
  
subplot(1,3,1)  
plot(t, mgnetX, 'r')  
grid on  
title('Magnetometro en X');  
xlabel('Tiempo [ms]');  
ylabel('Campo Magnetico [mG]');  
  
subplot(1,3,2)  
plot(t, mgnetY, 'r')  
grid on  
title('Magnetometro en Y');  
xlabel('Tiempo [ms]');  
ylabel('Campo Magnetico [mG]');  
  
subplot(1,3,3)  
plot(t, mgnetZ, 'r')  
grid on  
title('Magnetometro en Z');  
xlabel('Tiempo [ms]');  
ylabel('Campo Magnetico [mG]');  
  
figure  
  
%-----  
% Barometro  
%-----  
  
subplot(1,2,1)
```

```
plot(t, press, 'r')
grid on
title('Barometro');
xlabel('Tiempo [ms]');
ylabel('Presion Barometrica [mbar]');

%-----
% Termometro
%-----

subplot(1,2,2)
plot(t, temp, 'r')
grid on
title('Termometro');
xlabel('Tiempo [ms]');
ylabel('Temperatura [C]');

%-----
% Componentes del cuaternion
%-----

figure

subplot(2,2,1)
plot(t, Eta, 'r')
grid on
title('Parte escalar del cuaternion de medicion');
xlabel('Tiempo [ms]');
ylabel('Eta');

subplot(2,2,2)
plot(t, Ex, 'r')
grid on
```

```
title('Parte vectorial del cuaternion de medicion');
xlabel('Tiempo [ms]');
ylabel('Ex');

subplot(2,2,3)
plot(t, Ey, 'r')
grid on
title('Parte vectorial del cuaternion de medicion');
xlabel('Tiempo [ms]');
ylabel('Ey');

subplot(2,2,4)
plot(t, Ez, 'r')
grid on
title('Parte vectorial del cuaternion de medicion');
xlabel('Tiempo [ms]');
ylabel('Ez');
```

---

```
% TRIAD (Triaxial Attitude Determination)

% Busca encontrar la orientacion de un cuerpo a partir de dos vectores de
% medicion y dos vectores de referencia.

function quat_med = TRIAD(acc_ref, acc, mag_ref, mag)

    % Encontrar una triada de vectores ortogonales entre si, (r1, r2,
    % r3), en el sistema coordenado donde efectua la medicion.

    r1 = acc / norm(acc);

    r2 = cross(r1, mag) / norm(cross(r1,mag));

    r3 = cross(r1, r2);
```

---

## A. APÉNDICE

---

```
% Encontrar una triada de vectores ortogonales entre si, (s1, s2,
% s3), en el sistema coordenado de referencia.

s1 = acc_ref / norm(acc_ref);

s2 = cross(s1, mag_ref) / norm(cross(s1, mag_ref));

s3 = cross(s1, s2);

% Obtener la matriz de cosenos directores.

A = r1*s1' + r2*s2' + r3*s3';

% Se necesita obtener el cuaternion de medicion para el SR-UKF, para
% ello:

% Se obtienen las componentes de un vector V a partir de los elementos
% de la matriz
% de cosenos directores.

v1 = A(2,3) - A(3,2);
v2 = A(3,1) - A(1,3);
v3 = A(1,2) - A(2,1);

V = [v1; v2; v3];

% Se calcula un parametro "Gamma", a partir de la traza de
% la matriz de cosenos directores.

gamma = (trace(A) - 1)/2;
```

```
% Con "Gamma" se obtiene a Theta, el angulo de rotacion del cuerpo  
% rigido.
```

```
theta = acos(gamma);
```

```
% Se normaliza al vector V para obtener al vector Lambda. Lambda  
    representa el  
% vector unitario en torno al cual se parametriza la rotacion del  
% cuerpo.
```

```
L = V / norm(V);
```

```
% Se calcula la parte escalar del cuaternion.
```

```
Eta = cos(theta/2);
```

```
% Se calcula la parte vectorial del cuaternion.
```

```
Eps = L * sin(theta/2);
```

```
quat_med = [Eta; Eps];
```

---

## A.4. Esquemáticos y Diseños de PCBs

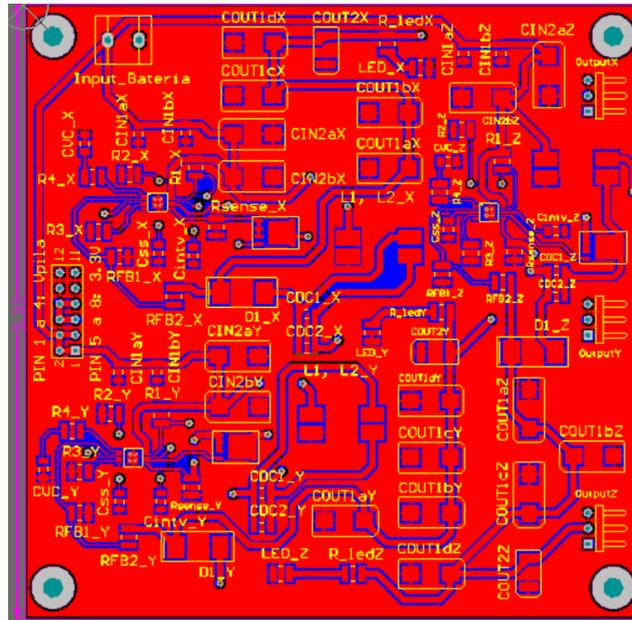


Figura A.1: PCB para la alimentación de motores BLDC con fuentes de 12 [V] / 2 [A].

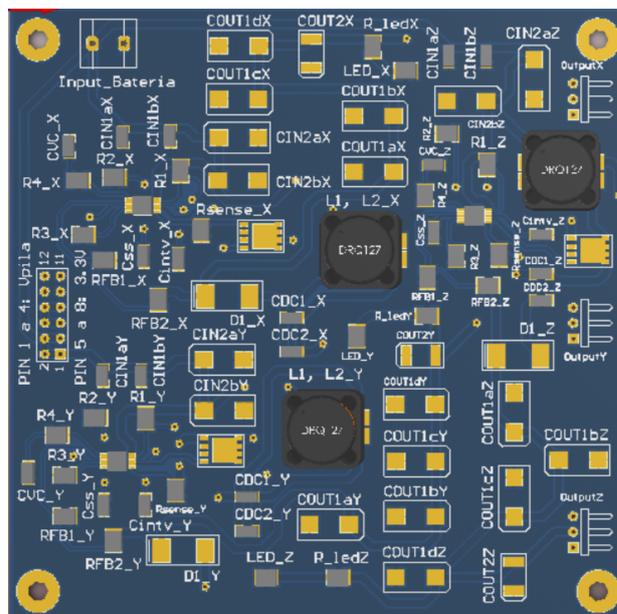


Figura A.2: Vista en 3D de la PCB para la alimentación de motores.



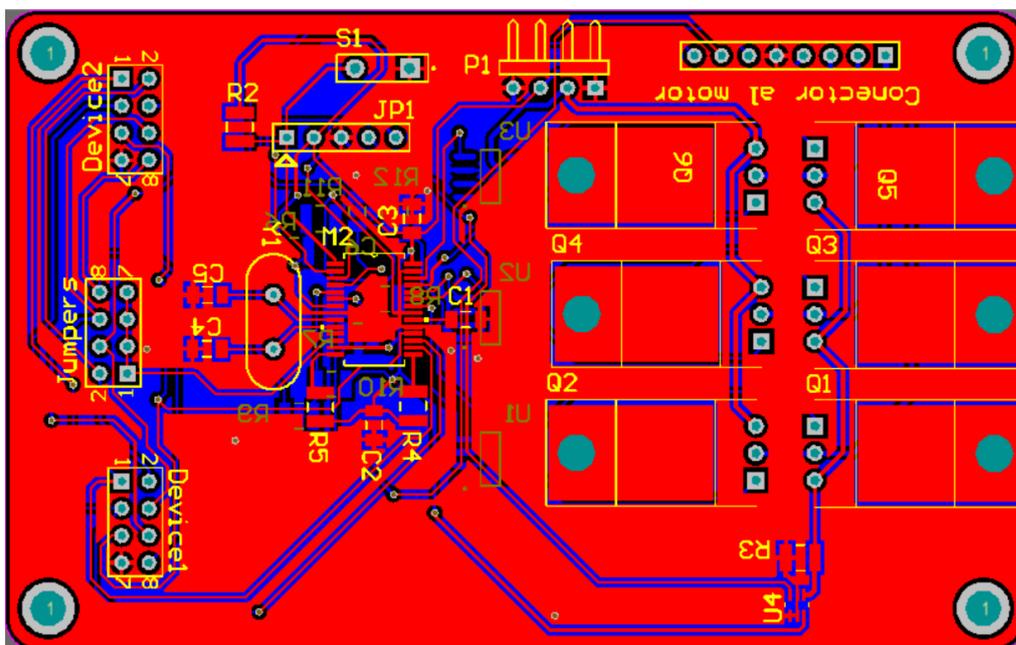


Figura A.5: PCB de drivers para motores BLDC.

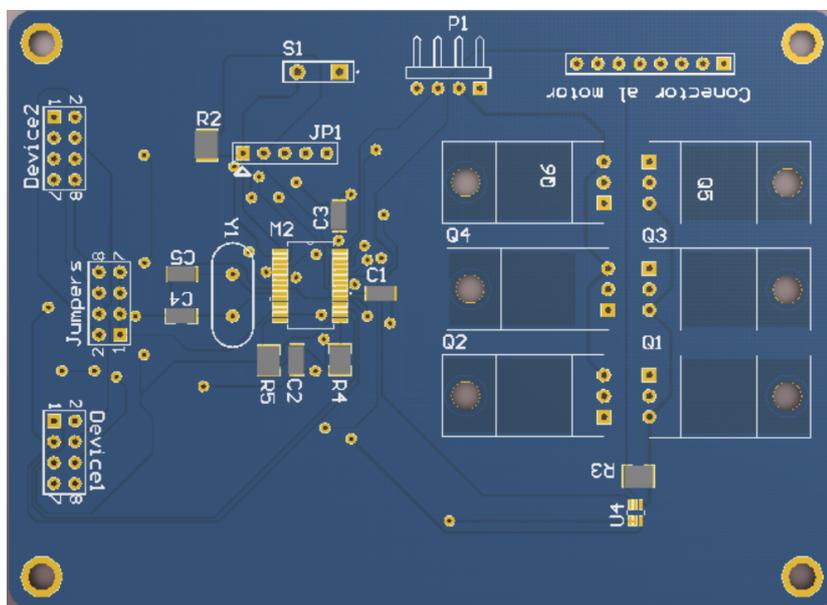
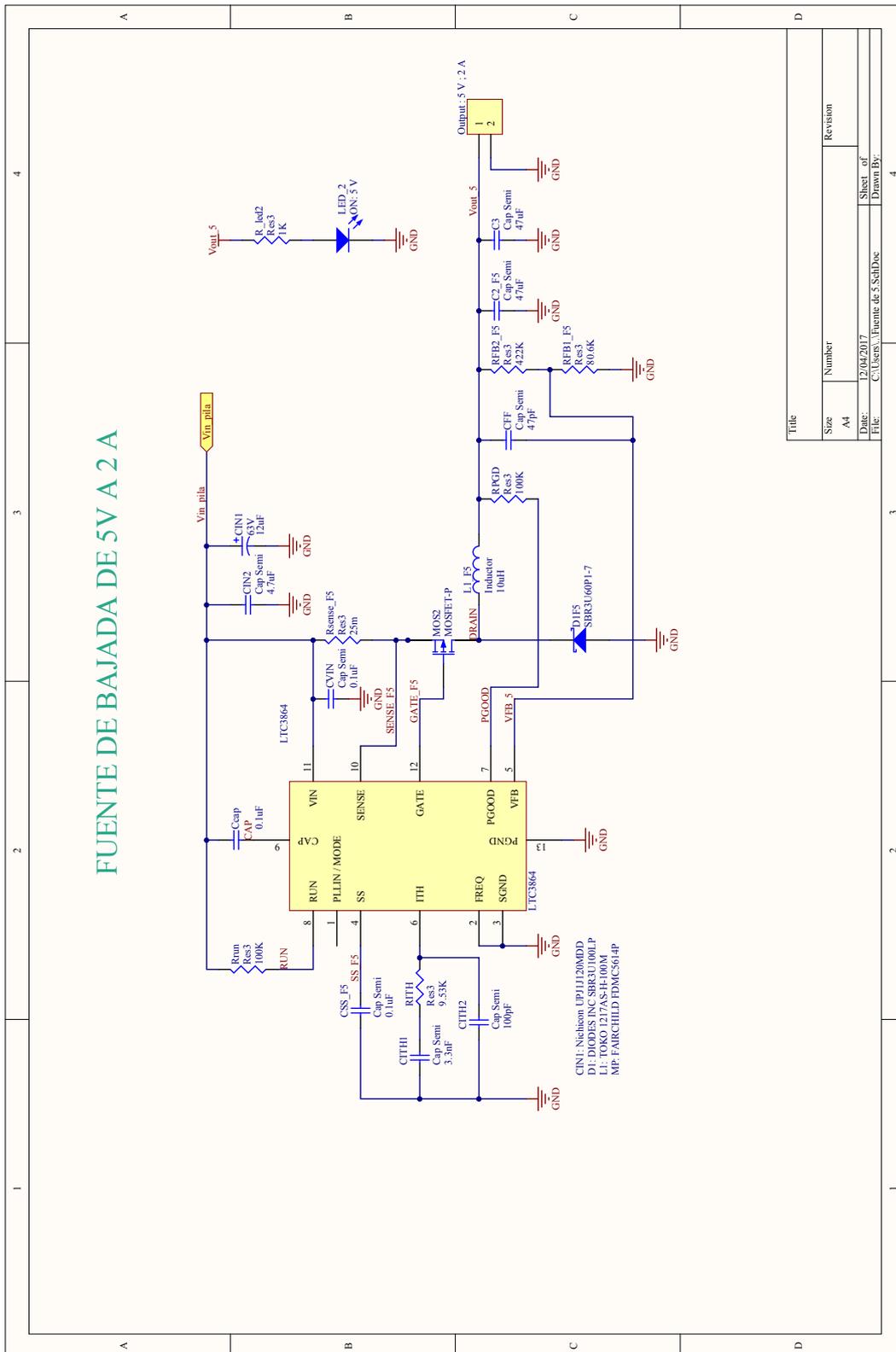


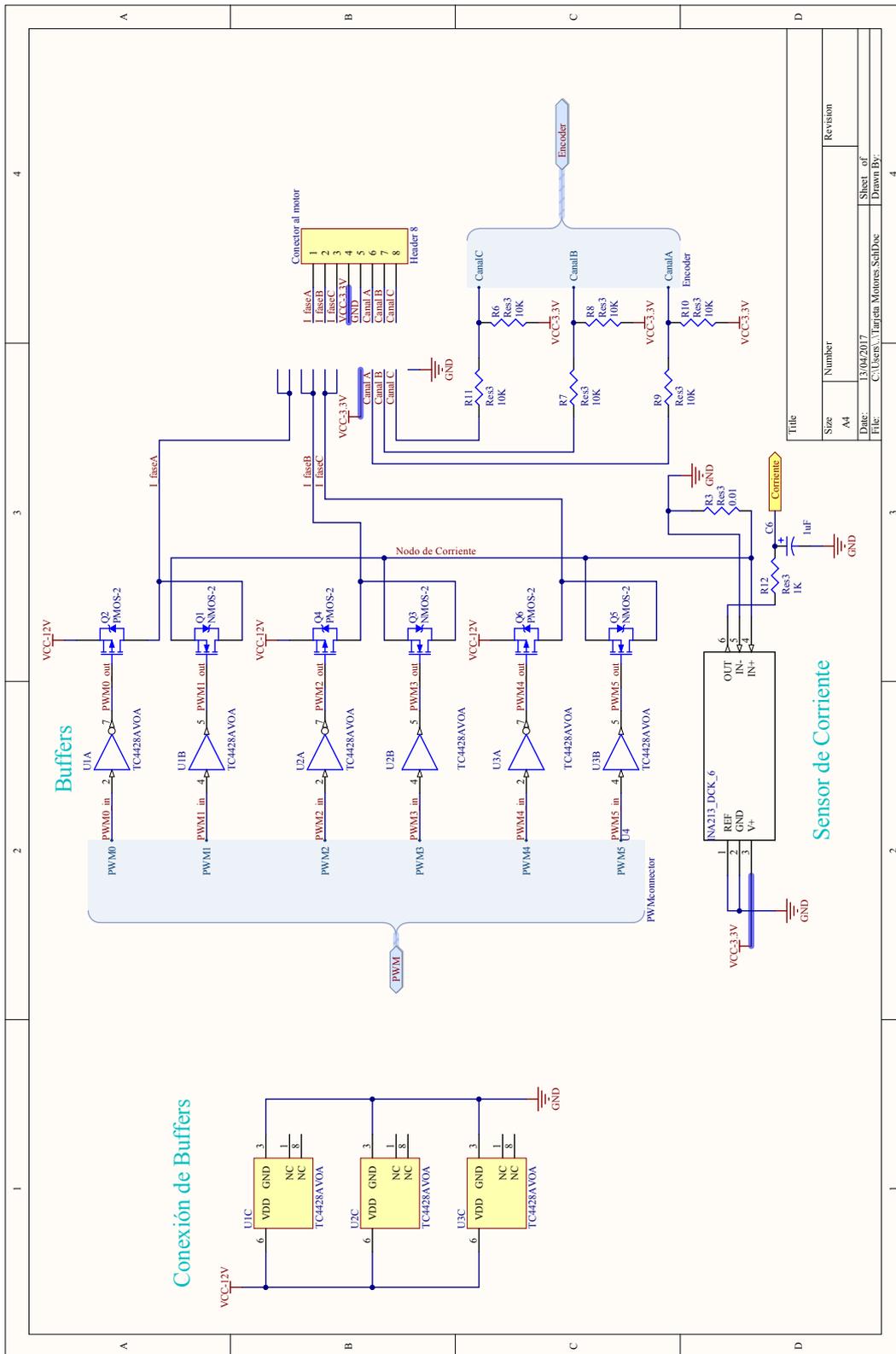
Figura A.6: Vista en 3D de la PCB de drivers para motores BLDC.













# Bibliografía

---

- [1] Mendoza-Bárceñas, M. A. (2014). *Desarrollo e integración de instrumentación mecatrónica para validar experimentalmente arquitecturas reconfigurables de control de orientación de satélites pequeños*. Tesis de doctorado. UNAM, México.
- [2] Jiménez-Madrigal, E. A. (2009). *Subsistemas de estabilización activa y sensores para un simulador satelital*. Tesis de licenciatura. UNAM, México.
- [3] Córdova-Alarcón, J. R. (2011). *Estimación y control de orientación para el nanosatélite HUMSAT-México*. Tesis de maestría. UNAM, México.
- [4] Griffin, R. (2014). *Zynq Workshop for Beginners*. Silica, EEUU.
- [5] Crockett, L., Elliot, R., Enderwitz, M., Stewart, R. (2014). *The Zynq Book*. Strathclyde Academic Media, Reino Unido.
- [6] Crockett, L., Elliot, R., Enderwitz, M., Stewart, R. (2014). *The Zynq Book: Tutorials*. Strathclyde Academic Media, Reino Unido.
- [7] Linear Technology Corporation (1997). *LTC1624: High Efficiency SO-8 N-Channel Switching Regulator Controller*. Linear Technology Corporation, EEUU.
- [8] Linear Technology Corporation (2012). *LTC3864: 60V Low Iq Step-Down DC/DC Controller with 100% Duty Cycle Capability*. Linear Technology Corporation, EEUU.
- [9] Linear Technology Corporation (2008). *LT3757/LT3757A: Boost, Flyback, SEPIC and Inverting Controller*. Linear Technology Corporation, EEUU.
- [10] Brown, W. (2002). *AN857: Brushless DC Motor Control Made Easy*. Microchip Technology Inc, EEUU.
- [11] Yedamale, P. (2003). *AN885: Brushless DC (BLDC) Motor Fundamentals*. Microchip Technology Inc, EEUU.
- [12] NXP Semiconductors (2007). *AN10661: Brushless DC motor control using the LPC2141*. NXP Semiconductors, Países Bajos.
- [13] Analog Devices (2014). *ADIS16448: Compact, Precision Ten Degrees of Freedom Inertial Sensor*. Analog Devices, EEUU.
- [14] Whitcomb-Crowell, C. (2011). *Development and Analysis of a Small Satellite Attitude Determination and Control System Testbed*. Tesis de maestría. MIT, EEUU.

- [15] Mendoza-Bárceñas, M. A., Vicente-Vivas, E., y Rodríguez-Cortés, H. (2014). *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, 12 (3): 370-383.
- [16] Kim, D., Park, S. Y., Kim, J. W., y Choi, K. H. (2008). *Development of a Hardware-In-Loop (HIL) Simulator for Spacecraft Attitude Control Using Momentum Wheels*. Journal of Astronomy and Space Sciences, 25 (4): 347-360.
- [17] Vijayakumar, G., Hemakumar, K. (2013). *SEPIC Converter based Photovoltaic System with Particle Swarm Optimization MPPT*. International Journal of Emerging Trends in Engineering Research, 1 (1): 5-10.
- [18] Susan-Florence, Y., Blesslin-Sheeba, T. (2014). *Energy Harvesting Capability of Air for Measurements using Self-Powered Wireless Sensor Network*. International Journal of Engineering Research and Applications, 8 (6): 30-34.
- [19] Eswari, N., Manisha, V., Geetha, R., Sudhakaran, M. (2016). *Design of Integrated ZVS Single-Inductor Multiple-Output using Synchronous Buck Converter for Multi-Load Applications*. International Research Journal in Advanced Engineering and Technology, 2 (2): 868-872.
- [20] Maxinez, D. G., Alcalá-Jara J. (2002). *VHDL: El arte de programar sistemas digitales*. CECSA, México.
- [21] Xilinx. (2012). *Zynq Architecture*. Presentación Técnica, EEUU.
- [22] Jiménez-Madrugal, E. A. (2012). *Filtrado digital de señales de navegación inercial para el Satélite HumSAT*. Tesis de maestría. UNAM, México.
- [23] Córdova-Alarcón, J. R. (2008). *Control de estabilización para picosatélite universitario*. Tesis de licenciatura. UNAM, México.
- [24] Sidi, M. J. (1997). *Spacecraft Dynamics and Control: A Practical Engineering Approach*. Cambridge Aerospace Series 7. Cambridge University Press, EEUU.
- [25] Prado, J., Bisiacchi, G., Reyes, L., Vicente, E., Contreras, F., Mesinas, M., Juárez, A. (2005). *Three-axis air-bearing based platform for small satellite attitude determination and control simulation*. Journal of Applied Research and Technology, 3(3): 222-237.
- [26] “Diseñan en la UNAM Dispositivos para Orientar y Controlar Satélites Artificiales” en: [http://www.dgcs.unam.mx/boletin/bdboletin/2015\\_055.html](http://www.dgcs.unam.mx/boletin/bdboletin/2015_055.html).
- [27] Reyes-Morales, R. (2012). *Sistema de orientación y estabilización para un satélite pequeño de percepción remota*. Tesis de licenciatura. UNAM, México.

- [28] Reyes-Morales, R. (2015). *Diseño de un esquema de control para la orientación en tres ejes de un micro-satélite de percepción remota con ruedas inerciales*. Tesis de maestría. UNAM, México.
- [29] Snider, R. E. (2010). *Attitude Control of a Satellite Simulator Using Reaction Wheels and a PID Controller*. Tesis de maestría. AFIT, EEUU.
- [30] Schwartz, J. L., Hall, C. D. (2003). *The Distributed Spacecraft Attitude Control System Simulator: Development, Progress, Plans*. NASA Space Flight Mechanics Symposium, Greenbelt, Maryland, EEUU.
- [31] Schwartz, J. L. (2004). *The Distributed Spacecraft Attitude Control System Simulator: From Design Concept to Decentralized Control*. Tesis de doctorado. Virginia Tech, EEUU.
- [32] “Ingeniero mexicano triunfa en MIT y NASA” - Agencia informativa del CONACYT en: <http://conacytprensa.mx/index.php/sociedad/personajes/3042-ingeniero-mexicano-triunfa-en-mit-y-nasa>.
- [33] “Facilities in the Advanced Dynamics, Validation & Control Research Laboratory” - University of Washington en: [https://www.aa.washington.edu/research/adv\\_ctrl/facilities](https://www.aa.washington.edu/research/adv_ctrl/facilities).
- [34] “Satellite Attitude, Determination and Control Test Bed (ADCS TB)” - Hawaii Space Flight Laboratory (HSFL) at the University of Hawaii at Manoa en: <http://www.hsfl.hawaii.edu/~manunes/portfolio.html>.
- [35] Microchip (2014). *dsPIC33FJ16(GP/MC)101/102 and dsPIC33FJ32(GP/MC)101/102/104: 16-Bit Digital Signal Controllers*. Microchip Technology, EEUU.
- [36] “Raspberry Pi 2 Model B” - Raspberry Pi en: <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>.
- [37] “SPHERES Satellites” - NASA en: <http://www.nasa.gov/spheres/satellites.html>.
- [38] “Attitude Dynamics and Control Laboratory” - The University of Michigan en: <http://www.umich.edu/~adcl/facilities.html>.
- [39] Vizcaíno-Torres, E. (2012). *Actualización del hardware del subsistema de control de orientación del satélite educativo SATEDU*. Tesis de licenciatura. UNAM, México.
- [40] Guzmán-Mercado, J. I. (2014). *Plataforma FPGA reconfigurable para implantar el sistema de control de apuntamiento del satélite SATEDU*. Tesis de licenciatura. UNAM, México.