



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**IMPLEMENTACIÓN DEL ALGORITMO DE
CONTROL DE POSICIÓN Y ESTABILIZACIÓN
DEL SATÉLITE QUETZAL BASADO EN UNA
PLATAFORMA FPGA**

T E S I S

QUE PARA OBTENER EL TÍTULO DE

INGENIERO ELÉCTRICO-ELECTRÓNICO

PRESENTA

ÁNGEL MANCILLA CARRASCO



DIRECTOR DE TESIS
DR. SAÚL DE LA ROSA NIEVES

CIUDAD UNIVERSITARIA, 2014



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



TELLESPOR MI RAZZA HABLA



TELLESPOR MI RAZZA HABLA

TELLESPOR MI RAZZA HABLA



Implementación del Algoritmo de Control de Posición y Estabilización del Satélite Quetzal basado en una Plataforma FPGA

@eln

Facultad de Ingeniería,
Universidad Nacional Autónoma de México,
Ciudad de México, D.F.

Jurado:

M.I. Juan José Carreón Granados
M.I. Larry Escobar Salguero
M.I. Ricardo Mota Marzano
Dr. Saúl de la Rosa Nieves
Dr. Jorge Alfredo Ferrer-Pérez

1 DE MAYO DE 2014

*Este trabajo está dedicado a Alicia V., Jesús R., Karla M., Adrián G.,
Rafael A y Leo F.
Siempre presentes.*

Agradecimientos

A mi madre, por formarme, por tu amor incondicional y cariño, gracias ma. A mi padre, porque eres la mejor persona que conozco, por educarme, por tu amor y cariño, gracias pa. A mi hermana, Karla, por todo lo que has hecho por mi, por todos los buenos momentos, y por el amor que demuestras a la familia, gracias. A mis hermanos, Rafael y Adrián, por cuidarme y acompañarme siempre, siempre, gracias. A Leo, quien es mi persona favorita, verte crecer es increíble, que este trabajo te anime a encontrar lo que te apasiona. Su apoyo, cariño y protección incondicional es lo que me impulsa, siempre los llevo en mente y corazón, sin ustedes, no soy nada.

Thalía, ∞ .

A la Facultad de Ingeniería de la Universidad Nacional Autónoma de México, en la que recibí conocimiento, amistades increíbles y la formación para seguir adelante.

Al Dr. Jorge Rodríguez Cuevas, quien me alentó con sus palabras. Al Dr. Cohen, gran profesor. Al maestro Lauro Santiago, y a los profesores que conforman el jurado. Mi profundo agradecimiento.

Quiero agradecer de manera especial a mi director de tesis, el Dr. Saúl de la Rosa Nieves, por darme la oportunidad de contribuir con el proyecto Quetzal, por su paciencia y dedicación para enseñarme, guiarme y dar forma a este trabajo. Muchas gracias.

Al Dr. Saúl Daniel Santillán, jefe del Centro de Alta Tecnología de la Facultad de Ingeniería y líder del proyecto Quetzal. Al Dr. Carlos Romo Fuentes.

Al Dr. Ricardo A. Vázquez, invaluable compañero de trabajo y amigo. Sé que alcanzarás todo lo que te propongas doctore.

Al Dr. Jorge Alfredo Ferrer-Pérez, quien creyó en mí dándome la oportunidad de viajar a MIT, me alentó a continuar con este trabajo en momentos en que nadie más lo hizo, y quien me ha impulsado a trabajar entregado para obtener lo que deseo. Porque continúes trabajando con pasión por la ciencia, impulsando a jóvenes de la misma forma en que lo hiciste conmigo y mis compañeros. Y por el gran aporte a este trabajo, muchas gracias Jorge.

A Mariana, Eric β , Ana, Eric T., Dana, Adrián F., Alfredo P., Eric C., Dominike, Alonso, Bren R., Juan Metal, Juan C., Ema, Jorge A., Ursu, Ari, Omar, Pris R. y Saúl.

A mis compañeros Iñaki, Andrés, Óscar, Emilio, Antonio Esteban, Mike, Miguel, Valero, Sócrates, Adrián, Ricardo, y César *Scooper*, de quienes aprendí muchísimo. A N. Tesla, obvio.

A la Red de Ciencia y Tecnología Espacial (REDCyTE) y al programa PAPIIT-IT103212 por la beca otorgada durante la realización de este trabajo.





Índice general

Índice de figuras	v
Índice de tablas	ix
1. Introducción	1
1.1. Antecedentes	3
1.2. Descripción general del satélite Quetzal UNAM-MIT	10
1.3. Objetivos Generales de la misión	13
1.4. Objetivos de la implementación del Algoritmo de Determinación de Posición y Control	13
2. Requerimientos de diseño para el subsistema de Determinación de Posición Y Control (ADCS)	15
2.1. Sistema de Comando y Manejo de Información	15
2.2. Sistema de Determinación de Posición y Control	18
2.2.1. Importancia del ADCS en una plataforma satelital	18
2.3. Requerimientos de diseño del ADCS	20
2.3.1. Requerimientos energéticos y de masa para el subsistema de Determi- nación de Posición y Control	22
2.3.2. Requerimientos debidos al ambiente espacial	22
2.3.2.1. Efectos del medio ambiente espacial	26
2.3.2.2. Efectos de radiación espacial en los dispositivos electrónicos	26
2.3.2.3. Efectos de Eventos Únicos	29
2.3.2.4. Niveles de dosis total de ionización por radiación	30

2.3.2.5.	Estimación de niveles TID y Efectos de Eventos Únicos en la órbita del satélite Quetzal	30
2.3.3.	Requerimientos para el cumplimiento de los objetivos académicos de la misión	33
2.4.	Metodología de diseño del ADCS	34
2.4.1.	Modos de Control y Requerimientos	34
2.4.2.	Cuantificación de perturbaciones del medio ambiente espacial	36
2.4.3.	Selección de métodos de control del satélite	39
2.4.4.	Selección de tamaño y hardware para ADCS	41
2.4.4.1.	Sensores	42
2.4.4.1.1.	Sensor de sol	42
2.4.4.1.2.	Sensor de campo magnético	43
2.4.4.1.3.	Sensor de inercia	43
2.4.4.2.	Actuadores	44
2.4.4.2.1.	Ruedas de reacción	45
2.4.4.2.2.	Torques magnéticos	46
2.4.5.	Definición del Algoritmo de Determinación de Posición y Control . .	46
2.5.	Algoritmo de Determinación de Posición y Control	47
2.5.1.	Antecedentes Matemáticos	47
2.5.2.	Algoritmo	51
2.5.3.	Tipos de Algoritmos	52
2.5.4.	Descripción del Algoritmo de Determinación de Posición y Control . .	52
3.	Implementación del algoritmo de Determinación de Posición y Control	59
3.1.	Definición de arquitectura para implementación del algoritmo de Determinación de Posición y Control	59
3.1.1.	Selección de tecnología	60
3.1.2.	Sistema embebido	60
3.1.2.1.	Microprocesador	62

3.1.2.2.	Microcontrolador	63
3.1.2.3.	Procesador de señales digitales (DSP)	63
3.1.2.4.	Circuito Integrado de Aplicación Específica (ASIC)	64
3.1.2.5.	Dispositivos Lógicos Programables	65
3.1.2.6.	FPGA	66
3.1.2.7.	Modelos de ejecución	66
3.1.2.8.	Elección de tecnología	67
3.1.3.	Tecnología FPGA	68
3.1.4.	Ventajas y desventajas de la Tecnología FPGA	72
3.1.5.	Determinación de la mejor solución	72
3.1.6.	<i>Soft Processor</i>	74
3.1.6.1.	PowerPC	75
3.1.6.2.	Xilinx Microblaze	76
3.1.7.	Programa Universitario Xilinx Virtex-II Pro XC2VP30	81
3.2.	Configuración de Arquitectura	84
3.2.1.	Configuración de Arquitectura para la implementación del Algoritmo de Determinación de Posición Y Control	84
3.2.2.	Resumen de implementación de arquitectura en XC2VP30 FPGA Virtex- II Pro	88
3.3.	Implementación del algoritmo de Determinación de Posición Y Control . . .	89
3.3.1.	Simulación numérica del algoritmo de Determinación de Posición Y Control	89
3.3.2.	Metodología de implementación del algoritmo de Determinación de Posición y Control en XUPV2P	102
3.4.	Comparación de resultados entre implementación en XUPV2P y simulación numérica	116
4.	Conclusiones	132
4.1.	Trabajo a futuro	133
A.	Derivación de algunas expresiones	140
A.1.	Representación numérica en punto fijo y punto flotante	140
A.2.	Polinomio Schmidt de Legendre	141

B. Códigos <i>main</i> de la Simulación e Implementación	142
B.1. Función <i>mainSoftware.m</i> de Simulación en Matlab	142
B.2. Funciones utilizadas por <i>mainSoftware.m</i> de Simulación	144
B.3. Función <i>plotData.m</i> para graficar datos de XUPV2P	145
B.4. Código <i>mainadcs.c</i> implementado en XUPV2P	147

Índice de figuras

1.1. Misiones satelitales por objetivo, <i>Space Mission Applications</i> [1]	8
1.2. Simulación orbital Satélite Quetzal 1000 km 135° de inclinación [2]	11
1.3. Simulación orbital Satélite Quetzal 600 km 40° de inclinación [2]	11
1.4. Simulación orbital Satélite Quetzal 800 km 105° de inclinación [2]	12
2.1. Diagrama de un ADCS completo [1].	19
2.2. Capas de la Atmósfera terrestre [3].	25
2.3. Proyección sobre la superficie terrestre de la órbita del satélite Quetzal y el tiempo local. Simulación obtenida con SPENVIS.	31
2.4. Curvas de profundidad de Dosis Total de Radiación para un semiconductor dentro de una esfera de aluminio con diferentes grosores de blindaje. Estimación obtenida con SPENVIS usando el modelo SHIELDOSE-2 para la misión Quetzal.	32
2.5. Flujo de partículas de alta energía para un volumen sensible paralelepípedo rectangular de $3.87\mu m \times 3.87\mu m \times 2.0\mu m$. Estimada con SPENVIS usando el algoritmo constant LET (CREME) para la misión Quetzal.	33
2.6. Efectos de las perturbaciones del ambiente espacial en ADCS [1].	38
2.7. Parámetros físicos del sistema: vector unitario nadir $\vec{\eta}$, marco de referencia Inercial (x_i, y_i, z_i) , marco de referencia Satelital (x_b, y_b, z_b) , vector Torque magnético en marco Orbital $(\tau_{mx}^b, \tau_{my}^b, \tau_{mz}^b)$, vector Torque gravitacional en marco Satelital $(\tau_{gx}^b, \tau_{gy}^b, \tau_{gz}^b)$, vector campo magnético en el marco Satelital (B_x^b, B_y^b, B_z^b) , vector Velocidad angular en el marco Inercial relativo al marco Satelital en el marco Satelital $(\omega_{ib}^b, \omega_{ib}^b, \omega_{ib}^b)$, marco de referencia Orbital (x_o, y_o, z_o) y Matriz de Inercia (I_x, I_y, I_z)	54
3.1. Arquitecturas Von Neumann y Harvard [4].	62
3.2. Diagrama de flujo de diseño de un ASIC [4].	65

3.3. Distribución de CLBs, IOBs, PIs, bloques RAM y multiplicadores en un FPGA Virtex [4].	69
3.4. Diagrama de flujo de implementación de un diseño en FPGA [5].	71
3.5. Organización del hardware de PowerPC [6].	76
3.6. Diagrama de bloques de <i>Soft Processor</i> Microblaze [7].	77
3.7. Diagrama de flujo de las aproximaciones y determinación de la solución óptima.	80
3.8. Diagrama de bloques de XUP Virtex®-II Pro [8].	82
3.9. Fotografía de XUP Virtex®-II Pro.	83
3.10. Mapa de memoria de configuración, velocidad de reloj y cantidad de memoria de la implementación.	86
3.11. Diagrama de bloques de la Implementación de Microblaze.	87
3.12. Diagrama de flujo de la simulación en Matlab.	92
3.13. Gráfica de ángulos de Euler para valores iniciales $\phi = -45.125$, $\theta = -45.987$, $\psi = 48.978$ generada por Matlab.	93
3.14. Gráfica de ángulos de Euler para valores iniciales $\phi = -45.125$, $\theta = -45.987$, $\psi = 48.978$ generada por Matlab, por separado.	93
3.15. Gráfica de ángulos de Euler para valores iniciales de $\phi = -45.125$, $\theta = -45.987$, $\psi = 48.978$ generada por Matlab, zoom a los últimos 500 puntos de simulación.	94
3.16. Gráfica de ángulos de Euler para valores iniciales $\phi = 12.825$, $\theta = 15.987$, $\psi = -8.197$ generada por Matlab.	95
3.17. Gráfica de ángulos de Euler para valores iniciales $\phi = 12.825$, $\theta = 15.987$, $\psi = -8.197$ generada por Matlab, por separado.	95
3.18. Gráfica de ángulos de Euler para valores iniciales de $\phi = 12.825$, $\theta = 15.987$, $\psi = -8.197$ generada por Matlab, zoom a los últimos 500 puntos de simulación.	96
3.19. Gráfica de ángulos de Euler para valores iniciales $\phi = 7.99$, $\theta = -5.987$, $\psi = 6.197$ generada por Matlab.	97
3.20. Gráfica de ángulos de Euler para valores iniciales $\phi = 7.99$, $\theta = -5.987$, $\psi = 6.197$ generada por Matlab, por separado.	97
3.21. Gráfica de ángulos de Euler para valores iniciales de $\phi = 7.99$, $\theta = -5.987$, $\psi = 6.197$ generada por Matlab, zoom a los últimos 500 puntos de simulación.	98
3.22. Gráfica de ángulos de Euler para valores iniciales $\phi = 25$, $\theta = 25$, $\psi = 25$ generada por Matlab.	99

3.23. Gráfica de ángulos de Euler para valores iniciales $\phi = 25, \theta = 25, \psi = 25$ generada por Matlab, por separado.	99
3.24. Gráfica de ángulos de Euler para valores iniciales de $\phi = 25, \theta = 25, \psi = 25$ generada por Matlab, zoom a los últimos 500 puntos de simulación.	100
3.25. Gráfica de los valores de campo magnético para la órbita seleccionada.	103
3.26. Diagrama del algoritmo implementado en XUPV2P.	107
3.27. Gráfica de ángulos de Euler para valores iniciales $\phi = -45.125, \theta = -45.987, \psi = 48.978$ generada por XUPV2P.	108
3.28. Gráfica de ángulos de Euler para valores iniciales $\phi = -45.125, \theta = -45.987, \psi = 48.978$ generada por XUPV2P, por separado.	108
3.29. Gráfica de ángulos de Euler para valores iniciales de $\phi = -45.125, \theta = -45.987, \psi = 48.978$ generada por XUPV2P, zoom a los últimos 250 segundos.	109
3.30. Gráfica de ángulos de Euler para valores iniciales $\phi = 12.825, \theta = 15.987, \psi = -8.197$ generada por XUPV2P.	110
3.31. Gráfica de ángulos de Euler para valores iniciales $\phi = 12.825, \theta = 15.987, \psi = -8.197$ generada por XUPV2P, por separado.	110
3.32. Gráfica de ángulos de Euler para valores iniciales de $\phi = 12.825, \theta = 15.987, \psi = -8.197$ generada por XUPV2P, zoom a los últimos 250 segundos.	111
3.33. Gráfica de ángulos de Euler para valores iniciales $\phi = 7.99, \theta = -5.987, \psi = 6.197$ generada por XUPV2P.	112
3.34. Gráfica de ángulos de Euler para valores iniciales $\phi = 7.99, \theta = -5.987, \psi = 6.197$ generada por XUPV2P, por separado.	112
3.35. Gráfica de ángulos de Euler para valores iniciales de $\phi = 7.99, \theta = -5.987, \psi = 6.197$ generada por XUPV2P, zoom a los últimos 250 segundos.	113
3.36. Gráfica de ángulos de Euler para valores iniciales $\phi = 25, \theta = 25, \psi = 25$ generada por XUPV2P.	114
3.37. Gráfica de ángulos de Euler para valores iniciales $\phi = 25, \theta = 25, \psi = 25$ generada por XUPV2P, por separado.	114
3.38. Gráfica de ángulos de Euler para valores iniciales de $\phi = 25, \theta = 25, \psi = 25$ generada por XUPV2P, zoom a los últimos 250 segundos.	115
3.39. Gráficas comparativas entre Matlab y XUPV2P de ángulos de Euler para valores iniciales de $\phi = -45.125, \theta = -45.987, \psi = 48.978$. Gráficas completas y con un zoom en los últimos 500 puntos.	120

3.40. Gráficas comparativas en escala logarítmica entre Matlab y XUPV2P de Ángulos de Euler de para valores iniciales de $\phi = -45.125$, $\theta = -45.987$, $\psi = 48.978$	121
3.41. Gráfica en escala logarítmica del error absoluto entre Matlab y XUPV2P de Ángulos de Euler para valores iniciales de $\phi = -45.125$, $\theta = -45.987$, $\psi = 48.978$	121
3.42. Gráficas comparativas en escala logarítmica entre Matlab y XUPV2P de Cuaterniones Unitarios para valores iniciales de $\phi = -45.125$, $\theta = -45.987$, $\psi = 48.978$	122
3.43. Gráfica en escala logarítmica del error absoluto entre Matlab y XUPV2P de Cuaterniones Unitarios para valores iniciales de $\phi = -45.125$, $\theta = -45.987$, $\psi = 48.978$	122
3.44. Gráficas comparativas entre Matlab y XUPV2P de ángulos de Euler para valores iniciales de $\phi = 12.825$, $\theta = 15.987$, $\psi = -8.197$. Gráficas completas y con un zoom en los últimos 500 puntos.	123
3.45. Gráficas comparativas en escala logarítmica entre Matlab y XUPV2P de Ángulos de Euler de para valores iniciales de $\phi = 12.825$, $\theta = 15.987$, $\psi = -8.197$	124
3.46. Gráfica en escala logarítmica del error absoluto entre Matlab y XUPV2P de Ángulos de Euler para valores iniciales de $\phi = 12.825$, $\theta = 15.987$, $\psi = -8.197$	124
3.47. Gráficas comparativas en escala logarítmica entre Matlab y XUPV2P de Cuaterniones Unitarios para valores iniciales de $\phi = 12.825$, $\theta = 15.987$, $\psi = -8.197$	125
3.48. Gráfica en escala logarítmica del error absoluto entre Matlab y XUPV2P de Cuaterniones Unitarios para valores iniciales de $\phi = 12.825$, $\theta = 15.987$, $\psi = -8.197$	125
3.49. Gráficas comparativas entre Matlab y XUPV2P de ángulos de Euler para valores iniciales de $\phi = 7.99$, $\theta = -5.987$, $\psi = 6.197$. Gráficas completas y con un zoom en los últimos 500 puntos.	126
3.50. Gráficas comparativas en escala logarítmica entre Matlab y XUPV2P de Ángulos de Euler de para valores iniciales de $\phi = 7.99$, $\theta = -5.987$, $\psi = 6.197$	127
3.51. Gráfica en escala logarítmica del error absoluto entre Matlab y XUPV2P de Ángulos de Euler para valores iniciales de $\phi = 7.99$, $\theta = -5.987$, $\psi = 6.197$	127
3.52. Gráficas comparativas en escala logarítmica entre Matlab y XUPV2P de Cuaterniones Unitarios para valores iniciales de $\phi = 7.99$, $\theta = -5.987$, $\psi = 6.197$	128

3.53. Gráfica en escala logarítmica del error absoluto entre Matlab y XUPV2P de Cuaterniones Unitarios para valores iniciales de $\phi = 7.99$, $\theta = -5.987$, $\psi = 6.197$	128
3.54. Gráficas comparativas entre Matlab y XUPV2P de ángulos de Euler para valores iniciales de $\phi = 25$, $\theta = 25$, $\psi = 25$. Gráficas completas y con un zoom en los últimos 500 puntos.	129
3.55. Gráficas comparativas en escala logarítmica entre Matlab y XUPV2P de Ángulos de Euler de para valores iniciales de $\phi = 25$, $\theta = 25$, $\psi = 25$	130
3.56. Gráfica en escala logarítmica del error absoluto entre Matlab y XUPV2P de Ángulos de Euler para valores iniciales de $\phi = 25$, $\theta = 25$, $\psi = 25$	130
3.57. Gráficas comparativas en escala logarítmica entre Matlab y XUPV2P de Cuaterniones Unitarios para valores iniciales de $\phi = 25$, $\theta = 25$, $\psi = 25$	131
3.58. Gráfica en escala logarítmica del error absoluto entre Matlab y XUPV2P de Cuaterniones Unitarios para valores iniciales de $\phi = 25$, $\theta = 25$, $\psi = 25$	131

Índice de tablas

1.1. Clasificación de plataformas satelitales por rango de dimensiones [9].	5
2.1. Etapas de diseño del SCMI [9].	17
2.2. Presupuestos energético, masa, potencia y costo del ADCS de CASTOR [10].	22
2.3. Elementos del ambiente espacial [3].	24
2.4. Efectos del ambiente espacial en los subsistemas de una plataforma satelital 1 de 2 [3].	27
2.5. Efectos del ambiente espacial en los subsistemas de una plataforma satelital 2 de 2 [3].	28
2.6. Criterios de rendimiento para los modos de control del ADCS [1].	36
2.7. Precisión de apuntamiento y maniobrabilidad para cada método de control de posición de ADCS [1].	40
2.8. Efectos de la precisión en el apuntamiento del satélite y ADCS [1].	41
2.9. Características del Sensor de Sol SS-411 [11].	42
2.10. Características del Sensor Magnético MicroMag3 [12].	43
2.11. Características de unidad IMU ADIS16365 [13].	44
2.12. Características de Rueda de reacción Interplanetary RW-0.0.60-28 [14].	45
3.1. Uso de recursos del FPGA XC2VP30 después de implementación.	88
3.2. Límite crítico de Frecuencia para los módulos del sistema implementado. . .	89
3.3. Comparativa entre valores de error absoluto promedio y correlación para val- ores iniciales de $\phi = -45.125$, $\theta = -45.987$, $\psi = 48.978$	117
3.4. Comparativa entre valores de error absoluto promedio y correlación para val- ores iniciales de $\phi = 12.825$, $\theta = 15.987$, $\psi = -8.197$	117
3.5. Comparativa entre valores de error absoluto promedio y correlación para val- ores iniciales de $\phi = 7.99$, $\theta = -5.987$, $\psi = 6.197$	118

3.6. Comparativa entre valores de error absoluto promedio y correlación para valores iniciales de $\phi = 25, \theta = 25, \psi = 25$ 118

Lista de Acrónimos

ADC	Analog-to-digital Converter
ADCS	Attitude Determination and Control Subsystem
AEM	Agencia Espacial Mexicana
ALU	Arithmetic Logic Unit
ARE	Algebraic Riccati Equation
ASCII	American Standard Code for Information Interchange
ASIC	Application Specific Integrated Circuit
BFF	Body Fixed Frame
CCCP	
C&DH	Command and Data Handling
CAD	Computer-Aided Design
CASTOR	Cathode/Anode Satellite Thruster for Orbital Repositioning
CAT	Centro de Alta Tecnología
CISC	Complex Instruction Set Computing
CLB	Configurable Logic Block
CMG	Control Moment Gyroscope
CMOS	Complementary Metal Oxide Semiconductor
CORDIC	Coordinate Rotation Digital Computer
COTS	Commercial Off-The-Shelf
CPU	Central Processing Unit
CREME	Cosmic Ray Effects on MicroElectronics
DSP	Digital Signal Processor
ECEF	Earth Centered Earth-Fixed Frame

ECIF	Earth Centered Inertial Frame
EEPROM	Electrically Erasable Programmable Read-Only Memory
EPROM	Erasable Programmable Read Only Memory
EPS	Electric Power Subsystem
FI	Facultad de Ingeniería
FPA	Floating Point Arithmetic
FPGA	Field Programmable Gate Array
FPU	Floating Point Unit
GEO	Geostationary Earth Orbit
GPR	General Purpose Register
GPS	Global Positioning System
HDL	Hardware Description Language
HK	House Keeping
IC	Integrated Circuit
IEEE	Institute of Electrical and Electronics Engineers
IGRF	International Geomagnetic Reference Field
IMU	Inertial Measurement Unit
IOB	In Out Block
LEO	Low Earth Orbit
LET	Linear Energy Transfer
LMB	Local Memory Bus
LQR	Linear Quadratic Regulator
MAC	Multiply-Accumulate Operation
MBU	Multi-bit Upset
MEO	Medium Earth Orbit
MISTI	MIT International Science and Technology Initiatives
MIT	Massachusetts Institute of Technology
MMU	Memory Management Unit
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor
MSR	Machine State Register

NASA	National Aeronautics and Space Administration
PI	Programmable Interconnection
PLB	Processor Local Bus
PLD	Programmable Logic Device
PowerPC	Performance Optimization With Enhanced RISC - Performance Computing
PS	Propulsion Subsystem
PSL	Propulsion Systems Laboratory
RadHard	Radiation-Hardened
RAM	Random-Access memory
REDCyTE	Red de Ciencia y Tecnología Espacial
RISC	Reduced Instruction Set Computing
RTOS	Real-time Operative System
S&MS	Structure and Mechanical Subsystem
SBU	Single Bit Upset
SCMI	Sistema de Comando y Manejo de Información
SEE	Single Event Effects
SEFI	Single Event Functional Interruption
SEGR	Single Event Gate Rupture
SEL	Single Event Latchup
SET	Single Event Transient
SEU	Single Event Upset
SPENVIS	Space Environment Information System
SPR	Special Purpose Register
SRAM	Static Random-Access Memory
SSL	Space Systems Laboratory
T&C	Telecommunication and Commands
TCS	Thermal Control Subsystem
TID	Total Ionizing Dose
UCF	User Constraints File
UNAM	Universidad Nacional Autónoma de México

URSS	Unión de Repúblicas Socialistas Soviéticas
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuits
XBD	Xilinx Board Description
XPS	Xilinx Platform Studio
XUP	Xilinx University Program
XUPV2P	Xilinx University Program Virtex-II Pro

Capítulo 1

Introducción

Esta tesis se enfoca en la implementación de una propuesta de algoritmos de Determinación de Posición y Control del satélite Quetzal UNAM-MIT basado en una plataforma FPGA. El algoritmo será parte del Sistema de Comando y Manejo de Información (SCMI) que atiende al Subsistema de Determinación de Posición y Control (ADCS, por sus siglas en inglés). Se incluye en el trabajo, como marco de la selección e implementación del algoritmo de Determinación y Control del satélite, los criterios de diseño del SCMI a partir de los requerimientos del sistema ADCS.

Este trabajo está dividido en 4 capítulos. En el capítulo 1 se presentan los antecedentes en donde se incluye un marco teórico, con las características de la tecnología satelital y tendencia de desarrollo actual. Posterior a esto se aborda la importancia que dicha tecnología juega en el desarrollo de un país. Finalmente se presenta el proyecto de colaboración internacional en el que este trabajo se desarrolla, así como los objetivos del trabajo realizado.

En el capítulo 2 se presentan las características del sistema de Comando y Manejo de Información (SCMI). Posteriormente, se describe la metodología de diseño del ADCS considerando las restricciones y objetivos de la misión, medio ambiente espacial y objetivos académicos. Se define el concepto de algoritmo y mencionan los tipos de algoritmo para un ADCS, se hace una descripción de los conceptos matemáticos relacionados con el algoritmo a implementar. El capítulo continúa con una detallada descripción del algoritmo seleccionado, enumerando las operaciones y tareas que debe realizar; así como la justificación del uso de las mismas.

En el capítulo 3 se desarrolla todo el trabajo práctico de la tesis, iniciando por la justificación en la selección del dispositivo base para la implementación del algoritmo ADCS y la determinación de la mejor solución mediante una comparación entre diferentes tipos de tecnología. Posteriormente, se describe la configuración de arquitectura para la implementación del algoritmo, seguido por la descripción de la simulación numérica del algoritmo y después se presenta su metodología de implementación en la plataforma seleccionada. Como punto final de este capítulo se hace una comparación entre los resultados de la simulación numérica y la implementación embebida.

En el capítulo 4 se discuten las conclusiones del trabajo realizado, se compara el desempeño de la implementación con los requerimientos planteados en los objetivos, se plantea trabajo a futuro y se proponen mejoras en el diseño.

1.1. Antecedentes

Plataforma Satelital

La palabra Satélite se conoce y usa bastante en la actualidad, independientemente de la formación académica o profesional ya que hace tiempo dejó de ser un elemento único de países selectos y un tema que se debata e investigue únicamente en grandes instituciones académicas y centros de investigación. Se trata de un tema de interés no sólo para ingenieros electrónicos, de comunicaciones, científicos y tecnócratas; sino que fascina a los aficionados, entusiastas y en gran medida a cualquier persona [15]. Una definición aceptada para satélite, en general, es cualquier cuerpo natural o artificial que orbita un cuerpo celeste tal como un planeta o estrella. En el presente contexto, se hace referencia únicamente a satélites artificiales que orbitan el planeta Tierra puestos en órbitas específicas dependiendo del objetivo y aplicación deseada. Plataforma satelital se refiere al conjunto de subsistemas que conforman al satélite y permiten que la carga útil funcione suministrando los requerimientos que le permitan a ésta cumplir con su objetivo. De acuerdo a Wiley et al. [9], una plataforma satelital se puede dividir en los siguientes subsistemas:

Carga útil

La carga útil, denominada payload en inglés, puede considerarse como un subsistema, sin embargo no forma parte de la plataforma satelital ya que es ésta la razón de ser de la plataforma. La carga útil es la combinación de software y hardware en el satélite que interactúa con el objeto que es la porción del mundo exterior que el satélite busca o con el cual experimenta para cumplir con los objetivos de la misión [9]. La carga útil es por lo general única en cada misión y es la razón fundamental para que el satélite vuele. El diseño, selección y mantenimiento de la misión depende en gran medida de la carga útil por lo que es un tema fundamental y crítico durante el desarrollo de un satélite. Algunos ejemplos de carga útil son transmisores, cámaras, radiómetros, medidores de altitud, medidores de partículas de alta energía, etc.

Subsistema de Determinación de Posición y Control (ADCS, por sus siglas en inglés)

Se encarga de estabilizar al satélite y orientarlo en direcciones deseadas a lo largo de la misión contrarrestando las perturbaciones externas a las que se encuentra expuesto el satélite. El ADCS se relaciona con el subsistema de Propulsión porque en ciertas misiones el ADCS puede emplear propulsores como actuadores.

Subsistema de Telemetría y Comandos (T&C, por sus siglas en inglés)

Provee las interfaces entre el satélite y la estación terrena. La información de la carga útil y sistema de monitoreo pasan del satélite a través del subsistema de Telemetría y Comandos a los operadores y usuarios en el centro de operación. Comandos de operación de la plataforma satelital y de la carga útil pasan a través de este subsistema.

Subsistema de Comando y Manejo de Información(C&DH, por sus siglas en inglés)

Realiza dos tareas primordiales. Recibe, valida, decodifica y distribuye comandos a otros subsistemas. Además, recoge, procesa y da formato a información proveniente de la carga útil y subsistema de monitoreo para su transmisión o uso en procesamiento a bordo del satélite.

Subsistema de Potencia (EPS, por sus siglas en inglés)

Provee, almacena, distribuye y controla la potencia eléctrica del satélite. Generalmente la energía que usa este subsistema es generada mediante paneles solares, aunque también se utilizan métodos nucleares de generación de energía en misiones que requieren un alto consumo de potencia.

Subsistema de Control Térmico (TCS, por sus siglas en inglés)

Mantiene todos los componentes de los diferentes subsistemas, incluyendo la carga útil, dentro de un rango de temperatura requerido para cada fase de la misión. Los límites de temperatura incluyen un nivel bajo por el que los componentes no deben pasar y un nivel alto que no debe ser superado. Todos los componentes tienen un rango térmico en el cual son operativos, si la temperatura de alguno de los componentes sale de este rango el sistema puede tener una falla crítica y la misión perderse.

Subsistema Estructural y de Mecanismos(S&MS, por sus siglas en inglés)

Soporta mecánicamente todos los otros subsistemas del satélite, sujeta el satélite al vehículo de lanzamiento, y provee separación activa (dinámica) del mismo mediante activación por artefactos explosivos. El diseño estructural debe satisfacer todos los requerimientos de rigidez y fuerza, bajo las condiciones de vibración durante el lanzamiento y a lo largo de la vida útil del satélite.

Subsistema de Propulsión (PS, por sus siglas en inglés)

Encargado de hacer maniobras de transferencia de órbita de baja a una más alta para misiones de operación o en trayectorias de intercepción planetaria. Además, proveen empuje para ajuste de posición, mantenimiento y corrección de órbitas. Todos estos subsistemas interactúan entre sí y dependen mutuamente de ellos para su correcto funcionamiento a lo largo de la vida útil del satélite, de igual forma durante su diseño. Todos ellos conforman la plataforma satelital.

Clasificación de Satélites

Los satélites se pueden clasificar por sus dimensiones, siendo los más grandes de varias toneladas hasta los más pequeños de algunas decenas de gramos.

Categoría	Rango de masa (Kg)
Satélites Grandes	> 1000
Satélites Medianos	500-1000
Satélites Mini	100-500
Satélites Micro	10-100
Satélites Nano	1-10
Satélites Pico	0.1-1
Satélites Femto	< 0.1

Tabla 1.1: Clasificación de plataformas satelitales por rango de dimensiones [9].

De acuerdo Wertz et.al. [1] los satélites también se pueden clasificar en función al tipo de órbita en que operan, las siguientes son las órbitas más usadas aunque no son las únicas:

Órbitas terrestres bajas (LEO, por sus siglas en inglés)

Son satélites con una altura orbital de entre 160 kilómetros y 2000 kilómetros, con un periodo orbital promedio de 120 minutos. Satélites en orbitas LEO en alturas menores a 400 km tienen vidas útiles cortas en comparación con órbitas más elevadas debido al arrastre atmosférico que existe a estas alturas. Las aplicaciones más comunes para este tipo de órbitas debido a su cercanía con la tierra, corto periodo orbital y relativamente fácil posicionamiento en dichas órbitas son las comunicaciones, percepción remota como captura de imágenes incluyendo aplicaciones civiles y militares, aplicaciones científicas como medición de gases atmosféricos entre otras. Al tener la posibilidad de lanzar un satélite en una órbita LEO se pueden realizar captura de imágenes con una resolución espacial, ampliando el espectro de aplicaciones para análisis de información.

Órbitas terrestres medias (MEO, por sus siglas en inglés)

Satélites con una altura orbital de entre 2000 kilómetros y por debajo de 35700 kilómetros, teniendo un periodo orbital que oscila en un rango de 2 a 22 horas aproximadamente, las aplicaciones más comunes son navegación, comunicaciones y ciencia del medio ambiente. Un ejemplo muy común de aplicación de satélites en órbita MEO es el sistema de posicionamiento global (GPS).

Orbitas terrestres geoestacionarias (GEO, por sus siglas en inglés)

Son satélites con una órbita circular a 35786 kilómetros de altura con respecto a la superficie de la Tierra que siguen la dirección de rotación de la Tierra. Su periodo orbital es de 24 horas. Dado a que su velocidad angular respecto a la tierra es cero simula no tener movimiento, manteniéndose en un punto fijo sobre un plano ecuatorial. La mayoría de las comunicaciones satelitales se realizan usando órbitas geoestacionarias. Los satélites geoestacionarios tienen una masa de varios miles de kilogramos, un costo de cientos de millones de dólares estadounidenses y pueden llegar a tener fuentes nucleares de generación de potencia eléctrica.

Tecnología Satelital

Sin duda la tecnología satelital es una de las más complejas y costosas que el hombre desarrolla debido a los altos niveles de riesgo tecnológico originado por la complejidad de sistemas, requerimientos de rutina para cuartos limpios de ensamblado y pruebas, medio ambiente hostil de operación, intensiva labor de producción, tecnología de integración y metodología complejas y costosas, los lanzamientos son muy delicados requieren mucha atención, dependen de una gran cantidad de variables para su éxito, mercado reducido hay pocos clientes, por lo mismo hay pocos proveedores, tiempos largos de desarrollo, crítica infraestructura de transporte, pruebas intensivas durante y después de la etapa de manufactura, altos niveles de confiabilidad, autonomía y efectividad requeridos [1]. Esta tecnología es cara debido a que el diseño, desarrollo, fabricación, prueba y operación de los vehículos de lanzamiento y los satélites mismos requieren mano de obra intensiva altamente capacitada, los costos laborales son un factor dominante en el diseño e implementación de misiones espaciales. Según datos de la Administración Nacional de Aeronáutica y del Espacio de Estados Unidos de Norteamérica (NASA, por sus siglas en inglés), en el año 2007 se estimó que el costo total anual-hombre de un ingeniero de desarrollo espacial es de \$252,000 dólares estadounidenses [1].

Una manera de mitigar los altos costos de la tecnología satelital es mediante el uso de componentes comerciales (COTS), que a pesar de no tener calificación militar o espacial, llegan a ser utilizados en el espacio mediante el seguimiento de pruebas de certificación y la aplicación de técnicas de tolerancia a fallas a los sistemas desarrollados con éste tipo de componentes, logrando así, brindar la confiabilidad que requieren los sistemas espaciales a una fracción del costo que se tendría usando componentes con calificación espacial.

Debido a que es capaz de proveer un balance entre rendimiento, costo y flexibilidad, muchos diseños de tecnología aviónica incluyendo satelital [16], hacen uso de dispositivos Programables de Arreglo de Compuertas (FPGA, por sus siglas en inglés), basados en memorias estáticas de acceso aleatorio (SRAM, por sus siglas en inglés). Los FPGA han crecido en importancia para aplicaciones espaciales debido a su capacidad de reconfiguración en órbita, bajo costo de desarrollo, alta flexibilidad de diseño, y alto desempeño y capacidad de procesamiento en paralelo [17].

La capacidad y rendimiento de FPGAs con disponibilidad para aplicaciones espaciales ha crecido de una manera constante por más de una década. Para dispositivos configurables una sola vez el incremento ha ido de aproximadamente 2 kgates a 72 kgates, para dispositivos reprogramables se ha incrementado de 40 kgates a 2 Mgates. De igual forma, dispositivos comerciales con calificación espacial alcanzan tolerancia a radiación de 1Mrad y capacidad de cientos de miles de celdas lógicas [18].

Normalmente, diseños de sistemas para aplicaciones espaciales basados en tecnología FPGA se hacen usando dispositivos que pueden ser programables una sola vez, dado que éstos han sido probados como menos sensibles a la radiación que los basados en memorias SRAM. Sin embargo, cuando la lógica interna del FPGA solo puede ser programada una sola vez, se limita el proceso de diseño y se elimina la posibilidad de reconfiguración. Por otro lado, los diseños basados en tecnología SRAM pueden ser programados miles de veces, creando un flujo de diseño más abierto, facilitando pruebas intensivas en hardware, y lo más importante permitiendo la reconfiguración de diseños. Sin embargo, al mismo tiempo los FPGAs basados en SRAM son inherentemente vulnerables a errores por radiación ionizante debido a partículas presentes en el medio ambiente espacial. A pesar de estas limitaciones los FPGAs comerciales basados en tecnología SRAM, ya han volado en misiones orbitales terrestres e interplanetarias [19–21], otros ejemplos de aplicación de FPGA en este tipo de sistemas son el Joint Strike Fighter [22], el Boeing 787 Dreamliner [23] y el Mars Rover de la NASA [24]. En la actualidad es un gran reto tecnológico generar diseños basados en FPGAs para resolver tareas de a bordo en misiones satelitales.

La tecnología satelital se encuentra estrechamente ligada al desarrollo espacial y es en cierta medida contradictoria tecnológicamente hablando, ya que por un lado mucha de la tecnología actualmente usada en aplicaciones de la industria electrónica, automotriz, médica, comunicaciones, etc., es derivada o ha sido fuertemente influenciada por desarrollos espaciales y satelitales; y por otro lado, la tecnología satelital no hace uso de los últimos desarrollos electrónicos dados los altos niveles de confiabilidad requeridos, siempre es preferible el uso de tecnología previamente probada como confiable. Un ejemplo simple pero ilustrativo de esta contradicción se encuentra en los micro procesadores usados en computadoras personales, cuya capacidad fácilmente puede llegar a varios GHz de frecuencia de reloj y 8 o 16 núcleos de procesamiento teniendo un costo de unos cientos de dólares; en cambio, un micro procesador que cumpla con los requerimientos de confiabilidad espacial con una fracción de la capacidad de procesamiento de uno comercial puede llegar fácilmente a varios miles de dólares por unidad.

Aplicaciones Satelitales

Las aplicaciones que tienen los satélites son variadas, desde comunicaciones, percepción remota, navegación, científica, exploración, militar, etc. En la Figura 1.1 de la página 8 se muestran ejemplos de satélites clasificados por aplicaciones específicas en cada área.

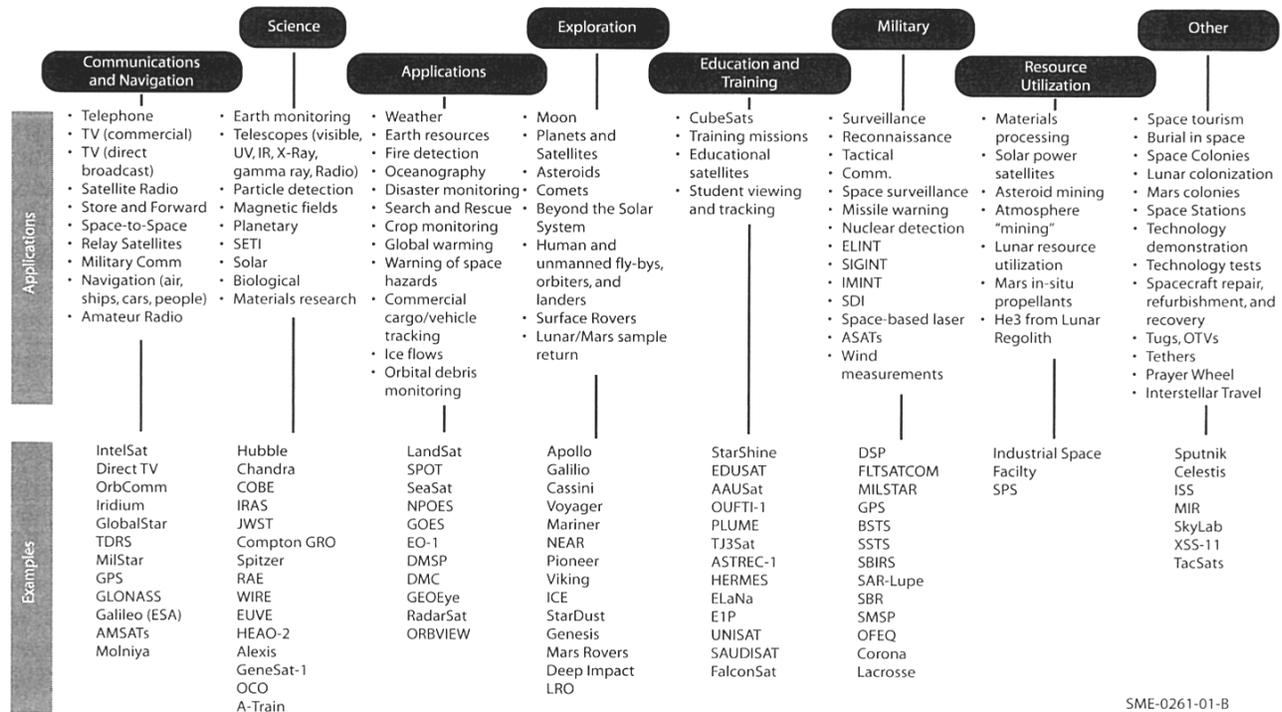


Figura 1.1: Misiones satelitales por objetivo, *Space Mission Applications* [1]

Desarrollo Espacial en México

A lo largo de las últimas décadas varios programas civiles de observación terrestre se han convertido en importantes herramientas para la toma de decisiones como manejo de recursos, planeación urbana y regional, producción agrícola y sistemas de monitoreo para evitar catástrofes naturales. Información obtenida por telemetría satelital es especialmente útil para países pequeños en su futuro desarrollo.

Programas de satelitales de monitoreo son tecnológicamente complejos y costosos en su inicio, por lo tanto difíciles de desarrollar con presupuestos limitados en países en desarrollo [25].

El sector espacial en un país resulta de vital importancia ya que conlleva un desarrollo tecnológico e industrial consigo. Además las comunicaciones y monitoreo dependen de tecnología satelital convirtiéndola en un aspecto estratégico para la seguridad y estabilidad de

un país. Si un país desarrolla su industria espacial ayuda a detonar otras industrias lo cual es vital para un país en vías de desarrollo como México.

La Agencia Espacial Mexicana (AEM) ha estado trabajando en establecer programas para incentivar el desarrollo tecnológico e investigación, para así atraer inversión hacia el sector espacial. De igual forma la AEM coordina esfuerzos de agencias gubernamentales, agencias internacionales, compañías, centros de investigación y universidades con el propósito de proveer soluciones de alto valor a problemas con implicaciones económicas y sociales. Estos objetivos serán alcanzados mediante el desarrollo de proyectos que permitan la creación de infraestructura básica para el monitoreo de diferentes fenómenos desde el espacio, además de ofrecer oportunidades para el desarrollo de instrumentos, componentes y subsistemas en los grupos de innovación. En el documento *Plan de Órbita: Mapa de ruta de la industria espacial mexicana*, desarrollado por ProMéxico y la AEM se discuten los hitos estratégicos, proyectos y programas de acción, así como la forma en que México pretende desarrollar la industria espacial nacional. Además la AEM como instrumento del Estado para contribuir a la soberanía nacional y a la solución de los grandes retos de la sociedad en seguridad, competitividad, educación, equidad, salud, inclusión digital y sustentabilidad ambiental. El documento se puede adquirir en la siguiente dirección electrónica:

http://promexico.gob.mx/work/models/comercio/Resource/130/1/images/Plano_obita_low.pdf

Siguiendo esta línea se propuso la creación de una red de instituciones académicas, universidades y sector industrial que integrara los esfuerzos nacionales para establecer una agenda de desarrollo del sector espacial en México. La REDCyTE (Red de ciencia y tecnología espacial) fue conformada por 140 académicos de 19 instituciones, 6 centros de desarrollo e investigación y 8 compañías [26].

Dentro de los académicos y grupos de trabajo que conforman la REDCyTE se encuentra el grupo Aeroespacial que encabeza el Dr. Saúl Daniel Santillán Gutiérrez adscrito al Centro de Alta Tecnología (CAT) de la Facultad de Ingeniería (FI) de la Universidad Nacional Autónoma de México (UNAM), dicho grupo trabaja en el desarrollo del satélite Quetzal UNAM-MIT [2]. Debido a la tendencia en el uso de tecnología FPGA para la implementación de aplicaciones espaciales [20], es de interés del grupo de Aeroespacial del CAT el desarrollar capacidades en el uso de esta tecnología con una perspectiva de utilización en el diseño de plataformas satelitales basadas en dicha tecnología. El trabajo de tesis que se presenta corresponde en parte al diseño de uno de los proyectos satelitales del grupo Aeroespacial del CAT, el satélite Quetzal. Esta tesis se enfoca en describir los requerimientos de diseño para un Sistema de Comando y Manejo de Información, e implementación de una propuesta de algoritmos de Determinación de Posición y Control del satélite Quetzal UNAM-MIT basándose en una plataforma FPGA.

1.2. Descripción general del satélite Quetzal UNAM-MIT

El proyecto satelital Quetzal UNAM-MIT surge a través de un acuerdo internacional de colaboración entre el CAT de la UNAM, el Laboratorio de Sistemas de Propulsión (SPL, por sus siglas en inglés), y el Laboratorio de Sistemas Espaciales (SSL, por sus siglas en inglés), siendo estos últimos pertenecientes al Instituto Tecnológico de Massachusetts (MIT, por sus siglas en inglés), para desarrollar un micro satélite que incorporará instrumentación para medir contaminantes atmosféricos en ciudades de México y América Latina.

Dicho proyecto surgió como una iniciativa para el desarrollo de capacidades en el área de tecnología espacial y la colaboración internacional entre ambas instituciones, para establecer una relación de largo plazo. La propuesta fue sometida a evaluación dentro del programa MIT International Science and Technology Initiatives (MISTI) del MIT. El programa MISTI consiste de fondos semilla para generar proyectos y colaboraciones internacionales, cuenta con un fondo general para proyectos en cualquier país y nueve fondos sectoriales.

El programa está abierto a todos los investigadores del MIT, quienes integran equipos de estudiantes de licenciatura y posgrado. Los fondos cubren visitas, reuniones de trabajo y talleres que facilitan las cooperaciones internacionales. Al proyecto satelital Quetzal presentado en el MISTI le fue otorgado el fondo máximo que se puede asignar en su categoría.

Las instituciones coordinadoras del proyecto son: el CAT por parte de la UNAM, el SPL y el SSL por parte del MIT. El Instituto de Geografía y el Centro de Ciencias de la Atmósfera son instituciones de la UNAM que apoyaron en la definición de la misión e instrumentación de subsistemas de la plataforma satelital [2].

La órbita de Quetzal deberá ser aquella, en la cual, el ángulo de visión del satélite cubra la zona de interés para que se pueda tomar imágenes de ésta por lo menos una vez al día. La órbita deberá tener una altitud sobre la zona de interés para tomar fotografías que garanticen la resolución y ancho de barrido requerido. La órbita deberá ser aquella en la que el pase del satélite sobre la zona de interés esté en el intervalo de 1 grado de apuntamiento del nadir. La órbita deberá ser aquella en la que el satélite pase dentro del rango de línea de vista de una estación terrena por lo menos una vez al día. En base a los requerimientos orbitales listados, las órbitas propuestas para el satélite Quetzal son orbitas circulares de 98° y 28.5° de inclinación a una altura de 686 km. Cabe mencionar que se han simulado distintas órbitas considerando diferentes altitudes y grados de inclinación para estimar el tiempo y frecuencia de paso sobre los puntos de interés de la misión.

Las simulaciones orbitales que se han hecho son a 600 km, 40° y 90° de inclinación, 800 km 45° y 105° de inclinación, y 1000 km 110° y 135° de inclinación. A continuación se presentan gráficos de las simulaciones a 1000 km, 600 km y 800 km respectivamente en donde se muestra con una línea verde el recorrido del satélite a lo largo del planeta y con franjas azules el área de cobertura que tendría.

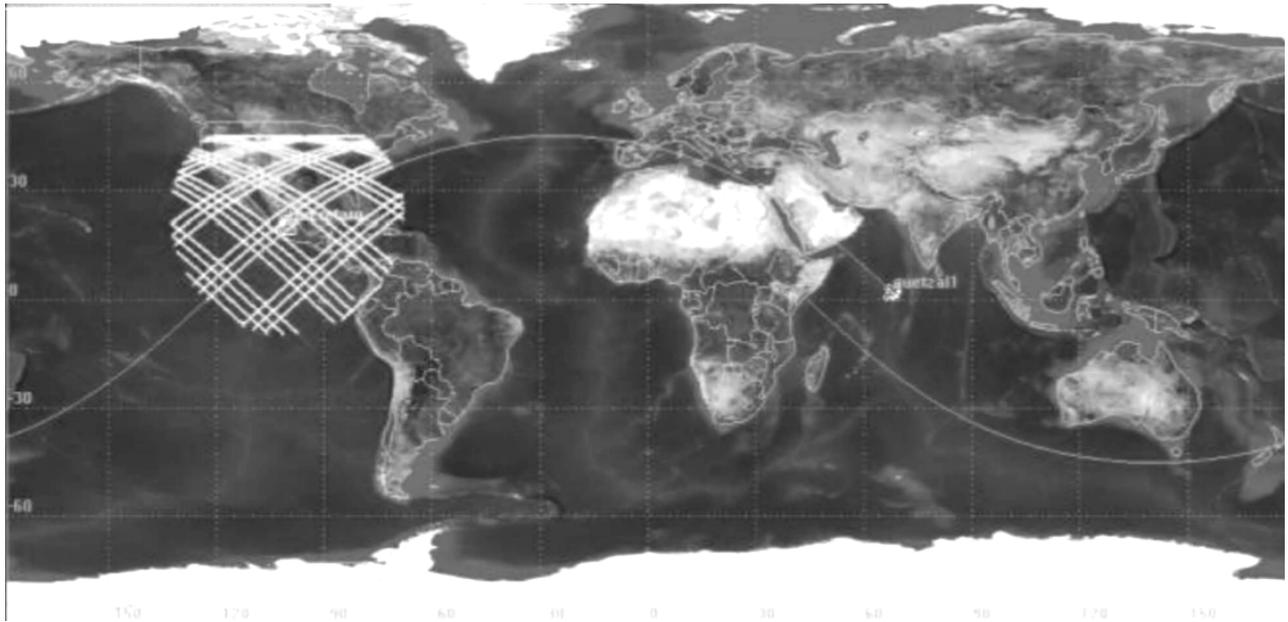


Figura 1.2: Simulación orbital Satélite Quetzal 1000 km 135° de inclinación [2]

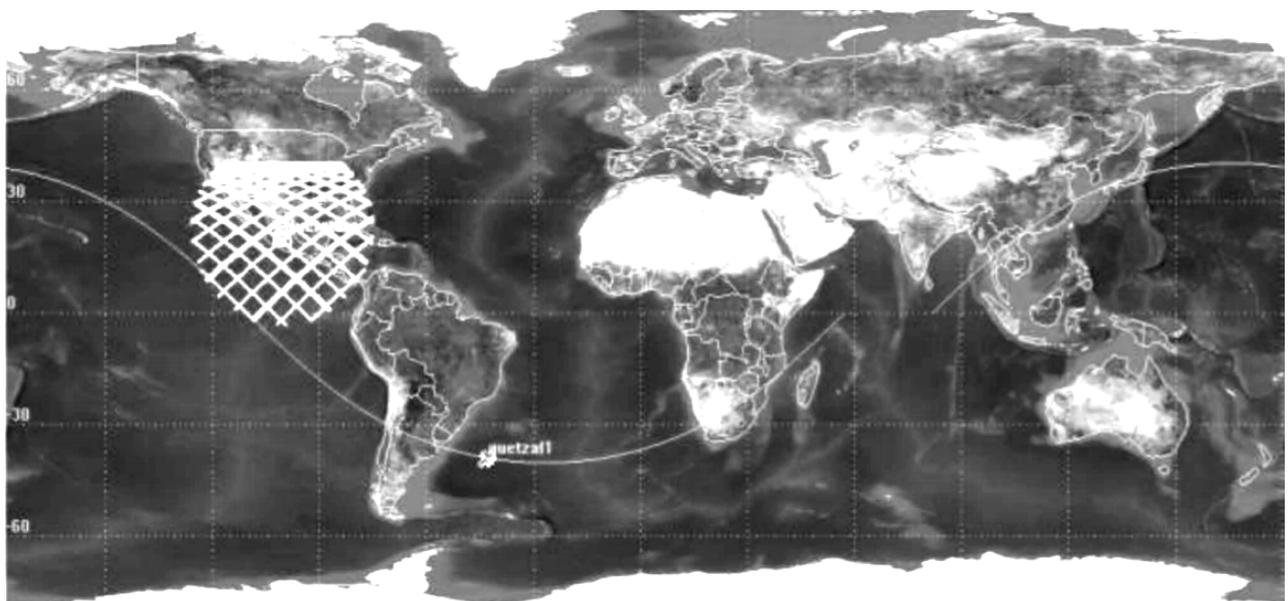


Figura 1.3: Simulación orbital Satélite Quetzal 600 km 40° de inclinación [2]

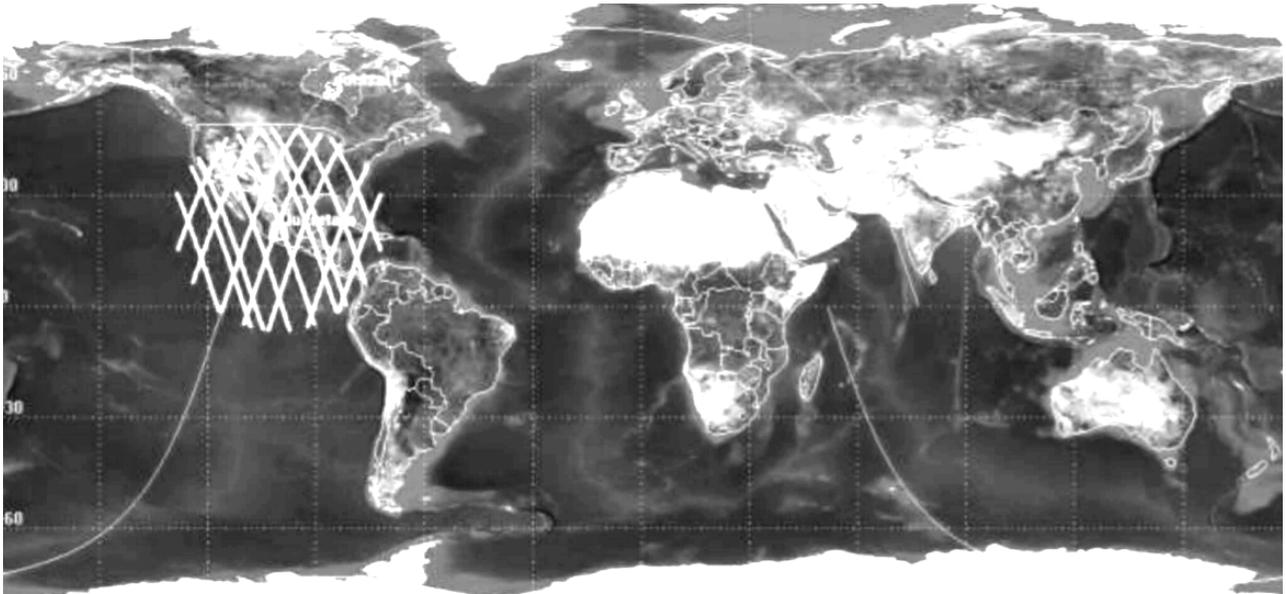


Figura 1.4: Simulación orbital Satélite Quetzal 800 km 105° de inclinación [2]

La plataforma satelital de Quetzal consistiría en los siguientes subsistemas, que serán desarrollados bajo los requerimientos y limitaciones de la carga útil seleccionada procurando el uso de elementos de bajo costo, COTS: navegación, determinación de posición y control, suministro de potencia, estructural, control térmico, telecomunicaciones, sistema de comando y manejo de información, telemetría y propulsión [2].

1.3. Objetivos Generales de la misión

La misión del satélite Quetzal se definió con base a las necesidades actuales para el monitoreo de contaminantes atmosféricos sobre ciudades mexicanas y de Latinoamérica. Se busca correlacionar las fuentes humanas y naturales en el impacto de la contaminación. El impacto directo social de Quetzal viene de acciones para mitigar las condiciones de contaminación debido a las emisiones generadas por los automóviles, volcanes, parques industriales, etc., con el objetivo de mejorar el *modus vivendi* de la sociedad.

En este sentido, uno de los objetivos más importantes del desarrollo del satélite Quetzal radica en el desarrollo y aplicación de tecnología satelital con alto impacto social. Por otro lado, existe una gran motivación para el desarrollo del satélite Quetzal dado que, en la actualidad hay diversos satélites que llevan a cabo misiones similares, con la particularidad, de que son demasiados grandes, costosos, complejos y que son prácticamente irrealizables para países emergentes en el área satelital. Los objetivos generales de la misión son:

- Intercambio académico.
- Desarrollo de recursos humanos en el área espacial enfocada a satélites pequeños.
- Visita de especialistas del MIT para cursos en la UNAM.
- Visita de alumnos de la UNAM al MIT para actividades.
- Diseño conceptual de la misión.
- Diseño preliminar de los subsistemas espaciales y carga útil.
- Requerimientos específicos del segmento terrestre.
- Análisis de riesgo, costos y planeación.
- Diseño, desarrollo y fabricación del satélite Quetzal.
- Generar aplicaciones espaciales con alto impacto social.

1.4. Objetivos de la implementación del Algoritmo de Determinación de Posición y Control

El trabajo desarrollado en esta tesis consiste en la implementación de un algoritmo de determinación de posición y control para un micro satélite. El objetivo de este trabajo es el uso de un microprocesador embebido en un FPGA para verificar su viabilidad como base de procesamiento para el desarrollo de los Algoritmos de Determinación y Control. Cabe mencionar que el alcance del trabajo se limita únicamente a la implementación del algoritmo en una plataforma FPGA.

Capítulo 2

Requerimientos de diseño para el subsistema de Determinación de Posición Y Control (ADCS)

En este capítulo se enumeran los requerimientos para diseñar un ADCS y se mencionan lo que es un Sistema de Comando y Manejo de Información.

2.1. Sistema de Comando y Manejo de Información

El sistema de comando y manejo de información (SCMI), tiene dos funciones primordiales. Recibe, valida, codifica y distribuye comandos a los subsistemas del satélite, ejecuta, procesa y da formato a los datos de telemetría del satélite para su transmisión o procesamiento a bordo del satélite.

Este sistema provee autonomía y control a las diferentes funciones del satélite, mediante un manejo de comandos de control y monitoreo a través de una interfaz única permitiendo el manejo de múltiples subsistemas.

El tamaño del SCMI es proporcional a la complejidad del satélite. La confiabilidad puede por sí misma duplicar el tamaño del hardware necesario en caso de ser necesario el uso de redundancia en el SCMI. Es importante definir el concepto de telemetría ya que es aplicado en el SCMI, de acuerdo al estándar 100 del *IEEE*, (Institute of Electrical and Electronics Engineers), telemetría se puede definir de la siguiente manera: medición intermedia con la intención de permitir que dicha medición pueda ser interpretada a distancia del detector primario, esto incluye suministro para convertir la medición en una cantidad representativa de otra, que puede ser transmitida para su medición a distancia.

Un SCMI combina y maneja información de telemetría de diversas fuentes para su uso abordo de la nave o transmisión a estación terrena. La mayoría de los SCMI son de tipo multiplexado, generan secuencias de entradas con un orden predeterminado y las organizan en un formato fijo de salida. Información proveniente de todas las entradas es digitalizada y formateada en tramas seriales de datos continuos para su bajada a través del sistema de transmisión [9]. El SCMI debe ser capaz de genera una tasa de muestreo suficiente en todos los datos para mantener una correcta comunicación entre el satélite y la estación terrena. La tasa de muestreo de datos es la suma de todas las tasas de muestreo de entradas más un ancho de banda adicional para la inserción de códigos de sincronización y un contador de identificación.

El diseño del SCMI se divide en 5 etapas: proceso de dimensionamiento del SCMI, identificación de los requerimientos y limitantes, determinación de la complejidad de las funciones del SCMI, determinación del nivel de complejidad de todo el SCMI, y estimación de tamaño, peso y consumo de potencia de cada componente. En la Tabla 2.1 de la página 17 se enumeran las etapas de diseño del SCMI, así como el procedimiento, información y factores a considerar en cada una de ellas.

El SCMI es uno de los últimos subsistemas en ser definidos. Es una herramienta usada para configurar, controlar o programar la carga útil y otros subsistemas, incluyendo el ADCS. El SCMI no puede ser completamente definido hasta que los requerimientos de los sistemas a monitorear sean establecidos.

Para evaluar la arquitectura de hardware y su habilidad para cumplir con los requerimientos del algoritmo de control, se usan programas denominados *Benchmark* que contienen una mezcla de instrucciones específicas de modo que pueda ser usado para comparar distintas arquitecturas de computadora [9].

Etapa	Procedimiento	Datos necesarios y consideraciones
Identificación de funciones a realizar por el SCMI.	Determinar o estimar que debe ser controlado y monitoreado en el satélite para completar la misión. Determinar que tareas deben ser realizadas a bordo del satélite.	Procesamiento de comandos: si es requerido procesamiento de comandos: -¿Cuál es la taza de comandos? -¿Cuántos canales son necesarios? -¿Existe computadora? -¿Se requiere almacenamiento para los comandos? Procesamiento de telemetría: si es requerido procesamiento de telemetría: -¿Cuántos canales? -¿Cuál es la taza de datos de telemetría? -¿Cuál es la taza de datos de la carga útil? -¿Es requerida una interfaz de computadora? Otros: ¿Se requiere un reloj de misión? ¿Se requiere un watchdog para la computadora? ¿Se requiere ADCS?
Identificación de requerimientos y limitaciones.	Determinar los parámetros, derivados de aspectos del diseño general del satélite, que impactan el SCMI.	Restricciones de plataforma satelital. Confiabilidad. Tiempo de vida de la misión. Ambiente de radiación. Diagrama de tiempos. Presupuesto.
Determinación de complejidad de funciones del SCMI.	Estimar la complejidad de cada función identificada en la etapa 1, aplicando las restricciones establecidas en la etapa 2.	Evaluar cada función por separado.
Determinación de la complejidad de todo el SCMI.	Colectar funciones en comandos y componentes de telemetría y determinar la complejidad compuesta de cada uno.	¿Hay algunas otras funciones como el tiempo de reloj de la misión combinadas en comandos de telemetría?
Estimación de tamaño, peso, y consumo de potencia para cada componente.	Aplicar resultados en etapa 4 para componentes de comando y telemetría.	Sistemas combinados

Tabla 2.1: Etapas de diseño del SCMI [9].

2.2. Sistema de Determinación de Posición y Control

También llamado Sistema de Posición y Control Orbital, Sistema de Posición Terrena, Navegación y Control, dependiendo de las necesidades particulares de la misión; en este trabajo se hará referencia a Subsistema de Determinación de Posición y Control (ADCS, por sus siglas en inglés).

Se define posición como la orientación en tres dimensiones con respecto a un marco de referencia. Sistema de posición incluye sensores, actuadores, aviónicas, algoritmos, software, y soporte terreno usado para determinar y controlar la posición del satélite [1].

La determinación de posición es el proceso de combinar información disponible por los sensores, con información conocida de la dinámica del satélite para proveer un solución única a la posición como función del tiempo, mediante procesamiento a bordo o posterior. Debido al desarrollo tecnológico muchos algoritmos de control cuyo procesamiento se realizaba posterior ahora se realizan a bordo. La determinación de posición se realiza mediante el uso de sensores que permiten relacionar información de referencias externas, como estrellas, el sol, la Tierra, la orientación del satélite. El uso de varios sensores mitiga el nivel de ruido introducido al sistema de determinación de posición. Esta tarea es compleja ya que existe una gran gama de combinaciones de sensores que se pueden hacer en función de los modos de operación, métodos de filtrado tales como el filtro de Kalman [1].

El control de posición es la combinación de la predicción y reacción de la dinámica rotacional del satélite. Se pueden diseñar e implementar controladores que aprovechen las perturbaciones naturales del medio en el que trabajará el satélite para lograr estabilización pasiva en su posición; el uso de este método depende de las necesidades de la misión en cuestión de grado de apuntamiento. Por otro lado, el uso de actuadores como propulsores, pares magnéticos y ruedas inerciales para generar controladores activos de posición producen una capacidad superior en el grado de apuntamiento del satélite.

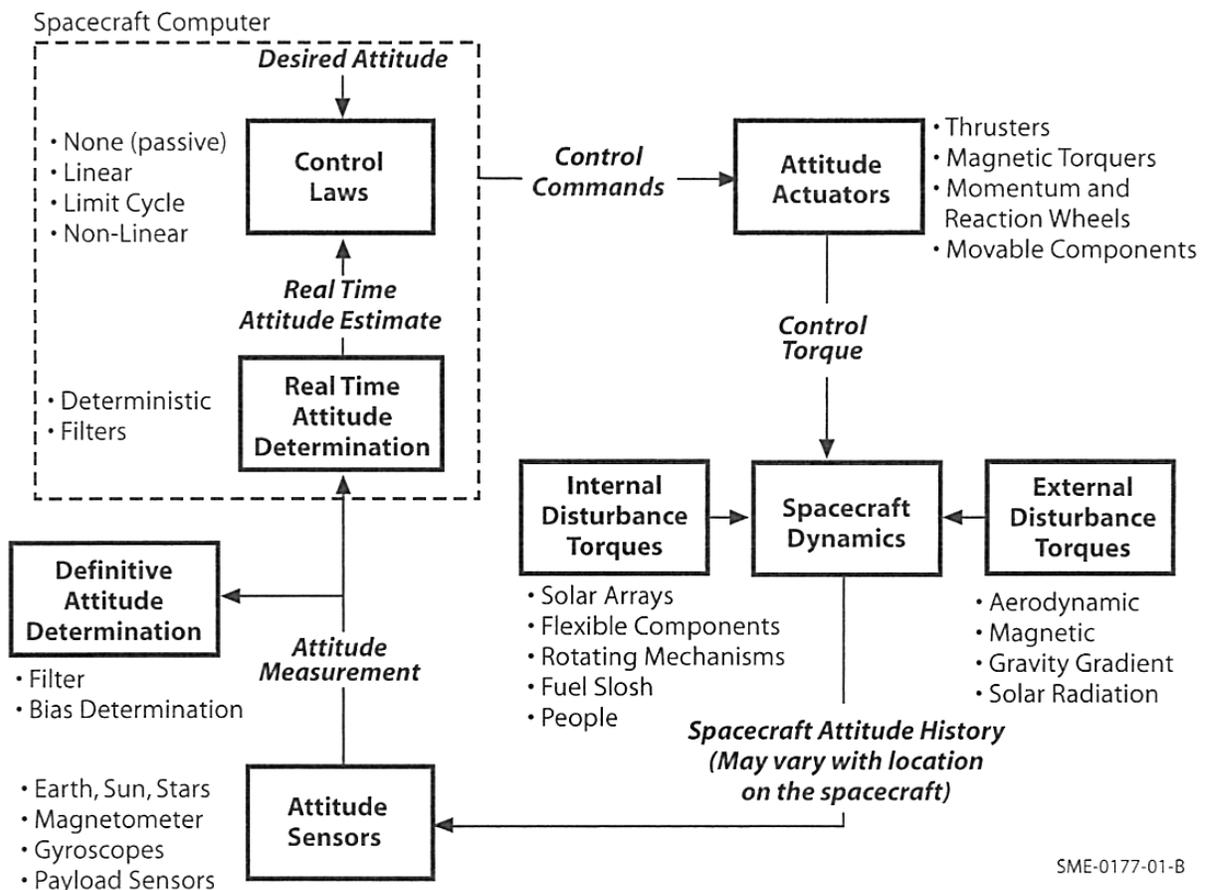
2.2.1. Importancia del ADCS en una plataforma satelital

El ADCS interactúa y tiene repercusiones en los requerimientos de la misión, el subsistema de potencia, propulsión, comunicaciones, estructural y térmico.

Dependiendo la carga útil se requerirán cierto grado de apuntamiento inercial o hacia la tierra, la precisión y grado de estabilidad del apuntamiento, mantenimiento de la órbita, autonomía y navegación a bordo. El subsistema de control térmico puede requerir maniobras especiales para cumplir sus objetivos. Las comunicaciones requieren un grado de apuntamiento, que también está en función de la cantidad de información a transmitir generada por la carga útil. El subsistema de potencia podría requerir modos de apuntamiento para lograr cumplir con las necesidades energéticas de la misión. Por otro lado, el ADCS influye en el

sistema de propulsión teniendo que considerar el tipo de propulsor que se selecciona, la carga de combustible y cantidad de impulso requerido. En el subsistema estructural, el ADCS modifica y se limita por los valores de centro de masa, inerciales, flexibilidad, ubicación de propulsores y sensores. Por último, el grado de carga que puede ser aportado y el tipo de regulación utilizado por el subsistema de potencia es afectado por el ADCS.

En aplicaciones de percepción remota como captura de imágenes, uso de espectrómetros, sistemas de comunicaciones, sistemas de navegación, se requiere un grado de apuntamiento muy alto y el cumplimiento de éste es fundamental para el éxito de la misión. Es así como los requerimientos de apuntamiento de la carga útil son el factor más importante en la determinación del tipo de actuadores y sensores. En la Figura 2.1 se muestra un ADCS completo detalladamente.



SME-0177-01-B

Figura 2.1: Diagrama de un ADCS completo [1].

2.3. Requerimientos de diseño del ADCS

Como se menciona en Wiley et al. [9], el sistema de comando y manejo de información (SCMI) tiene la tarea de coleccionar información relativa al estado de salud de los subsistemas del satélite para la toma de decisiones a bordo o para su envío a la estación terrestre y recibir comandos, además se puede requerir que el SCMI tenga que procesar y almacenar gran cantidad de información. Estas funciones demandan arquitecturas especiales, que ofrezcan autonomía y confiabilidad. El funcionamiento del sistema de control de posición y estabilización es crítico para la comunicación y operación del satélite.

En la etapa inicial de diseño para el satélite Quetzal se toman como base los requerimientos del satélite CASTOR (*Cathode/Anode Satellite Thruster for Orbital Repositioning*) [10] para generar los primeros prototipos de subsistemas. Estos requerimientos son los siguientes:

Primero, la nave debe orientar el motor principal paralelo al vector velocidad de la órbita deseada. El propulsor principal debe ser orientado en línea o en oposición al vector velocidad. Cualquier imprecisión en el apuntamiento resultará en un decremento de la eficiencia direccional de propulsión y una pérdida en ΔV . Se requiere alcanzar 97% de eficiencia en tiempo promedio en la dirección de propulsión.

Segundo, la nave debe orientar los paneles solares en dirección del sol de modo que los rayos del sol sean normales al área de los paneles solares. El grado de apuntamiento requerido para la generación de potencia debe mantener 97% de eficiencia, lo que equivale a tener un ángulo de error de 14° . Finalmente la nave debe ser controlada en base a la ejecución de los algoritmos de control diseñados para su misión.

La orientación instantánea de la nave debe ser conocida para mantener estos valores de eficiencia. Con base en [10] la relación necesitada entre la precisión de actuación y la precisión de estimación es 5 a 1 [9]. De este modo la precisión de estimación de posición apropiada es un grado en cada eje; por lo tanto son requeridos 4 sensores de sol y un sensor magnético. Una unidad de medición de inercia (IMU) es requerida para integrar aceleraciones y determinar la posición en periodos en que los sensores no estén disponibles. Además, en [10] se deriva mediante un análisis cómo un error en la determinación de la posición crea un error en la estimación de posición. De tal modo que tener un error de entrada de referencia de 166 arc segundos (0.05 grados), lo que conduce a que tener dichos valores en la medición llevará a 0.05 grados aceptables de error en la posición.

De este modo se tienen los siguientes puntos principales a considerar:

- ↔ Estimar y controlar la posición del satélite para garantizar una eficiencia del 97% en el vector de propulsión.
- ↔ Estimar la posición del satélite con 1° de su posición real en 3 ejes.
- ↔ Controlar la posición del satélite con 5° respecto a las entradas en los tres ejes.

- ↷ Ser capaz de controlar la posición del satélite en un rango dinámico de 360° en los tres ejes.
- ↷ Ser capaz de realizar giros de 180° en el cuerpo del satélite en un periodo de 10 minutos.
- ↷ Efectuar estimación y control de la posición del satélite a una frecuencia mínima de 1 Hz.
- ↷ Reducir el giro del satélite a una magnitud manejable durante el despliegue inicial y la salida desde el eclipse.
- ↷ Ser capaz de aplicar torque de 0.033N-m.

Por otro lado, simulaciones y pruebas nos brindan información sobre los requerimientos de los sensores [10]:

- ↷ El vector de propulsión puede ser de 97.5 % en promedio.
- ↷ El sensor de sol debe ser caracterizado para tener determinador de error.
- ↷ Un rango de 5° en el control puede ser alcanzado en operación.
- ↷ Ruedas de reacción proveen movimiento rotacional alrededor de 3 ejes por 360° ; redundancia es obtenida usando torques magnéticas.
- ↷ Frecuencia de estimación de la posición es ejecutada a 1 Hz. Los sensores son muestreados a frecuencias iguales o mayores a 5 Hz cada uno. La información del GPS es muestreada a 5 Hz.

A partir de los requerimientos de diseño, en base a las necesidades de la misión CASTOR, se derivan varias tareas para el desarrollo del sistema ADCS.

En este trabajo de tesis nos enfocaremos sólo al control magnético del ADCS. Por tanto, el alcance y objetivos de este trabajo se limitan a la implementación de una propuesta de algoritmo de Determinación de Posición y Control con pares magnéticos como actuadores para una plataforma satelital micro, que servirá como base para un futuro desarrollo del ADCS del satélite Quetzal. Los resultados serán comparados con una simulación numérica hecha en software dedicado.

2.3.1. Requerimientos energéticos y de masa para el subsistema de Determinación de Posición y Control

De acuerdo a los sensores y actuadores que cumplen con las necesidades de apuntamiento y eficiencia para la misión se calculan la cantidad de masa y potencia requeridas por el ADCS que atenderá dicho sistema. En la Tabla 2.2 se muestran los presupuestos energéticos, de masa, de potencia y costo para el satélite CASTOR. Cabe mencionar que los costos que se presentan son de componentes comprados, los únicos componentes fabricados son las bobinas magnéticas.

Componente	Cantidad	Potencia nominal [W]	Potencia máxima [W]	Masa total [Kg]	Costo total [\$USD]
Sensor de sol	4	0.1	0.3	0.136	48000
Sensor magnético	1	0.0024	0.003	0.022	480
Ruedas de reacción	3	1.5	21	0.675	90000
Bobinas magnéticas	3	3	6	0.75	750
IMU	1	0.165	0.285	0.02	620
Unidad GPS	1	0.5	0.8	0.02	23010
Receptor GPS	1			0.012	
Total		5.2674	28388	1.885	164860

Tabla 2.2: Presupuestos energético, masa, potencia y costo del ADCS de CASTOR [10].

2.3.2. Requerimientos debidos al ambiente espacial

Como se mencionó en el punto 2.1, el medio ambiente espacial requiere ser considerado en el diseño del SCMI debido a que afecta el comportamiento de los subsistemas que conforman la plataforma satelital.

El medio ambiente espacial se define como la región más allá de la atmosfera terrestre, donde fuentes de energía no terrestre comienzan a dominar y dónde el satélite opera.

El ambiente espacial cercano a la tierra incluye la atmosfera superior de la Tierra, ionosfera, cinturones de radiación y magnetosfera. El ambiente espacial cercano a la Tierra y condiciones atmosféricas influyen fuertemente en el rendimiento y vida útil de operación de sistemas espaciales, afectando su tamaño, peso, complejidad y costo.

Ciertas partículas presentes en el ambiente espacial interactúan con los sistemas satelitales limitando su desempeño y potencia, pudiendo también ocasionar fallas críticas en los sistemas. Es por esto que un SCMI tiene requerimientos debido al ambiente espacial, estos

requerimientos influyen directamente en la selección de tecnología, tamaño de la misma y diseño de ADCS. De acuerdo a Bedingfield [3], el ambiente espacial se puede dividir en 9 medios específicos: Termosfera neutral, ambiente térmico, ambiente de plasma, ambiente de meteoritos y basura espacial, ambiente solar, ambiente de radiación ionizante, ambiente de campo magnético, ambiente de campo gravitacional, y mesosfera.

En la Tabla 2.3 se enumeran los medios específicos, su definición, su relación con elementos de diferentes subsistemas y los modelos en que están basados.

	Definición	Cuestiones programáticas	Modelos/Bases de datos
Termósfera Neutral	Densidad Atmosférica, Variaciones de Densidad, Composición atmosférica (Oxígeno atómico), Vientos Atmosféricos.	Diseño de Sistemas GN&C, Degradación de Materiales/Erosión de Superficie (fluencias de oxígeno atómico), Arrastre/Decaimiento, Tiempo de Vida, Anulación de Colisiones, Apuntamiento de Sensores, Diseño Esperimental, Errores en Posicionamiento Orbital, Seguimiento de Pérdida.	Jacchia/MET, MSIS, LIFTIM, Upper Atmospheric Wind Models.
Ambiente Térmico	Radiación Solar (Albedo y variaciones de OLR), Transferencia de radiación, Transmiancia Atmosférica.	Diseño de Sistema de Control Térmico activo y pasivo, Selección del tamaño y material del radiador, Posicionamiento de potencia, Diseño de Arreglo solar.	Bases de datos ERBE, ERB, NIMBUS, ISSCP, Modelos Climáticos, Modelos de Circulación General (GCM).
Plasma	Plasma Ionosférico, Plasma Auroral, Plasma Magnetosférico.	EMI, Diseño de S/C de Sistemas de Potencia, Determinación de Materiales, S/C Calentamiento, S/C de Carga/Arqueo.	Modelos Ionosféricos de Referencia Internacional, NASCAP/LEO, NASCAP/-GEO, POLAR
Meteoritos y Escombros Orbitales	Flujo M/OD, Distribución de dimensiones, Distribución de masas, Distribución de velocidad, Direccionalidad.	Anulación de Colisiones, Supervivencia de tripulación, Efectos secundarios de ección, Diseño Estructural/Blindaje, Materiales/Deterioro de Paneles solares.	Modelos de Flujo.
Medio Ambiente Solar	Dinámica y física solar, Tormentas Geomagnéticas, Predicciones de Actividad solar, Índices Solares/-Magnéticos, Constantes solar, Espectro solar.	Predicción solar, Tiempo de vida/Evaluaciones de arrastre, Cargas/Calentamiento por reingreso, Entradas para otros modelos, Operaciones de contingencia.	Modelo MSFC EL Laboratory, Datos de predicción NOAA, Modelos estadísticos, Base de datos solar.
Radiación Ionizante	Radiación por Protones/-Electrones atrapados, Rayos Galácticos Cósicos (GCR), Eventos de Partículas Solares.	Niveles de Radiación, Efectos electrónicos por Dosis en partes, Efectos electrónicos por single event upset, Niveles de dosis en materiales, Niveles de dosis en humanos.	CREME, AE-8MIN, AE-8MAX, AP-8MIN, AP-8MAX, Radbelt, Solpro, SHIELDDOSE.
Campo Magnético	Campo Magnético Natural	Corrientes inducidas en estructuras grandes, Localización de la Anomalía del Atlántico Sur, Localización de cinturones de radiación.	IGRF2010.
Campo Gravitacional	Campo Gravitacional Natural.	Mecánica/Rastreo Orbital.	GEM-T1, GEM-T2.
Mesósfera	Densidad Atmosférica, Variaciones de Densidad, Vientos.	Reingreso, Selección de Materiales, Diseño experimental de atado.	Earth-GRAM 95, UARS, Mars-GRAM 3.34.

Tabla 2.3: Elementos del ambiente espacial [3].

Las diferentes capas de la atmósfera de la Tierra se pueden distinguir en la Figura 2.2.

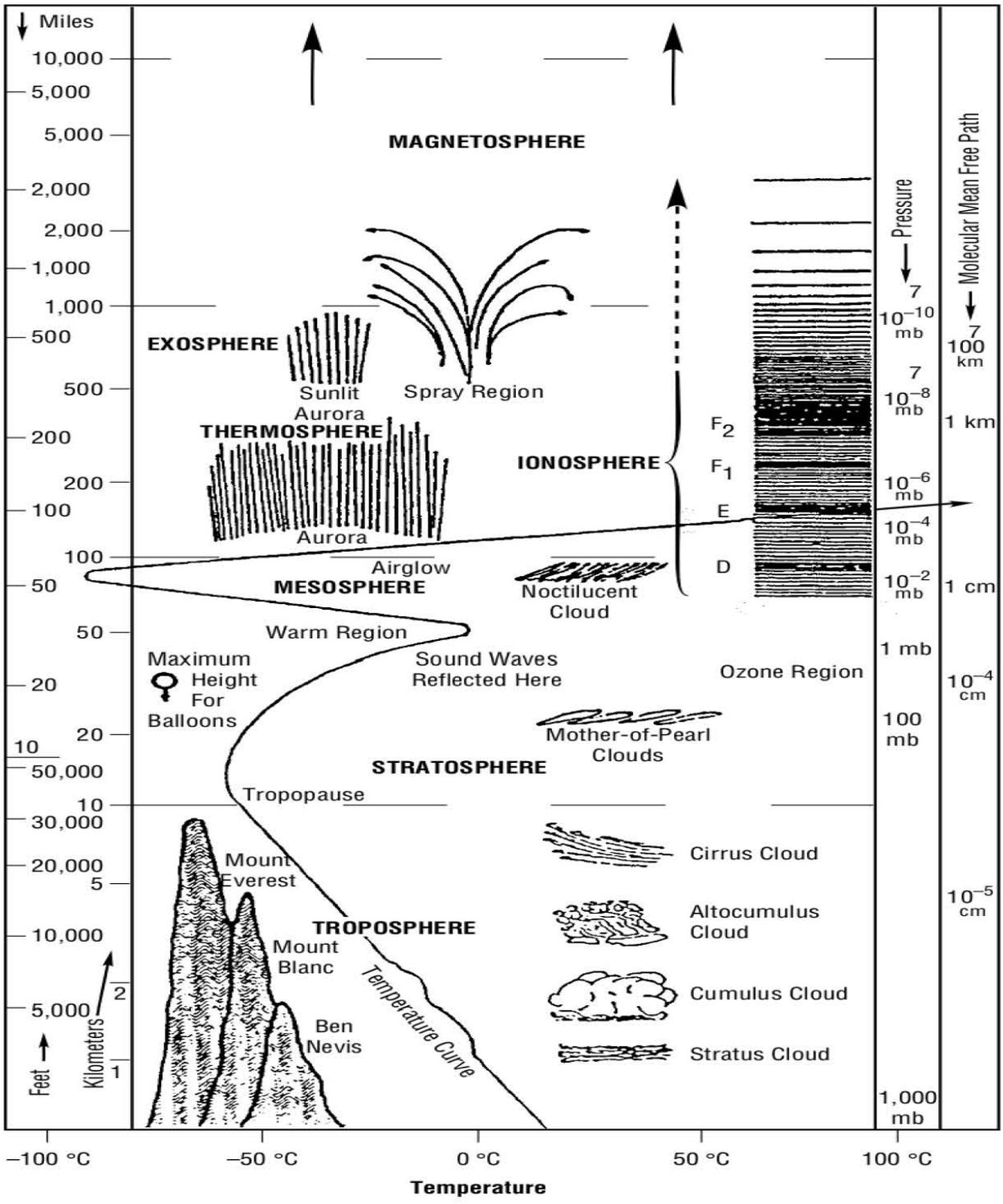


Figura 2.2: Capas de la Atmósfera terrestre [3].

2.3.2.1. Efectos del medio ambiente espacial

Dados los estados de la atmosfera espacial, existe una gran cantidad de efectos e interacciones entre dichos estados y los subsistemas que conforman una plataforma satelital. Los efectos de estas interacciones pueden influir el diseño y generar funcionamiento erróneo en los subsistemas. En las Tablas 2.4 y 2.5 de las páginas 27 y 28 respectivamente, se muestran las interacciones y efectos del ambiente espacial en los subsistemas de una plataforma satelital.

2.3.2.2. Efectos de radiación espacial en los dispositivos electrónicos

Entre los efectos producidos por el medio ambiente espacial para el diseño de sistemas electrónicos se da especial atención a los efectos por radiación espacial. Los efectos de la radiación en dispositivos electrónicos se puede clasificar en dos tipos: efectos de dosis total de ionización (TID, por sus siglas en inglés), y efectos de evento simple (SEE, por sus siglas en inglés). Efectos TID son descritos como los efectos debidos a la acumulación de partículas cargadas en los niveles de dopaje de los dispositivos electrónicos semiconductores, esta acumulación se da a lo largo del tiempo en que los dispositivos electrónicos, a bordo de satélites, experimentan el impacto de partículas cargadas durante su viaje en órbita, y a partir de los niveles específicos de acumulación de partículas se presentan efectos según el dispositivo electrónico que es impactado. Por otro lado, los efectos SEE se refieren a la alteración instantánea del circuito electrónico como resultado del impacto de una partícula cargada de alta energía que interactúa de tal manera con el material que lo daña.

	Medios ambientes espaciales			
Subsistemas Satelitales	Termósfera Neutral	Ambiente Térmico	Plasma	Meteoritos/Desechos Orbitales
Aviónica		Diseño Térmico	Transtorno debido a EMI por Arqueo, S/C de Carga	EMI debido a Impactos
Potencia Eléctrica	Degradación del rendimiento de arreglo de Paneles	Diseño de Arreglo de Paneles, Asignación de Potencia, Rendimiento del Sistema de Potencia	Cambio en Potencial Flotante, Perdidas de Corrientes, Reatracción de Contaminantes	Daño de celdas solares
GN&C/ Apuntamiento	Diseño del Sistema de Apuntamiento		Torques debido a potencial inducido	Anulación de Colisiones.
Materiales	Selección de Materiales, Degradación de Materiales	Selección de Materiales	Arqueo, Propiedades de efectos de contaminación en superficie	Propiedades de degradación de superficies ópticas.
Óptica	S/C de incandescencia, Interferencia con sensores	Diseño de influencias ópticas	Reatracción de Contaminantes, Cambio en las propiedades en superficies ópticas	Degradación de propiedades de superficies ópticas.
Propulsión	Composición de arrastre/Requerimientos de combustible		Cambio en Potencial Flotante debido a disparos de propulsores haciendo contacto con el Plasma	Anulación de colisiones, Blindaje adicional incrementa requerimientos de combustible, Ruptura de tanques presurizados.
Estructuras		Superficies térmicamente sensibles, Fatiga, Vibraciones inducidas térmicamente	Tamaño estructural influencia el S/C de Carga	Daño estructural, Diseño de blindaje, Masa de S/C, Supervivencia de tripulación.
Telemetría, Seguimiento y Comunicaciones	Posibles errores en seguimiento, Posibles errores de pérdida.		EMI debido a arqueo.	EMI debido a impactos.
Control Térmico	Cargas por reingreso/Calentamiento, Degradación de superficie debido a Oxígeno Atómico.	Diseño del sistema de control térmico activo y pasivo, Dimencionamiento del Radiador, Puntos de congelamiento.	Reatracción de contaminantes, Cambio en propiedades de emitancia.	Cambio en propiedades Térmicas/Ópticas.
Operaciones de Misión	Lineas de tiempo de relanzamiento, Valoración de tiempo de vida de S/C.	Influencia Planeación/Sequenciación de la misión.	Lineas de tiempo de servicio (EVA).	Supervivencia de tripulación.

Tabla 2.4: Efectos del ambiente espacial en los subsistemas de una plataforma satelital 1 de 2 [3].

	Medios ambientes espaciales				
Subsistemas Satelitales	Ambiente Solar	Radiación Ionizante	Campo Magnético	Gravitacional	Mesósfera
Aviónica	Diseño Térmico	Degradación: SEU, errores de Bit, cambios de Bit	Efectos Potenciales Inducidos		
Potencia Eléctrica	Diseño de Arreglos de Paneles Solares, Distribución de Potencia.	Disminución en salida de celdas solares.	Efectos Potenciales Inducidos		
GN&C/ Apuntamiento	Influencia Densidad y Arrastre.		Dimencionamiento de Torques Magnéticos	Estabilidad y Control, Torques gravitacionales.	Efectos en GN&C por reingreso.
Materiales	Exposición solar UV necesitada para selección de Materiales.	Degradación de Materiales.			Degradación de Materiales debido a Interacciones Atmosféricas.
Óptica	Información necesaria para diseño Óptico.	Oscurecimiento de ventanas y fibras ópticas.			
Propulsión	Influencia Densidad y Arrastre			Influencia el consumo de combustible	
Estructuras	Influencia el posicionamiento de Estructuras térmicamente sensibles.		Induce corrientes en estructuras grandes.	Presupuesto de propelente	Diseño de estructura de atadura.
Telemetría, Seguimiento y Comunicaciones	Precisión de seguimiento, Influencia Densidad y Arrastre		Localización de la Anomalía del Atlántico.	Puede inducir errores de seguimiento.	
Control Térmico	Influencia cargas/-calentamiento por reingreso.				
Operaciones de Misión	Lineas de tiempo de la misión, Planeación de la misión	Lineas de Tiempo de reemplazo de tripulación.			

Tabla 2.5: Efectos del ambiente espacial en los subsistemas de una plataforma satelital 2 de 2 [3].

2.3.2.3. Efectos de Eventos Únicos

Los efectos SEE son perturbaciones eléctricas causadas por el impacto de partículas de energía sobre las capas semiconductoras de un dispositivo electrónico. Partículas pesadas de ion son los causantes, generalmente, de los SEE. Esta energía es medida como energía lineal de transferencia (LET, por sus siglas en inglés), cuyas unidades son $MeV - mg/cm^2$. Existen diferentes tipos de efectos SEE [27]:

Eventos de trastorno simple (SEU, por sus siglas en inglés), ocurren cuando una partícula cargada genera que una o más celdas de memoria o celdas de configuración cambien de estado, si sólo una celda cambia este SEU se conoce como *Single Bit Upset* (SBU), en el caso en que se modifiquen varias celdas se conoce como *Multi-bit Upset* (MBU).

Eventos Simples Transitorios (SET, por sus siglas en inglés), ocurren cuando una partícula cargada causa un pico de corriente o tensión. Si la duración de este evento dura lo suficiente, puede quedar fijo en un *flip-flop* y propagarse a través del circuito generando un daño mayor.

Evento Simple de Latchup (SEL, por sus siglas en inglés), es el estado alto de corriente de una estructura PNPN en un dispositivo semiconductor complementario de óxido metálico (CMOS), donde el corto circuito es mantenido debido a la realimentación del dispositivo. Dependiendo la duración del SEL, éste puede causar daño permanente en el dispositivo afectado.

Evento Simple de Interrupción de Funcionalidad (SEFI, por sus siglas en inglés), es una anomalía de un elemento interno de memoria que causa pérdida en la funcionalidad del dispositivo.

Evento Simple de Ruptura de Compuerta (SEGR, por sus siglas en inglés), afecta en mayor medida a los transistores de efecto de campo metal-óxido-semiconductor (MOSFET).

2.3.2.4. Niveles de dosis total de ionización por radiación

La dosis total de ionización (TID) es una degradación acumulativa en los dispositivos electrónicos, debido en mayor medida a las cargas depositadas por electrones y protones en el material de dispositivos electrónicos. Como resultado de esta acumulación de carga en regiones semiconductoras, el TID genera cambios en los estados de voltajes en compuertas, e incremento en corrientes de fuga [27]. La TID tiene como unidad de medida el rad, siendo un rad definido como la radiación absorbida por el dispositivo. Un rad equivale a $6.24E4eV$ o $10nJ$ de energía depositada por gramo de material.

Se denomina elementos endurecidos o *RadHard* a los componentes electrónicos que son diseñados para soportar efectos TID, esta tolerancia se especifica en *rads*, generalmente *krads* [28].

2.3.2.5. Estimación de niveles TID y Efectos de Eventos Únicos en la órbita del satélite Quetzal

En base a las órbitas propuestas para la misión del satélite Quetzal, en el punto 1.2, se pueden hacer estimaciones de los niveles de radiación que experimentarán los dispositivos electrónicos a bordo del satélite.

Para las estimaciones de los efectos de radiación en órbita utilizamos la plataforma el Sistema de Información del Medio Ambiente Espacial de la Agencia Espacial Europea (SPENVIS), este sistema es una interfaz en internet (www.spENVIS.oma.be) con modelos del entorno espacial y sus efectos, incluyendo los rayos cósmicos, los cinturones de radiación, partículas de alta energía del sol, plasma, gases y micro partículas.

Para la estimación de los efectos de radiación espacial en semiconductores específicamente utilizamos los modelos de fuentes de radiación y sus efectos que ofrece SPENVIS. Para la estimación del flujo de partículas solares y el espectro de Rayos Cósmicos Galácticos se seleccionó el modelo CREME-96.

Para hacer estimaciones con SPENVIS es necesario determinar el periodo en que estará el satélite en órbita. Para el satélite Quetzal, para los fines de esta estimación, se fijara como fecha de puesta en órbita el 1 de Junio del 2017 y se realizar un análisis de los efectos de radiación durante 1 año en órbita.

En l Figura 2.3 de la página 31 se presenta la estimación de la órbita más probable para la misión, órbita circular a 98° de inclinación y altura de 686 km, con un tiempo local en el nodo ascendente de 13:00 horas.

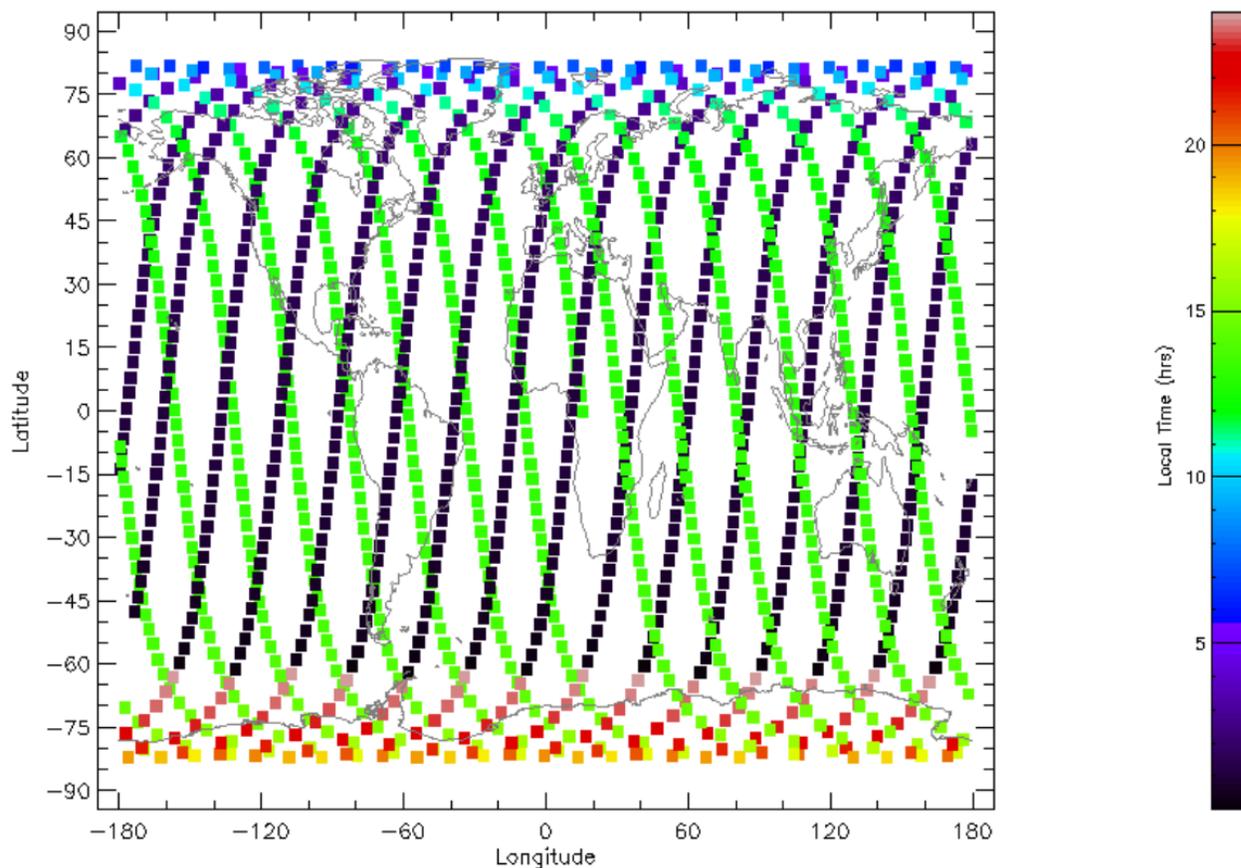


Figura 2.3: Proyección sobre la superficie terrestre de la órbita del satélite Quetzal y el tiempo local. Simulación obtenida con SPENVIS.

Para la determinación del blindaje óptimo contra los Efectos de Ionización Total, se obtiene las curvas de profundidad de ionización con SPENVIS usando el modelo SHIELDOSE-2. Estas curvas permiten estimar la radiación acumulada en los componentes semiconductores a bordo del satélite en un periodo determinado. En la Figura 2.4 de la página 32 se muestran las curvas de profundidad de radiación para la misión Quetzal. En la gráfica se muestra una comparación entre la dosis de radiación y el grosor del blindaje para un dispositivo electrónico. Se muestran las gráficas para diferentes tipos de partículas. Como se observa el grado de mitigación de radiación alcanza un máximo y a partir de ese punto aunque el grosor del blindaje aumente, no hay una reducción significativa en la cantidad de radiación.

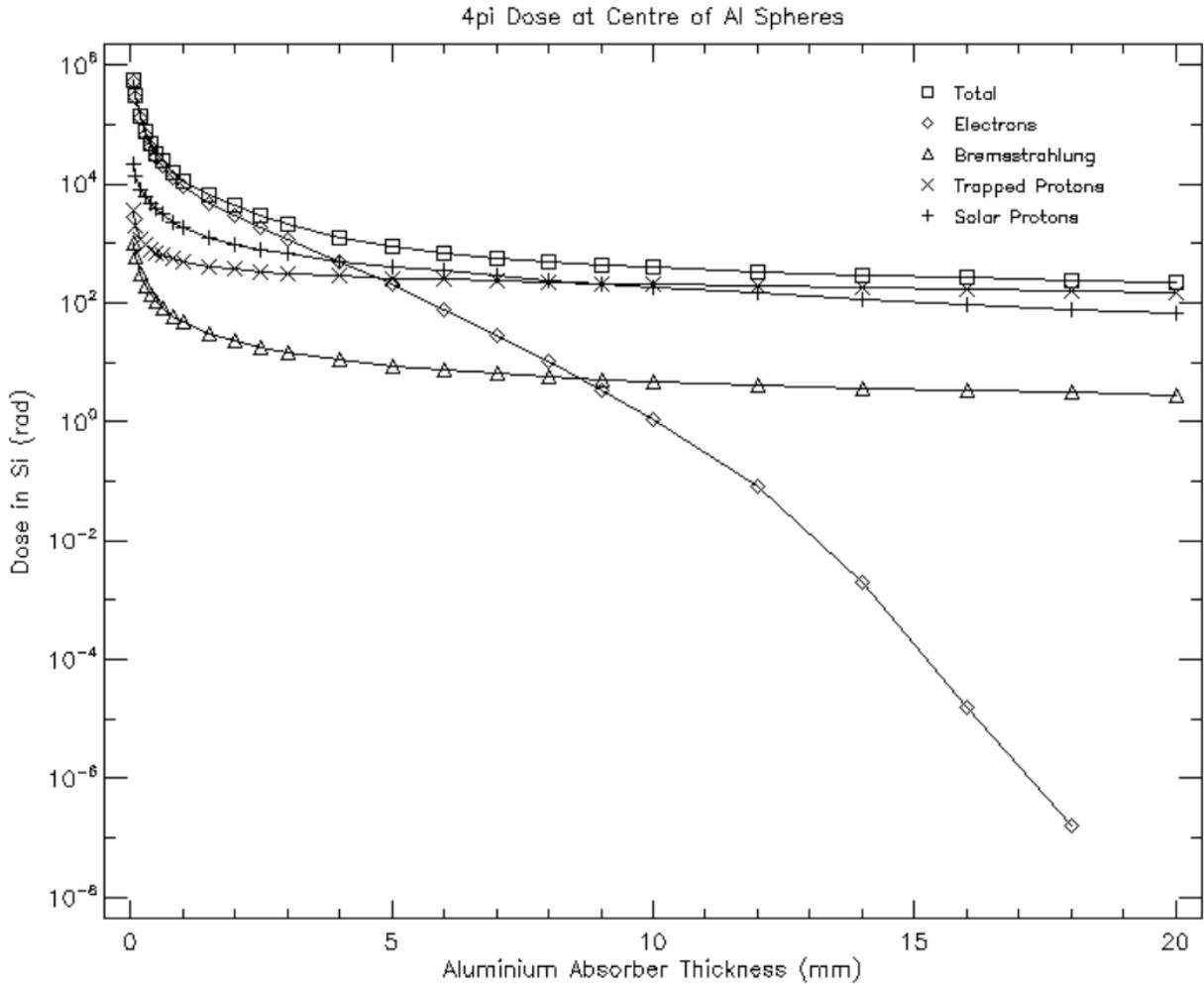


Figura 2.4: Curvas de profundidad de Dosis Total de Radiación para un semiconductor dentro de una esfera de aluminio con diferentes grosores de blindaje. Estimación obtenida con SPENVIS usando el modelo SHIELDOSE-2 para la misión Quetzal.

Para estimar la probabilidad de ocurrencia de los efectos SEE mediante SPENVIS se usa el algoritmo *Constant LET (CREME)*, este algoritmo permite obtener el flujo de partículas de alta energía que pueden impactar a un área sensible de un semiconductor, a partir de este flujo y con el valor LET de las partículas y el valor LET de umbral del componente semiconductor se obtienen modelos probabilísticos para obtener el número estimado de posibles errores SEE, en especial los errores SEU. En la Figura 2.5 de la página 33 se muestra el flujo de partículas de alta energía que pueden impactar a un componente semiconductor en la misión Quetzal.

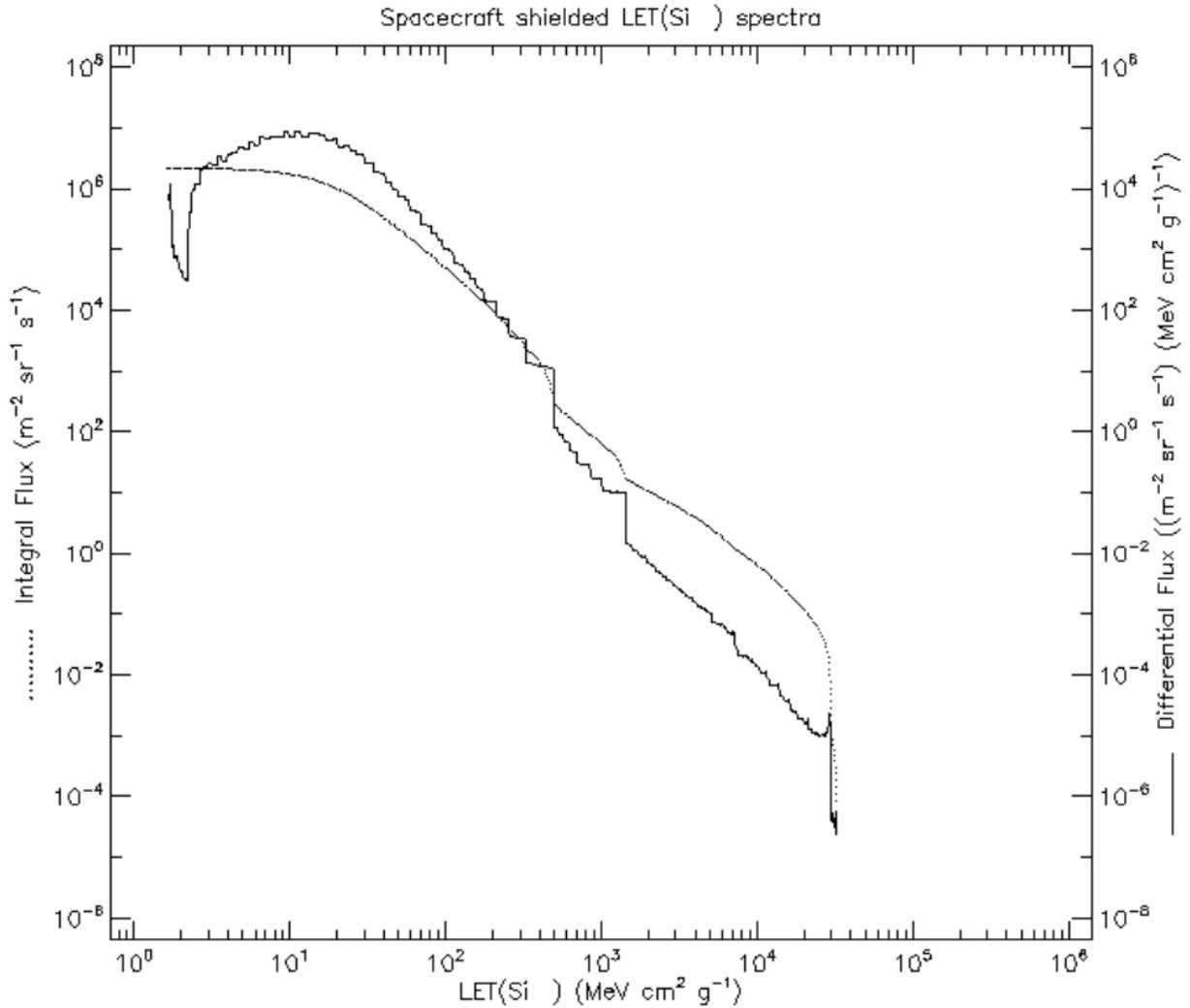


Figura 2.5: Flujo de partículas de alta energía para un volumen sensible paralelepípedo rectangular de $3.87\mu\text{m} \times 3.87\mu\text{m} \times 2.0\mu\text{m}$. Estimada con SPENVIS usando el algoritmo constant LET (CREME) para la misión Quetzal.

2.3.3. Requerimientos para el cumplimiento de los objetivos académicos de la misión

Es de interés académico para el desarrollo del presente proyecto implementar los algoritmos de control basado en tecnología FPGA y en el marco de su implementación. De este modo los requerimientos académicos de este trabajo son: 1) desarrollar la metodología de diseño para la implementación de los algoritmos de control basándose en un FPGA. 2) generar una propuesta de esquema tolerante a fallas para la implementación del algoritmo.

2.4. Metodología de diseño del ADCS

El diseño del ADCS es un proceso iterativo que puede ser dividido en 5 etapas:

1. Definición de modos de control, en esta etapa los requerimientos y perfil de la misión son evaluados, así como el tipo de inserción del vehículo de lanzamiento para definir una lista de los diferentes modos de control, requerimientos y limitantes a lo largo de la misión.
2. Cuantificación de perturbaciones, en esta etapa son considerados la geometría y órbita del satélite, modelos de campo magnético y solar, así como el perfil de la misión para determinar los valores de torques y externos que afectarán la dinámica del satélite [29].
3. Selección del tipo de control de posición del satélite, considerando la carga útil, los requerimientos de potencia y térmicos, dirección de apuntamiento, perturbaciones y nivel de precisión requerido se selecciona un método de control en 3 ejes, control de giro, por gradiente gravitacional, etc [30].
4. Selección y dimensionamiento del Hardware de ADCS, considerando la geometría, masa, requerimientos de precisión, geometría de la órbita, tiempo de vida estimada para la misión, condiciones de ambiente espacial, dirección de apuntamiento, velocidad de giro, detección de fallas y redundancia se seleccionan los tipos de sensores tales como de sol, de tierra, de estrella, inerciales, GPS, etc. De igual forma son seleccionados los actuadores, tales como ruedas de inercia, propulsores, torques magnéticos, etc. Es también en este punto donde se selecciona el método de procesamiento de información aviónica.
5. Como punto final se definen los algoritmos de Determinación de Posición y Control. En esta etapa del diseño del ADCS se hace consideración en el rendimiento en el método de estabilización, conocimiento de posición y precisión de apuntamiento, se debe hacer un balance entre los limitantes del sistema (térmico, potencia, tiempo de vida, sensibilidad a variaciones del sistema y capacidad de procesamiento del satélite). Como resultado en esta etapa se generan los algoritmos y parámetros para cada modo de determinación de posición y control, así como la lógica para cambiar entre cada uno de estos [31].

2.4.1. Modos de Control y Requerimientos

Los modos de control son diseñados para alcanzar series de requerimientos. Es un proceso iterativo en el que una cierta configuración de hardware de ADCS debe ser usado de diferentes formas para cumplir con todos los requerimientos. Debido a que el ADCS será dependiente de otros subsistemas, como el de potencia, estructural, las necesidades de éste impondrán a su vez requerimientos a estos subsistemas, incluyendo propulsión, control térmico y estabilidad estructural.

Las necesidades de la misión pueden imponer requerimientos al ADCS, una necesidad considerada como crítica es el control orbital, que requeriría que el satélite controle su posición durante la inserción de órbita. Durante la inserción de órbita se pueden generar grandes pares de perturbación, que pueden derivar en un diseño del ADCS con grandes actuadores por toda la misión.

Una vez que el satélite se encuentra en la órbita los requerimientos de apuntamiento propios de la carga útil suelen ser los que dominan el comportamiento del ADCS. Esto puede requerir control de posición relativo al planeta, inercial o respecto al giro.

Existen 6 modos de control de posición:

Adquisición: Determinación inicial de posición y estabilización del satélite para comunicación con la estación terrena y generación de potencia justo después del lanzamiento. También puede ser usado para recuperación en caso de perturbación o interrupción en el subsistema de potencia o emergencia.

Inserción de órbita: Periodo durante y después del impulso cuando el satélite es puesto en la órbita final. Incluye las opciones de no uso de control del satélite, estabilización de giro simple del motor del cohete, y control de posición completa usando un sistema de propulsión con combustible líquido.

Misión normal: También denominado *on-Station*, este modo es usado durante la mayor parte del desarrollo de la misión. Los requerimientos de este modo deben guiar el diseño del ADCS.

Giro: Reorientación del satélite cuando se es requerido.

Contingencia o modo seguro: Usado en situación de emergencia si el modo *on-Station* falla o es deshabilitado. Este modo por lo general usará menos energía o menos componentes para cumplir con los requerimientos mínimos de potencia y térmicos.

Especial : Los requerimientos de este modo pueden necesitar objetivos especiales en ciertos periodos, tales como cuando el satélite pasa a través de una sombra ocasionada por un cuerpo celeste.

Cada modo de control del ADCS requiere especificar sus necesidades de rendimiento, en la Tabla 2.6 de la página 36 se muestran los criterios de rendimiento para los diferentes modos de control, incluye su definición y ejemplos posibles para una misión teórica.

Criterio	Definición	Ejemplos/Comentarios
Precisión	Conocimiento de y control sobre la posición del vehículo relativa a un la posición de un objetivo como definido relativo a una referencia absoluta.	0.25 grados, 3σ , a veces incluye determinación de errores con errores de control.
Rango	Rango de movimiento angular sobre el que rendimiento de determinación y control debe ser alcanzado.	Cualquier posición dentro de 30 grados de nadir. Siempre que las velocidades angulares sean menores que 2 grados/segundo.
Fluctuación	Límite especificado en movimiento angular de alta frecuencia.	0.1 grados sobre 60 segundos, 1 grado/s, 1 a 20 Hz; previene desenfoco excesivo de datos de sensores.
Desvío	Límite en lentitud, movimiento angular de baja frecuencia.	0.01 grados sobre 20 minutos, 0.05 grados máximo; usado cuando el vehículo puede desviarse del objetivo con comandos de entrada poco frecuentes.
Respuesta Transitoria	Tiempo de asentamiento permitido o máximo sobre paso cuando cuando se adquieren nuevos objetivos o se recupera de perturbaciones.	10 % de sobre paso máximo, decremento a menos de 0.1 grados en 1 minuto; puede también limitar excursiones desde una trayectoria trazada entre objetivos.

Tabla 2.6: Criterios de rendimiento para los modos de control del ADCS [1].

2.4.2. Cuantificación de perturbaciones del medio ambiente espacial

Las condiciones ambientales en las que operar el satélite definen el método de control que es efectivo en el satélite. La selección del modo de control y hardware para el ADCS se encuentran en función de los torques de perturbación debido al ambiente espacial.

Cuatro fuentes de torques de perturbación son consideradas en el desarrollo del ADCS: efectos debidos a gradientes gravitacionales, torques debido al campo magnético en los satélites, efectos por radiación solar, y torques debido a interacciones aerodinámicas para satélites en órbitas LEO. Las otras perturbaciones al satélite son o muy pequeñas relativos a las cuatro mencionadas, tales como presión de emisiones infrarrojas, o son muy limitados en tiempo [3].

Presión por radiación solar es un torque generado cuando el satélite absorbe la luz solar que impacta en él y se modela de la siguiente forma:

$$T_S = \frac{\Phi}{c} A_S (1 + q) (c p_s - c m) \cos(\varphi) \quad (2.1)$$

Donde T_S es la presión solar por radiación, Φ es la constante solar con un valor promedio de $1.366 W/m^2$, $c = 3 \times 10^8 m/s$ es la velocidad de la luz, A_S es el área en que impacta la luz solar, q es un factor de reflexión con un rango entre 0 y 1, φ es el ángulo de incidencia del sol, $c p_s$ y $c m$ son los centros de presión solar y de masa respectivamente.

Arrastre atmosférico es un torque generado por las partículas de la atmosfera en órbitas LEO, se modela de la siguiente forma:

$$T_a = \frac{1}{2} \rho C_d A_r V^2 (c p_a - c m) \quad (2.2)$$

Donde T_a es el torque por arrastre atmosférico, ρ es la densidad atmosférica en kg/m^3 , C_d es el coeficiente de arrastre cuyo valor va entre 2.0 y 2.5 dependiendo el satélite, A_r es el área de arrastre en m^2 , V es la velocidad de la órbita del satélite en m/s , $c p_a$ y $c m$ son los centros de presión atmosférica y de masa respectivamente, medidos en m .

Campo magnético es generado por el núcleo líquido de la Tierra, tiene un importante efecto en la dinámica del satélite. El máximo torque magnético se modela de la siguiente manera:

$$T_m = DB = D \left(\frac{M}{R^3} \lambda \right) \quad (2.3)$$

Donde T_m es el torque magnético, D es el momento residual de dipolo en $A.m^2$, y B es la intensidad de campo magnético en Teslas ¹.

Gradiente gravitacional es causado cuando el centro de gravedad del satélite no está alineado con su centro de masa. El centro de gravedad del satélite en órbita depende de su posición relativa a la tierra. Una expresión simplificada para el torque de gradiente gravitacional en dirección del eje Z es la siguiente:

$$T_g = \frac{3\mu}{2R} |I_z - I_y| \sin(2\theta) \quad (2.4)$$

Donde T_g es el torque por gradiente gravitacional, μ es la constante gravitacional de la Tierra igual a 3.986×10^{14} en m^3/s^2 , R es la distancia desde el centro de la Tierra en m , θ es el ángulo entre la vertical local y el eje Z principal, I_z e I_y son los momentos de inercia en Y y Z en $kg.m^2$.

¹† Salve Nikola Tesla ¶

En la Figura 2.6 se muestran los efectos del ambiente espacial para diferentes tipos de órbitas y el impacto que tienen en los requerimientos de diseño del ADCS.

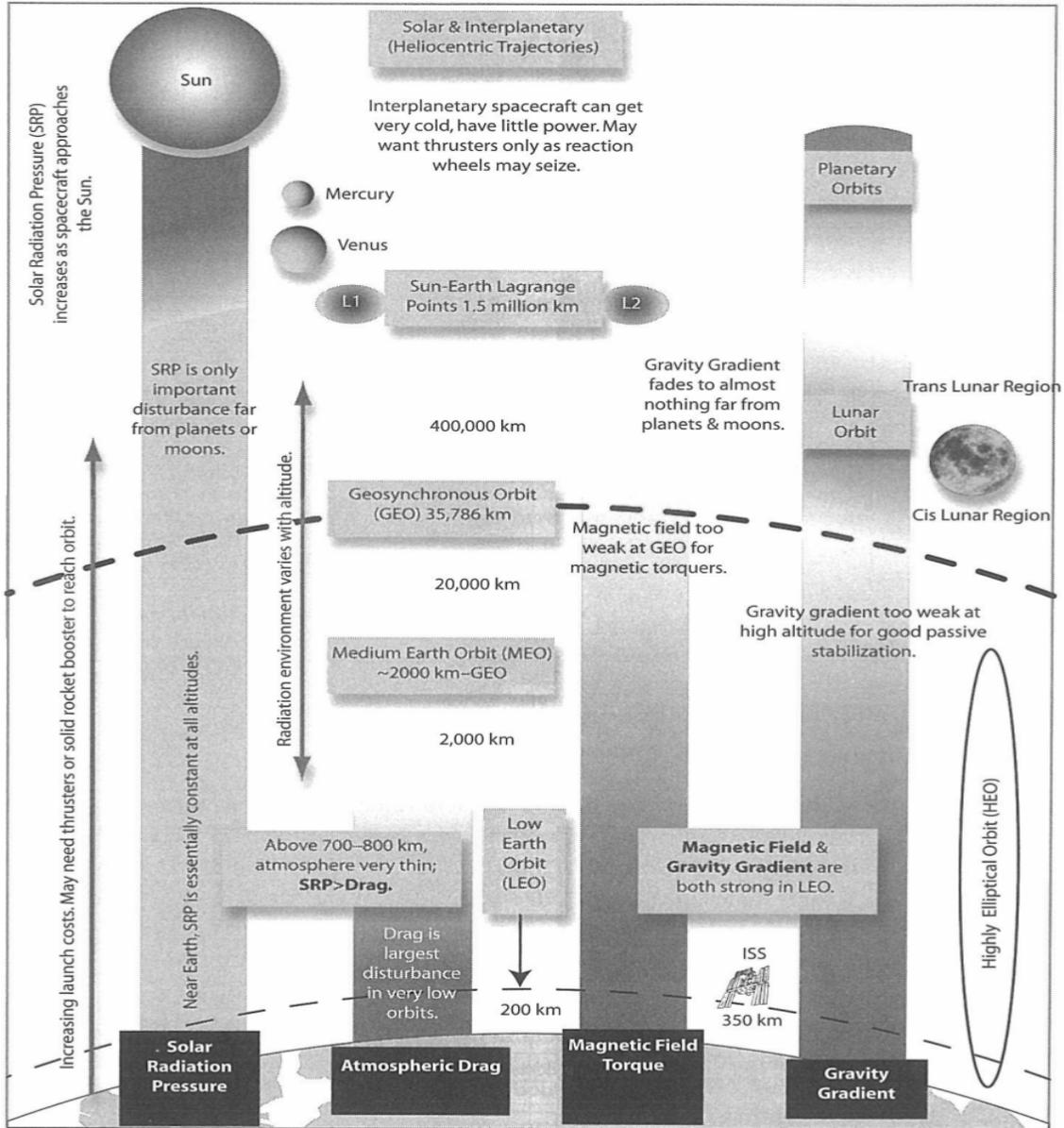


Figura 2.6: Efectos de las perturbaciones del ambiente espacial en ADCS [1].

2.4.3. Selección de métodos de control del satélite

Una vez definidos los requerimientos debidos al ambiente espacial en el que operará el satélite se pueden seleccionar los métodos de control adecuados dependiendo los objetivos de la misión. De acuerdo a Larson et al. [32] y Wertz et al. [1], los métodos de control pasivos y activos para controlar la posición del satélite son los siguientes: gradiente gravitacional, por control de momento angular, gradiente gravitacional más momento angular, magnético pasivo, por estabilización de giro, estabilización de giro doble, magnético activo con filtrado, de momento angular cero con uso de propulsores, de momento angular cero con uso de ruedas de inercia. Cada uno de estos métodos tiene grados de apuntamiento, maniobrabilidad, precisión y restricciones que deben ser consideradas para cada uno de los diferentes modos de control necesarios para el ADCS durante toda la misión. En la Tabla 2.7 de la página 40 se muestra un resumen de las capacidades de precisión, limitantes y opciones de apuntamiento de cada método de control.

Cabe mencionar que la precisión juega un papel fundamental al momento de seleccionar el método de control para el ADCS, la Tabla 2.8 de la página 41 muestra los grados de precisión en el método de control y su efecto en el satélite y ADCS.

Tipo de Control	Opciones de Apuntamiento	Maniobrabilidad de Posición	Presión típica	Límites en tiempo de vida
Gradiente-Gravitacional	Sólamete en la vertical local de la Tierra.	Muy limitado.	± 5 grados (2 ejes).	Ninguno.
Gradiente-Gravitacional + Rueda de reacción	Sólamete en la vertical local de la Tierra.	Muy limitado.	± 5 grados (3 ejes).	Tiempo de vida de los rodamientos.
Magnético pasivo	Norte/Sur Sólolamete.	Muy limitado.	± 5 grados (2 ejes).	Ninguno.
Amortiguamiento de velocidad + Adquisición de vector de objetivo	Usualmente potencia solar o Comunicación con la Tierra	Generalmente usado como modo robusto de emergencia.	$\pm 5 - 15$ grados (2 ejes).	Ninguno.
Estabilización de giro simple	Inercialmente fijo en cualquier dirección.	Re-apuntamiento con maniobras de presión; muy lento con torques, más rápido con propulsores.	± 0.1 grados a ± 1 grado en 2 ejes (proporcional a velocidad de giro).	Propelente de propulsor (En caso de usar propulsores).
Estabilización de giro doble	Limitado únicamente por articulación en la plataforma de desprendimiento	Re-apuntamiento con maniobras de presión; muy lento con torques, más rápido con propulsores.	± 0.1 grados a ± 1 grado en 2 ejes (proporcional a velocidad de giro).	Propelente de propulsor (En caso de usar propulsores).
1 Rueda de reacción	Apuntamiento vertical local u objetivos inerciales.	Maniobras rápidas posibles alrededor del vector aceleración. Re-apuntamiento del vector de aceleración tal como con estabilización de giro.	± 0.1 grados a ± 1 grado.	Propelente (de ser usado). Vida de sensor y rodamientos de ruedas.
Magnético Activo con Filtrado	Cualquiera, pero puede tener desviaciones en periodos cortos.	Lento; más rápido a bajas altitudes.	± 1 grado a ± 5 grados (dependiendo sensores).	Tiempo de vida de los sensores.
Momento Cero (Únicamente propulsor)	No tiene limitaciones.	Sin restricciones. Altas frecuencias posibles.	± 0.1 grados a ± 5 grados.	Propelente.
Momento Cero (3 Ruedas)	No tiene limitaciones.	No tiene limitaciones.	± 0.0001 grados a ± 1 grado (Determinado por sensores y procesador).	Propelente (de ser usado). Vida de sensores y rodamiento de ruedas.
Momento Cero (CMG)	No tiene limitaciones. Corto periodo de vida de CMG puede requerir gran cantidad de redundancia.	Sin restricciones. Altas frecuencias posibles.	± 0.001 grados a ± 1 grado	Propelente (de ser usado). Vida de sensores y soporte de CMG.

Tabla 2.7: Precisión de apuntamiento y maniobrabilidad para cada método de control de posición de ADCS [1].

Precisión requerida (3σ)	Efectos en satélite	Efectos en ADCS
>5 grados	Permite mayor ahorro en costo. Permite estabilización por gradiente gravitacional (GG).	Sin determinación de posición. No requiere sensores para estabilización por GG. Los únicos actuadores necesarios son Motor del Boom, interruptor del GG, y ruedas de reacción. Con determinación de posición. Sensores de Sol y Sensores Magnéticos son adecuados para determinación de posición @ ≥ 2 grados. Precisiones mayores pueden requerir seguidores de estrella o sensores de horizonte.
1 grado a 5 grados	GG no es factible. Estabilización de giro factible si es rígido, posición inercialmente fija es aceptable. Las necesidades de la Carga Útil pueden requerir plataforma de desprendimiento. Estabilización en 3 ejes funcionará.	Sensor de Sol y sensor de horizonte pueden ser adecuados para sensado. Presión para estabilización en 3 ejes puede ser alcanzada con control de banda muerta RCS pero ruedas de reacción ahorrarán propelente para misiones largas. Propulsores son adecuados como actuadores. Torques magnéticos (y sensores magnéticos) son útiles.
0.1 grados a 1 grado	Estabilización de 3 ejes por ruedas de inercia factible. Estabilización de giro doble también factible.	Necesidad de presión en referencia de posición deriva en requerimiento de seguidor de estrella o sensor de horizonte y posiblemente giroscopios. Ruedas de reacción con propulsores para descarga de momento y control grueso. Torques magnéticos factibles en satélites ligeros (sensor magnético requerido).
<0.1 grados	Estabilización en 3 ejes es necesario. Puede requerir plataforma articulada y aislada de vibraciones para la Carga Útil con sensores separados.	Mismo que anterior para 0.1 grados a 1 grado pero necesita sensor de estrella y giroscopios de mejor clase. Leyes de control y necesidades de cómputo son más complejas. Rendimiento de flexibilidad de estructura muy importante.

Tabla 2.8: Efectos de la precisión en el apuntamiento del satélite y ADCS [1].

2.4.4. Selección de tamaño y hardware para ADCS

Hardware de ADCS se refiere a actuadores y sensores que cumplen con los mínimos requerimientos de desempeño para que todos los modos de operación del ADCS funcionen, permitiendo que el satélite realice su tarea adecuadamente. En esta etapa se prefiere la selección de componentes estándar de ser posible. Si los componentes estándar no se encuentran disponibles, componentes especializados deben ser diseñados y construidos.

2.4.4.1. Sensores

Los sensores juegan un papel fundamental en el desarrollo del ADCS, ya que brindan la información a procesar por el sistema de control. Con ellos se determina la posición, velocidad y dirección entre otros factores fundamentales en los sistemas de control.

La arquitectura de sensores considerada en la misión CASTOR en [10] consiste en un sistema GPS con calificación espacial para determinar la posición, 4 sensores de sol con calificación espacial, unidad de medición de inercia de tres ejes (IMU), y un sensor magnético. A continuación se describen brevemente las funciones de los diferentes sensores, así como información específica de dichos sensores, seleccionados para la misión CASTOR; y los cuales serán fundamentales para la propuesta de diseño de Quetzal.

2.4.4.1.1. Sensor de sol El Sensor de sol de dos ejes que mide el brillo del sol permite el uso de equipo confiable, simple, barato, y con consumo mínimo de potencia. Los sensores analógicos de sol están basados en la salida sinusoidal de corriente de una celda solar de silicio como función del ángulo de incidencia solar. La corriente de salida del sensor es proporcional al coseno del ángulo de incidencia solar. En la Tabla 2.9 se muestran las características del sensor de Sol de 2 ejes SS-411.

Sensor de Sol de 2 ejes SS-411		
Rendimiento	Precisión	$\pm 0.1^\circ$ en un rango $\pm 70^\circ$.
	Ancho de banda	5 soluciones de vector por segundo.
Características físicas	Masa	34 g
	Dimensiones	$34 \times 32 \times 21mm$
	Tensión de alimentación	5-50 V_{DC}
	Corriente de alimentación	Consumo promedio $7.5mA$ Consumo pico $27mA$
	Rango de temperatura	$-25^\circ C$ a $50^\circ C$
	Tolerancia a Radiación	$20kRad$
	Vibración	$16gms$ vibración aleatoria $5000g$ en impacto
Comandos/Telemetría	Interfases	UART, SPI, I ² C, CAN, RS285

Tabla 2.9: Características del Sensor de Sol SS-411 [11].

2.4.4.1.2. Sensor de campo magnético Son sensores vectoriales los cuales miden la fuerza y dirección del campo magnético terrestre para determinar la orientación de la nave espacial con respecto al campo magnético local. Son ampliamente usados ya que son ligeros, de bajo consumo de potencia, baratos y pueden operar sobre un amplio rango de temperaturas. Como desventajas presentan poca precisión para altitudes mayores a 1000 km donde la fuerza del campo terrestre es menor los errores no pueden ser despreciados. Información sobre las pruebas de conexión y pruebas de precisión se indican en [10]. En la Tabla 2.10 de la página 43 se muestran las características del Sensor Magnético MicroMag3 seleccionada para el proyecto CASTOR.

Sensor Magnético MicroMag3		
Rendimiento	Rango de medición de campo	$\pm 1100\mu T$
	Medición de alta resolución de campo	$0.015\mu T$
	Velocidad de muestreo rápido	2000 muestras por segundo
Características físicas	Masa	10g (estimados)
	Dimensiones	$25.4 \times 25.4 \times 19mm$
	Tensión de alimentación	$3 V_{DC}$
	Corriente de alimentación	Menor a $500\mu A$
	Rango de temperatura	$-25^{\circ}C$ a $50^{\circ}C$
	Tolerancia a Radiación	No especificada
	Vibración	No especificada
Comandos/Telemetría	Interfases	Protocolo SPI

Tabla 2.10: Características del Sensor Magnético MicroMag3 [12].

2.4.4.1.3. Sensor de inercia Los sensores de inercia combinan acelerómetros y giroscopios. Un acelerómetro mide fuerza específica y un giroscopio mide velocidad angular, ambos sin una referencia externa. Sensores que miden velocidad, aceleración o velocidad angular de un cuerpo con respecto a características del ambiente no son sensores de inercia. La mayoría de acelerómetros mide fuerza específica a lo largo de ejes únicos. Similarmente, la mayoría de giroscopios mide velocidad angular alrededor de un solo eje. Una unidad de medición inercial (IMU) combina múltiples acelerómetros y giroscopios. En la Tabla 2.11 de la página 44 se muestran las características de la Unidad de Medición Inercial seleccionada para el proyecto CASTOR modelo ADIS16365 de *AnalogDevices*.

Unidad de Medición Inercial (IMU) de tres ejes ADIS16365		
Rendimiento	Giroscopio digital de 3 ejes	$\pm 75^\circ/s, \pm 150^\circ/s, \pm 300^\circ/s$
	Resolución	14 bits
	Ancho de Banda	330 Hz
	Acelerómetro	$\pm 18g$
Características físicas	Masa	10g(estimados)
	Dimensiones	$25.4 \times 25.4 \times 19mm$
	Tensión de alimentación	$5 V_{DC}$
	Corriente de alimentación	Modo normal $33mA$ Modo rápido $33mA$ Modo reposo $500\mu A$
	Rango de temperatura	$-25^\circ C$ a $50^\circ C$
	Tolerancia a Radiación	No especificada
	Vibración	No especificada
Comandos/Telemetría	Interfases	Protocolo SPI

Tabla 2.11: Características de unidad IMU ADIS16365 [13].

Es importante considerar el tipo de convertidor analógico digital que se usa, así como el tipo de muestreo que se aplica a las señales provenientes de los sensores, ya que esto influye en el óptimo funcionamiento de la etapa de sensado. Haciendo una correcta selección y mediante la utilización de un método adecuado de muestreo se puede reducir el ruido introducido al sistema en esta etapa del procesamiento de la señal, sin la necesidad de introducir hardware de filtrado adicional reduciendo así el consumo energético y cantidad de masa del ADCS.

2.4.4.2. Actuadores

Un actuador es un mecanismo que provee algún tipo de torque controlado para el sistema de posición y estabilización. En controles activos, se ejecutan algoritmos para tomar decisiones continuas por lo que hardware especial es requerido. Las fuentes más comunes de torque son jets de gas, bobinas magnéticas y ruedas de reacción o inercia. Sistemas pasivos de control hacen uso de torques provenientes del ambiente para mantener la orientación de la nave. Boom gravitacionales, imanes permanentes y velas solares son sistemas de control pasivos comunes. La determinación del uso de sistemas activos o pasivos de control depende de las necesidades de apuntamiento y estabilidad de la misión, carga útil, requerimientos de potencia, restricciones de masa, características orbitales y la rapidez de respuesta requerida para el sistema de control. En el satélite CASTOR las magnitudes de los torques de control son dimensionados de tal forma que el satélite sea capaz de completar los cambios de estado

requeridos en cada orbita. Además, los paneles solares deben volver a adquirir luz solar rápidamente (tiempo de apuntamiento de paneles solares debe ser mucho menor al periodo en que el satélite tiene luz solar) después de entrar en zona con iluminación solar después de zona de eclipse. De igual modo en [10] se especifica que los requerimientos de torque están basados en los peores casos de torques de perturbación y la capacidad de reorientación en 10 minutos. Para este punto de diseño conceptual de la misión del satélite Quetzal se han considerado ruedas de inercia y bobinas magnéticas como actuadores. Al igual que con los sensores los actuadores no han sido definidos para la misión, sin embargo se diseñará en base a los requerimientos de CASTOR.

2.4.4.2.1. Ruedas de reacción Son dispositivos usados para la conservación de momento angular. El cual depende del momento de una rueda giratoria. Consiste en cualquier rueda giratoria o disco usado para transferir o almacenar momento. Es capaz de almacenar momento variable alrededor de su eje de rotación, el cual es normalmente fijo a la nave. En la Tabla 2.12 se muestran las características de la Rueda de reacción *Interplanetary RW-0.0.60-28*.

Rueda de reacción Interplanetary RW-0.0.60-28		
Rendimiento	Momento Nominal	60mN.m.s @ 6500 RPM
	Torque Nominal	>5mN.m @ 28 V _{DC}
	Modo de Control	Velocidad o Torque control en CPU
Características físicas	Masa	0.225 Kg
	Dimensiones	75 × 65 × 38mm
	Tensión de alimentación	7.5–35 V _{DC} (Max 50 V _{DC})
	Potencia de alimentación	7 W @ Torque completo, máxima velocidad 0.5 W @ 5000 RPM en estado estable 0.2 W @ 2000 RPM en estado estable
	Rango de temperatura	-40°C a 70°C
	Tolerancia a Radiación	>15 Krad en dosis total
	Vibración	>12 grms
Comandos/Telemetría	Interfases	UART, SPI, I ² C, CAN, RS485, SMBus

Tabla 2.12: Características de Rueda de reacción Interplanetary RW-0.0.60-28 [14].

2.4.4.2.2. Torques magnéticos Son bobinas que al ser excitadas con una tensión eléctrica generan una corriente, que a su vez crea un dipolo magnético. La fuerza y dirección del dipolo depende de la cantidad y dirección de la corriente que fluye por la bobina, el número de vueltas de alambre y el área total dentro de la bobina. De igual forma es función del material que se encuentra dentro de la bobina.

Los Torques magnéticos son usados para generar momentos de dipolos magnéticos para control de posición y momento angular. De igual forma sirven para compensar perturbaciones debidas al ambiente espacial. Tienen un bajo consumo de potencia. Son aplicables en órbitas LEO principalmente.

Las bobinas para la misión CASTOR fueron hechos por los diseñadores de CASTOR ya que su fabricación es relativamente fácil. Su tamaño les permite des saturar las ruedas de reacción en aproximadamente una órbita, esto debido a que las ruedas de reacción van acumulando energía en forma de cantidad de momento angular y dicha energía debe ser disipada. Las bobinas deben tener un dipolo magnético de aproximadamente $3 A.m^2$ Cálculos del diseño y características de las bobinas son detallados en [10].

Una bobina con las características mostradas en [10] tiene una impedancia aproximada de 366 ohms. La potencia máxima requerida por cada bobina es de 2 W . En el peor de los casos el consumo de potencia sería de 6 W , siendo este caso muy improbable el consumo promedio será de 1.5 W . La interfaz de las bobinas es mediante un controlador doble de puente H modelo NJM2670. Dicho controlador tiene un rango de tensión de 0 V_{DC} a 60 V_{DC} , con lo que se logra un rango de control de corriente de 5 mA a 1500 mA .

2.4.5. Definición del Algoritmo de Determinación de Posición y Control

La definición del algoritmo de determinación de posición y control es el último paso en el desarrollo del ADCS. Generalmente la definición del algoritmo se inicia con un sistema de control simple que se va refinando al agregar o modificar lazos de realimentación para generación de datos de posición y aceleración, modificando ganancias y restricciones, hasta llegar a un sistema de control en tres ejes. Durante todo el proceso de diseño del algoritmo del ADCS se requieren simulaciones matemáticas para confirmar los modelos de diseño, incluyendo modelos de error en los sensores, así como perturbaciones internas y externas.

Teoría de control lineal es aplicada en análisis y diseño preliminar, además de que el sistema puede ser analizado de una manera adecuada usando ecuaciones diferenciales lineales con coeficientes constantes, sin embargo aunque el sistema puede describirse mediante modelos lineales los efectos no lineales son inherentes y deben ser introducidos para mejorar el rendimiento del sistema. El diseño del algoritmo debe reconocer los efectos de interacción que tienen la ganancia de control, la capacidad del sistema de control para compensar perturbaciones, precisión del sistema de control y ancho de banda del mismo.

2.5. Algoritmo de Determinación de Posición y Control

El alcance y objetivos de este trabajo se limitan a la implementación de una propuesta de algoritmo de Determinación de Posición y Control con pares magnéticos como actuadores para una plataforma satelital de clase micro, que servirá como base para un futuro desarrollo del ADCS del satélite Quetzal.

2.5.1. Antecedentes Matemáticos

Para entender el funcionamiento del algoritmo ADCS es necesario tener noción de algunos conceptos matemáticos que a continuación se enumeran de una manera breve, si se requiere profundizar en el estudio de estos conceptos se pueden revisar las derivaciones matemáticas de las ecuaciones en las siguientes fuentes [33–37].

Marco de referencia fijo al satélite (BFF, por sus siglas en inglés)

El origen de este marco de referencia es el centro de masa del satélite. También conocido como marco satelital. Los ejes coordenados X_B (roll), Y_B (pitch) y Z_B (yaw) definen la orientación de la posición. Consideremos a \vec{r} un vector que va desde el centro de la tierra hasta el centro de masa del satélite, y a \vec{v} como el vector velocidad del satélite. Por convención X_B se selecciona en la misma dirección del vector de velocidad \vec{v} , perpendicular a la órbita, Y_B es en dirección de $\vec{r} \times \vec{v}$, lo que significa que es perpendicular al plano orbital, finalmente Z_B sigue la regla de la mano derecha [37].

Marco de referencia inercial centrado en la tierra (ECIF, por sus siglas en inglés)

Este marco es importante debido a que las leyes de movimiento y gravedad aplicadas al satélite están definidas en este marco de referencia y porque muchos satélites son apuntados inercialmente con este marco. Este marco de referencia es definido en relación al eje de rotación y plano orbital de la Tierra alrededor del Sol. También conocido como marco inercial.

Marco de referencia centrado y fijo a la Tierra (ECEF, por sus siglas en inglés)

Como su nombre lo indica este marco se encuentra centrado en la Tierra. En este marco el eje X_{ecef} es en dirección desde el centro de la Tierra apuntando al primer meridiano y ecuador. El eje Z_{ecef} está en dirección desde el centro de la Tierra hacia el polo norte. Y_{ecef} sigue la regla de la mano derecha para determinar su dirección.

Cuaterniones Unitarios

Un cuaternión puede ser representado como un vector complejo que contiene una parte real y tres imaginarias.

$$q = (q_0 \quad q_1 \quad q_2 \quad q_3)^T = \begin{pmatrix} q_0 \\ \vdots \\ q_3 \end{pmatrix}; \quad q_0 \in \mathfrak{R}; \quad q_1 : q_3 \in \mathfrak{S} \quad (2.5)$$

$$q_0 = \cos(\theta/2) \quad (2.6)$$

$$q_{1:3} = \vec{\gamma} \cos(\theta/2) \quad (2.7)$$

Donde $\vec{\gamma}$ es un vector unitario y θ el ángulo de rotación alrededor del vector unitario $\vec{\gamma}$.

Los Cuaterniones cuentan con las siguientes propiedades:

$$\bar{q} = [q_0 \quad q_{1:3}]^T \quad (2.8)$$

$$\|q\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} \quad (2.9)$$

$$q^{-1} = \frac{\bar{q}}{\|q\|} \quad (2.10)$$

Cuando un cuaternión que tiene norma 1 se denomina cuaternión unitario y puede ser usado para la representación de la posición de un cuerpo rígido.

$$q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1 \quad (2.11)$$

Operador Skew-symmetric

Para un vector \vec{x} que pertenece al espacio vectorial de tres elementos R^3 , se define el operador matricial $\mathbf{S}(\vec{x})$ como:

$$\mathbf{S}(\vec{x}) = \begin{pmatrix} 0 & -x_1 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{pmatrix} \quad (2.12)$$

Usando el operador *Skew-symmetric*, el producto cruz entre dos vectores $\vec{v} \times \vec{y}$ puede ser representado como una multiplicación matricial:

$$\vec{v} \times \vec{y} = \mathbf{S}(\vec{v})\vec{y} \quad (2.13)$$

Matriz de rotación

Se define como matriz de rotación a una matriz que al ser multiplicada con un vector rota el vector manteniendo su magnitud. El grupo ortogonal especial de matrices de rotación de 3×3 se denota como $\mathbf{SO}(3)$.

$$\forall R \in \mathbf{SO}(3) \longrightarrow \det(R) = \pm 1 \quad y \quad R^{-1} = R^T \quad (2.14)$$

Si $v \in \mathbf{R}^3$ expresa un vector en un marco de coordenadas, y $v' \in \mathbf{R}^3$ expresa al mismo vector en otro marco de coordenadas entonces:

$$v' = Rv, \quad v = R^T v' \quad (2.15)$$

Si R_b^a representa la matriz de rotación de las coordenadas hacia las coordenadas b , entonces:

$$v^a = R_b^a v_b \quad (2.16)$$

Transformación de marco orbital a marco satelital

La posición del satélite puede ser determinada al estimar la matriz de rotación entre el marco satelital y el marco orbital. Usando el operador matricial $\mathbf{S}()$ se puede definir la matriz de rotación entre estos marcos de referencia R_{orbit}^{body} de la siguiente forma:

$$R_{orbit}^{body} = I + 2q_0\mathbf{S}(q_{1:3}) + 2\mathbf{S}^2(q_{1:3}) \quad (2.17)$$

Donde:

$$R_{orbit}^{body} = (R_{body}^{orbit})^T \quad (2.18)$$

Matriz de inercia

La matriz de inercia alrededor del origen se define de la siguiente manera:

$$I_O = \begin{pmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{yx} & I_y & -I_{yz} \\ -I_{zx} & -I_{zy} & I_z \end{pmatrix} \quad (2.19)$$

$$I_x = \int (y^2 + z^2)\rho_m dV; \quad I_{xy} = I_{yx} = \int xy\rho_m dV \quad (2.20)$$

$$I_y = \int (x^2 + z^2)\rho_m dV; \quad I_{xz} = I_{zx} = \int xz\rho_m dV \quad (2.21)$$

$$I_z = \int (x^2 + y^2)\rho_m dV; \quad I_{yz} = I_{zy} = \int yz\rho_m dV \quad (2.22)$$

Cuando los ejes inerciales y del marco de referencia coinciden, la matriz de inercia se reduce a lo siguiente:

$$I = \begin{pmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{pmatrix} \quad (2.23)$$

Controlador Lineal Cuadrático

Considerando el sistema lineal:

$$\dot{x} = Ax + Bu \quad (2.24)$$

$$y = Cx \quad (2.25)$$

El controlador LQR consiste en encontrar una ganancia K

$$u = -Kx \quad (2.26)$$

Tal que minimice la función de costo L

$$L = \frac{1}{2} \int_0^{\infty} (x^T Qx + u^T Ru) dt \quad (2.27)$$

Donde Q y R son matrices positivas, $x^T Qx$ representa el costo de rendimiento, $u^T Ru$ representa el costo de consumo de energía [34]. Las matrices Q y R generalmente se seleccionan como matrices diagonales, de acuerdo a Svartveit [38] los valores de las matrices de peso Q y R pueden ser calculadas como:

$$Q = \text{diagonal}(q_i); \quad q_i = \frac{1}{\Delta u_i} \quad (2.28)$$

$$R = \text{diagonal}(r_i); \quad r_i = \frac{1}{\Delta x_i} \quad (2.29)$$

Donde Δu_i representa el máximo momento dipolar del actuador y Δx_i representa la desviación de estado. Sin embargo como se explica en Bevilacqua et al. [39] los valores de estas matrices diagonal pueden ser ajustados para lograr el rendimiento requerido por el controlador.

2.5.2. Algoritmo

Un algoritmo es un método efectivo para calcular una función o lograr un objetivo previamente propuesto. Es el procedimiento finito paso a paso, con instrucciones bien definidas para procesar información y realizar tareas concretas o solucionar problemas. Cualquier tarea compleja de procesamiento de información debe ser separada en operaciones simples para ser ejecutadas por la unidad central de procesamiento o algún procesador específico. Algoritmo es la manera en que las operaciones simples son ejecutadas para realizar una tarea compleja. Para brindar confiabilidad al sistema de control se usan algoritmos establecidos, probados, ya que el uso de nuevos algoritmos involucra riesgos técnicos, agregan costo y una implementación más lenta [32].

Determinar la posición y el control cubre varios métodos de control. En la integración cinemática se determina la posición actual mediante la integración de aceleración proveniente de giroscopios. Usando determinación de errores se estima que tan lejos se encuentra la orientación de la nave con respecto a lo deseado. El principio de operación de control por par magnético se basa en la determinación de los valores de campo magnético en cada punto y generación de dipolos magnéticos adecuados para que su interacción con el campo magnético estabilice la nave. Para naves con estabilización de giro se mantiene la posición mediante control de precesión [9].

Se puede decir que un algoritmo de control de posición recibe la tarea compleja de mantener el grado de apuntamiento a lo largo de diferentes etapas de la misión satelital y divide dicha tarea en tareas menos complejas, tomando como entradas aceleraciones, posiciones, velocidades y teniendo como salida pares de control. Por otro lado la posición del satélite es referenciada a un marco terrestre fijo con el objetivo de obtener datos de ángulos de la velocidad angular, de este modo es necesario un método de conversión para demostrar el vector velocidad correctamente.

2.5.3. Tipos de Algoritmos

El algoritmo de control es por lo regular matemáticamente exigente y requiere alta precisión y estrictas líneas de tiempo [9]. Para la ejecución del algoritmo de control se requiere desarrollo de software o hardware.

Los métodos más comunes para algoritmos de determinación de posición y control son los modelos de ángulos de Euler y cuaterniones. Los modelos basados en ángulos de Euler han sido probados como muy eficientes debido a que los modelos linealizados de Euler son controlables, y todos los métodos de control lineal de sistemas son directamente aplicables. Las desventajas del uso de ángulos de Euler radican en que los diseños basados en modelos linealizados pueden no estabilizar al satélite original no linealizado, esto es, el diseño puede dejar de funcionar si el satélite se encuentra en un punto demasiado lejos de donde se realizó la linealización [34].

Aunque el método de ángulos de Euler es confiable, simple y suficientemente preciso para aplicaciones, en ciertos momentos ocurren singularidades en transformaciones de coordenadas debido a un bloqueo de ejes. Estas singularidades ocurren por los indistinguibles cambios en los primer y tercer ángulos de Euler cuando el segundo se encuentra en algún valor crítico [37]. De este modo y para evitar estas singularidades que pueden llevar a ejecuciones erróneas del sistema de control, la representación o modelo basado en Cuaterniones puede ser usada como método más adecuado de representación de coordenadas, ya que ésta nos brinda confiabilidad respecto a la aparición de singularidades.

El método de Cuaterniones reducidos no depende de secuencias de rotación, debido a la estructura especial del modelo linealizado del satélite, se pueden aplicar modelos de diseño del ADCS, tales como regulador lineal cuadrático (LQR) y diseño por asignación robusta de polos de una manera relativamente sencilla.

2.5.4. Descripción del Algoritmo de Determinación de Posición y Control

El algoritmo de control que se presenta para su implementación corresponde al modo de operación del ADCS denominado on-Station, que es el modo siguiente después del modo de Inserción de órbita por lo que se asume que el satélite ya se encuentra en la órbita para la que fue diseñado. Siendo un satélite con órbita de tipo LEO se puede aprovechar la gran influencia que tiene el campo magnético de la Tierra para controlar la posición del satélite.

De este modo el algoritmo a implementar es de tipo magnético activo con filtrado y estabilización en tres ejes, con 0.1 grados de control de apuntamiento, en el que se combina información de posición del satélite, generalmente provista por un GPS, y el ADCS determina el campo magnético local basado en modelos matemáticos a bordo del satélite previamente

cargados. Esta información es combinada con sensores de campo magnético para lograr una adecuada determinación de posición y control.

El ADCS requiere manejar valores tan grandes como la masa de la tierra y altura de órbita; y valores tan pequeños como los de campo magnético de la tierra. Usando un formato simple de 32 bits en punto flotante se puede manejar un rango muy amplio de valores con una precisión muy grande. Considerando este rango dinámico de valores que presenta el procesamiento de información, la mejor opción para la implementación del procesamiento es en formato de punto flotante con el cual se logra una precisión adecuada.

El algoritmo consiste en los modelos matemáticos de la dinámica y cinemática del satélite, determinación del valor de campo magnético, cálculo de ganancia de control mediante LQR, pares de perturbación y control.

En la figura 2.7 de la página 54 se muestra un diagrama del sistema físico con sus parámetros asociados.

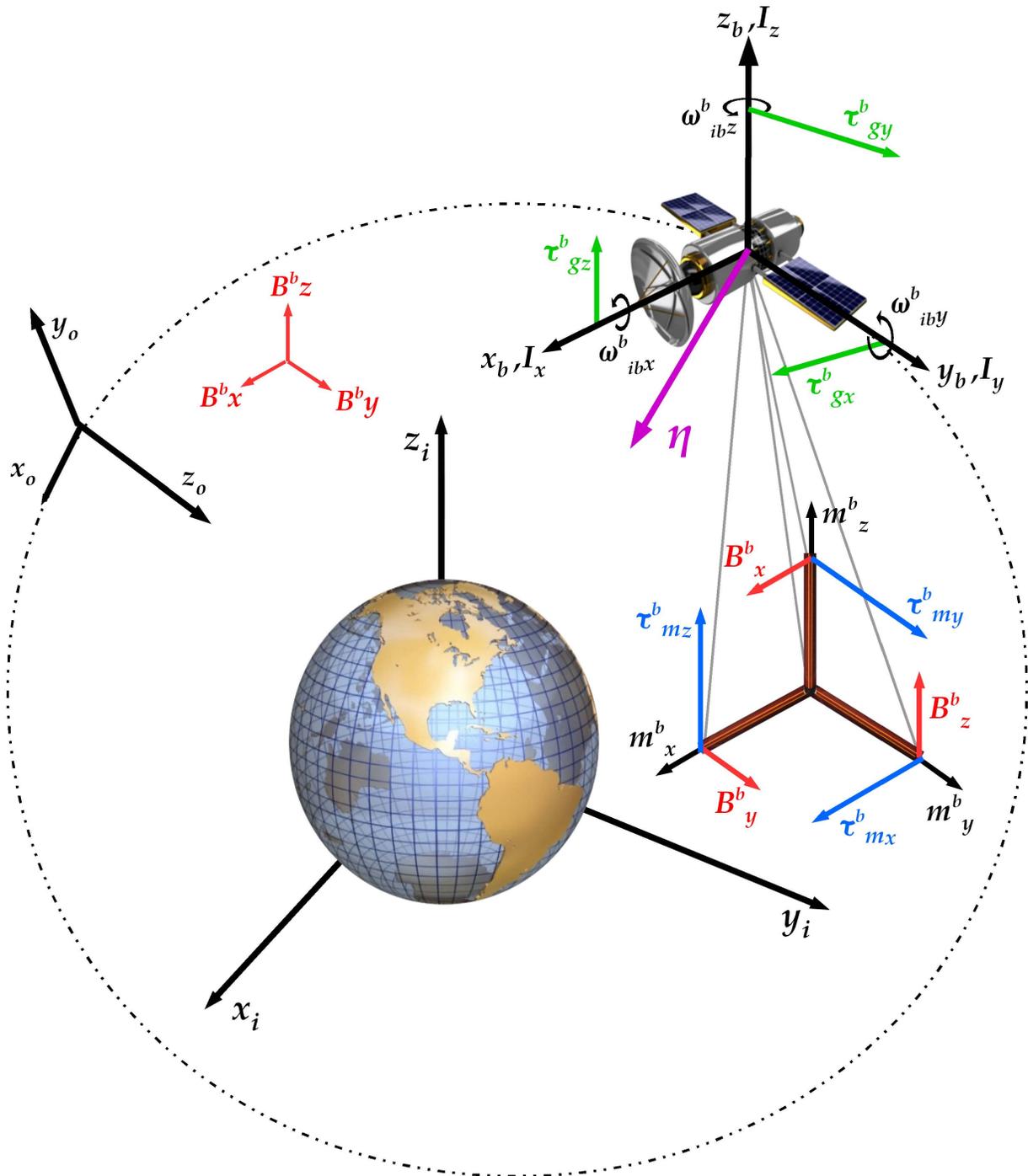


Figura 2.7: Parámetros físicos del sistema: vector unitario nadir $\vec{\eta}$, marco de referencia Inercial (x_i, y_i, z_i) , marco de referencia Satelital (x_b, y_b, z_b) , vector Torque magnético en marco Orbital $(\tau^b_{mx}, \tau^b_{my}, \tau^b_{mz})$, vector Torque gravitacional en marco Satelital $(\tau^b_{gx}, \tau^b_{gy}, \tau^b_{gz})$, vector campo magnético en el marco Satelital (B^b_x, B^b_y, B^b_z) , vector Velocidad angular en el marco Inercial relativo al marco Satelital en el marco Satelital $(\omega^b_{ibx}, \omega^b_{iby}, \omega^b_{ibz})$, marco de referencia Orbital (x_o, y_o, z_o) y Matriz de Inercia (I_x, I_y, I_z) .

Dinámica

La dinámica del satélite se puede describir con la siguiente ecuación [33]:

$$I\dot{\omega}_{bi}^b + \omega_{bi}^b \times (I\omega_{bi}^b) = \tau^b \quad (2.30)$$

Donde $\tau^b = \tau_g^b + \tau_m^b$ es el torque debido al campo magnético y gravitacional. I es el momento de inercia, ω_{bi}^b es el vector de la velocidad angular en el marco del satélite relativo al marco inercial en el marco satelital. Utilizando el operador matricial *Skew-symmetric*, la dinámica se puede expresar de la siguiente forma:

$$I\dot{\omega}_{bi}^b + \mathbf{S}(\omega_{bi}^b)I\omega_{bi}^b = \tau^b \quad (2.31)$$

Cinemática

La cinemática del satélite describe su posición y orientación, se puede expresar de manera compacta usando Cuaterniones de la siguiente forma [33]:

$$\dot{q} = \frac{1}{2} \begin{pmatrix} -q_{1:3}^T \\ q_0 I_{3 \times 3} + \mathbf{S}(q_{1:3}^T) \end{pmatrix} \omega_{bo}^b \quad (2.32)$$

Donde $I_{3 \times 3}$ es la matriz de inercia, $\mathbf{S}()$ es el operador *Skew-symmetric*, $q_{0:3}$ son los valores de cuaterniones instantáneos y ω_{bo}^b es la velocidad angular del marco satelital relativo al marco orbital en el marco de referencia satelital.

Pares de perturbación y control

El torque magnético puede ser expresado de la siguiente forma [33]:

$$\tau^b = \tau_g^b + \tau_m^b \quad (2.33)$$

$$\tau_m^b = \vec{m}^b \times \vec{B}^b \quad (2.34)$$

Donde \vec{m}_b es el dipolo magnético generado por el actuador, \vec{B}^b es el campo magnético en el marco satelital, que se puede expresar de la siguiente forma:

$$\vec{B}^b = (B_x^b, B_y^b, B_z^b)^T \quad (2.35)$$

$$\tau_m^b = S(m^b)\vec{B}^b = \begin{pmatrix} B_z^b m_y - B_y^b m_z \\ B_x^b m_z - B_z^b m_x \\ B_y^b m_x - B_x^b m_y \end{pmatrix} \quad (2.36)$$

El torque gravitacional τ_g^b puede ser expresado de la siguiente forma:

$$\tau_g^b = 3\omega_0^2 \vec{\eta} \times (I\vec{\eta}) \quad (2.37)$$

$$\omega_0^2 = \frac{\mu}{R_0^3} \quad (2.38)$$

Donde $\mu = 3.986 \times 10^{14} m^3/s^3$ es el coeficiente gravitacional de la Tierra, R_0 es la distancia del centro de la tierra al centro del satélite, $\vec{\eta}$ es el vector unitario nadir y I es la matriz de inercia. En el marco satelital la ecuación 2.37 se puede expresar de la siguiente forma:

$$\tau_g^b = 3\omega_0^2 c_3^b \times (Ic_3^b) \quad (2.39)$$

Donde c_3^b es la tercera columna de la matriz de rotación R_{orbit}^{body} de la ecuación 2.17.

Modelo de campo magnético

El modelo empleado es el IGRF, que calcula el campo magnético de la Tierra en cualquier punto de su superficie. Se puede calcular de la siguiente forma:

$$B_r = -\frac{\partial V}{\partial r} = \sum_{n=1}^{\infty} \left(\frac{a}{r}\right)^{n+2} (n+1) \sum_{m=0}^n P_n^m(\theta) [g_n^m \cos(m\varphi) + h_n^m \sin(m\varphi)] \quad (2.40)$$

$$B_\theta = -\frac{1}{r} \frac{\partial V}{\partial \theta} = -\sum_{n=1}^{\infty} \left(\frac{a}{r}\right)^{n+2} (n+1) \sum_{m=0}^n \frac{\partial P_n^m(\theta)}{\partial \theta} [g_n^m \cos(m\varphi) + h_n^m \sin(m\varphi)] \quad (2.41)$$

$$B_\varphi = -\frac{\partial V}{r \partial r \sin \theta} = \frac{1}{\sin \theta} \sum_{n=1}^{\infty} \left(\frac{a}{r}\right)^{n+2} (n+1) \sum_{m=0}^n P_n^m(\theta) [-mg_n^m \cos(m\varphi) + mh_n^m \sin(m\varphi)] \quad (2.42)$$

Donde $a = 6371.2 \times 10^3$ es el radio de la Tierra, r es la distancia desde el centro de la Tierra al satélite, $P_n^m(\theta)$ representa los polinomios Schmidt normalizados asociados de Legendre, θ

es la co-latitud y φ la longitud, g_n^m y h_n^m son constantes gaussianas. El polinomio de Schmidt asociado de Legendre puede ser consultado en el Apéndice A.

Modelo matemática linealizado

Dado que un modelo no lineal es más complejo de operar se buscan puntos en los que se puedan linealizar simplificando así la complejidad de cálculos. La linearización del modelo se puede revisar en [33]. El modelo linealizado es el siguiente:

$$\dot{x}(x) = Ax(t) + B(t)u(t) \quad (2.43)$$

$$x = (q_1, \dot{q}_1, q_2, \dot{q}_2, q_3, \dot{q}_3) \quad (2.44)$$

$$u = (m_x, m_y, m_z)^T \quad (2.45)$$

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -4k_x\omega_0^2 & 0 & 0 & 0 & 0 & (1-k_x)\omega_0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -3k_y\omega_0^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -(1-k_z)\omega_0 & 0 & 0 & -k_z\omega_0^2 & 0 \end{pmatrix} \quad (2.46)$$

$$B = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{2I_x}B_z^o & -\frac{1}{2I_x}B_y^o \\ 0 & 0 & 0 \\ -\frac{1}{2I_y}B_z^o & 0 & \frac{1}{2I_y}B_x^o \\ 0 & 0 & 0 \\ \frac{1}{2I_x}B_y^o & -\frac{1}{2I_x}B_x^o & 0 \end{pmatrix} \quad (2.47)$$

$$k_x = \frac{I_y - I_z}{I_x} \quad (2.48)$$

$$k_y = \frac{I_x - I_z}{I_y} \quad (2.49)$$

$$k_z = \frac{I_y - I_x}{I_z} \quad (2.50)$$

Donde I_x, I_y, I_z representan los momentos de inercia en los ejes x, y, z respectivamente, ω_0 es la velocidad angular inicial, B_x^o, B_y^o, B_z^o representan los componentes del campo magnético en el marco orbital.

Ganancia de control

La ganancia de control se calcula haciendo uso del controlador LQR, este proceso involucra la solución de la **Ecuación Algebraica de Riccati** (ARE, por sus siglas en inglés). Teniendo en cuenta el problema de minimización planteado en el controlador LQR se tiene lo siguiente:

$$L = \frac{1}{2} \int_0^{\infty} (\vec{x}^T Q x + \vec{u}^T R u) dt \quad (2.51)$$

Para dar solución a la función de minimización L se requiere resolver la ARE:

$$\dot{R}(t) = -R(t)A - A^T R(t) + R(t)B(t)P^{-1}B(t)R(t) - Q(t) \quad (2.52)$$

De este modo la ganancia de control se calcula de la siguiente forma:

$$\vec{u}(t) = -P^{-1}B(t)^T R(t)\vec{x}(t) \quad (2.53)$$

Donde $R(t)$ es la solución a la ecuación (2.52) y $x(t)$ es el vector de estados del sistema descrito por (2.44). La ganancia de control se puede escribir de una manera compacta:

$$\vec{u}(t) = -K\vec{x}(t)^T \quad (2.54)$$

Donde K es la matriz de ganancia óptima.

La ARE se encuentra constantemente en muchas aplicaciones en diferentes áreas de la ingeniería y ciencia, incluyendo modelos de colas [40], modelos numérico de soluciones a ecuaciones de transporte [41], juegos diferenciales [42], sistemas de control de viento para edificios altos [43], optimización de trayectorias [44], control óptimo [45, 46].

Resolver la ARE es fundamental para la solución de modelos de filtrado, y diseño de controladores de sistemas lineales dinámicos, sin embargo el cálculo de la solución de la ARE en aplicaciones de control resulta matemáticamente demandante [47]. Existe una amplia bibliografía dedicada a la implementación de algoritmos de solución algebraica y numérica de la ARE, se pueden consultar las siguientes fuentes para profundizar en su estudio [48–53].

Capítulo 3

Implementación del algoritmo de Determinación de Posición y Control

Como parte de la metodología propuesta para el diseño del ADCS de Quetzal, en el presente trabajo se presenta la implementación del algoritmos de ADCS para un satélite pequeño usando bobinas magnéticas como actuadores, usando un dispositivo programable de arreglo de compuertas (FPGA, pos sus siglas en inglés) como base de procesamiento, esto como un primer prototipo de implementación para el proyecto satelital QUETZAL. El dicho trabajo consistirá en el algoritmo de control desarrollado en Kaplan [33]. Dicho algoritmo debe ser ejecutado con eficiencia y la suficiente precisión. Se utilizarán modelos matemáticos para la determinación de los valores de campo magnético previamente evaluados en una computadora; posteriormente se evaluará el modelo de control para ser comparado con una simulación descrita en Matlab.

Las tareas en la implementación consisten en la determinación de los modelos de campo magnético, cálculo de ganancia óptima mediante un LQR para realizar la realimentación de estados y controlar el sistema.

3.1. Definición de arquitectura para implementación del algoritmo de Determinación de Posición y Control

Para implementar un algoritmo se requiere una arquitectura específica que cumpla con los requerimientos de capacidad de procesamiento, precisión y eficiencia, así como confiabilidad requeridos. A continuación se describe el proceso de definición de la arquitectura para la implementación del algoritmo.

3.1.1. Selección de tecnología

Existen cierto equilibrio entre las cualidades de diferentes tecnologías como microprocesadores, FPGAs y circuitos integrados de aplicación específica (ASIC por sus siglas en inglés), incluyendo software vs hardware, generalidad vs rendimiento, costo vs rendimiento, y seguridad vs generalidad. Los microprocesadores ejecutan software, que es relativamente sencillo de programar pero viene con altos costos en cuando a diseño de software. ASIC es hardware para aplicaciones específicas, tienen alto rendimiento y son costosos en su proceso de diseño y fabricación. FPGA son más difíciles de programar que microprocesadores, pero son menos costosos que los ASIC, además, los FPGA pueden alcanzar mayor capacidad de rendimiento que los microprocesadores, pero no tanto como los ASICs [16] por lo tanto son un punto intermedio entre las soluciones de mayor desempeño ASICs, y de mayor facilidad de implementación, microprocesadores.

Para hacer una selección de tecnología se tiene que definir sistema embebido, software y hardware, su relación y el tipo de soluciones en base a cada tecnología, así como las combinaciones de estas. A continuación se describen diferentes tecnologías para la solución de aplicaciones embebidas, sus ventajas y desventajas.

3.1.2. Sistema embebido

Desde sistemas simples de uso cotidiano como videojuegos, teléfonos, hasta sofisticados equipos satelitales y militares, el desarrollo de sistemas embebidos ha llegado a ser parte fundamental de la sociedad [54].

Un sistema embebido es una computadora especializada. Dicha computadora se conforma por una unidad de procesamiento o procesador como núcleo, periféricos de entrada y salida; además de unidades de procesamiento especializado junto con componentes de memoria para almacenamiento de programas y datos que se ejecutan en el procesador o unidad de procesamiento.

Debido a que la mayoría de los sistemas digitales en estos días son sistemas embebidos, la tendencia de la práctica de diseño digital involucra el desarrollo de interfaces electrónicas entre los núcleos de procesamiento y los circuitos de aplicación específica para realizar tareas no realizadas por el procesador núcleo.

Generalmente un sistema embebido es un componente que forma parte de un sistema más grande y complejo. Aunque un sistema embebido puede ser capaz de realizar tareas más complejas que las que normalmente desempeña, está limitado a desempeñar una tarea específica, tales como el procesamiento de una imagen o la ejecución de algoritmos de control. Una de las diferencias fundamentales entre sistemas embebidos y sistemas de propósito general radica en la necesidad de hardware y software de propósito especial para aplicaciones embebidas.

El diseño de sistemas embebidos, diseño de circuitos electrónicos para aplicaciones específicas, está basado en series de decisiones así como del uso de técnicas de desarrollo basadas en la selección de la representación de datos, generación o selección de algoritmos, selección de plataformas de hardware, partición entre software y hardware, generación de programas, síntesis de nuevo hardware, consolidación, co-emulación y generación de prototipo [4]. Estas actividades están directamente relacionadas con el estudio de algoritmos aritméticos y circuitos, en especial en aplicaciones que requieren una gran cantidad de procesamiento.

La representación numérica en el desarrollo de una aplicación específica puede ser elegida para que se sea la más conveniente sin limitarse a tipos de dato flotante, fijo o de doble precisión, sistemas no estándar pueden ser utilizados.

Hardware se define como la implementación física de una máquina de cómputo. Usualmente es una colección de circuitos electrónicos y quizá algunos componentes mecánicos. Además, los componentes que conforman el hardware son activos concurrentemente. Cuando las entradas varían, cambios en los circuitos se propagan a través de la máquina de una manera predecible aunque no necesariamente sincronizada.

Software es una especificación que describe el comportamiento de la máquina. Es escrito en un lenguaje de programación y esa representación del comportamiento deseado de la máquina es llamada programa. De igual forma es información y por lo tanto no se manifiesta por sí misma en el mundo físico [54].

Uno de los aspectos que presenta mayor reto en el diseño de sistemas embebidos es el hecho de que dicho diseño requiere tanto de componentes en software como de hardware.

En el desarrollo de software, el modelo de ejecución de las operaciones se realiza de forma secuencial el beneficio de este modelo de ejecución secuencial radica en que el programador queda liberando de la compleja labor de sincronización de la ejecución del software. Por el contrario, el desarrollo de hardware es por defecto paralelo y los ingenieros de diseño deben incluir explícitamente la sincronización en su diseño, siendo el diseño de hardware más complejo que el diseño de software.

El uso de una solución basada en un modelo de flujo de datos (solución en hardware) cuenta con muchas ventajas como son la capacidad de generar procesos que se ejecutan en paralelo, reconfiguración parcial o total, gran capacidad de procesamiento, etc. Conlleva a su vez mayor complejidad en el proceso de desarrollo. Dependiendo en la frecuencia con que la estructura del sistema puede cambiar, los tres tipos de implementación son sistemas dedicados, sistemas reconfigurables, y sistemas programables. A continuación se mencionan de manera breve diferentes tipos de tecnología con las que se pueden solucionar sistemas embebidos.

3.1.2.1. Microprocesador

Existen dos clasificaciones principales para microprocesadores en base al tipo de arquitectura, las de set de instrucciones complejo (CISC) y las de set reducido de instrucciones (RISC), la mayoría de los procesadores actuales son una combinación de ambos modelos. Otro modelo de clasificación es de acuerdo al tipo de acceso a memoria, siendo los modelos Von Neumann y Harvard los fundamentales. En el modelo Von Neumann la memoria es compartida entre las instrucciones, o programa y los datos; utiliza un bus para datos y otro para direccionamiento como comunicación entre la memoria y el procesador. La información debe ser extraída de manera secuencial limitando el ancho de banda de operación. Por otro lado, el modelo de arquitectura Harvard usa diferentes memorias para sus instrucciones y para sus datos, requiriendo un bus dedicado para cada una, esto permite que la información e instrucciones sean extraídas simultáneamente. Diferentes anchos de bus son posibles permitiendo a la memoria de programa y datos estar mejor optimizados con respecto a los requerimientos de la arquitectura [4]. Además existe la arquitectura Harvard modificada, en la que el programa y los datos son ordenados (*cached*) separadamente pero son almacenados en una memoria conectada por un solo bus.

El diseño Von Neumann es más simple que el Harvard, teniendo un uso más eficiente de la memoria por lo que es más usado en procesadores de propósito general. Por el otro lado, la arquitectura Harvard es más usada en procesadores de señales digitales (DSP) y microprocesadores para aplicaciones específicas en la transmisión de datos, ya que permite un más grande y predecible ancho de banda en memoria. En la Figura 3.1 se muestran diagramas de ambas arquitecturas.

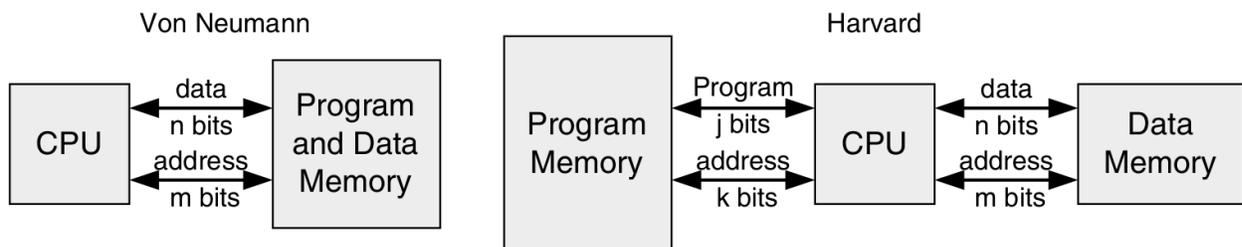


Figura 3.1: Arquitecturas Von Neumann y Harvard [4].

3.1.2.2. Microcontrolador

Son computadoras en un solo chip ya que cuentan con un núcleo de procesamiento, memoria y puertos de entrada y salida, son relativamente lentos y con poca memoria, pero son baratos y pueden interactuar fácilmente con dispositivos en el mundo real mediante interfaces. Generalmente emplean arquitectura RISC de 8, 16 o 32 bits. Actualmente los microcontroladores cuentan con memorias mayores, convertidores digitales-analógicos, analógicos-digitales, puertos de comunicación como CAN, USB o Ethernet, así como comunicación vía infra rojo o por radio.

3.1.2.3. Procesador de señales digitales (DSP)

El análisis de señales digitales requiere la representación de señales de manera digital; procesadores digitales son usados para analizar, modificar, o extraer información de señales analógicas o digitales: el procesador de señales digitales (DSP, por sus siglas en inglés) es un dispositivo electrónico específicamente diseñado para procesamiento de señales digitales. Las señales digitales son obtenidas de señales físicas mediante sensores, transductores y convertidores analógico-digitales (ADC, por sus siglas en inglés). Una vez que estas señales son procesadas pueden ser convertidas nuevamente a señales analógicas mediante convertidores digital-analógicos [4].

Los DSP tienen un amplio espectro de aplicaciones en diversas áreas tales como: procesamiento de imágenes, incluyendo reconocimiento de patrones, visión robótica, mejoramiento de imagen, reconocimiento facial etc. Instrumentación y control, en donde se usan para realizar análisis espectral, control de posición, reducción de ruido, compresión de datos, sistemas guía, y procesamiento de GPS. En procesamiento de audio y video se hace reconocimiento de voz, síntesis de voz, conversión de texto a audio, audio digital, ecualización, etc. Aplicaciones militares, como aseguramiento de comunicaciones, procesamiento de radares y sonares, y sistemas para guiar misiles. En telecomunicaciones se usan para hacer cancelación de eco, ecualización adaptativa, telefonía celular, sistemas inalámbricos y módems.

Un DSP generalmente involucra multiplicadores y sumadores por hardware, que pueden ser usados en paralelo con una sola instrucción mediante el uso de unidades de multiplicación y acumulación (MAC). La diferencia principal entre un procesador de señales digitales y uno de propósito general se relaciona con el espectro de problemas que cada uno puede solucionar. Cadenas de datos infinitas, para ser procesadas en tiempo real con procesos aritméticamente intensivos con pocos saltos de control, son consideradas aplicaciones para DSPs.

3.1.2.4. Circuito Integrado de Aplicación Específica (ASIC)

Circuitos integrados de aplicación específica (ASIC) son circuitos integrados especialmente diseñados para tareas específicas. Dicha tecnología tiene ventajas tales como velocidad incrementada, bajo consumo de potencia, bajo costo en producción en masa, dificultad para generar ingeniería inversa por lo que el diseño es seguro, mejor control de las características de entrada y salida, y un diseño más compacto. Sin embargo, también tiene muchas desventajas en comparación con otras tecnologías; alto costo para una producción reducida, muy alto costo de inversión en cuanto a herramientas CAD, estaciones de trabajo y mano de obra, una vez que el diseño se encuentra en silicón no puede ser modificado. La tecnología ASIC puede clasificarse en ASIC completamente especializado, parcialmente especializado, y circuitos integrados programables.

En ASIC completamente especializado todas las capas de máscara son especializadas. Esta tecnología ofrece el más alto rendimiento y el menor tamaño, con la desventaja de incremento de tiempo de diseño, mayor complejidad y costo, además conlleva el mayor riesgo de falla. Este diseño tiene sentido únicamente cuando no hay librerías o IP cores disponibles, o cuando se requiere muy alto nivel de rendimiento en la aplicación. Microprocesadores y memorias fueron en un inicio completamente especializados, ahora la industria los está convirtiendo en parcialmente especializados mediante técnicas ASIC también en estas áreas.

En ASIC parcialmente especializados son realizados a nivel de compuertas. Así pierden algo de la flexibilidad de un diseño completamente especializado, esto con la finalidad de reducir costos y tiempo de diseño. Soluciones parcialmente especializadas se categorizan en arreglo de compuertas y celdas estándar.

El flujo de diseño de ASIC se conforma por los siguientes pasos:

Entrada de diseño mediante un lenguaje de descripción de hardware tales como VHDL, Verilog o mediante un diagrama esquemático. Síntesis de lógica, en base al lenguaje de descripción de hardware o esquemático, se extrae la topología de conexión electrónica, esto es, una descripción de las celdas lógicas y sus interconexiones. Partición de sistema divide un sistema muy largo en piezas de tamaño ASIC. Simulación previa a configuración en la que se prueba el funcionamiento del circuito. Una etapa de acomodamiento de los diferentes bloques del circuito en el chip. Una etapa en la que las celdas se colocan en cada bloque. Generación de las conexiones entre las celdas y los bloques. Extracción en la que se determinan los valores de resistencia y capacitancia entre las interconexiones y calculan retardos para propósitos de simulación. Simulación posterior al ordenamiento físico donde se revisa que el circuito funcione después de incluir los retardos creados por las interconexiones hechas. Finalmente se verifica que la disposición del circuito cumpla con las especificaciones de las reglas de diseño. En la Figura 3.2 de la página 65 se muestra el diagrama de flujo de diseño de un ASIC.

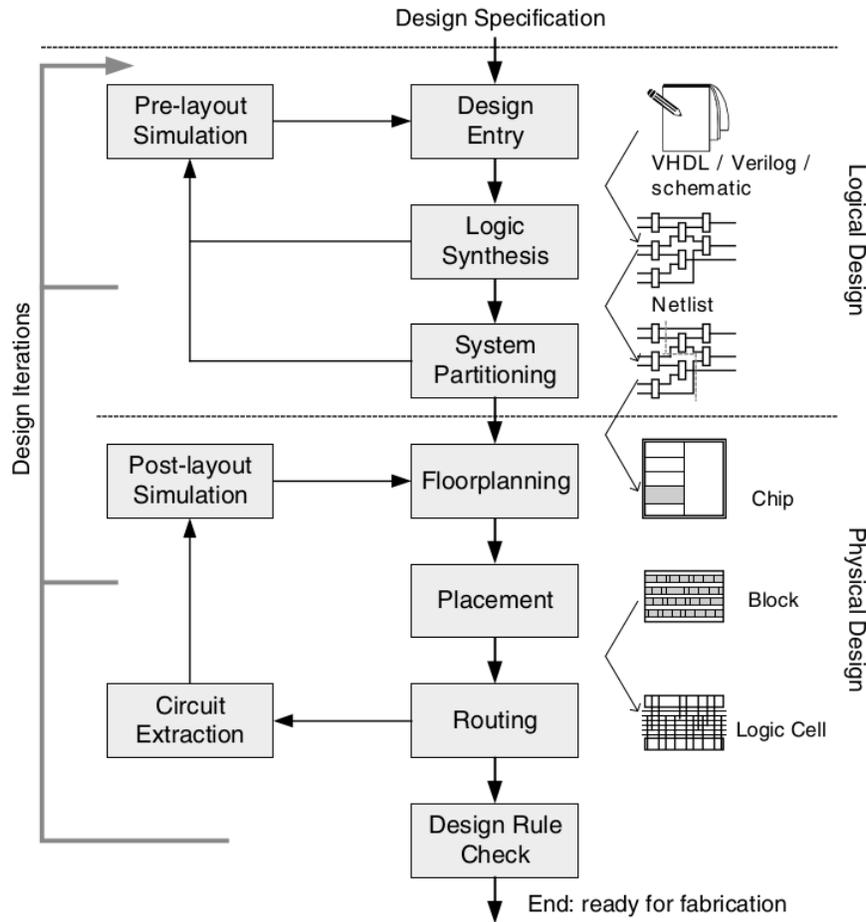


Figura 3.2: Diagrama de flujo de diseño de un ASIC [4].

3.1.2.5. Dispositivos Lógicos Programables

Los dispositivos lógicos pueden ser clasificados en dos categorías: fijos y programables. Circuitos lógicos fijos son permanentes, ejecutan la misma función o serie de funciones y una vez manufacturados no pueden ser modificados, como los ASICs. Por otro lado dispositivos lógicos programables (PLDs por sus siglas en inglés) son estándar y pueden ser modificadas en cualquier momento para desempeñar cualquier número de funciones. Los circuitos pueden ser modificados en cualquier momento del proceso de diseño, ajustándolos a los requerimientos de la aplicación. Los PLDs están basados en tecnología reconfigurable, memoria regrabable, así para modificar el diseño el dispositivo sólo tiene que ser reprogramado.

3.1.2.6. FPGA

Un FPGA es un circuito integrado diseñado para ser configurado por el ingeniero de diseño. Este tipo de dispositivos se programa usando un lenguaje de descripción de hardware. Los FPGA tienen ventajas significativas de rendimiento en comparación con microprocesadores debido a su arquitectura altamente paralela y significativa flexibilidad. El nivel de paralelismo a nivel hardware que puede desarrollar un FPGA en aplicaciones le permite ejecutar dichas aplicaciones más rápido por varios órdenes de magnitud que procesadores embebidos o estaciones de trabajo de alto nivel. Considerando una función entre millones de operaciones por watt, los FPGA han demostrado una ventaja de más de 1000 veces la relación de consumo de potencia/rendimiento sobre los procesadores comerciales más poderosos.

En el pasado la tecnología FPGA solía ser usada para aplicaciones de baja complejidad, volumen y rapidez, sin embargo ahora los FPGA pueden alcanzar más de 500 *MHz* de velocidad de reloj. Además de que el incremento en la densidad electrónica y el adicionamiento de otras características, como procesadores embebidos, bloques específicos de procesamiento de señales; los FPGA están siendo cada vez más adecuados para cualquier tipo de diseño y aplicación.

Pueden ser usados para implementar prácticamente cualquier diseño de hardware. El uso común que tenían estos dispositivos era el de hacer prototipos de pruebas de hardware que serían eventualmente implementados en ASICs. Sin embargo, el uso de dispositivos FPGA ha ido incrementándose para su uso como plataformas finales de diseño.

Además de la capacidad de implementación de hardware para aplicaciones específicas se puede hacer uso de microprocesadores embebidos en el FPGA, teniendo la capacidad de conectar dichos microprocesadores embebidos a diseños en hardware teniendo una solución que combina los dos modelos de solución de aplicaciones embebidas. Esta dualidad resulta muy útil al incorporar software simplificando diseños que resultarían muy complejos de realizar usando solo descripción de hardware.

3.1.2.7. Modelos de ejecución

Una vez definidos los tres tipos de tecnología empleado en el diseño de aplicaciones embebidas: dispositivos de aplicación específica, dispositivos programables y dispositivos de aplicación general; podemos definir dos modelos de ejecución:

Modelo secuencial de ejecución en el que el circuito ALU es multiplexado en tiempo para ejecutar cada operación secuencialmente. Una parte substancial del circuito es usado para decodificar y controlar los pasos de tiempo, generalmente este modelo es conocido como modelo de cómputo de programa almacenado tipo von Neumann.

El modelo no secuencial puede tener una variedad de topologías, pero es claramente diferente al modelo secuencial por la ausencia de un controlador de propósito general que maneje las

secuencias y el multiplexado en tiempo de circuitos fijos. Este modelo es conocido como de ejecución de flujo de datos.

Es importante señalar que la velocidad de procesamiento de los dos modelos no puede ser comparada simplemente contando el número de ciclos ya que las frecuencias de reloj de ambas son diferentes.

Típicamente un periodo de ciclo de reloj en un FPGA puede ser de 5 a 10 veces más largo que uno implementado en procesador con la misma tecnología, sin embargo en un ciclo de reloj en una solución basada en FPGA se pueden realizar cientos y hasta miles de procesos en paralelo en contraste con el proceso secuencial en un microprocesador. Teniendo en cuenta toda esta información, podemos asegurar que el software se ejecuta en un procesador donde éste es hardware que implementa un modelo de ejecución secuencial. A su vez el hardware será la especificación que usaremos para configurar el FPGA y no usa el modelo secuencial de ejecución [54].

De este modo podemos decir que existen dos tipos de soluciones para un sistema de procesamiento de información: soluciones por software y soluciones por hardware. Ambas dependen de los requerimientos que la aplicación tiene, del tipo de dispositivo utilizado para realizar el procesamiento de información y del modelo de ejecución.

3.1.2.8. Elección de tecnología

Al comparar las capacidades con las que cada tecnología cuenta y en base a que los requerimientos del ADCS incluyen procesamiento de información proveniente de múltiples sensores, comunicación con estación terrena, comunicación con Sistema de Posicionamiento Global (GPS), interacción estrecha con otros subsistemas, ejecución de algoritmos aritméticamente demandante, control de actuadores, flexibilidad para reconfigurar el sistema, capacidad de brindar confiabilidad mediante el uso de técnicas de tolerancia a fallas como la implementación de redundancia, y el uso de elementos comerciales para su diseño, la tecnología FPGA se destaca como la mejor opción para el desarrollo del ADCS, ya que ofrece capacidad suficiente para cumplir con todos los requerimientos al combinar soluciones por software y por hardware en un mismo dispositivo. Además es ideal para la implementación de técnicas de tolerancia a fallas y uso de elementos comerciales, reconfiguración del sistema parcial o completo dependiendo de futuros requerimientos del diseño.

3.1.3. Tecnología FPGA

FPGA es un circuito integrado (IC) que contienen bloques lógicos configurables junto con interconexiones entre dichos bloques lógicos. Permiten configurar un circuito electrónico digital de manera arbitraria después de que el chip de silicón ha sido creado, probado e instalado. El circuito que es configurado en el FPGA es llamado diseño.

La arquitectura básica de un FPGA consiste en un arreglo de dos dimensiones de bloques lógicos y flip-flops, entradas-salidas, e interconexiones entre dichos bloques. Además de estos elementos, dispositivos actuales cuentan con hardware dedicado a uno o varios microprocesadores embebidos. Existen tres tipos de tecnología base con la que los FPGA se fabrican [4]:

Tecnología basada en SRAM: las conexiones del FPGA se hacen usando transistores de paso, de compuerta o multiplexores que son controlados por celdas SRAM. Esta tecnología permite configuración rápida del circuito, su mayor desventaja es el tamaño del chip ya que requiere tecnología RAM y una fuente externa no volátil para cargar la configuración del dispositivo, esto significa que:

Número uno: la tecnología RAM es físicamente más grande que otro tipo de tecnologías.

Número dos: al ser volátil, la información almacenada depende de una fuente de energía constante para mantenerse, por lo que es requerida una memoria no volátil para cargar nuevamente la información en caso de ser requerido. Esto impacta directamente en el tamaño del chip. Pueden ser programados un número ilimitado de veces.

Tecnología Antifuse: un elemento antifuse se mantiene en un estado de alta impedancia hasta que es programado en un estado de baja impedancia. Esta tecnología puede ser usada una sola vez limitando la capacidad de reconfiguración que presenta la tecnología SRAM. Es menos costosa que la tecnología RAM.

Tecnología EPROM/EEPROM: siendo del tipo no volátil, esto es que la información contenida no se pierde si la fuente de energía es interrumpida, este método es el mismo usado en memorias EPROM/EEPROM donde la configuración es almacenada en el dispositivo sin una memoria externa, por lo que la reprogramación del circuito no es posible.

Todos los dispositivos FPGA de Xilinx contienen los siguientes recursos básicos: bloques lógicos configurables (CLBs) que contienen lógica combinatorial y registros. Bloques de entrada y salida (IOBs) que son la conexión entre el FPGA y el mundo exterior. Interconexiones programables (PIs). Bloques RAM. Buffers de tres estados, de reloj global, multiplicadores etc. En la Figura 3.3 de la página 69 se muestra un diagrama con los elementos que conforman un FPGA de Xilinx.

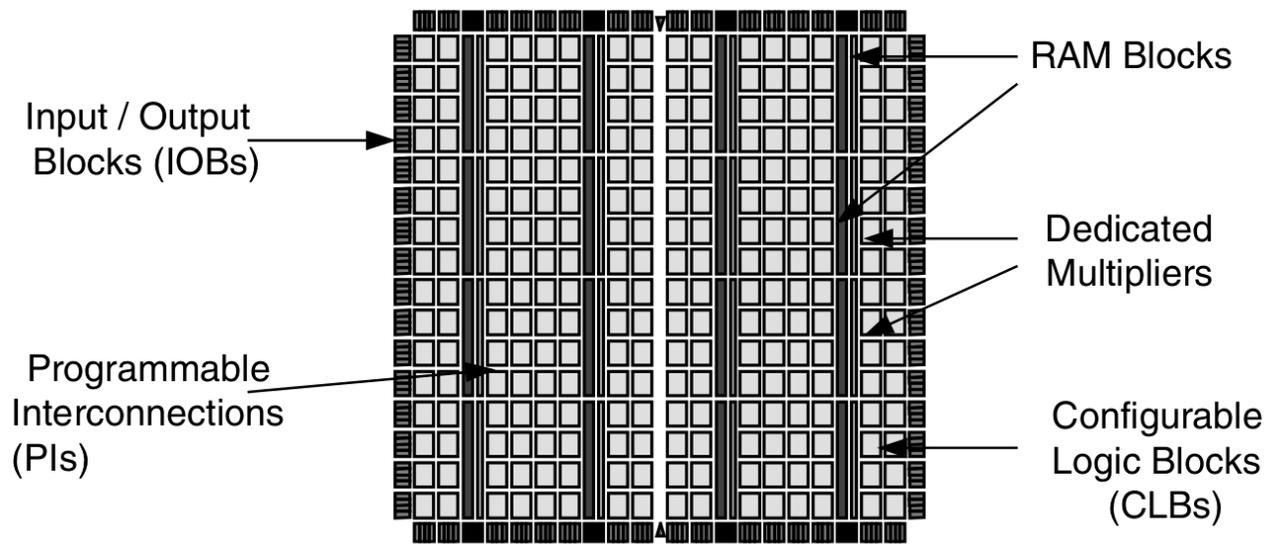


Figura 3.3: Distribución de CLBs, IOBs, PIs, bloques RAM y multiplicadores en un FPGA Virtex [4].

A continuación se describen los diferentes bloques que conforman un FPGA Xilinx.

Bloque lógico configurable (CLBs) El bloque básico de construcción de dispositivos de Xilinx se denomina *Slice*, los dispositivos Virtex y Spartan tienen dos slices en cada CLB. Los CLBs contienen tablas de búsqueda (LUTs) y elementos de almacenamiento. Los generadores de funciones Xilinx son implementados usando LUTs de cuatro entradas, además cada LUT puede ser programada como RAM síncrona de 16x1 bits. Dos LUTs pueden ser combinadas con un slice para crear RAM síncrona de 16x2 o 32x1 bits, o RAM síncrona de 16x1 bits de puerto dual. Finalmente una LUT puede generar un registro de corrimiento de 16 bits, ideal para captura de datos a alta velocidad. Los elementos de almacenamiento en un slice pueden ser configurados como *flip-flops* tipo D [4].

Bloque de entrada salida (IOBs) Este bloque incluye entradas y salidas que soporta una amplia variedad de estándares de señales. Los IOBs son programables y se pueden categorizar de la siguiente manera: pista de entrada, en la que un buffer en la pista de entrada del IOB conecta las señales directamente a la lógica interna o a través de un flip-flop de entrada opcional. Pista de salida, que incluye un buffer de tres estados de salida que guía la señal de salida por la pista, también puede ser guiada a través de un flip-flop de tres estados de salida. Bloque bidireccional que puede ser una combinación de cualquiera de las configuraciones de entrada o salida.

Bloque RAM Los FPGA de Xilinx incluyen varios bloques de memoria RAM, que son organizadas en columnas a lo largo del dispositivo. El número de bloques va desde 8 hasta algunos cientos, dependiendo de la capacidad del dispositivo y familia. Dentro de cada bloque RAM en el FPGA se encuentran tablas de búsqueda configurables con las que se pueden implementar funciones lógicas o usarse como bloques de memoria RAM. También se usan para inicializar datos.

Recursos aritméticos en FPGAs Xilinx Los FPGA incluyen circuitos dedicados sumadores, multiplicadores y contadores, por lo que estas operaciones se realizan de una manera más rápida que si se implementan usando LUTs y ruteo normal solamente. Además, lógica dedicada de acarreo provee capacidades de acarreo en operaciones aritméticas a alta velocidad.

Lenguaje de descripción de Hardware (HDL, por sus siglas en inglés)

Es un lenguaje de computadora diseñado específicamente dedicado para describir circuitos electrónicos. Puede describir operaciones, estructuras así como estímulos de entrada para verificar la operación de un diseño. La sintaxis y semántica de los HDL incluye notaciones explícitas para expresar tiempo y concurrencia, atributos del hardware [4]. Usando apropiadamente cualquier HDL y mediante programas denominados sintetizadores que determinan las operaciones lógicas del hardware y producen los archivos de configuración interna del FPGA (*netlist*) de hardware genérico, se puede implementar las conexiones internas de los circuitos para lograr operaciones específicos. VHDL es, junto con Verilog, uno de los más importantes HDL. El proceso de implementación de un circuito de aplicación específica en un FPGA es complejo y va desde el diseño de este, pasando por la representación, síntesis, generación de ruteo del circuito, configuración del archivo de generación para finalizar con la implementación del circuito. La Figura 3.4 en la página 71 presenta el diagrama de flujo de generación de una aplicación usando FPGA. Dada la complejidad de este proceso sólo se menciona de una manera breve en este trabajo, si se desea una profundización en el estudio de diseño de circuitos usando dispositivos FPGA y ASIC se sugiere revisar las siguientes fuentes [4, 16, 54–60].

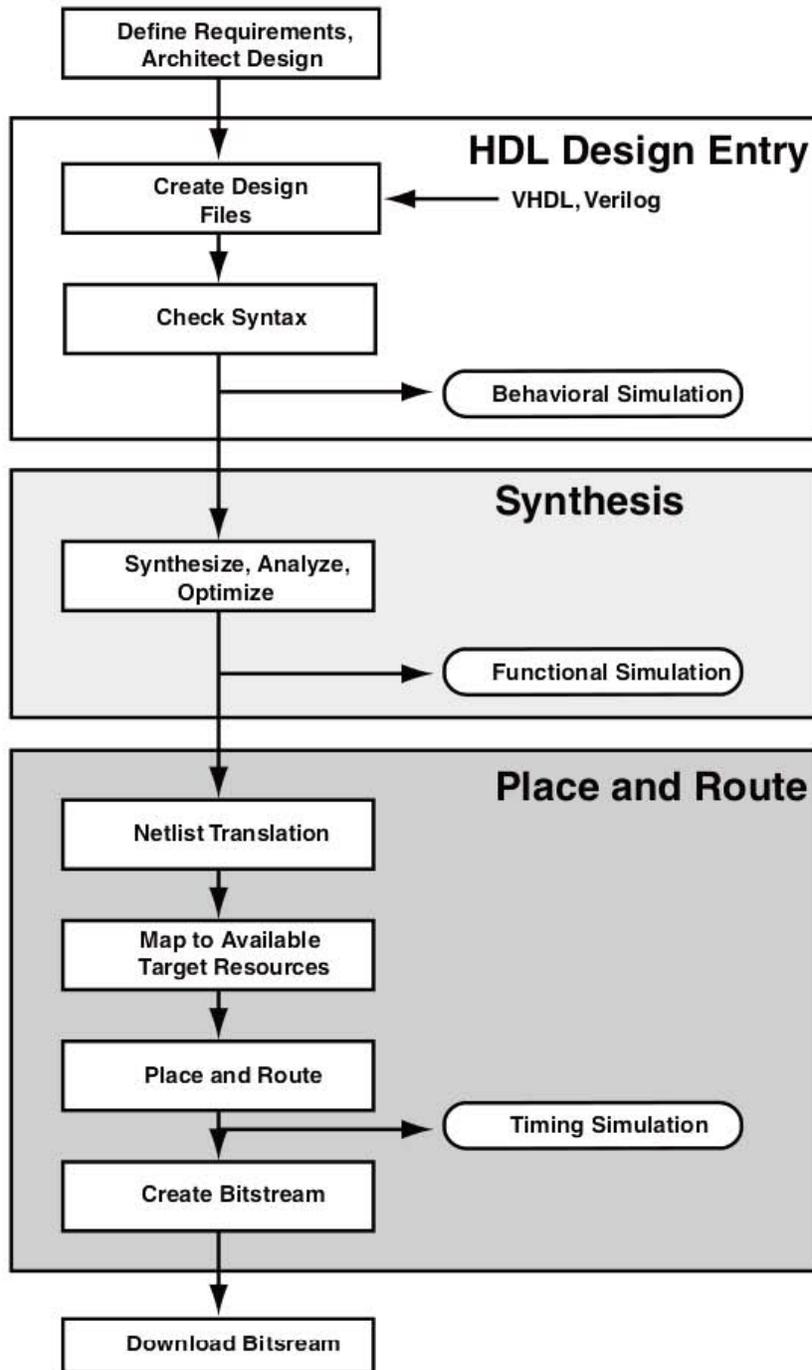


Figura 3.4: Diagrama de flujo de implementación de un diseño en FPGA [5].

3.1.4. Ventajas y desventajas de la Tecnología FPGA

Las principales ventajas que presenta la tecnología FPGA radican en su capacidad de configuración y reconfiguración. Dada la gran flexibilidad que tiene para la implementación de diseños lógicos ésta tecnología puede ser aprovechada para realizar procesos paralelos, contando con núcleos que realicen tareas similares o completamente distintas se puede alcanzar una gran capacidad de procesamiento.

De igual forma si se requiere estas arquitecturas pueden ser modificadas para adaptarlas en base a los requerimientos que presente la aplicación en que se trabaja. Algunas desventajas de la tecnología FPGA en comparación con otras como los microcontroladores es que su costo es mayor. Además de que la sincronización de los procesos corre a cargo del diseñador, por lo que la implementación de una aplicación usando esta tecnología puede resultar en un mayor grado de complejidad que la que tendría usando microcontroladores.

Otra importante desventaja de dispositivos FPGA comerciales (COTS) en aplicaciones espaciales es debido a su alta integración electrónica, esto los hace muy sensibles a los efectos que produce la radiación debido al ambiente para el cual se pretenden trabajar. Sin embargo, dadas las capacidades de reconfiguración y flexibilidad de adoptar arquitecturas paralelas, son ideales para la aplicación de técnicas de tolerancia a fallas pudiendo así cumplir con los requerimientos de confiabilidad.

3.1.5. Determinación de la mejor solución

El uso de dispositivos FPGA de tipo COTS es una opción atractiva para el diseño de plataformas satelitales micro en las que los presupuestos no son tan grandes como en aplicaciones de comunicaciones en órbitas GEO y MEO cuyos satélites tienen costos de cientos de millones de dólares. De igual forma, se describieron algunas características de la tecnología FPGA como la capacidad de reconfiguración y alta capacidad de procesamiento en paralelo que hacen a la tecnología FPGA más atractiva para el desarrollo de este trabajo, dichas características convierten al FPGA en un dispositivo ideal para la aplicación de técnicas de tolerancia a fallas. Por todo esto los dispositivos FPGA COTS son una tecnología con gran potencial para dar solución a aplicaciones de tipo espacial.

Todo el procesamiento de la información se puede realizar en dos formatos numéricos que tienen diferentes propiedades, dichos formatos son el punto fijo y el punto flotante. Elegir el formato adecuado determina la precisión y eficiencia con que se desarrolla la tarea, siendo el punto fijo el que menor precisión ofrece y la mayor eficiencia, no así el punto flotante que requiere una mayor cantidad de procesamiento pero ofrece una precisión muy superior al punto fijo.

La implementación de aplicaciones usando números en formato de punto flotante es muy cómoda para diseñadores de software ya que simplifica el proceso de programación, no así para los diseñadores de hardware dado que la implementación de procesamiento de información usando formato de punto flotante requiere casi 3 veces el hardware que requeriría un formato de punto fijo. Sin embargo la ventaja es la precisión que siempre se mantiene constante con un gran rango dinámico, donde el formato de punto fijo pierde precisión.

En comparación con una implementación basada en software, una precisión numérica mayor es un problema más grande para FPGAs y ASICs [60]. En un diseño en FPGA para implementar aritmética de punto flotante se utilizan recursos de la lógica programable, pero dichas implementaciones consumen área considerable y tienen a requerir hasta 20 etapas de pipeline antes de tener rendimiento aceptable por lo que son convenientes sólo para ciertas aplicaciones de cálculo numérico científico intensivo [61].

La primera aproximación en la propuesta de diseño del sistema fue la implementación discreta de la lógica para el manejo de aritmética de punto flotante, esta propuesta fue descartada debido a la complejidad que requería esta tarea. Dadas las características de la aplicación y del algoritmos en cuanto a número de operaciones, tipo de operaciones -en su mayoría vectorial y matricial-, requerimientos de precisión numérica y el uso de tecnología, la implementación usando únicamente lógica combinacional y secuencial resultó en un diseño muy grande, complejo e ineficiente.

La siguiente aproximación fue el uso de paquetes incluidos en el Estándar VHDL IEEE 2008, en dichos paquetes, se incluyen librerías para el manejo de aritmética de punto fijo y flotante [62].

Desafortunadamente Xilinx ISE aún no cuenta con el estándar 2008 de VHDL, por lo que se buscaron alternativas de software para poder sintetizar (genera los circuitos físicamente dentro de la estructura del FPGA que realicen las operaciones requeridas) el diseño. La alternativa más adecuada pareció ser Active-HDL, un software de diseño, simulación y verificación de sistemas basados en FPGA. Active-HDL es propiedad y es desarrollado por Aldec, empresa dedicada al diseño de sistemas ASICs y FPGA.

Active-HDL está más desarrollado como software que Xilinx ISE en cuanto a simulación, es por eso que ya incluye el estándar 2008 de VHDL dentro de sus librerías, sin embargo, se encuentra atrás de Xilinx ISE en cuanto a las herramientas de síntesis de diseño, por lo que aunque se puede simular prácticamente cualquier diseño usando las librerías de manejo de punto flotante y fijo la síntesis aún no es garantizada.

3.1.6. *Soft Processor*

La opción adecuada considerando estas restricciones y requerimientos resultó en el uso de un microprocesador embebido en el FPGA. Estos dispositivos embebidos se denominan procesadores suaves (soft processors). Usando este dispositivo embebido se puede lograr un buen rendimiento en el desempeño del algoritmo, con la precisión requerida, al usar un *Soft Processor* se tienen las ventajas de ambas soluciones de sistemas embebidos. Por una parte se tiene una implementación relativamente más sencilla usando herramientas de programación y de ser necesario se tendría lógica para implementar algún diseño específico necesario.

La plataforma basada en un microprocesador embebido es completamente configurable en base a las necesidades de diseño, puede ser reconfigurado y expandido con periféricos externos para generar comunicación y procesamiento de información de sensores, actuadores y otros subsistemas de manera paralela. En caso de ser requerido se pueden implementar algoritmos Cordic, transformadas de Fourier u operadores especiales en hardware para acelerar el cálculo liberando capacidad de procesamiento en el *Soft Processor*.

Hoy en día dispositivos FPGA de gran capacidad pueden implementar procesadores suaves ocupando únicamente 1% de sus recursos excediendo los 200 DMIPS de rendimiento. Todos los fabricantes de dispositivos FPGA basados en tecnología RAM proveen procesadores suaves optimizados para sus arquitecturas. Altera tiene el Nios II, Xilinx Microblaze, Lattice el Mico32, y todos siguen arquitectura RISC. Existen otros procesadores embebidos en el mercado que pueden ser implementados en dispositivos FPGA de prácticamente cualquier fabricante tales como PowerPC¹ y ARM, algunos de estos no son propiamente procesadores suaves, ya que están implementados en hardware dedicado dentro del FPGA [59], esto significa que aunque son procesadores embebidos dentro del chip de un FPGA, no son descritos e implementados utilizando los recursos lógicos y de memoria del FPGA, estos procesadores, como el PowerPC, tienen hardware dedicado dentro del FPGA por lo que no son configurables como por ejemplo el Microblaze que se implementa usando los recursos lógicos y de memoria del FPGA.

Como se menciona en Ratter [19] y [24] los dispositivos FPGA llevan años siendo usados en aplicaciones espaciales. En especial el Virtex-II Pro se ha usado no sólo por misiones gubernamentales, también se ha empleado como núcleo de procesamiento en misiones académicas en satélites pequeños como se describe en Huber et al. [63] y Quinn et al. [21]. Además de las misiones en las que se ha usado el Virtex-II Pro de Xilinx integra hardware dedicado a dos procesadores embebidos PowerPC y la posibilidad de usar el *Soft Processor* Microblaze, que está optimizado para la arquitectura de estos FPGA. Se decidió usar este dispositivo para la implementación del algoritmo y como base para un futuro desarrollo del ADCS y otros subsistemas.

Es importante mencionar que el dispositivo Virtex-II Pro es considerado como maduro por Xilinx, esto quiere decir que dispositivos como Virtex 5 y Virtex 6, por mencionar dos

¹Performance Optimization With Enhanced RISC Performance Computing

familias, superan en capacidad de recursos por mucho a Virtex-II Pro, sin embargo dada las necesidades de confiabilidad en aplicaciones espaciales los dispositivos probados suelen ser mejor opción para la implementación de soluciones.

Algunos otros procesadores suaves son: TSK3000A y TSK51/52 de Altium, OpenSPARC de SUN, Cortex-M1 de ARM, LEON3 y LEON4 de Aeroflex Gaisler, OpenRISC de OpenCores, y ARC de ARC Internation y *Synopsys*.

3.1.6.1. PowerPC

El PowerPC 400 es un microprocesador de 32 bits con arquitectura RISC propiedad de IBM. Puede trabajar hasta 667 MHz y generar 1334 DMIPS como mínimo de rendimiento.

El PowerPC 405 es una implementación de 32 bits de la arquitectura de ambiente embebido PowerPC que se deriva de la arquitectura PowerPC 400. La arquitectura PowerPC provee un modelo de software que garantiza la compatibilidad de implementación de la familia de microprocesadores PowerPC. Esta arquitectura está optimizada para su implementación como procesador embebido. Soporte para aritmética de punto flotante puede ser incorporado por hardware o software.

El PowerPC cuenta con 32 registros de 32 bits de propósito general (GPR), 32 registros de 32 bits de propósito especial (SPR) que proveen acceso adicional a recursos del procesador, como registro del contador, recursos para depuración, temporizadores, registros de interrupción, entre otros. La mayoría de estos registros pueden ser usados únicamente por software con privilegios. Cuenta también con un registro de 32 bits de estado de máquina (MSR) que controla la operación del procesador [6].

La organización del hardware del PowerPC se divide de la siguiente manera:

- ↪ Unidad de procesamiento central con un pipeline de 5 etapas que consiste en búsqueda, decodificación, ejecución, respuesta, y etapas cargado de respuestas. Cuenta además con una unidad aritmética lógica (ALU) y una unidad de acumulación y multiplicación (MAC).
- ↪ Unidad de manejo de memoria (MMU) que soporta varios tamaños de paginado de 1 KB, 4 Kb, 16 KB, 256 KB, 1 MB, 4 MB y 16 MB. Cuenta además con varios atributos de protección de almacenamiento y opciones de control de acceso.
- ↪ Unidades separadas de cache de instrucciones y cache de datos.
- ↪ Soporte para depuración, incluyendo una interfaz JTAG.
- ↪ Tres temporizadores programables.

La Figura 3.5 en la página 76 muestra la organización del hardware de PowerPC.

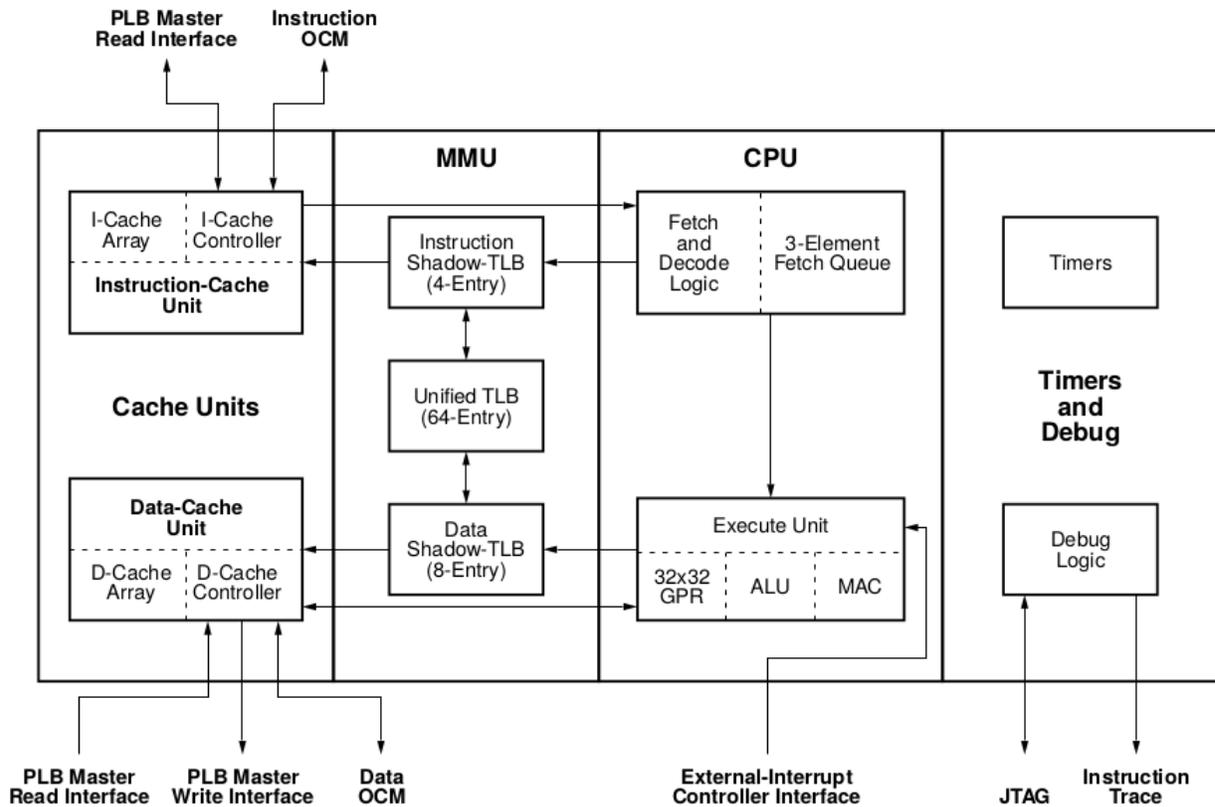


Figura 3.5: Organización del hardware de PowerPC [6].

3.1.6.2. Xilinx Microblaze

Microblaze es un núcleo procesador suave RISC de 32 bits diseñado y optimizado para ser usado en familias de FPGA Xilinx [7].

El Microblaze es altamente configurable, permitiendo seleccionar capacidades específicas para cada aplicación. Por otro lado sus características fijas son: 32 registros de 32 bits de propósito general, instrucciones de 32 bits de palabra con tres y dos modos de operación, bus de direccionamiento de 32 bits. Además, se pueden habilitar funcionalidades adicionales, tales como una Unidad de Procesamiento de Punto Flotante (FPU), optimización de área en la que se implementa un pipeline de 3 etapas (Fetch, Decode, Execute) disminuyendo la capacidad de procesamiento y liberando recursos del FPGA para lógica, u optimización para aumentar capacidad de procesamiento implementando un pipeline de 5 etapas (*Fetch, Decode, Execute,*

Access Memory, Writeback) a un costo de aumento de área en el FPGA, multiplicador por hardware de 64 bits, conversión de aritmética de punto flotante e instrucciones de raíz cuadrada, Unidad de Manejo de Memoria (MMU) para poder ejecutar sistemas operativos en tiempo real, unidad de corrimiento por hardware, división por hardware, instrucciones de comparación de patrones, entre otras [7]. En la Figura 3.6 de la página 77 se muestra el diagrama describiendo las capacidades fijas y opcionales de Microblaze. La capacidad de memoria y velocidad de reloj dependen del dispositivo en el que es implementado, teniendo hasta 256 KB disponibles para instrucciones y datos compartidos, con un reloj @ 100MHz en el dispositivo Virtex-II Pro.

Adicionalmente el Microblaze incorpora características de tolerancia a fallas opcionales. Provee detección de errores para los bloques RAM internos, detección y corrección de errores (ECC) en bloques LMB RAM. Cuando las características de tolerancia a fallas son habilitadas, todos los errores suaves en bloques de memoria RAM son detectados y corregidos, lo que disminuye significativamente la intensidad de falla. Debido a los requerimientos de precisión que presenta el algoritmo el uso de aritmética de punto flotante es necesario. Las operaciones con aritmética de punto flotante son intensivamente demandantes para procesadores de propósito general por lo que es necesario el uso de una unidad de punto flotante que mejore la capacidad de procesamiento del Microblaze.

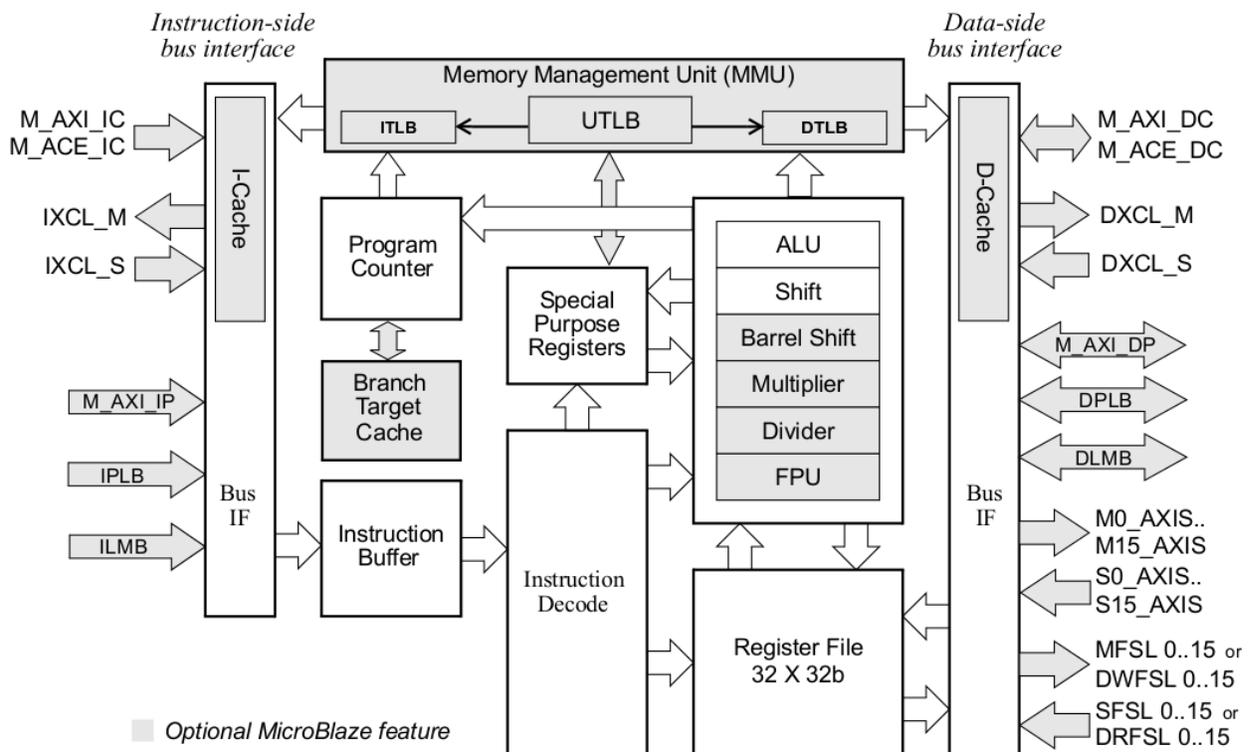


Figura 3.6: Diagrama de bloques de *Soft Processor* Microblaze [7].

La unidad de procesamiento de punto flotante (FPU) de Microblaze está basada en el estándar 754 de la IEEE (*Institute of Electrical and Electronics Engineers*):

- ↪ Usa el formato IEEE 754 de precisión simple, incluyendo definición de infinito, no número (NaN) y cero.
- ↪ Soporta instrucciones de suma, resta, multiplicación, división, comparación, conversión y raíz cuadrada.
- ↪ Implementa redondeo al modo más cercano.

Un número en formato simple de punto flotante IEEE 754 está compuesto por los siguientes tres elementos:

- ↪ Un bit de signo.
- ↪ Exponente de 8 bits.
- ↪ Fracción de 32 bits, también conocida como mantisa o significante.

Esta FPU incluye las siguientes operaciones aritméticas: adición, sustracción, multiplicación, división, raíz cuadrada. También incorpora operaciones de comparación y conversión: menor que, igual a, menor o igual, mayor que, no igual, mayor o igual, conversión de entero signado a flotante, y de flotante a entero signado.

La FPU usa excepciones por hardware, cuando son habilitadas, excepciones son activadas para todas las condiciones estándar de IEEE: subdesbordamiento, desbordamiento, división por cero, y operaciones ilegales. Operaciones de punto flotante de doble precisión son emuladas por software.

Determinación y selección de solución

Microblaze es la opción ideal para la implementación del algoritmo ADCS por las características de cantidad de memoria, flexibilidad de configuración, rendimiento, unidad de punto flotante (FPU), características de tolerancia a fallas, cantidad de núcleos dedicados para el control de interfaces de comunicación, controladores de memoria, dispositivos de entrada y salida de propósito general, y capacidad de implementación de diseños discretos.

Inicialmente debido a los requerimientos de aritmética y precisión que el sistema presenta se planteó una solución basada en Hardware con la utilización del estándar IEEE 2003 para VHDL como HDL y usando el software Xilinx ISE. Esta primera aproximación a la solución resultó en alta complejidad, con un gran tiempo de desarrollo, simulación y depuración por lo que fue desechada. Bajo estas condiciones el circuito era sintetizable usando las herramientas de Xilinx.

La segunda propuesta de solución también estaba basada en Hardware utilizando VHDL como HDL, con la diferencia que se utilizaría el estándar IEEE 2008 para VHDL. Este estándar, a diferencia del 2003, incluye la capacidad de manejo de aritmética de punto flotante (FPA, por sus siglas en inglés) lo cual solucionaba el problema de alta complejidad de diseñar lógica para aritmética de punto flotante desde cero reduciendo el tiempo de implementación. Mediante el software Active-HDL que incluye el estándar 2008 para VHDL se describieron algunas operaciones matriciales requeridas por el sistema, sin embargo aunque Active-HDL cuenta con herramientas de simulación para el estándar 2008 no cuenta con las herramientas de síntesis de circuitos, por lo que no fue posible implementar esta propuesta de solución.

Finalmente se optó por realizar una solución basada en software mediante la utilización de un procesador embebido en el FPGA y lenguaje C++ para la programación del algoritmo. Bajo estas condiciones el circuito fue sintetizable usando Xilinx ISE cumpliendo con los requerimientos de precisión, además se tiene la capacidad de combinar diseños discretos en hardware con soluciones basadas en software.

En la Figura 3.7 de la página 80 se muestran las diferentes aproximaciones a la solución que se propusieron y cuáles fueron los factores que determinaron su aplicación o rechazo.

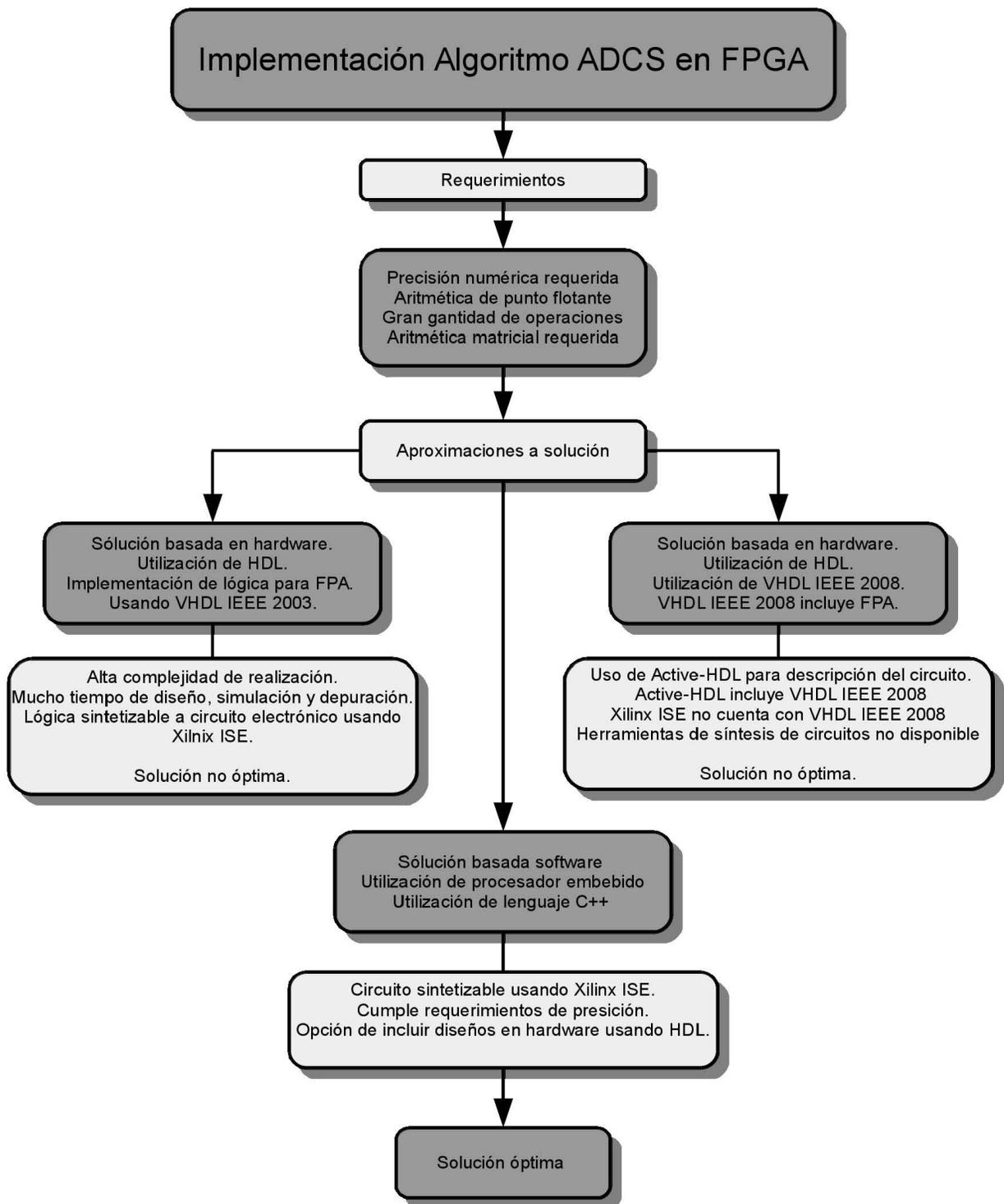


Figura 3.7: Diagrama de flujo de las aproximaciones y determinación de la solución óptima.

3.1.7. Programa Universitario Xilinx Virtex-II Pro XC2VP30

Xilinx ofrece un sistema de desarrollo poderoso, versátil y de bajo costo para todos los niveles de programas educativos de ingeniería. El Programa Universitario Xilinx (XUP) Virtex-II Pro ² (XUPV2P) más que una tarjeta de desarrollo es todo un sistema de desarrollo basado en el FPGA Virtex-II Pro, que puede ser usado desde cursos introductorios de ingeniería hasta proyectos avanzados de investigación. Se compone del Virtex-II Pro integrado a una colección de componentes y periféricos que pueden ser usados para crear complejos sistemas y demostrar la capacidad del Virtex-II Pro [8].

El XUP Virtex-II Pro cuenta con las siguientes características:

- ↔ FPGA Virtex-II Pro con 30,816 celdas lógicas, 136 multiplicadores de 18 bits, 2,448 Kb de bloques RAM
- ↔ Dos núcleos PowerPC @ 300 MHz.
- ↔ Hasta 2 GB de memoria DDR SDRAM
- ↔ Sistema controlador ACE y conector tipo Type II CompactFlash para configuración del FPGA y almacenamiento de datos
- ↔ Puerto serial RS-232 DB9
- ↔ Dos puertos seriales PS-2
- ↔ Tres puertos seriales ATA, dos puertos Host y uno objetivo
- ↔ 100 MHz de sistema de reloj, 75 MHz de reloj SATA
- ↔ Audio CODEC AC-97 con amplificador de audio y salida para audífono o altavoz
- ↔ Dispositivo Ethernet 10/100 PHY

En la Figura 3.8 de la página 82 se muestra un diagrama de bloques del XUP Virtex-II Pro, al centro se encuentra el FPGA las conexiones con los diferentes módulos de control, comunicación, de tiempo y fuente de potencia. La Figura 3.9 de la página 83 muestra una foto del XUP Virtex-II Pro (XUPV2P).

Información específica de todas las características del FPGA Virtex-II Pro se pueden encontrar en [64].

²Virtex®-II Pro

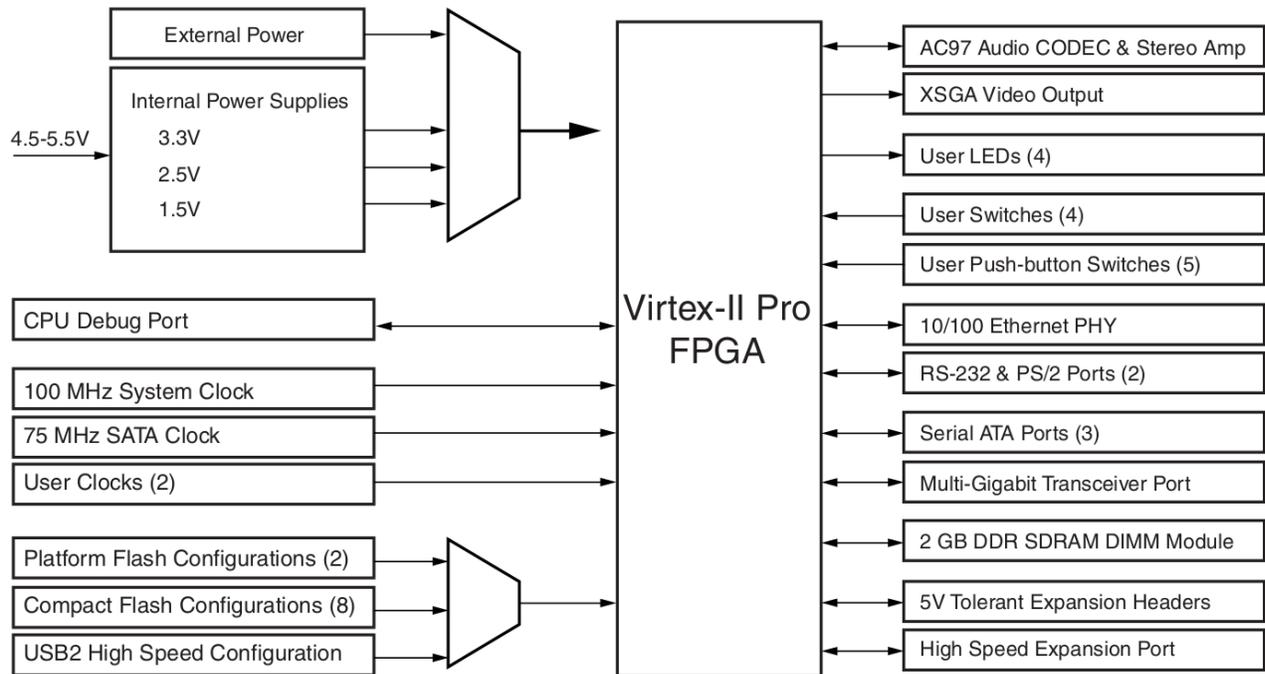


Figura 3.8: Diagrama de bloques de XUP Virtex®-II Pro [8].



Figura 3.9: Fotografia de XUP Virtex®-II Pro.

3.2. Configuración de Arquitectura

La configuración de la arquitectura se hizo usando el software Xilinx Plataform Studio (XPS) versión 10.0.03 que cuenta con soporte para XUP Virtex-II Pro en su versión para sistema operativo Windows de 32 bits. Disponible también en versiones de 64 bits para plataforma Linux y Windows.

3.2.1. Configuración de Arquitectura para la implementación del Algoritmo de Determinación de Posición Y Control

Como se ha mencionado, se hizo uso de Microblaze dada la flexibilidad que presenta, se configuró para brindar el mayor desempeño posible debido a la gran cantidad de operaciones con aritmética de punto flotante requeridas, además se incluyó en la configuración capacidad de comunicación serial vía RS232 para poder hacer una comparación entre los resultados de la implementación hecha en XUPV2P y una simulación en Matlab. Ningún otro puerto o entrada de propósito general fue utilizado.

Se debe conseguir el archivo comprimido de configuración *EDK-XUP-V2ProPack.zip*, de la dirección <http://www.digilentinc.com/Data/Products/XUPV2P/EDK-XUP-V2ProPack.zip>, el cual incluye los archivos de Descripción de Tarjeta Xilinx (XBD), que define el contenido de una tarjeta de desarrollo particular y cómo interactúa con el FPGA de la tarjeta, en este caso las configuraciones para Microblaze y *PowerPC*, y su interacción con todos los puertos de la tarjeta XUPV2P. El archivo XBD tiene las siguientes características [7]:

- ↪ Bloques que definen las interfaces del FPGA soportadas por la tarjeta.
- ↪ Cada bloque tiene una lista de atributos, parámetros y puertos.
- ↪ Información de conectividad entre diferentes puertos y módulos.
- ↪ Información de archivos UCF para configuración de cada pin del FPGA.

Los archivos UCF son archivos ASCII que especifican las restricciones en el diseño. Las restricciones afectan como el diseño lógico es implementado en el dispositivo FPGA. Estos archivos de configuración son diseñados específicamente por el fabricante de la tarjeta de desarrollo, en este caso Xilinx. Se pueden diseñar tarjetas propias con los puertos que se requieran para diseños específicos, en tal caso se tendrían que generar archivos de configuración UCF y XBD propios.

No se hará un paso a paso de la configuración de Microblaze, sólo se mencionarán los puntos más importantes.

Una vez teniendo los archivos UCF y XBD se crea un nuevo proyecto en XPS. Se selecciona el dispositivo XUPV2P y Microblaze como núcleo. Se seleccionaron 64 KB de memoria compartida para instrucciones y datos, y habilitó la unidad de punto flotante (FPU). Se seleccionaron 100 MHz de frecuencia de reloj. También se seleccionó el periférico XPS UAR-LITE con un baudaje de 115200 bits por segundo, 8 bits de datos sin paridad. El mapa de memoria es generado automáticamente por XPS, teniendo los resultados que se muestran en la Figura 3.10 contenida en la página 86.

Se incluyó una FPU, multiplicador de enteros de 32 bits y habilitó un pipeline de 5 etapas que incrementa el tamaño del diseño en el FPGA pero maximiza su rendimiento. Una vez seleccionados los elementos es necesario incluir las librerías necesarias para que los puertos interactúen con el Microblaze de manera transparente para el diseñador.

La Figura 3.11 de la página 87 muestra un diagrama de bloques donde se detallan el tipo de núcleos, interfaces y buses configurados. Se pueden observar el bus local de memoria de datos (*dlmb*, LMB), que es un bus local de conexión rápida para conectar los puertos I y D de Microblaze a los periféricos y bloques RAM (BRAM). El bus local de memoria de instrucciones (*ilmb*, LMB) es un bus local de rápida conexión para conectar los periféricos y BRAM a Microblaze. El bus local de procesador (*mb_plb*, PLB) que es un bus local que incluye unidad de control de bus, temporizador *watchdog*, y unidades separadas de direccionamiento, escritura, y lectura de datos. Bloques RAM incluyendo *dlmb_cntlr* e *ilmb_cntlr*, (*lm_bram*, BRAM) son módulos de memoria configurable que se conectan a una variedad de controladores de interfaz BRAM. *RS232-Uart_1* es un módulo genérico UART para bus PLB46. *Proc_sys_reset_0* es un módulo de manejo de restablecimiento. *Clock_generator_0* es el módulo generador de reloj para el sistema del procesador. Finalmente se puede observar el módulo *microblaze_0* que es el procesador de 32 bits Microblaze.

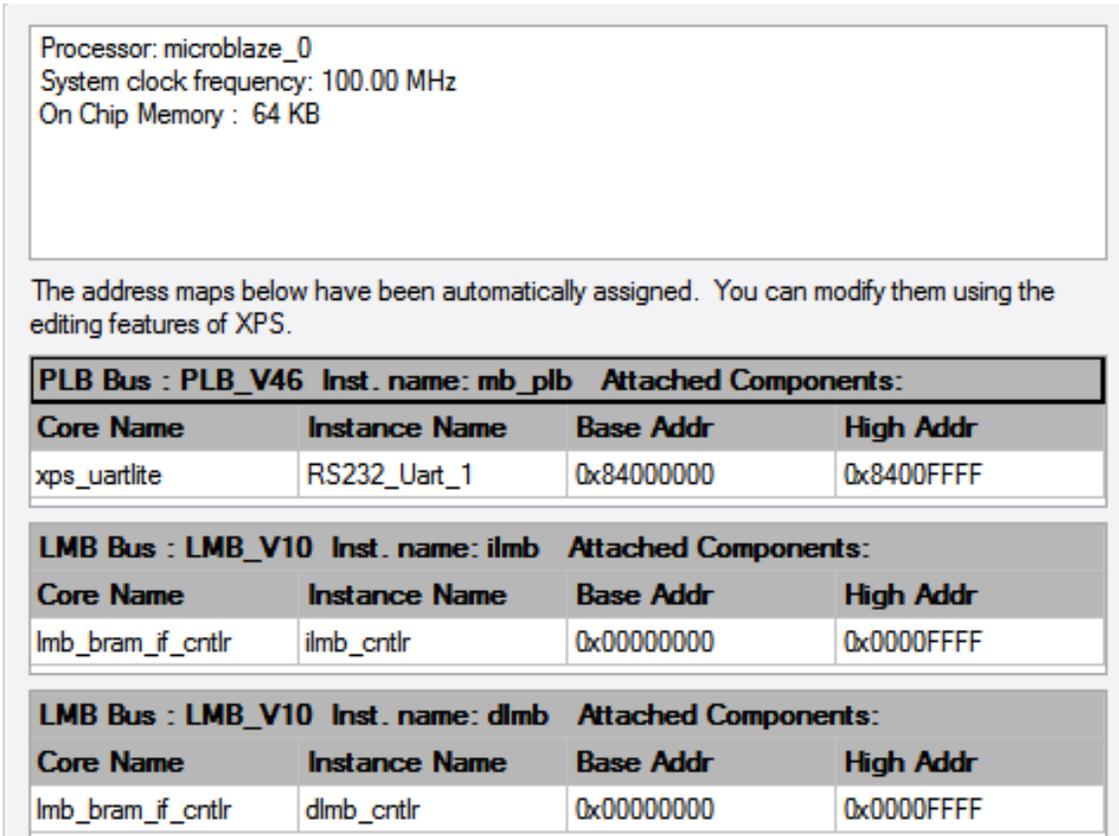


Figura 3.10: Mapa de memoria de configuración, velocidad de reloj y cantidad de memoria de la implementación.

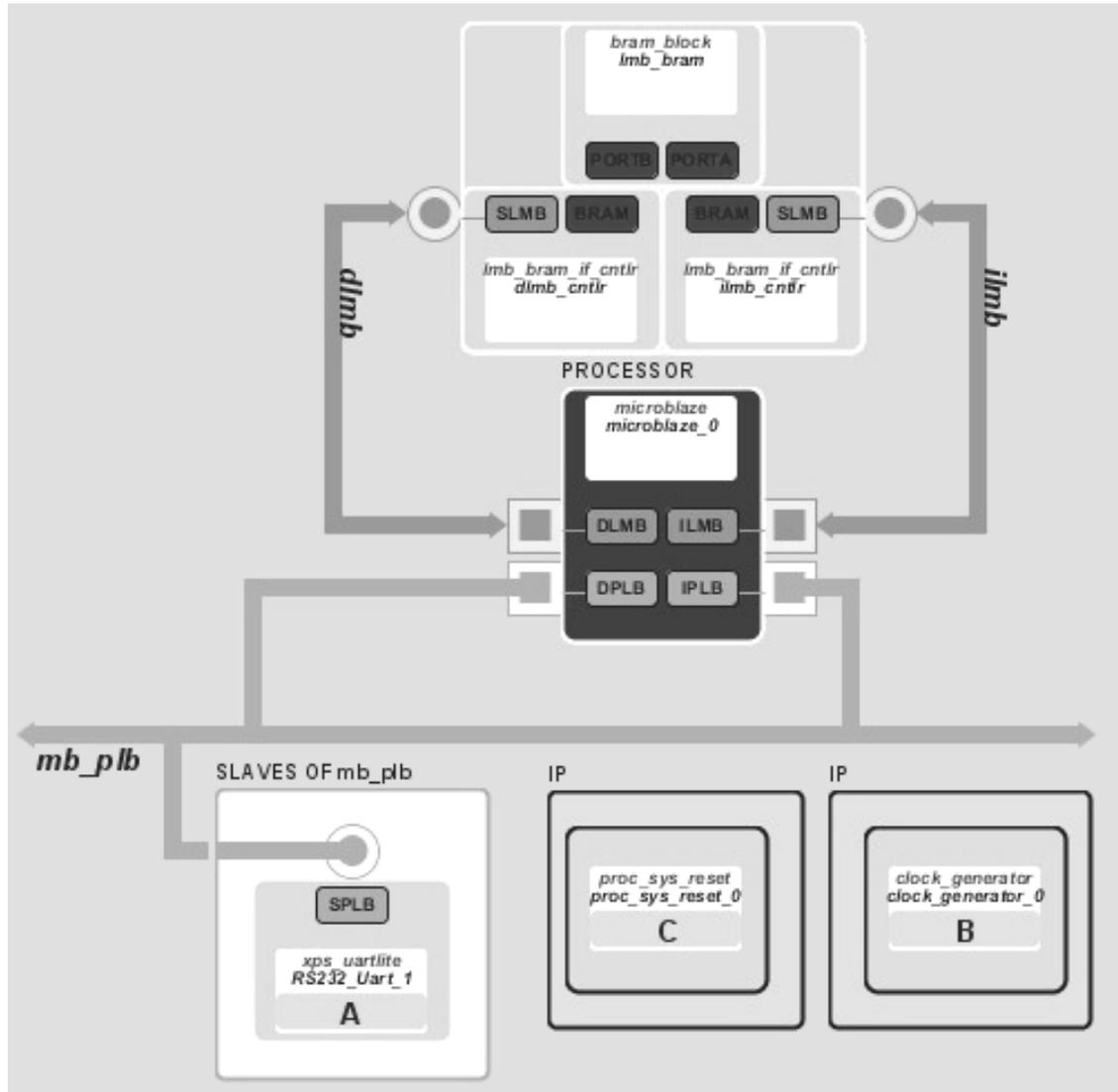


Figura 3.11: Diagrama de bloques de la Implementación de Microblaze.

3.2.2. Resumen de implementación de arquitectura en XC2VP30 FPGA Virtex-II Pro

Bajo las configuraciones mencionadas en el punto 3.2.1., los resultados en la implementación de Microblaze en el FPGA XC2VP30 son los siguientes:

Tipo de recurso	Utilizado #	Disponible #	Porcentaje %
Slices	2524	13696	18
Slices Flip Flops	2532	27392	9
LUTs de 4 entradas	3877	27392	14
IOs	11	N/A	N/A
Bonded IOs	11	556	1
GCLKs	1	16	6
DCMs	1	8	12
MULT18x18s	8	136	5
BRAMS	32	136	23

Tabla 3.1: Uso de recursos del FPGA XC2VP30 después de implementación.

Como se puede ver en la Tabla 3.1 el dispositivo utilizó aproximadamente el 11 % del total de sus recursos disponibles para implementación de lógica, sin mencionar que los dos núcleos PowerPC no fueron empleados. Dada la capacidad que tiene este dispositivo es posible hacer un arreglo de varios núcleos de procesamiento que pueden ejecutar tareas de manera simultánea, el límite es fijado por la capacidad del dispositivo.

Información de los límites de frecuencia de reloj de los diferentes módulos que conforman al sistema se muestran en la Tabla 3.2. Se observa que el valor crítico de frecuencia se presenta en el módulo *microblaze_0*, por lo que esta es la frecuencia máxima a la que el sistema puede trabajar bajo las condiciones de configuración seleccionadas. Esta máxima frecuencia depende de las condiciones físicas del dispositivo y la configuración seleccionada para el módulo. Con las herramientas de software adecuadas se puede hacer una optimización de las trayectorias internas de los circuitos teniendo así la posibilidad de mejorar la frecuencia máxima de operación.

Módulo	Puerto CLK	Frecuencia Máxima [MHz]
microblaze_0	DCACHE_FSL_OUT_CLK	146.278
RS232_Uart_1	SPLB_Clk	219.298
mb_plb	PLB_Clk	277.350
proc_sys_reset_0	Slowest_sync_clk	278.746
Ilmb	LMB_Clk	341.530
dlmb	LMB_Clk	341.530
clock_generator_0	CLKIN	341.530

Tabla 3.2: Límite crítico de Frecuencia para los módulos del sistema implementado.

3.3. Implementación del algoritmo de Determinación de Posición Y Control

Una vez configurada la plataforma se debe implementar el algoritmo. El proceso completo de implementación del algoritmo consistió en tres etapas: la primera etapa fue la generación de una simulación mediante la implementación del algoritmo usando un lenguaje numérico computacional. Se hizo uso de Matlab por la capacidad que tiene de manejar matrices, generar gráficas, por la cantidad de funciones de control que contiene y además porque ya se tenía experiencia en su uso lo cual redujo el tiempo de desarrollo de la simulación. Como segunda etapa, se hizo una conversión de las funciones en lenguaje Matlab que conformaban la simulación a C++ estándar para comparar su resultado. Finalmente, las funciones realizadas en C++ estándar se ajustaron a C++ empleado para programar Microblaze.

Cabe mencionar que la simulación pudo haber sido hecha usando otro lenguaje de programación como Python, Octave, Fortran, C++ o Simulink.

3.3.1. Simulación numérica del algoritmo de Determinación de Posición Y Control

La simulación se realizó usando Matlab 2013a en su versión para plataforma Linux de 64 bits. La distribución de GNU/Linux utilizada fue Debian.

La simulación tiene como objetivo evaluar el modelo matemático del satélite y llevar su posición desde los valores iniciales ($\phi = \phi_0, \theta = \theta_0, \psi = \psi_0$) hasta hasta una posición

($\phi = 0, \theta = 0, \psi = 0$) donde se encuentra estabilizado. Se seleccionaron tercias de valores iniciales para comparar datos entre la simulación e implementación que cubrieran un rango de valores representativos y tratando de que sean diferentes para evaluar la simulación e implementación.

Los valores de tercias seleccionados son los siguientes:

$$\mapsto \phi = -45.125, \quad \theta = -45.987, \quad \psi = 48.978$$

$$\mapsto \phi = 12.825, \quad \theta = 15.987, \quad \psi = -8.197$$

$$\mapsto \phi = 7.99, \quad \theta = -5.987, \quad \psi = 6.197$$

$$\mapsto \phi = 25, \quad \theta = 25, \quad \psi = 25$$

Como se ha descrito, el algoritmo a implementar se divide en los siguientes bloques o funciones: dinámica, cinemática, modelo de medio ambiente, cálculo de ganancia mediante LQR. Aunque Matlab es un ambiente para manipulación de matrices y vectores, que incorpora de manera transparente para el usuario operaciones de este tipo y dado que el algoritmo requiere el uso de este tipo de operaciones decidí realizar todas las funciones que conforman cada bloque del algoritmo de manera discreta, esto es, programar funciones para multiplicación de matrices, vectores, inversión de matrices, producto punto, producto cruz, etc., con el fin hacer más parecida la ejecución de la simulación en Matlab con la implementación hecha en C++.

Las funciones principales son: `dynamics`, `kinematics`, `environmentmodel` y `lqrcontrol`. Además hay una función para convertir ángulos de Euler a cuaterniones y cuaterniones a ángulos de Euler. En la Figura 3.12 de la página 92 se muestra un diagrama de flujo de la simulación y la interacción de las diferentes funciones.

Dynamics: evalúa los vectores velocidad angular ($w_b_ib_{3 \times 1}$) y aceleración angular ($w_b_ib_dot_{3 \times 1}$) del satélite en el marco del satélite respecto al marco inercial. Sus argumentos son el torque ($tau_{3 \times 1}$), la matriz de Inercia ($Imatrix_{3 \times 3}$), la matriz inversa de la matriz de inercia y la velocidad angular en el mismo marco de referencia ($w_b_ib_{3 \times 1}$). Hace uso de las siguientes funciones: *VecCross*, *MatrixMul*, *InverseMatrix*.

Kinematics: evalúa los valores actuales de los vectores de cuaterniones ($q_{4 \times 1}$) y su derivada ($q_dot_{4 \times 1}$), la matriz de rotación del marco satelital al orbital ($r_b_o_{3 \times 3}$), y la velocidad angular del marco satelital al orbital satelital ($w_b_ob_{3 \times 1}$). Sus argumentos son la velocidad angular del satélite en el marco satelital respecto al marco inercial ($w_b_ib_{3 \times 1}$), el vector de cuaterniones ($q_{4 \times 1}$) y la velocidad angular inicial (w_0). Hace uso de las siguientes funciones: *RotationMatrix*, *skewsymmetric*, *MatrixMul*.

Environmentmodel: evalúa los valores de campo magnético en el marco satelital ($b_b_{3 \times 1}$) y orbital ($b_o_{3 \times 1}$), así como el torque gravitacional ($tau_g_{3 \times 1}$). Sus argumentos son la

matriz de rotación en el marco satelital respecto al orbital ($r_b_o_{3 \times 1}$), velocidad angular inicial (w_0), matriz de inercia ($Imatrix_{3 \times 1}$) y altitud del satélite. Hace uso de las siguientes funciones: *slat*, *slon*, *igrf11magm*, *VecCross*. La función *igrf11magm* es parte del *Aerospace Toolbox* de Matlab, y calcula el valor de campo magnético de la Tierra usando un modelo de grado 11 del Campo de Referencia Geomagnético Internacional (IGRF). Es válido para una altura de entre 1 km y 600 km.

lqrcontrol: evalúa la ganancia óptima ($kgain_{3 \times 6}$) y torque de control ($m_{3 \times 1}$) mediante el uso de un controlador LQR. Sus argumentos son el vector de estados ($xvector_{1 \times 6}$), el campo magnético ($r_b_o_{3 \times 1}$), la velocidad angular inicial (w_0), y el vector de inercia del satélite. Hace uso de la función *lqr*, incluida en el Control System Toolbox de Matlab, dicha función evalúa la ganancia óptima de un sistema lineal mediante la minimización de una función cuadrática de costo.

Euler2quat: evalúa los cuaterniones unitarios a partir de los ángulos de Euler.

Quat2eul: evalúa los valores de ángulos de Euler a partir de los cuaterniones unitarios.

Después de la Figura 3.12, comenzando en la página 93 y hasta la página 100, se muestran las gráficas de los resultados obtenidos para los diferentes valores de ángulos de Euler iniciales, donde Muestras [n] representa el número de puntos en la simulación.

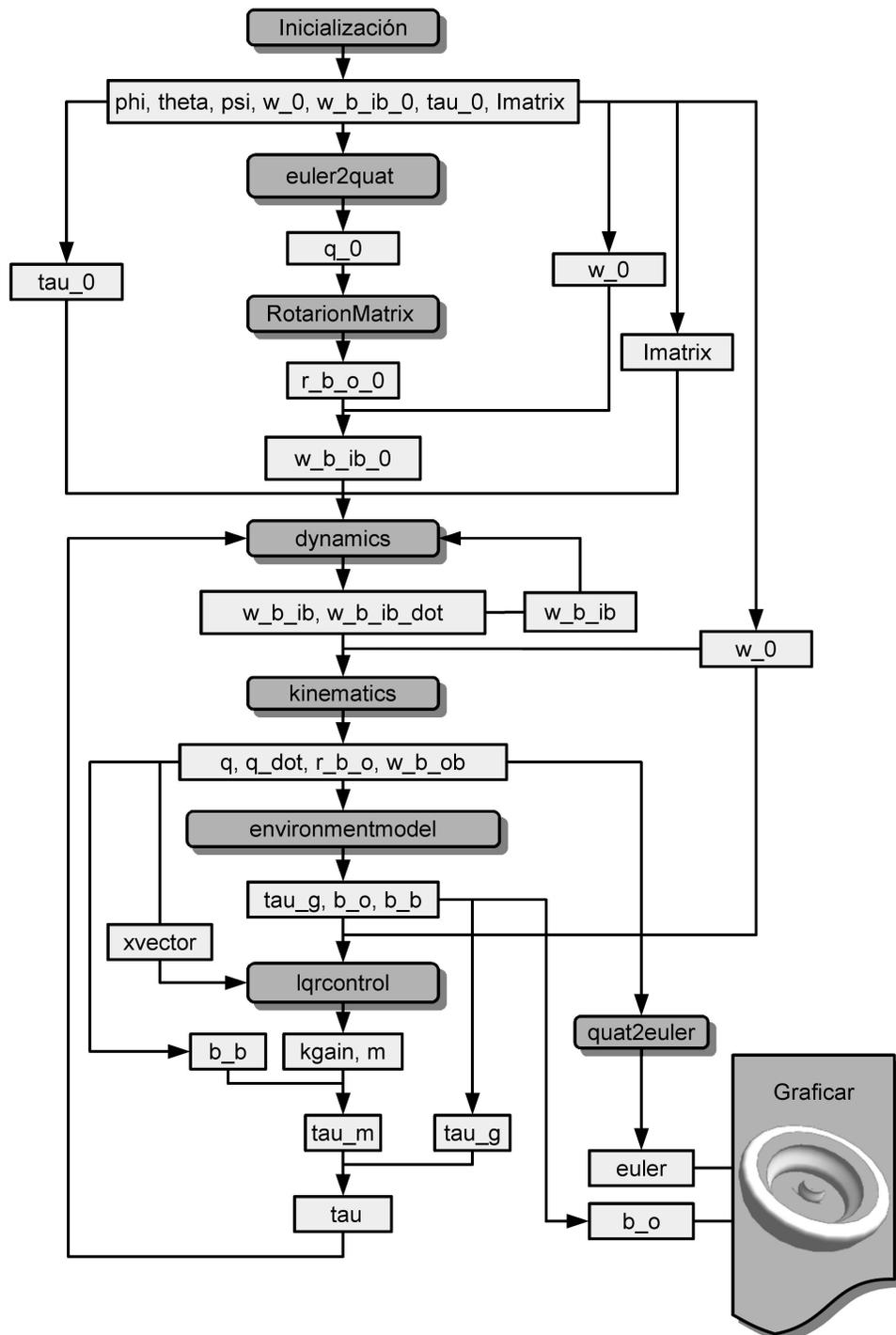


Figura 3.12: Diagrama de flujo de la simulación en Matlab.

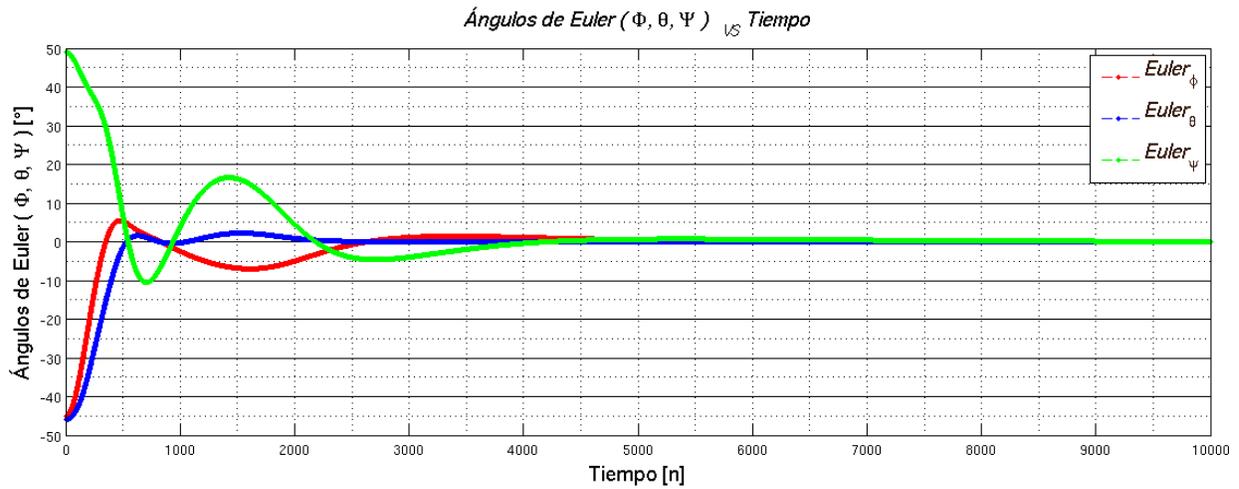


Figura 3.13: Gráfica de ângulos de Euler para valores iniciais $\phi = -45.125$, $\theta = -45.987$, $\psi = 48.978$ gerada por Matlab.

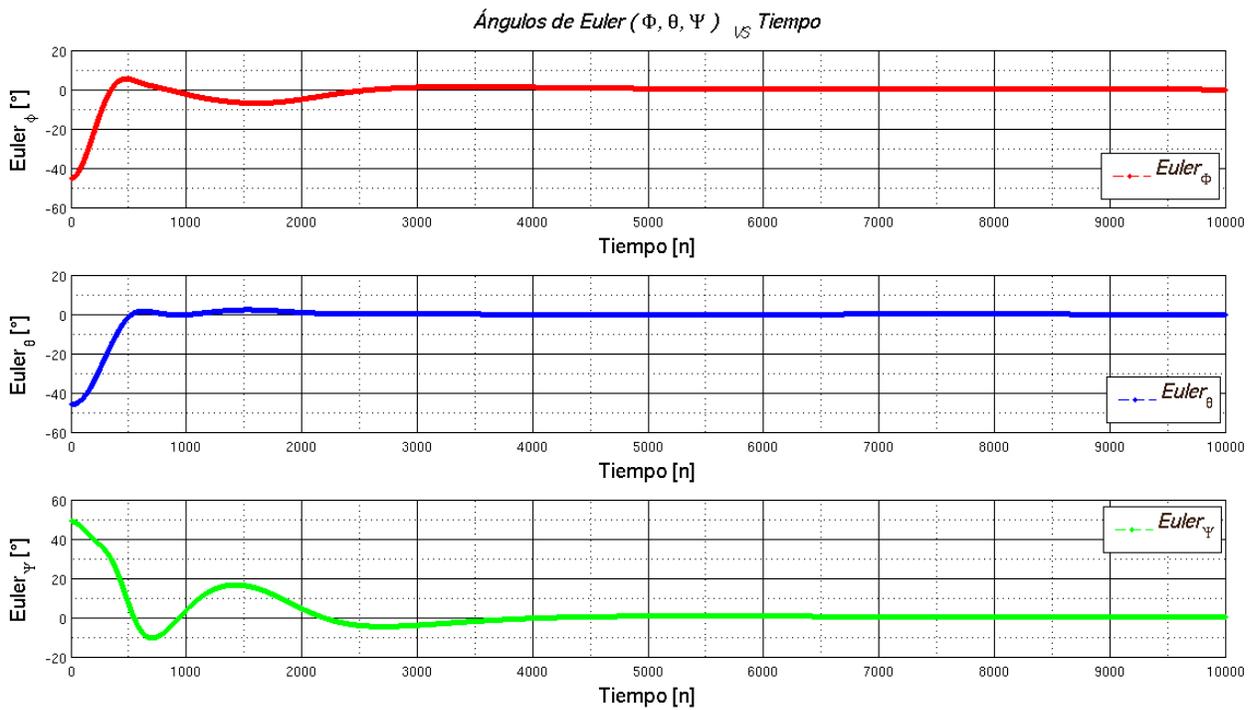


Figura 3.14: Gráfica de ângulos de Euler para valores iniciais $\phi = -45.125$, $\theta = -45.987$, $\psi = 48.978$ gerada por Matlab, por separado.

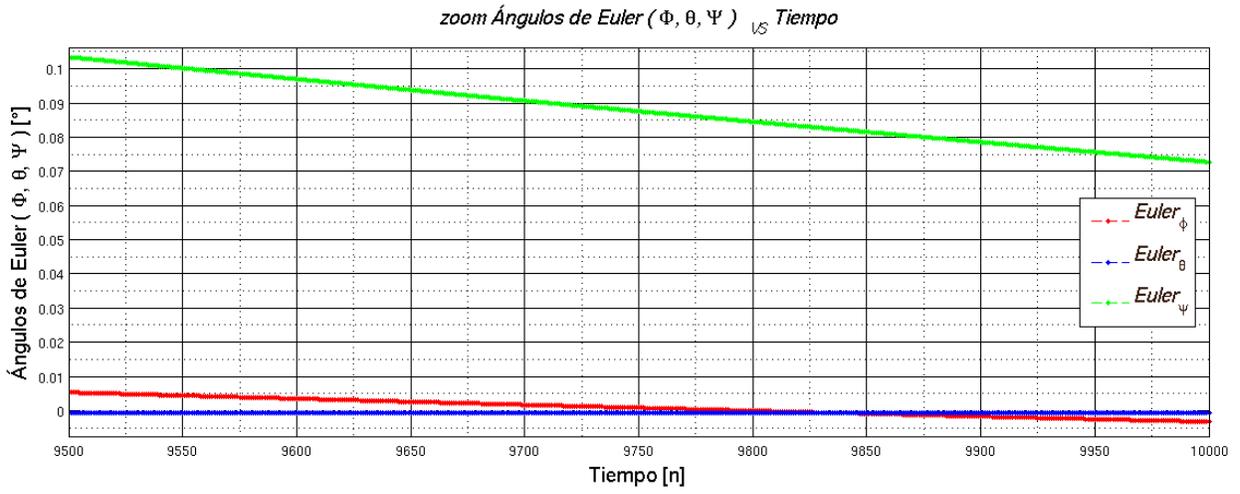


Figura 3.15: Gráfica de ángulos de Euler para valores iniciales de $\phi = -45.125$, $\theta = -45.987$, $\psi = 48.978$ generada por Matlab, zoom a los últimos 500 puntos de simulación.

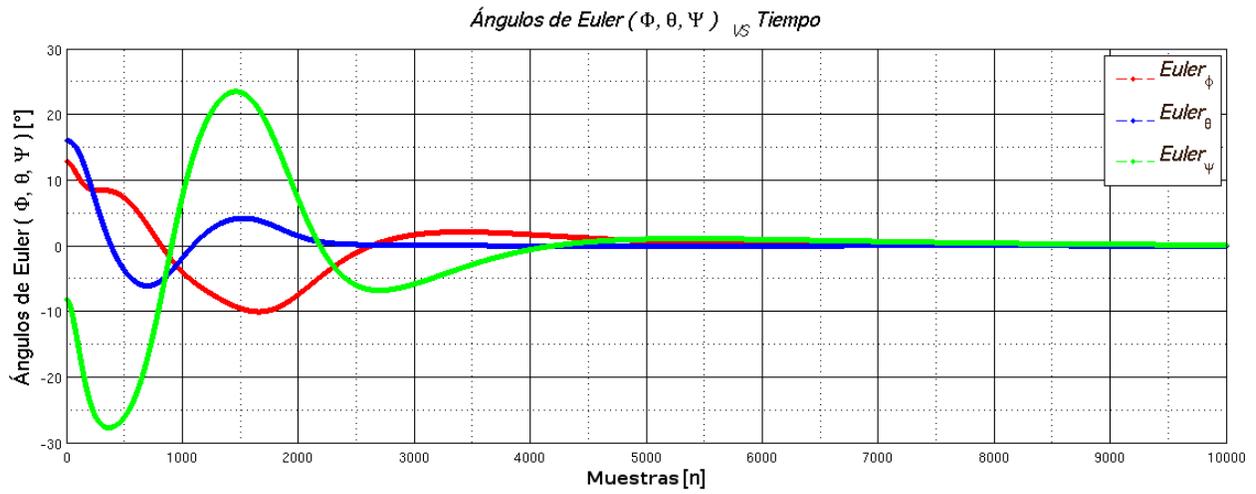


Figura 3.16: Gráfica de ângulos de Euler para valores iniciais $\phi = 12.825$, $\theta = 15.987$, $\psi = -8.197$ gerada por Matlab.

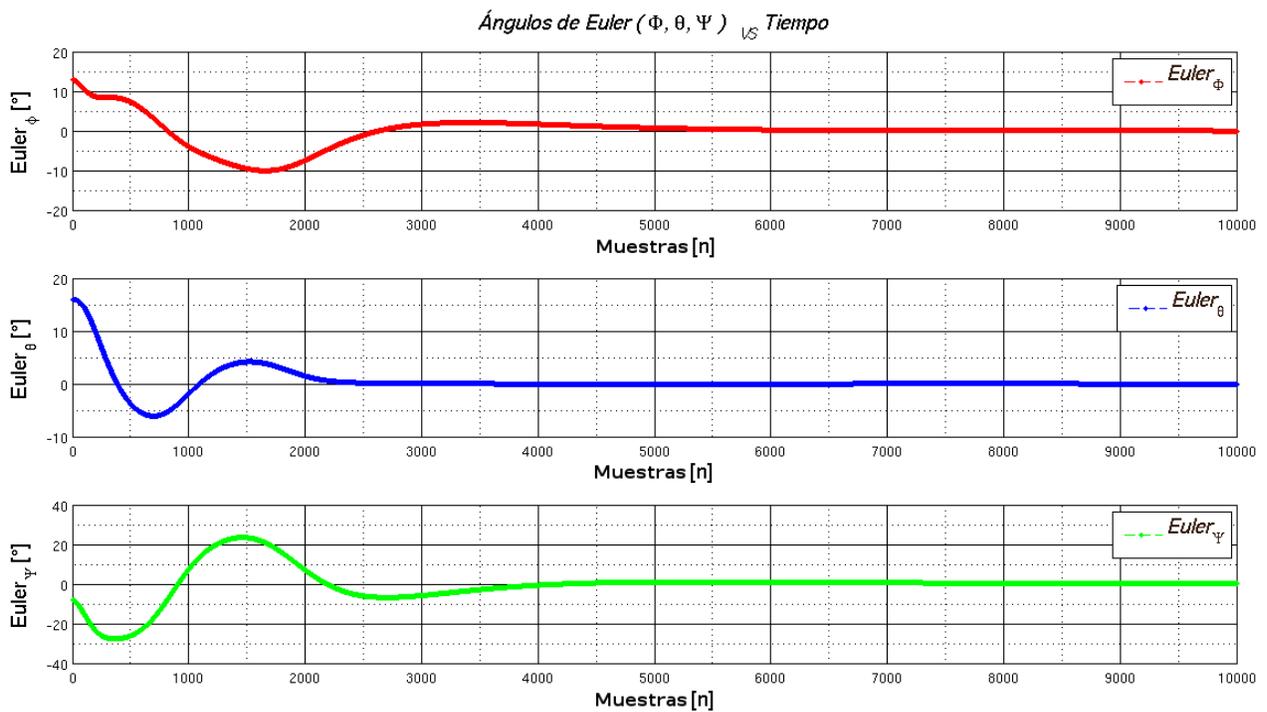


Figura 3.17: Gráfica de ângulos de Euler para valores iniciais $\phi = 12.825$, $\theta = 15.987$, $\psi = -8.197$ gerada por Matlab, por separado.

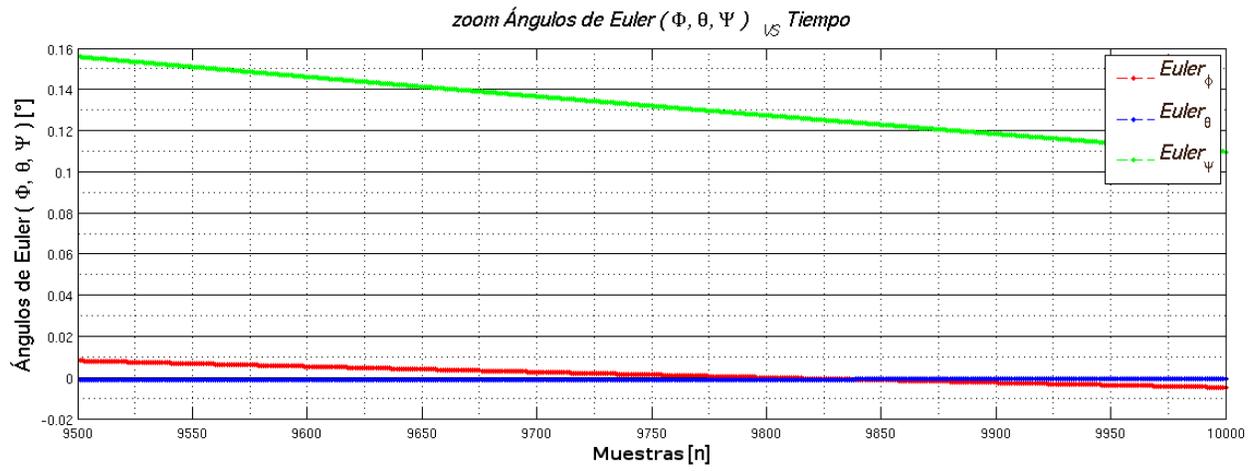


Figura 3.18: Gráfica de ángulos de Euler para valores iniciales de $\phi = 12.825$, $\theta = 15.987$, $\psi = -8.197$ generada por Matlab, zoom a los últimos 500 puntos de simulación.

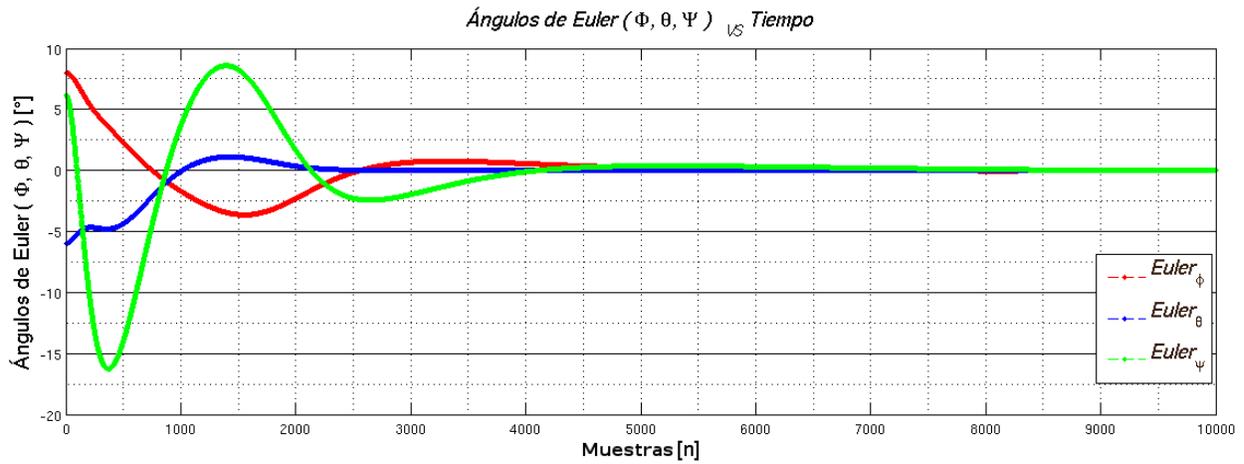


Figura 3.19: Gráfica de ângulos de Euler para valores iniciais $\phi = 7.99$, $\theta = -5.987$, $\psi = 6.197$ gerada por Matlab.

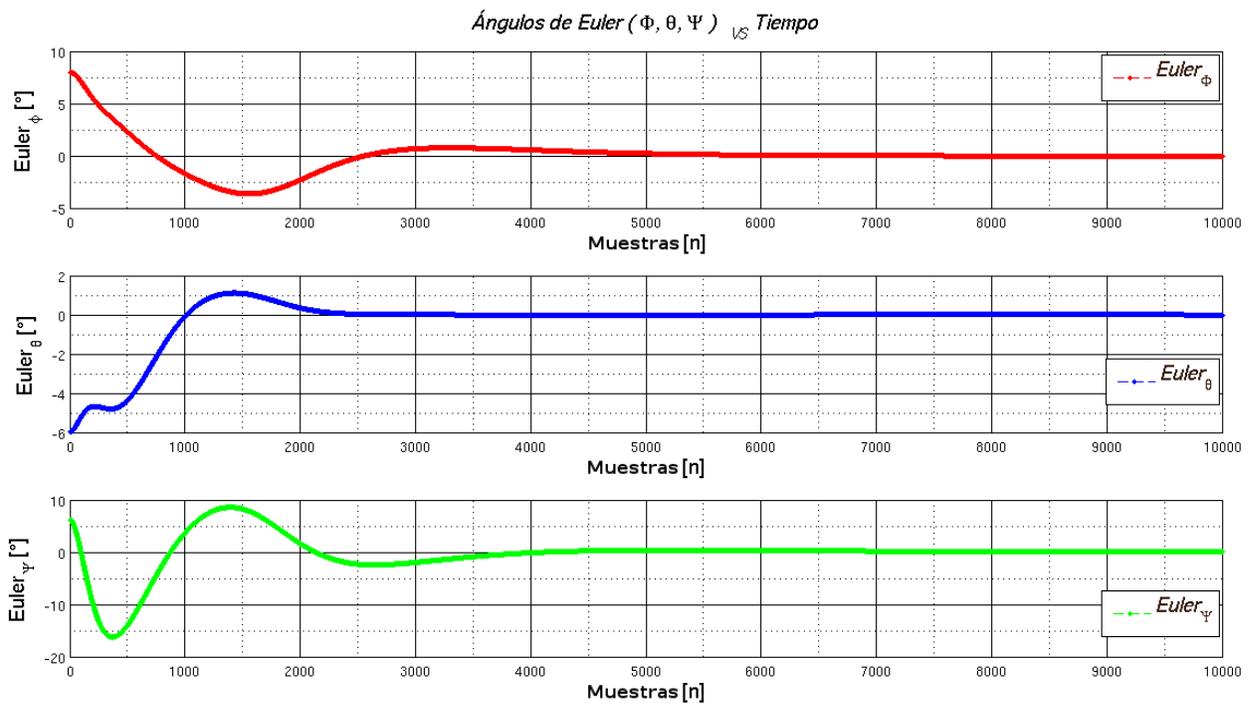


Figura 3.20: Gráfica de ângulos de Euler para valores iniciais $\phi = 7.99$, $\theta = -5.987$, $\psi = 6.197$ gerada por Matlab, por separado.

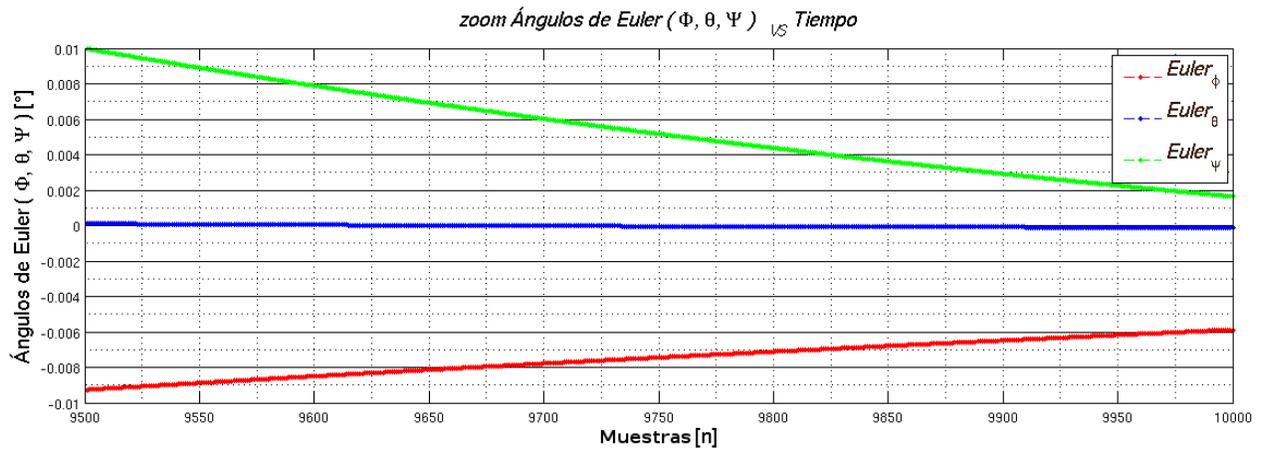


Figura 3.21: Gráfica de ángulos de Euler para valores iniciales de $\phi = 7.99$, $\theta = -5.987$, $\psi = 6.197$ generada por Matlab, zoom a los últimos 500 puntos de simulación.

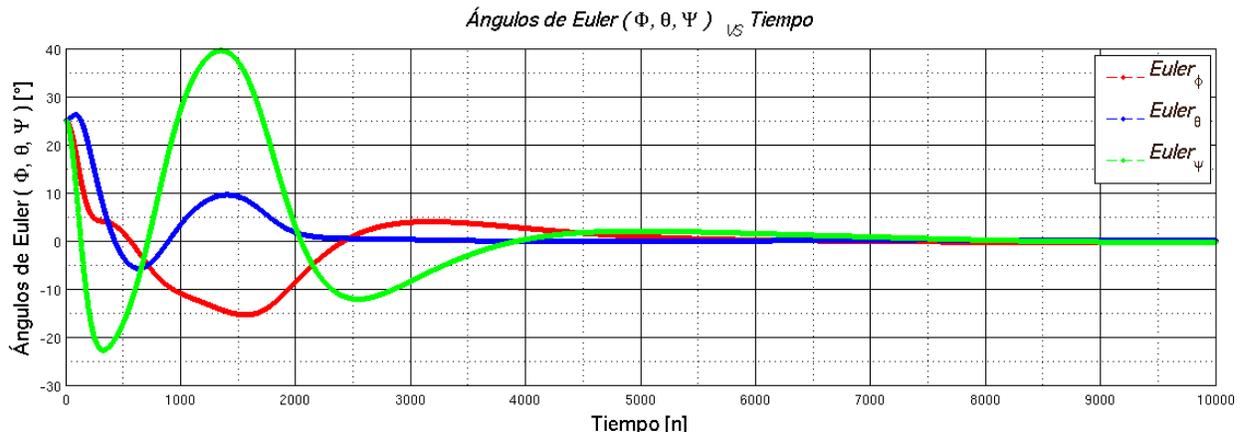


Figura 3.22: Gráfica de ángulos de Euler para valores iniciales $\phi = 25$, $\theta = 25$, $\psi = 25$ generada por Matlab.

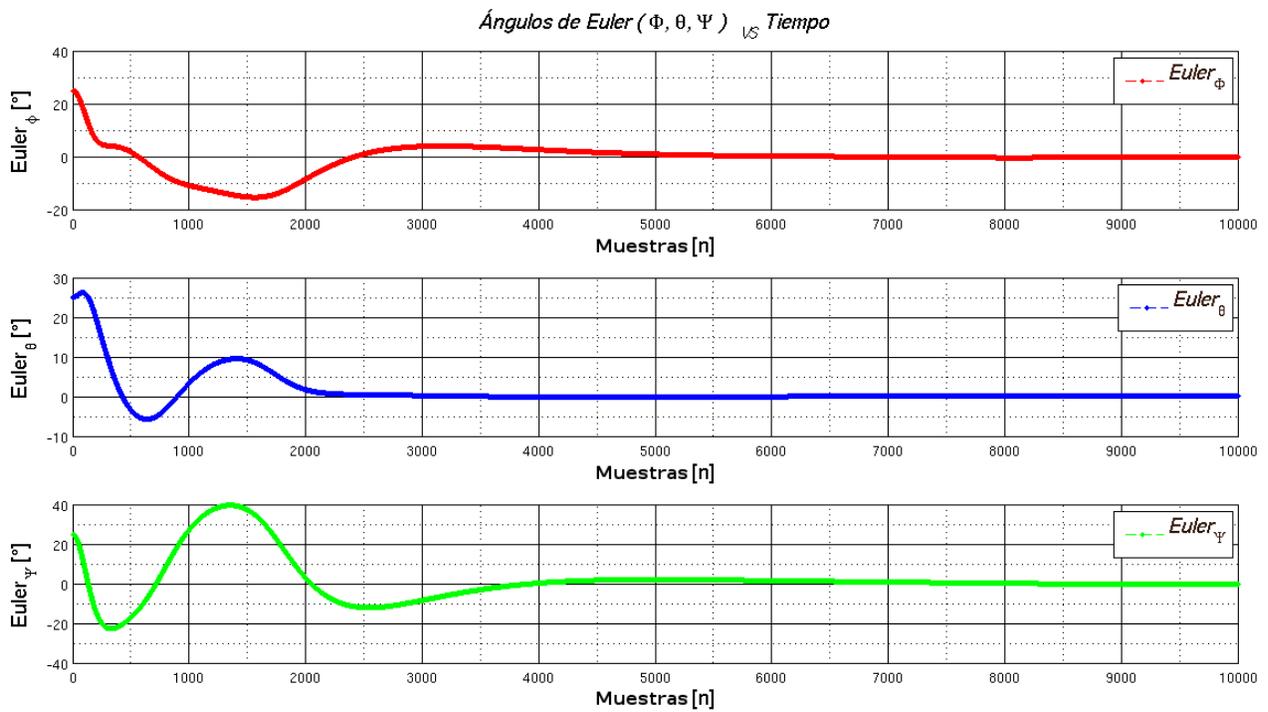


Figura 3.23: Gráfica de ángulos de Euler para valores iniciales $\phi = 25$, $\theta = 25$, $\psi = 25$ generada por Matlab, por separado.

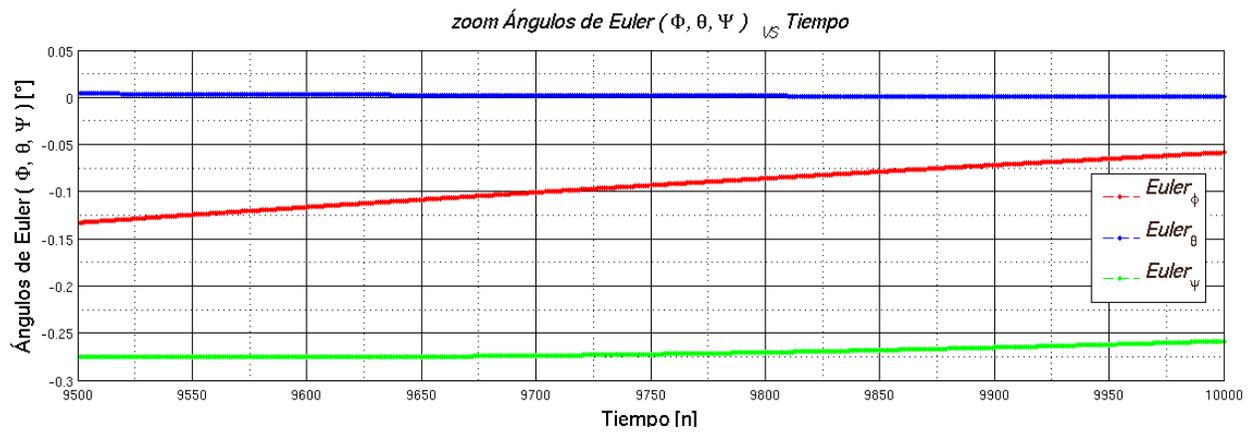


Figura 3.24: Gráfica de ángulos de Euler para valores iniciales de $\phi = 25$, $\theta = 25$, $\psi = 25$ generada por Matlab, zoom a los últimos 500 puntos de simulación.

Como se ve en las gráficas de las Figuras 3.13 a 3.24 el sistema se encuentra prácticamente estabilizado en $n = 5000$, dependiendo de los ángulos de Euler iniciales. En la primera imagen de la serie de gráficas presentadas para cada combinación de ángulos de Euler iniciales, se muestra una superposición de las gráficas correspondientes a cada ángulo de Euler. En la segunda imagen de la serie presentada, se muestran las gráficas para cada ángulo de Euler por separado. Finalmente, en la última imagen de la serie de gráficas presentadas se muestra un zoom a los últimos 500 puntos de la simulación donde se aprecia que el valor es prácticamente cero y tiende hacia éste.

El rendimiento en la respuesta del sistema mostrado en las gráficas de las Figuras 3.13 a 3.24 se logró gracias a la función lqr . Dicha función requieren las matrices A, B, Q y P en la forma $lqr(A, B, Q, P)$, donde Q y P son matrices simétricas positivas que definen el rendimiento del sistema penalizando la desviación que el sistema tiene respecto a los valores esperados.

Una primera selección de las matrices Q y P se puede hacer mediante la regla de Bryson [65], en la que se usa una matriz diagonal con valores $1/Q_{ii}$ y $1/P_{ii}$, donde Q_{ii} y P_{ii} son los valores máximos aceptables en la función de minimización. Aunque por lo general el uso de la regla de Bryson da buenos resultados, normalmente se emplea como un punto de inicio para posteriormente aplicar un método iterativo hasta lograr el rendimiento esperado en el sistema. Para la simulación se usaron los siguientes valores de matrices $QMatrix_{6 \times 6}$ y $PMatrix_{3 \times 3}$:

$$Qdes = (180/(14\pi))^2$$

$$Pdes = (1/2)^2$$

$$QMatrix_{6 \times 6} = diagonal[7Qdes, Qdes, Qdes, Qdes, Qdes, Qdes]$$

$$PMatrix_{3 \times 3} = diagonal[Pdes, Pdes, Pdes]$$

Estos valores fueron ajustados para que la respuesta deseada fuera la más rápida por lo que el sistema se logra estabilizar prácticamente en $n = 5000$. Estos valores fueron seleccionados únicamente con el propósito de simulación. Para el diseño de un ADCS final se tendrían que seleccionar las matrices $QMatrix_{6 \times 6}$ y $PMatrix_{3 \times 3}$ en función de los requerimientos, capacidades y objetivos de los diferentes modos de control a utilizar.

3.3.2. Metodología de implementación del algoritmo de Determinación de Posición y Control en XUPV2P

Una vez teniendo la simulación en Matlab se procedió a convertir estas funciones a C++ y analizó su resultado. Como se mencionó el controlador LQR hace uso de la función *lqr* del *Control System Toolbox*, esta función junto con *igrf11magm* del *Aerospace Toolbox* son los puntos fundamental para realizar el controlador, ya que determinan la ganancia de control óptima y el campo magnético requerido para estabilizar el satélite.

La función *igrf11magm* evalúa el valor de campo magnético para cualquier punto de la superficie terrestre, como se mencionó en el capítulo 2, el cálculo del valor de campo magnético es computacionalmente exigente y dado que la órbita por la que viaja el satélite es constante y completamente definida antes del vuelo, se pueden calcular los valores de campo magnético para la órbita y precargarlos al sistema [1]. Lo mismo ocurre con el cálculo de la ganancia óptima que es computacionalmente demandante y depende de los valores constantes del sistema, y los valores variantes del vector de campo magnético, por lo que la ganancia depende del punto de la órbita en el que se encuentre el satélite. Una vez calculada la ganancia y el vector de estados se genera el par de control.

Teniendo en cuenta esto, utilicé la herramienta *cftool* de *Curve Fitting Toolbox* para generar polinomios que representen las curvas correspondientes a los valores de campo magnético y de ganancia para la órbita definida. Esta herramienta cuenta con diferentes tipos de aproximación como son exponencial, Fourier, Gauss, Interpolación, Polinomial, Suma de Potencias, Suma de Senos entre otras. Cada tipo de polinomio genera una mejor o peor aproximación para determinado tipo de gráfica. Se puede lograr una muy buena aproximación usando, por ejemplo, un polinomio de Gauss de grado 3 que usando uno de tipo Fourier de grado 8 para cierta gráfica, esto genera un número menor de coeficientes a almacenar.

La herramienta *cftool* genera los coeficientes del polinomio correspondiente a una curva previamente calculada, la información de estos modelos precargados se puede combinar con información proveniente de sensores magnéticos y GPS para lograr una determinación óptima, como se menciona en Wertz et al. [1]. Las funciones se pueden separar en secciones y hacer una combinación de diferentes polinomios de diferentes grados para generar la mejor aproximación posible. Para las curvas que representan el campo magnético y la matriz de ganancia las mejores aproximaciones son Fourier, Gauss y Suma de senos.

Los siguientes son los polinomios utilizados con el formato de los vectores de coeficientes generados para la implementación del modelo de campo magnético y ganancia de control:

Polinomio *Fourier* y formato de su vector de coeficientes correspondientes:

$$Fourier^n(\chi) = a_0 + \sum_{\vartheta=1}^n a_{\vartheta} \cos(\vartheta\chi\omega) + b_{\vartheta} \sin(\vartheta\chi\omega) \quad (3.1)$$

$$Fourier_coefvec^n = \{a_0, a_1, b_1, a_2, b_2, \dots, a_n, b_n, \omega\} \quad (3.2)$$

Polinomio *Gauss* y formato de su vector de coeficientes correspondientes:

$$Gauss^n(\chi) = \sum_{\vartheta=1}^n a_{\vartheta} e^{-\left(\frac{\chi - b_{\vartheta}}{c_{\vartheta}}\right)^2} \quad (3.3)$$

$$Gauss_coefvec^n = \{a_1, b_1, a_2, b_2, \dots, a_n, b_n\} \quad (3.4)$$

Polinomio *SineSum* y formato de su vector de coeficientes correspondientes:

$$SineSum^n(\chi) = \sum_{\vartheta=1}^n a_{\vartheta} \sin(b_{\vartheta}\chi + c_{\vartheta}) \quad (3.5)$$

$$SineSum_coefvec^n = \{a_1, b_1, c_1, a_2, b_2, c_2, \dots, a_n, b_n, c_n\} \quad (3.6)$$

Donde n representa el grado de cada polinomio.

En la Figura 3.25 se muestran las gráficas de cada componente del campo magnético para la órbita definida. Las curvas generadas por la función *igrf11magm* representan el campo magnético para la órbita del satélite y la mejor aproximación se logró usando combinaciones de polinomios de tipo *Fourier* de diferentes grados.

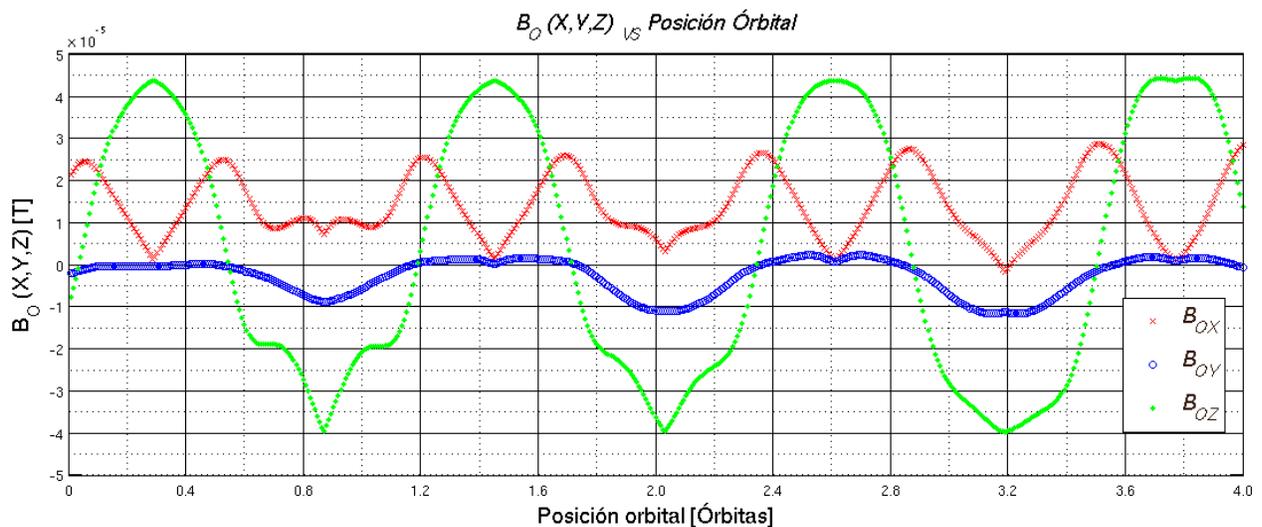


Figura 3.25: Gráfica de los valores de campo magnético para la órbita seleccionada.

Para un tiempo igual a $t = 10000$ en el que el satélite se encuentra completamente estabilizado el campo magnético se puede describir con los siguientes polinomios:

$$bo_x = \left\{ \begin{array}{ll} Fourier^2 & \text{para } t < 7500 \text{ con } bo_xpvec \\ Fourier^2 & \text{para } t \geq 7501 \text{ con } bo_xsvec \end{array} \right\} \quad (3.7)$$

$$bo_y = Fourier^8 \quad \text{con } bo_yvec \quad (3.8)$$

$$bo_z = Fourier^8 \quad \text{con } bo_zvec \quad (3.9)$$

Donde $Fourier^n$ representa el polinomio de Fourier de grado n , bo_xpvec , bo_xsvec , bo_yvec , y bo_zvec representan los vectores de coeficientes correspondientes a cada polinomio. El componente de campo magnético se divide en dos polinomios de segundo orden porque así se logra una mejor aproximación a su curva que teniendo un polinomio octavo grado, siendo éste el máximo grado que se logra en las aproximaciones usando *cftool*. Propiamente la información que se almacena son vectores de coeficientes, que son los argumentos a una función *polynomialF* que genera el valor del polinomio requerido Gauss, Fourier o SineSum.

Continuando con el modelo de campo magnético, se tienen los siguientes vectores de coeficientes:

$$bo_xpvec = \{2.358e^{-6}, 2.9e^{-6}, 3.871e^{-6}, 9e^{-7}, 2.055e^{-7}, 0.004334\}$$

$$bo_xsvec = \{2.199e^{-6}, 6.783e^{-7}, -1.671e^{-6}, 2.79e^{-7}, -2.481e^{-7}, 0.001143\}$$

$$bo_yvec = \{5.189^{-7}, 4.247^{-7}, -1.227^{-6}, -8.409^{-7}, -9.112^{-7}, -9.675^{-7}, \\ 1.464^{-7}, -2.52^{-7}, 6.033^{-7}, 2.391^{-7}, 3.004^{-7}, 1.829^{-7}, -2.947^{-8}, \\ 2.454^{-8}, -7.037^{-8}, -1.536^{-8}, -1.151^{-8}, 0.0004189\}$$

$$bo_zvec = \{-0.000609, 7.714^{-5}, 0.001139, 0.0008859, -0.0001319, -0.0001348, -0.0005933, \\ -0.0003174, 0.0001312, .5.803^{-5}, 0.0001381, 4.558^{-5}, -2.479^{-5}, \\ -7.298^{-6}, -1.021^{-5}, -1.135^{-6}, 1.14^{-6}, 0.0003142\}$$

Es claro que el número de coeficientes de cada vector está determinado por el grado del polinomio, para el polinomio de Fourier se tiene $l = 2(n + 1)$ siendo l la longitud del vector y n el orden del polinomio usado para la aproximación. Algo muy importante que debe ser considerado cuando se utiliza este método es el correcto manejo de los coeficientes ya que un signo o valor equivocado generaría una función que no estaría representando el valor esperado.

Ya se ha mencionado que para controlar el sistema se usó la función *lqr* del *Control System Toolbox* de Matlab, dicha función evalúa la matriz de ganancia óptima K con la cual se calcula el par de control del sistema y tiene la siguiente forma:

$$Kmatrix_{3 \times 6} = \begin{pmatrix} km_{11} & km_{12} & km_{13} & km_{14} & km_{15} & km_{16} \\ km_{21} & km_{22} & km_{23} & km_{24} & km_{25} & km_{26} \\ km_{31} & km_{32} & km_{33} & km_{34} & km_{35} & km_{36} \end{pmatrix} \quad (3.10)$$

Los valores para cada punto de la órbita definida fueron graficados y se siguió el mismo procedimiento usado para determinar el modelo de campo magnético. Se generaron polinomios para cada elemento de $Kmatrix_{3 \times 6}$ representando los valores a lo largo de la órbita definida. Todos los elementos km_{nm} están conformados por una combinación de polinomios Gauss, Fourier o SineSum.

Para el elemento km_{11} de la matriz $Kmatrix_{3 \times 6}$, se tiene la siguiente combinación:

$$km_{11} = \left\{ \begin{array}{ll} Fourier^5 & \text{para } t < 7500 \text{ con } km_{11}pvec \\ Fourier^6 & \text{para } t \geq 7501 \text{ con } km_{11}svec \end{array} \right\} \quad (3.11)$$

Con los vectores de coeficientes correspondientes:

$$\begin{aligned} km_{11}pvec &= \{-1.2, -0.5832, 1.005, 0.7213, 0.3782, 0.7175, -0.3027, 0.1892, \\ &\quad -0.3353, -0.01137, -0.09207, 0.0006311\} \\ km_{11}svec &= \{11.52, 13.36, 17.32, -5.268, 15.72, -9.432, 2.433, -3.161, \\ &\quad -2.879, 0.2084, -1.284, 0.1958, -0.08506, 0.001198\} \end{aligned}$$

El mismo procedimiento se realizó para todos los coeficientes de $Kmatrix_{3 \times 6}$. Como se mencionó en el capítulo 2, la señal de control se calcula como:

$$m_{3 \times 1} = -Kmatrix_{3 \times 6} \times Xvector_{6 \times 1}^T \quad (3.12)$$

Donde $Xvector_{6 \times 1}$ es el vector de estados del sistema que depende de la posición instantánea del satélite.

La implementación consta de las siguientes funciones: *dynamics*, *kinematics*, *statevector*, *envModel*, *lqr*, *VecCross* y *q2e*. A excepción de las funciones *lqr* y *envModel*, todas las demás

funciones son iguales a las utilizadas en la simulación de Matlab. La Figura 3.26 de la página 107 se muestra un diagrama de flujo del sistema implementado en XUPV2P. Las funciones *lqr* y *envModel* hacen uso de la función *polynomialF* que genera el valor del polinomio de Fourier, Gauss o SineSum, teniendo como argumentos el vector de coeficientes, el orden del polinomio y el tiempo. Una vez que el sistema inicia envía sus resultados mediante comunicación serial UART a una computadora donde se almacenan en un archivo con extensión *.dat* para posterior generar la graficar de correspondiente.

Una vez que el sistema inicia envía sus resultados mediante comunicación serial UART a una computadora donde se almacenan en un archivo con extensión *.dat* para posteriormente generar la grafica correspondiente.

Después de que los datos son transmitidos a la computadora y almacenados en el archivo con extensión *.dat* son graficados mediante la función *plotData.m* implementada usando Matlab. En las Figuras 3.27 a 3.38 se muestran las gráficas de los valores de ángulos de Euler generadas por XUPV2P para diferentes valores iniciales.

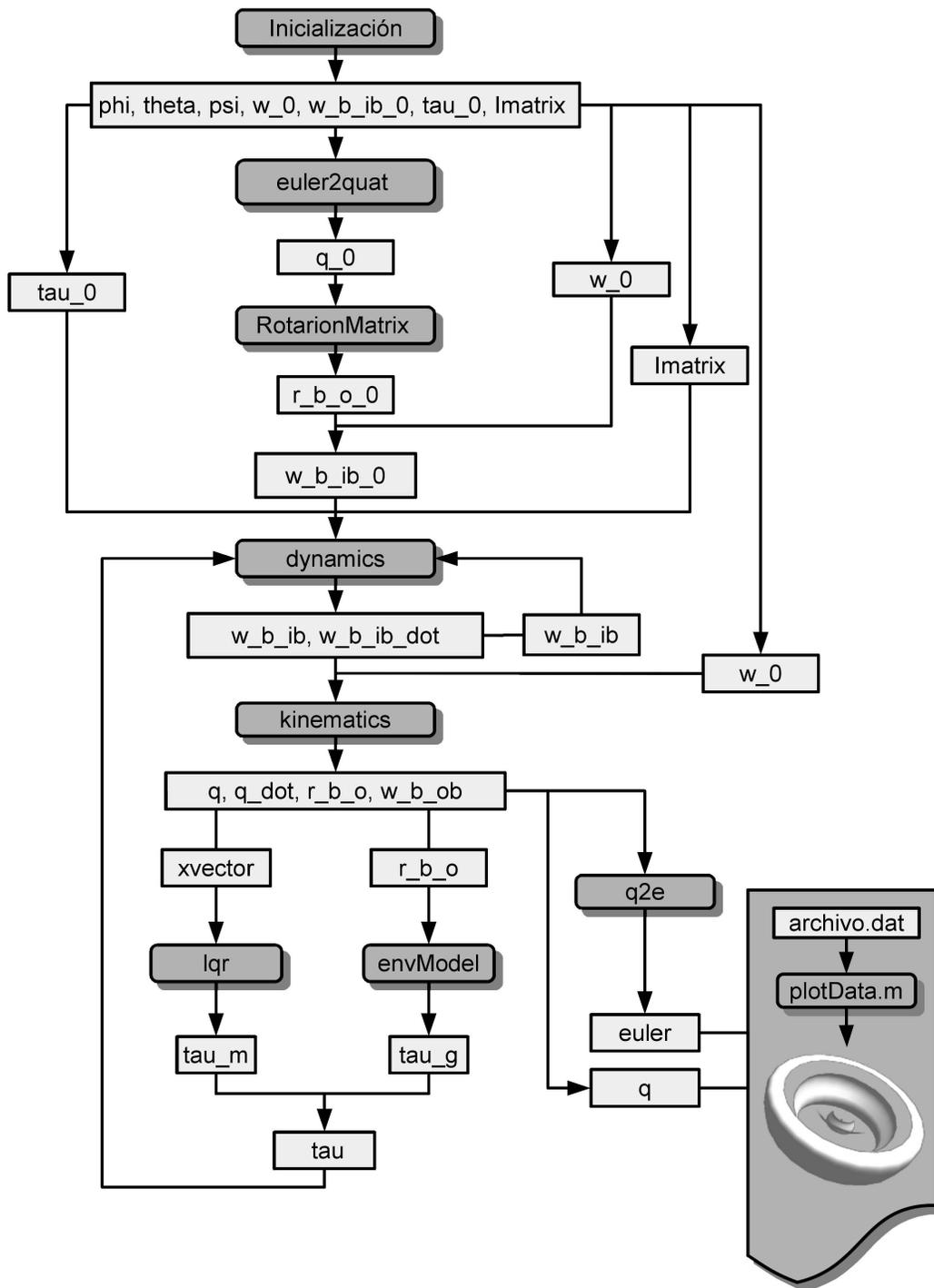


Figura 3.26: Diagrama del algoritmo implementado en XUPV2P.

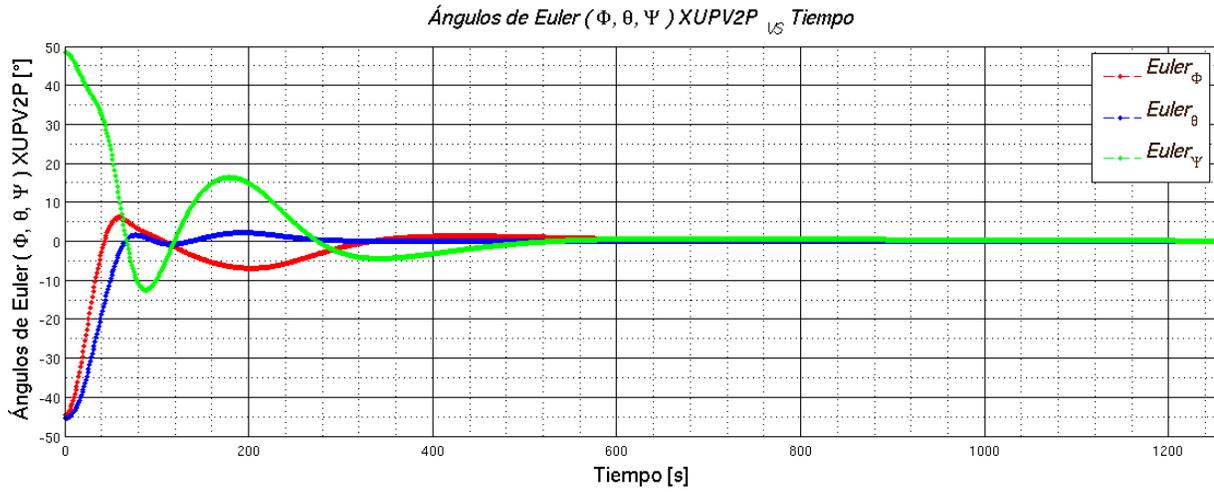


Figura 3.27: Gráfica de ângulos de Euler para valores iniciais $\phi = -45.125$, $\theta = -45.987$, $\psi = 48.978$ gerada por XUPV2P.

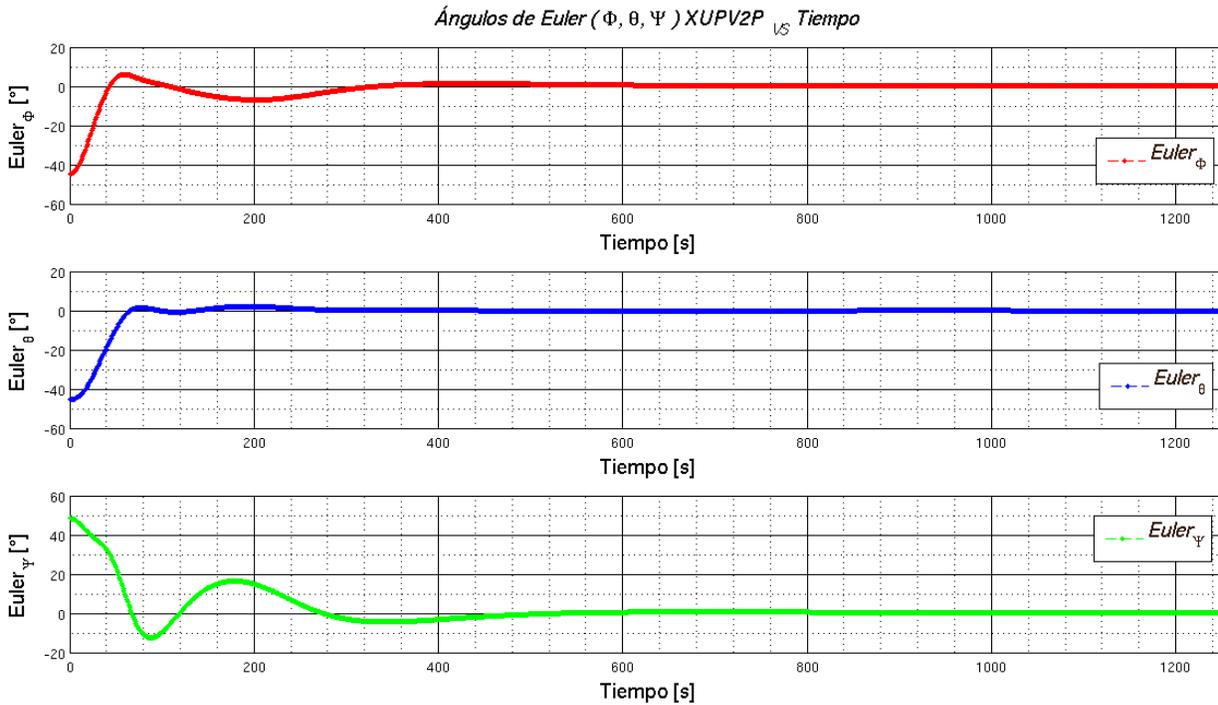


Figura 3.28: Gráfica de ângulos de Euler para valores iniciais $\phi = -45.125$, $\theta = -45.987$, $\psi = 48.978$ gerada por XUPV2P, por separado.

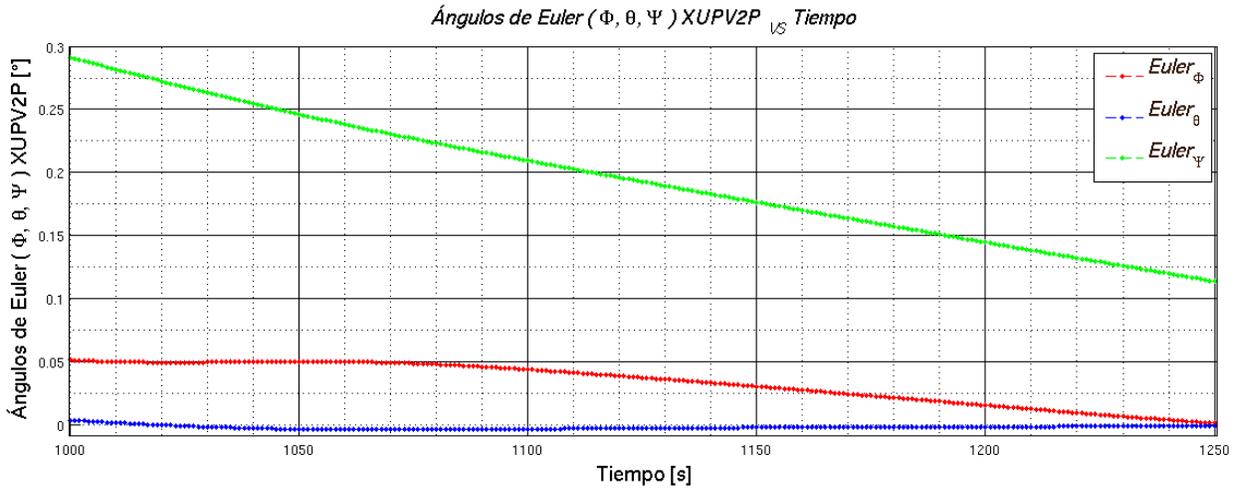


Figura 3.29: Gráfica de ángulos de Euler para valores iniciales de $\phi = -45.125$, $\theta = -45.987$, $\psi = 48.978$ generada por XUPV2P, zoom a los últimos 250 segundos.

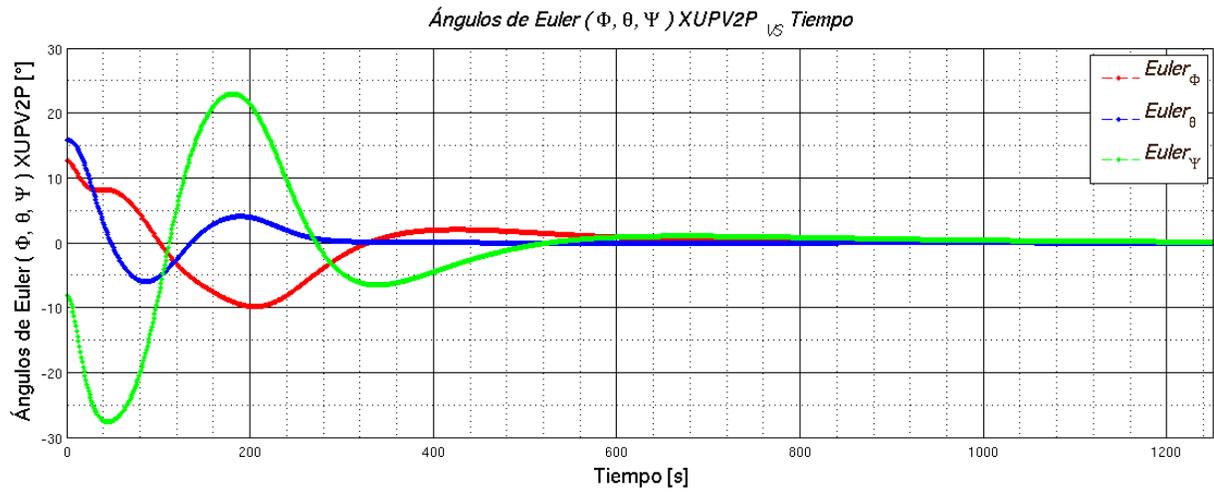


Figura 3.30: Gráfica de ângulos de Euler para valores iniciais $\phi = 12.825$, $\theta = 15.987$, $\psi = -8.197$ generada por XUPV2P.

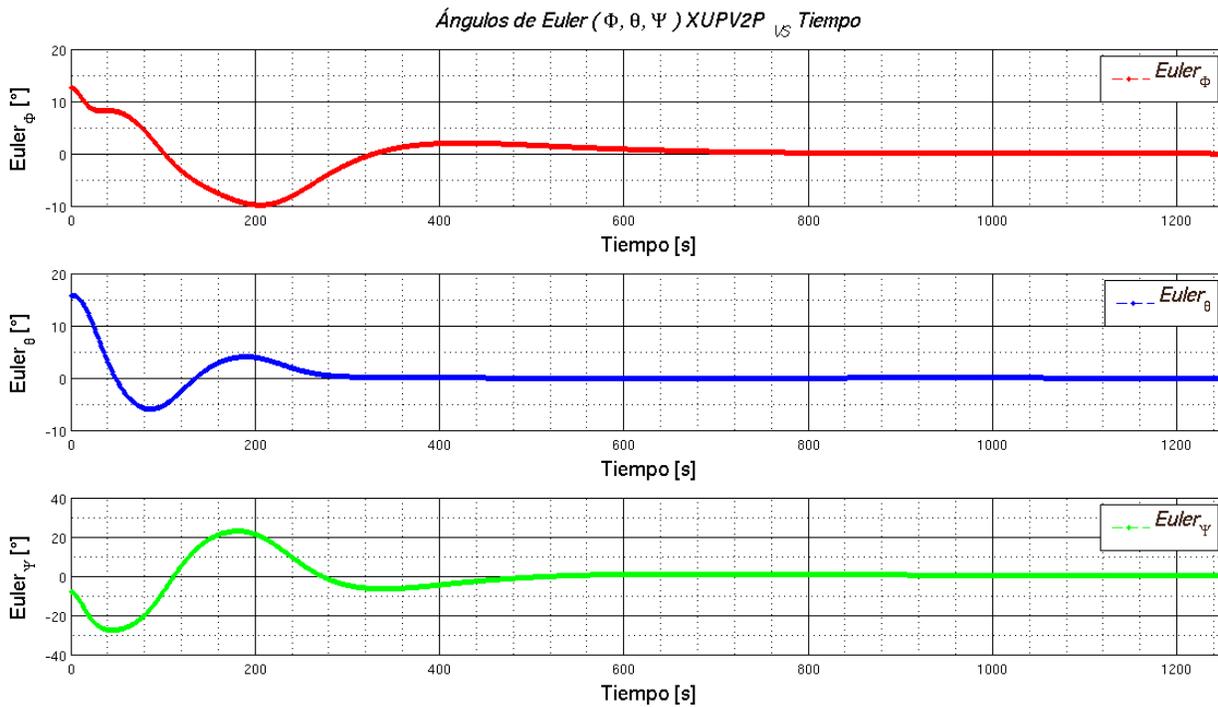


Figura 3.31: Gráfica de ângulos de Euler para valores iniciais $\phi = 12.825$, $\theta = 15.987$, $\psi = -8.197$ generada por XUPV2P, por separado.

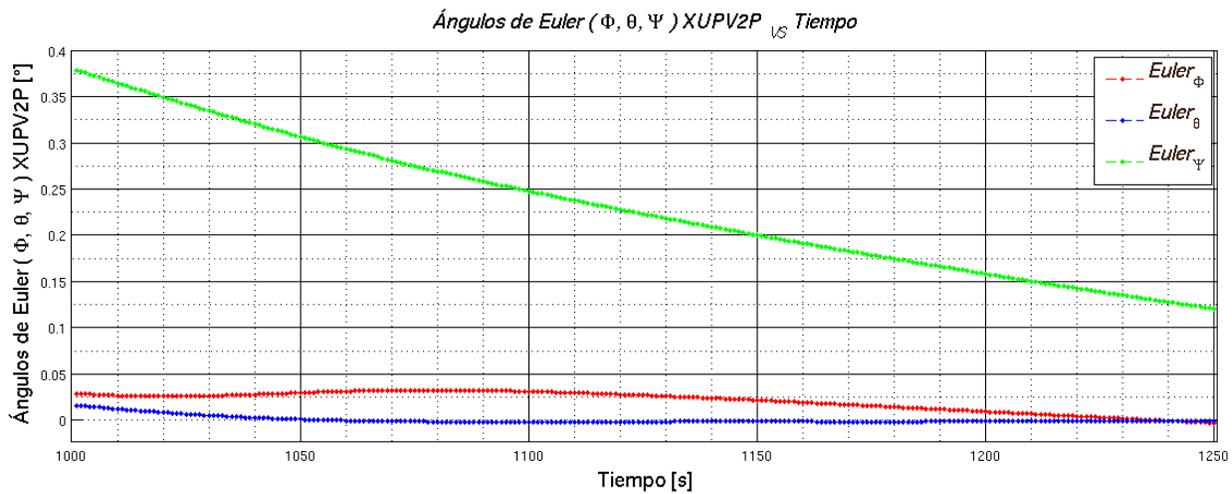


Figura 3.32: Gráfica de ángulos de Euler para valores iniciales de $\phi = 12.825$, $\theta = 15.987$, $\psi = -8.197$ generada por XUPV2P, zoom a los últimos 250 segundos.

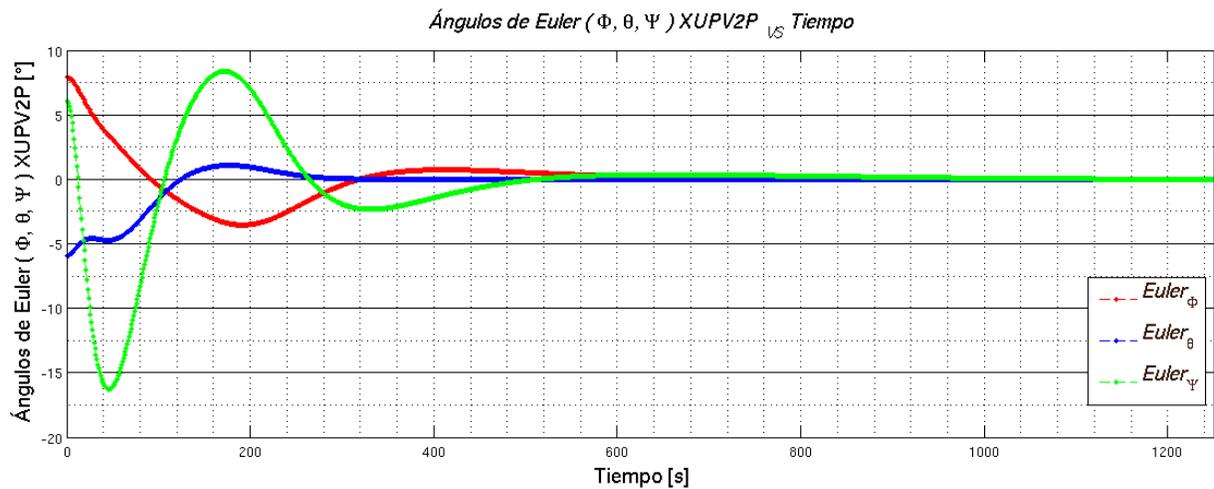


Figura 3.33: Gráfica de ângulos de Euler para valores iniciais $\phi = 7.99$, $\theta = -5.987$, $\psi = 6.197$ gerada por XUPV2P.

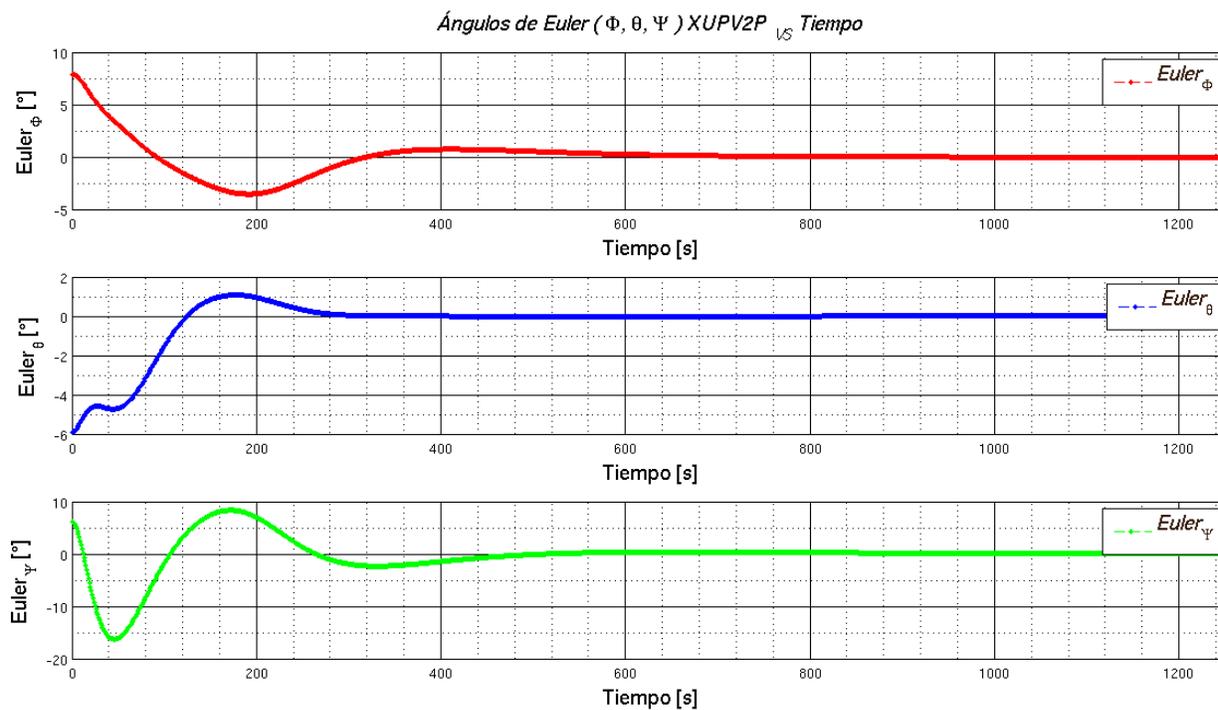


Figura 3.34: Gráfica de ângulos de Euler para valores iniciais $\phi = 7.99$, $\theta = -5.987$, $\psi = 6.197$ gerada por XUPV2P, por separado.

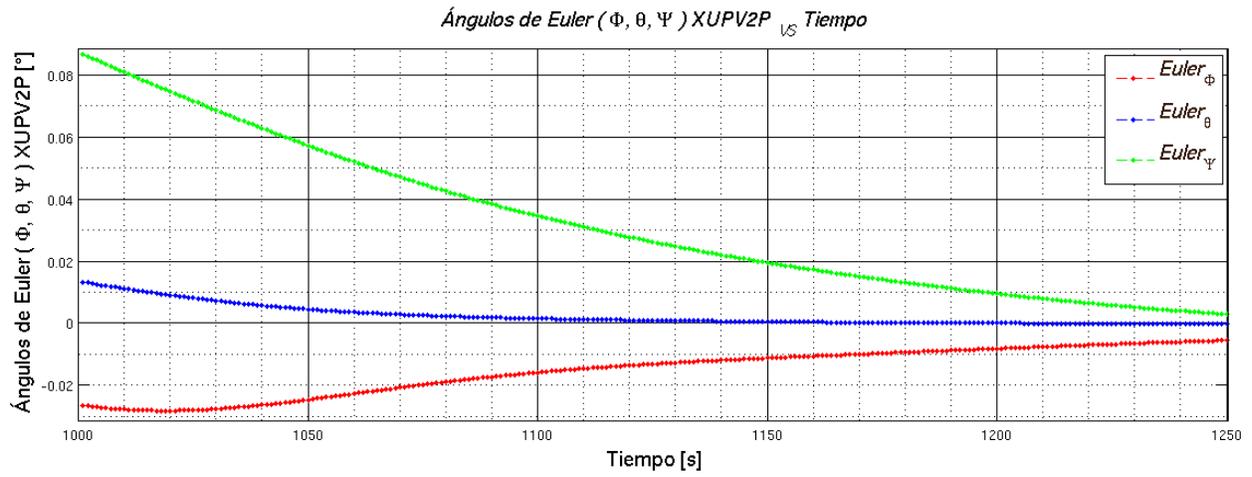


Figura 3.35: Gráfica de ángulos de Euler para valores iniciales de $\phi = 7.99$, $\theta = -5.987$, $\psi = 6.197$ generada por XUPV2P, zoom a los últimos 250 segundos.

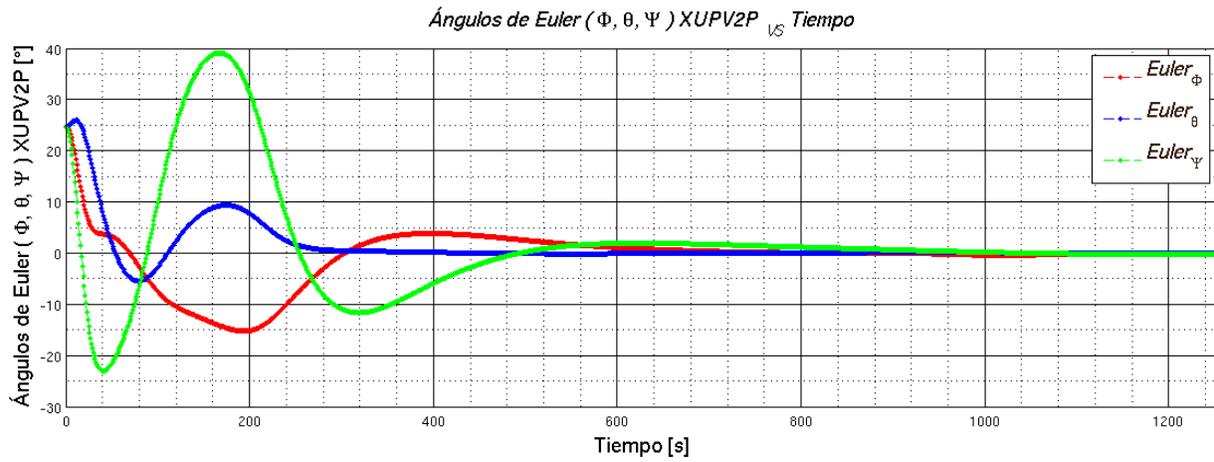


Figura 3.36: Gráfica de ângulos de Euler para valores iniciais $\phi = 25$, $\theta = 25$, $\psi = 25$ gerada por XUPV2P.

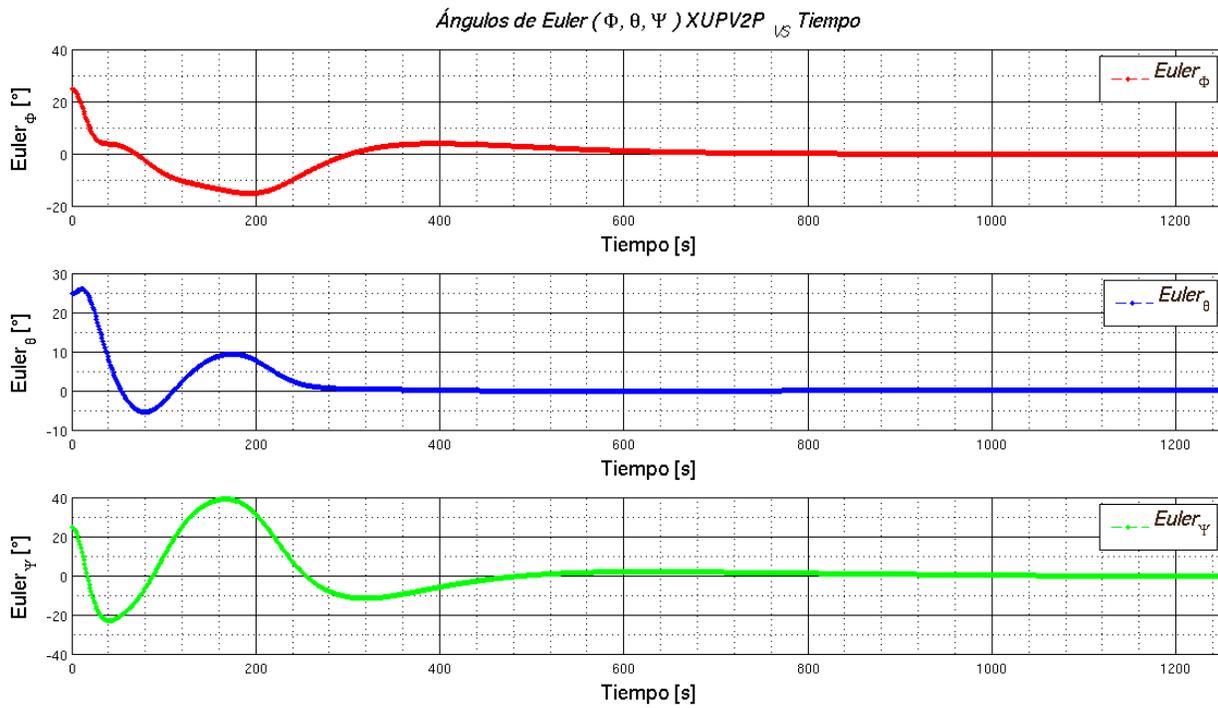


Figura 3.37: Gráfica de ângulos de Euler para valores iniciais $\phi = 25$, $\theta = 25$, $\psi = 25$ gerada por XUPV2P, por separado.

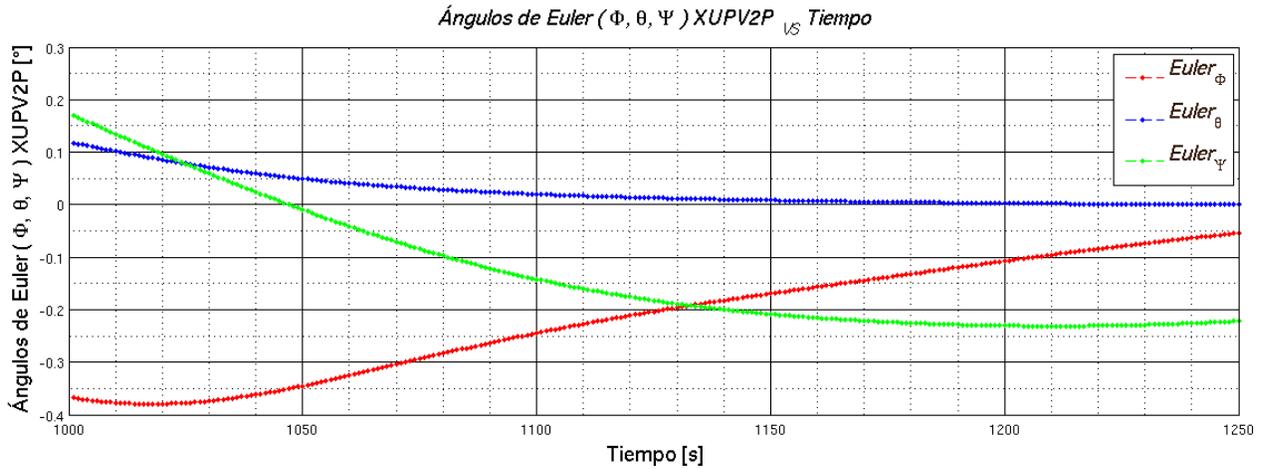


Figura 3.38: Gráfica de ángulos de Euler para valores iniciales de $\phi = 25$, $\theta = 25$, $\psi = 25$ generada por XUPV2P, zoom a los últimos 250 segundos.

La frecuencia en la obtención de datos por XUPV2P es de 8 Hz, valor mayor a los 5 Hz recomendados para un ADCS en Wertz et al. [1]. Si se desea implementar este sistema lo que involucraría el manejo de sensores y actuadores se podría fácilmente mantener un valor de frecuencia de funcionamiento cercano a 8 Hz, teniendo 3 Hz como rango disponible para el procesamiento de información de sensores y activación de actuadores. En las Figuras 3.27 a 3.38 en las que se muestran las gráficas de Ángulos de Euler generadas por la implementación en XUPV2P. Para genera estas gráficas se usaron 10000 puntos para poder compararlos con la misma cantidad de puntos en la simulación. Estos datos muestreados a 8Hz resultaron en 1250 segundos de datos almacenados de XUPV2P. Considerando $F_S = 8Hz$ como la frecuencia de muestreo de los datos, se tiene un periodo de muestreo $T_S = 125ms$, por lo que en la comparación entre los datos de Matlab y XUPV2P se hace considerando el tiempo discreto nT_S como punto de comparación.

3.4. Comparación de resultados entre implementación en XUPV2P y simulación numérica

Para evaluar los resultados obtenidos entre la simulación en Matlab y la implementación en XUPV2P, y como medida de evaluación del desempeño de la implementación se realizaron comparaciones de gráficas, de valores de error absoluto promedio y de grado de correlación entre los datos obtenidos por ambas fuentes (Matlab y XUPV2P). Las variables a comparar son los valores de ángulos de Euler (ϕ, θ, ψ) y de cuaterniones (q_0, q_1, q_2, q_3) unitarios. Esto debido a que, por un lado, los cuaterniones unitarios representan internamente la posición del satélite y son los valores con los que opera propiamente el algoritmo como se expuso en el capítulo 2. Por otro lado, aunque los ángulos de Euler no son empleados en el cálculo interno de posición, sí son la manera más sencilla de visualizar la posición del satélite y con la que cualquier lector estará más familiarizado.

A modo de evaluar los resultados se generaron y almacenaron tramas de datos para los mismos valores iniciales de ángulos de Euler generados por la simulación y por la implementación. Se seleccionaron 10000 puntos de simulación e implementación debido a que dadas las condiciones del sistema y dependiendo los valores de ángulos iniciales, el satélite se encontraba prácticamente estabilizado en $(\phi = 0, \theta = 0, \psi = 0)$ para 5000 puntos, por tanto al tomar 10000 puntos la estabilización del sistema está garantizada.

Para cada simulación se generaron tres gráficas para los valores de ángulos de Euler, la primera mostrando los tres ángulos, la segunda mostrando los ángulos por separado y la tercera mostrando nuevamente los tres ángulos con un zoom a los últimos 500 puntos de la simulación para mostrar con detalle los valores finales de la misma. Se generaron tres gráficas con las mismas características para los datos generados por la implementación.

El cálculo del error absoluto promedio y de grado de correlación se realizó tanto para los valores de ángulos de Euler (ϕ, θ, ψ) , como para los de Cuaterniones Unitarios (q_1, q_2, q_3, q_4) . A continuación se definen el error absoluto promedio y coeficiente de correlación para posteriormente mostrar los resultados para los diferentes valores iniciales de ángulos de Euler empleados. Estos resultados se muestran en las Tablas 3.3 a 3.6.

El error absoluto promedio (MAE, por sus siglas en inglés) es una cantidad que indica la proximidad de valores que existe entre dos series de datos y se define de la siguiente manera:

$$MAE = \frac{1}{n} \sum_{i=1}^n |s_i - r_i| \quad (3.13)$$

Donde s_i es el valor obtenido de la simulación y r_i el valor real obtenido en la implementación.

El coeficiente de correlación es una cantidad que indica el grado de dependencia entre dos tramas de datos. El valor del coeficiente de correlación oscila entre -1 y 1, siendo el valor de 1 representativo de dos gráficas idénticas. El tipo de correlación que se usó fue el de Pearson por simplicidad y está definida de la siguiente manera:

$$Corr_{XY} = \frac{cov(X, Y)}{\sigma_X \sigma_Y} \quad (3.14)$$

Donde $cov(X, Y)$ es la covariancia entre X, Y y $\sigma_X \sigma_Y$ la desviación estándar de X, Y respectivamente.

En siguientes tablas se muestran los resultados del cálculo de Error Absoluto Promedio y Correlación entre los datos obtenidos por la simulación en Matlab y XUPV2P para diferentes valores de Ángulos de Euler iniciales.

	Euler			Cuaterniones			
	Φ	Θ	Ψ	q_1	q_2	q_3	q_4
MAE	$8.4497e^{-2}$	$2.8707e^{-2}$	$1.5122e^{-1}$	$1.0592e^{-4}$	$7.3056e^{-4}$	$3.0816e^{-4}$	$1.2898e^{-3}$
Corr	$9.9965e^{-1}$	$9.9994e^{-1}$	$9.9918e^{-1}$	$9.9992e^{-1}$	$9.9894e^{-1}$	$9.9993e^{-1}$	$9.9879e^{-1}$

Tabla 3.3: Comparativa entre valores de error absoluto promedio y correlación para valores iniciales de $\phi = -45.125$, $\theta = -45.987$, $\psi = 48.978$.

	Euler			Cuaterniones			
	Φ	Θ	Ψ	q_1	q_2	q_3	q_4
MAE	$3.9084e^{-2}$	$1.9248e^{-2}$	$1.3343e^{-1}$	$1.5339e^{-4}$	$3.6052e^{-4}$	$1.1560e^{-4}$	$1.2352e^{-3}$
Corr	$9.9990e^{-1}$	$9.9987e^{-1}$	$9.9956e^{-1}$	$9.9929e^{-1}$	$9.9890e^{-1}$	$9.9992e^{-1}$	$9.9849e^{-1}$

Tabla 3.4: Comparativa entre valores de error absoluto promedio y correlación para valores iniciales de $\phi = 12.825$, $\theta = 15.987$, $\psi = -8.197$.

	Euler			Cuaterniones			
	Φ	Θ	Ψ	q_1	q_2	q_3	q_4
MAE	$2.5638e^{-2}$	$9.3721e^{-3}$	$7.7605e^{-2}$	$3.6626e^{-5}$	$2.3646e^{-4}$	$8.4053e^{-5}$	$7.1439e^{-4}$
Corr	$9.9965e^{-1}$	$9.9994e^{-1}$	$9.9918e^{-1}$	$9.9992e^{-1}$	$9.9894e^{-1}$	$9.9993e^{-1}$	$9.9879e^{-1}$

Tabla 3.5: Comparativa entre valores de error absoluto promedio y correlación para valores iniciales de $\phi = 7.99$, $\theta = -5.987$, $\psi = 6.197$.

	Euler			Cuaterniones			
	Φ	Θ	Ψ	q_1	q_2	q_3	q_4
MAE	$6.6011e^{-2}$	$4.2563e^{-2}$	$2.1161e^{-1}$	$3.1319e^{-4}$	$6.2686e^{-4}$	$2.1961e^{-4}$	$1.9438e^{-3}$
Corr	$9.9984e^{-1}$	$9.9978e^{-1}$	$9.9931e^{-1}$	$9.9886e^{-1}$	$9.9982e^{-1}$	$9.9990e^{-1}$	$9.9822e^{-1}$

Tabla 3.6: Comparativa entre valores de error absoluto promedio y correlación para valores iniciales de $\phi = 25$, $\theta = 25$, $\psi = 25$.

Como se puede ver en los datos de las Tablas 3.3 a 3.6 los valores de MAE entre la simulación y la implementación para los valores de Cuaterniones Unitarios tienen un rango que va desde $3.6626e^{-5}$ hasta $1.9438e^{-3}$ lo que significa que la variación es insignificante. Por otro lado los valores de ángulos de Euler no supera $1.6162e^{-1}$ grados y tiene un valor más bajo de $9.3721e^{-3}$ grados. Esto indica que existe menos de un grado de diferencia en promedio entre los datos obtenidos en la simulación y la implementación.

Los resultados obtenidos en la correlación que se observan en las Tablas 3.3 a 3.6 son superiores a 0.99 para todas las comparaciones lo que indica que la utilización de polinomios de Fourier y Gauss como el método de obtención del modelo de campo magnético y de la ganancia de control son muy precisos. Bajo estos resultados se podrían emplear polinomios de aproximación de menor grado con lo que se disminuiría el número de coeficientes y operaciones necesarias para el cálculo de la ganancia y modelo de campo magnético.

En la Figura 3.39 se muestran las gráficas de comparación entre Ángulos de Euler generados por Matlab y por XUPV2P. En las gráficas mostradas del lado izquierdo se superponen los valores de cada ángulo y en las del lado derecho se hace un zoom a los últimos 500 puntos de comparación en el que se aprecia la diferencia mínima que existe entre los resultados de una manera más detallada.

En la Figura 3.40 se muestran las gráficas de comparación entre Ángulos de Euler generados nuevamente por Matlab y XUPV2P en una escala logarítmica con el afán de resaltar las diferencias que existen entre las gráficas.

En la Figura 3.42 se muestran el mismo tipo de gráficas que en la Figura 3.40 pero para los valores de Cuaterniones Unitarios, se pueden apreciar las mínimas diferencias que existen entre los valores obtenidos por Matlab y XUPV2P.

Las Figuras 3.41 y 3.43 muestran básicamente las magnitudes del error entre los datos generados por Matlab y XUPV2P para Ángulos de Euler y Cuaterniones Unitarios respectivamente. En estas gráficas se puede ver que el MAE crece, alcanza un máximo de 1.5 grados aproximadamente para los Ángulos de Euler, y decae a valores menores a $1.0e^{-2}$ grados. El MAE se comporta de manera similar en los Cuaterniones Unitarios, donde se alcanza un valor máximo de MAE de $1.0e^{-2}$.

En las Figuras 3.44 a 3.58 se muestran gráficas con la misma información que en las Figuras 3.39 a 3.43 para diferentes valores de Ángulos de Euler iniciales. Los resultados mostrados en estas gráficas 3.39 a 3.58 muestran la viabilidad de la implementación para lograr la precisión requerida.

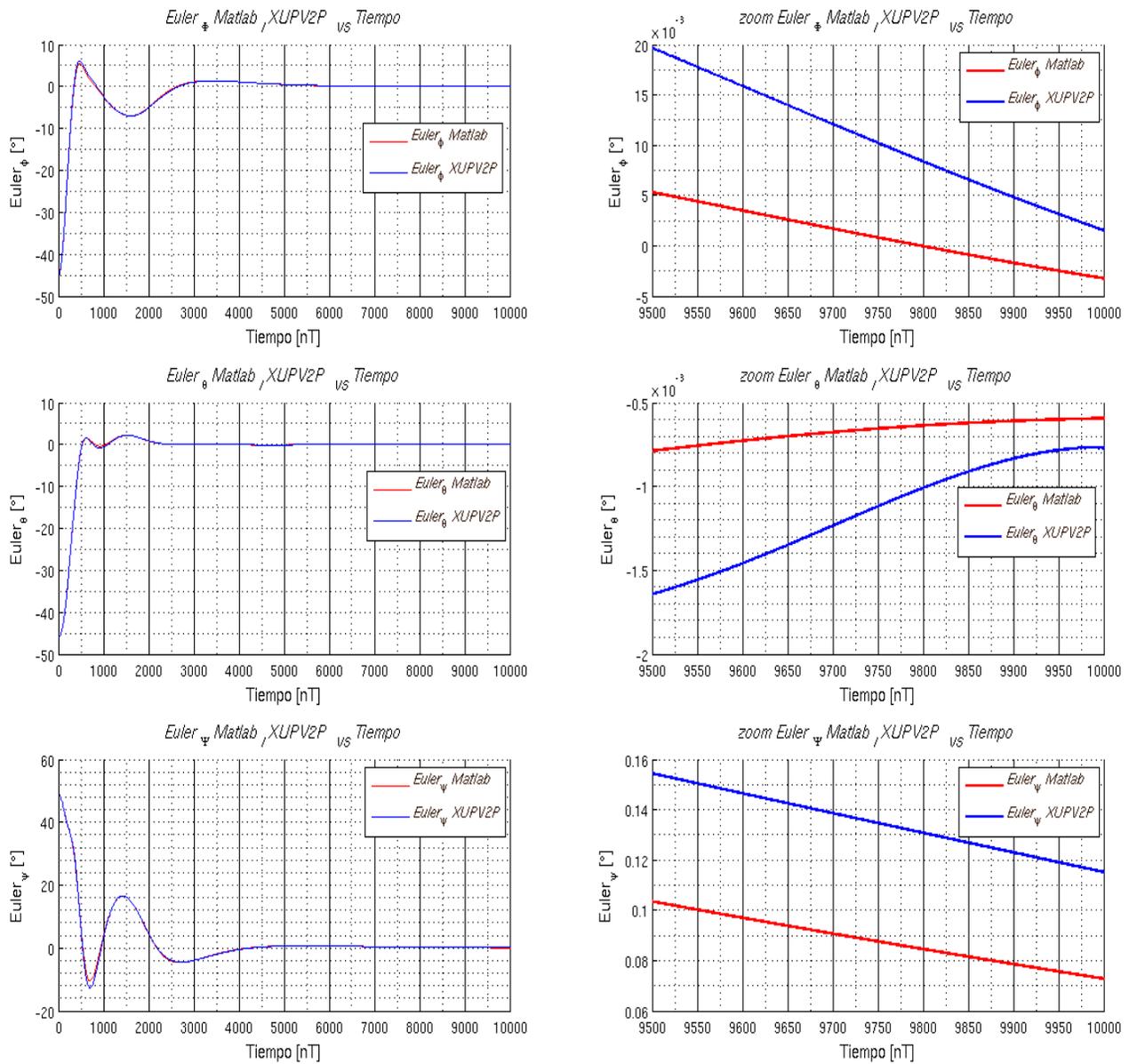


Figura 3.39: Gráficas comparativas entre Matlab y XUPV2P de ángulos de Euler para valores iniciales de $\phi = -45.125$, $\theta = -45.987$, $\psi = 48.978$. Gráficas completas y con un zoom en los últimos 500 puntos.

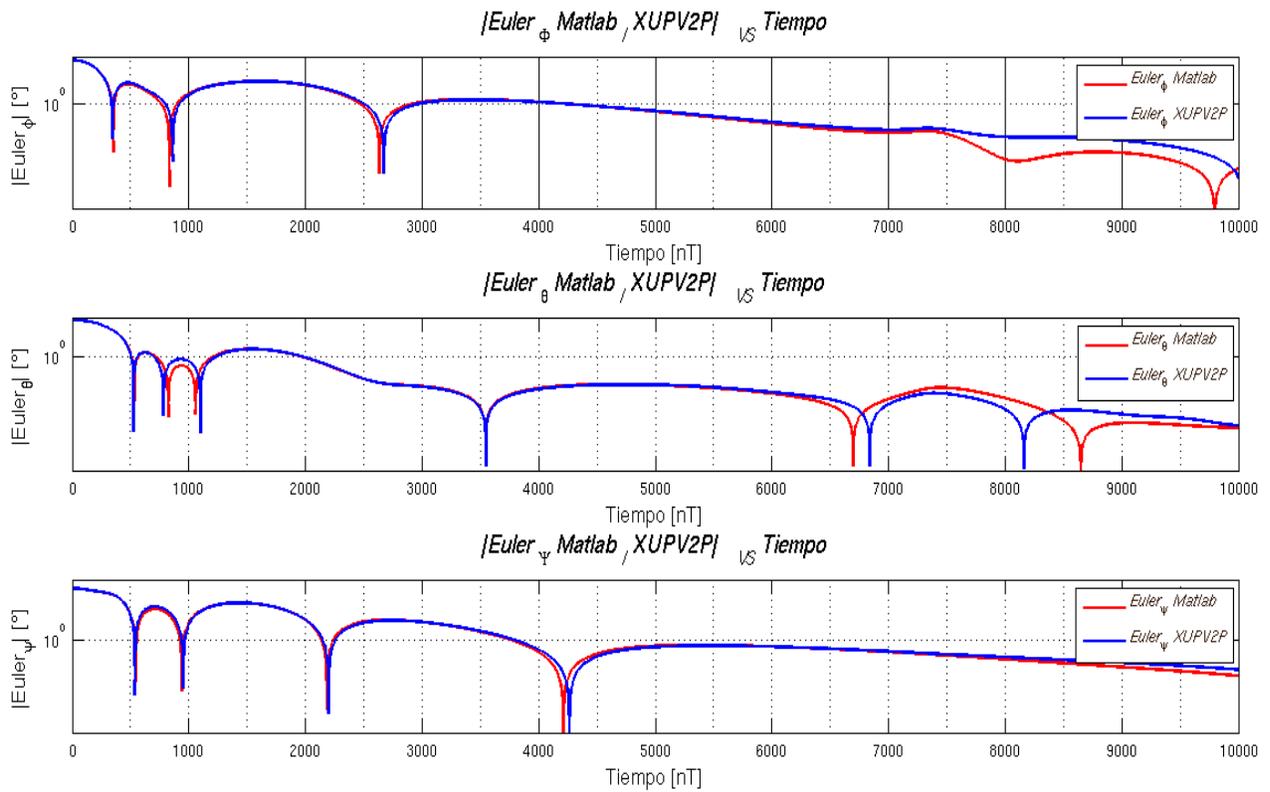


Figura 3.40: Gráficas comparativas en escala logarítmica entre Matlab y XUPV2P de Ángulos de Euler de para valores iniciales de $\phi = -45.125$, $\theta = -45.987$, $\psi = 48.978$.

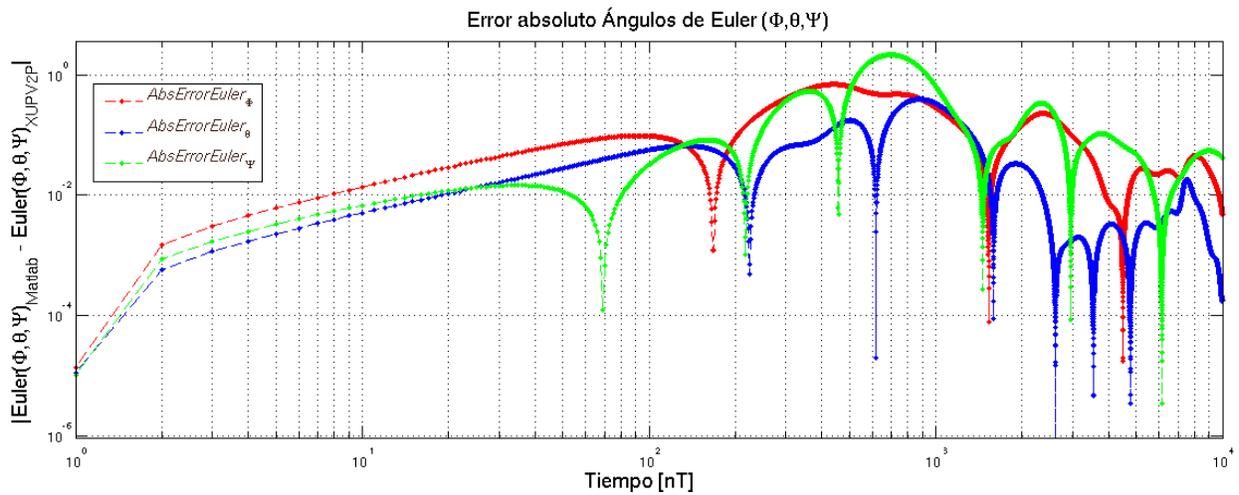


Figura 3.41: Gráfica en escala logarítmica del error absoluto entre Matlab y XUPV2P de Ángulos de Euler para valores iniciales de $\phi = -45.125$, $\theta = -45.987$, $\psi = 48.978$.

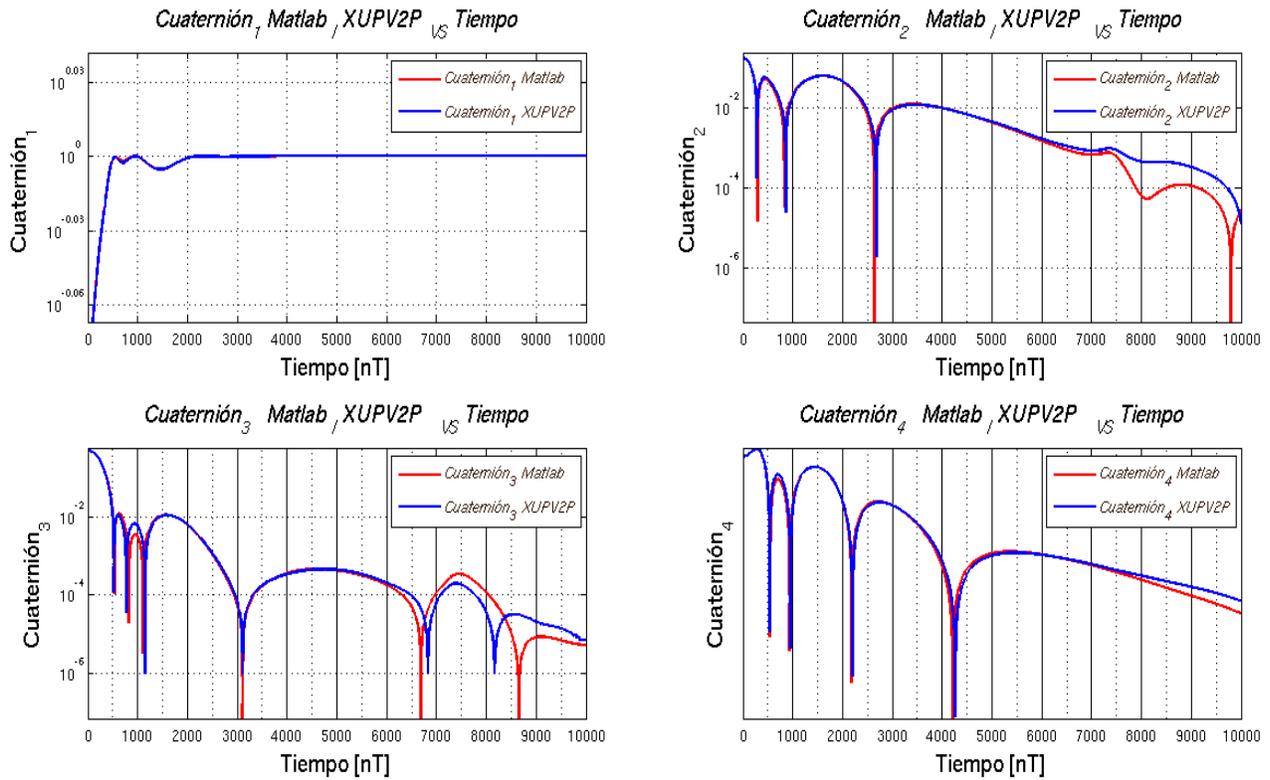


Figura 3.42: Gráficas comparativas en escala logarítmica entre Matlab y XUPV2P de Cuaterniones Unitarios para valores iniciales de $\phi = -45.125$, $\theta = -45.987$, $\psi = 48.978$.

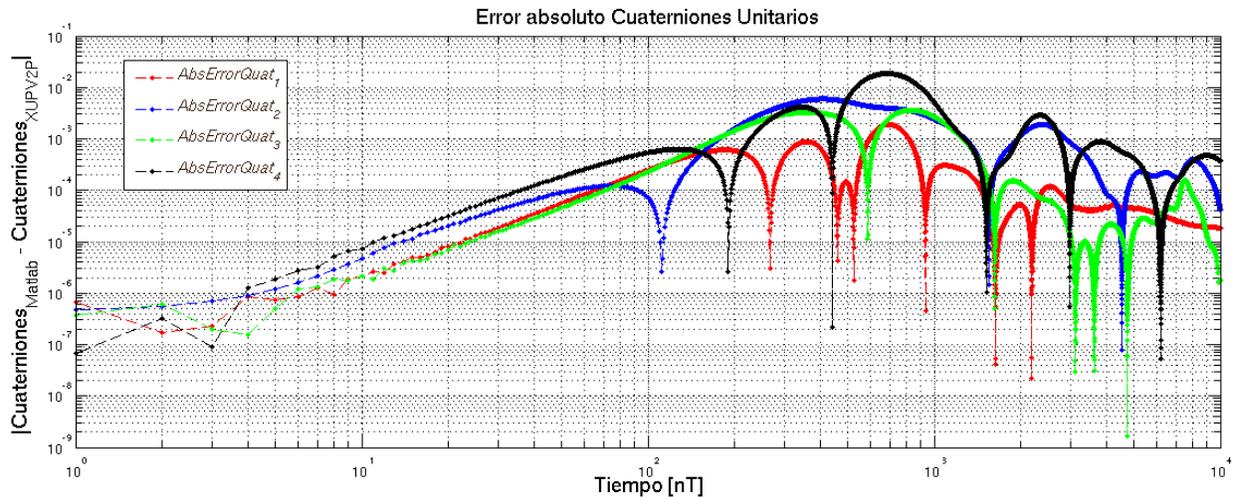


Figura 3.43: Gráfica en escala logarítmica del error absoluto entre Matlab y XUPV2P de Cuaterniones Unitarios para valores iniciales de $\phi = -45.125$, $\theta = -45.987$, $\psi = 48.978$.

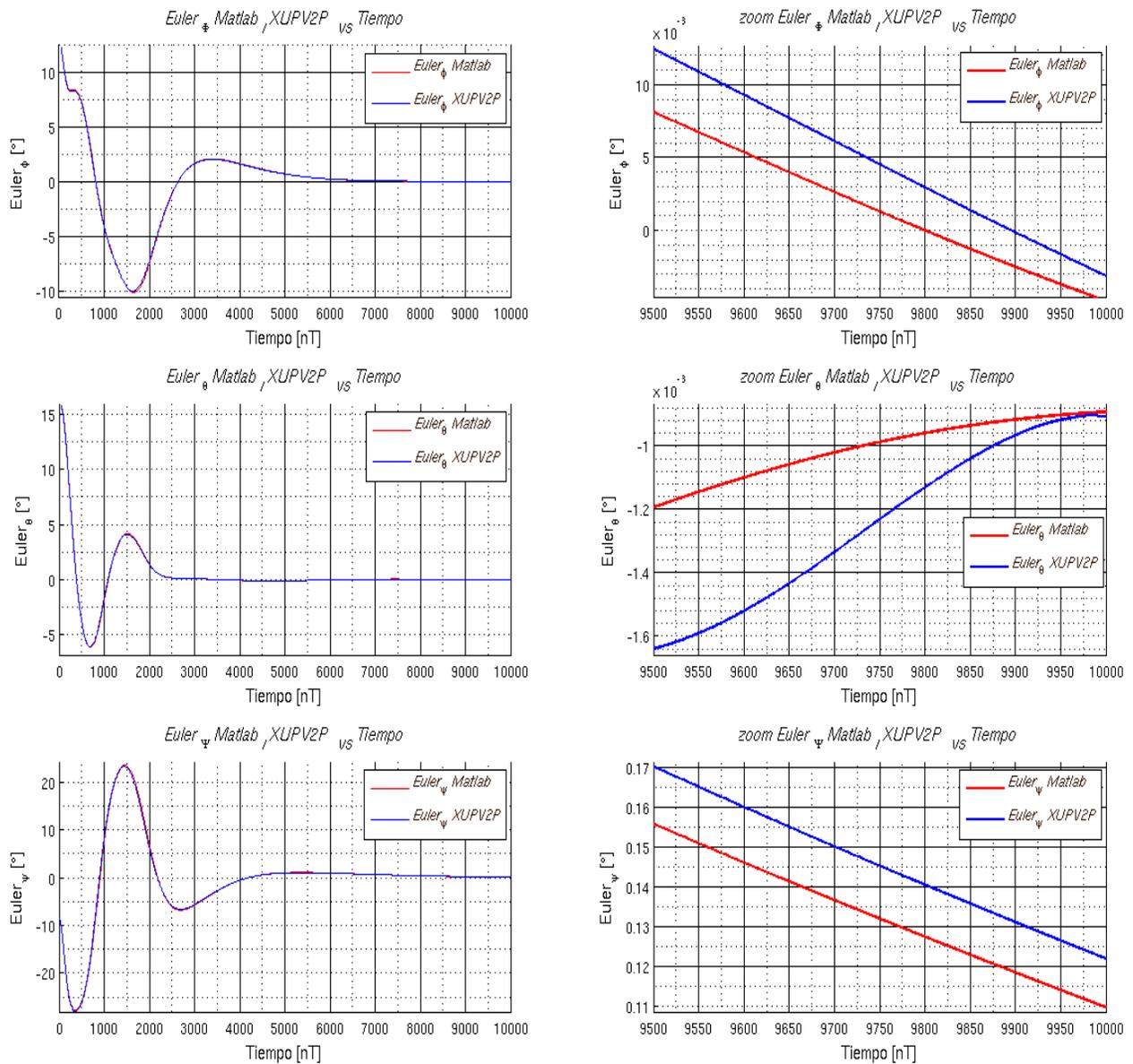


Figura 3.44: Gráficas comparativas entre Matlab y XUPV2P de ángulos de Euler para valores iniciales de $\phi = 12.825$, $\theta = 15.987$, $\psi = -8.197$. Gráficas completas y con un zoom en los últimos 500 puntos.

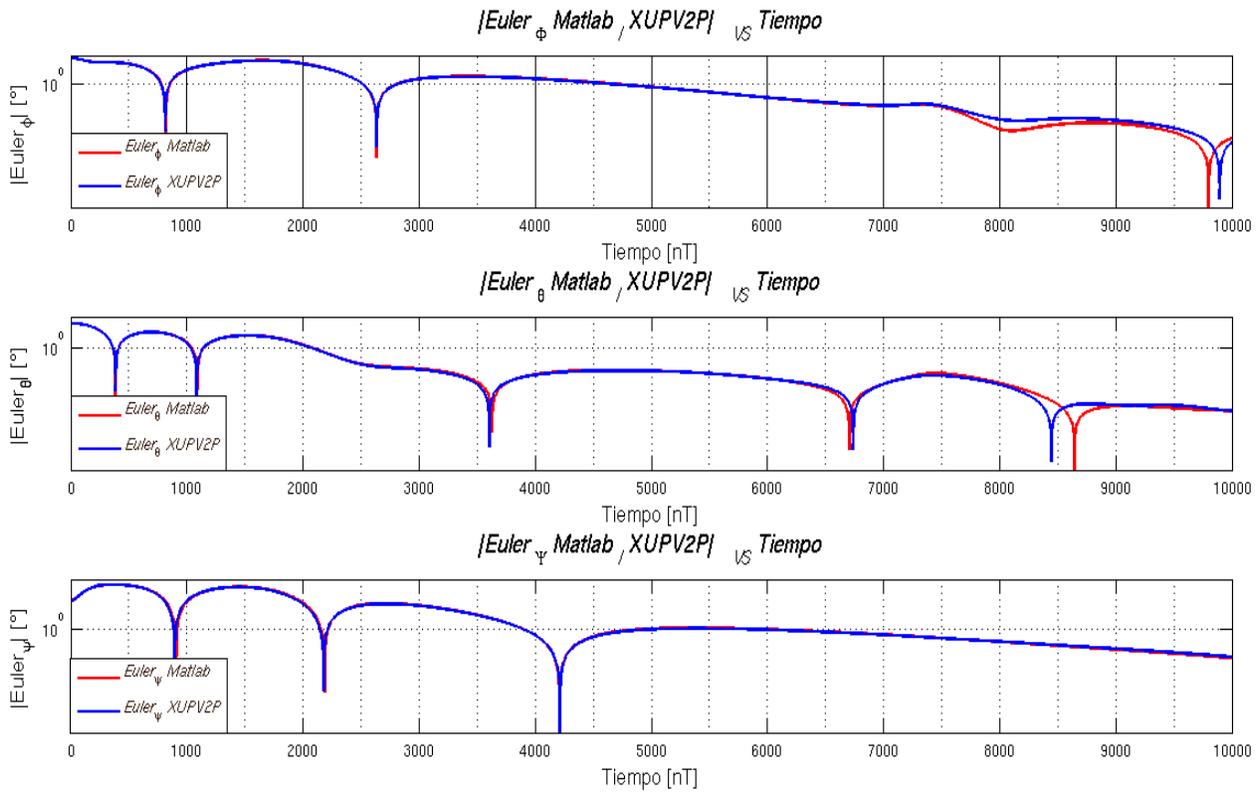


Figura 3.45: Gráficas comparativas en escala logarítmica entre Matlab y XUPV2P de Ángulos de Euler de para valores iniciales de $\phi = 12.825$, $\theta = 15.987$, $\psi = -8.197$.

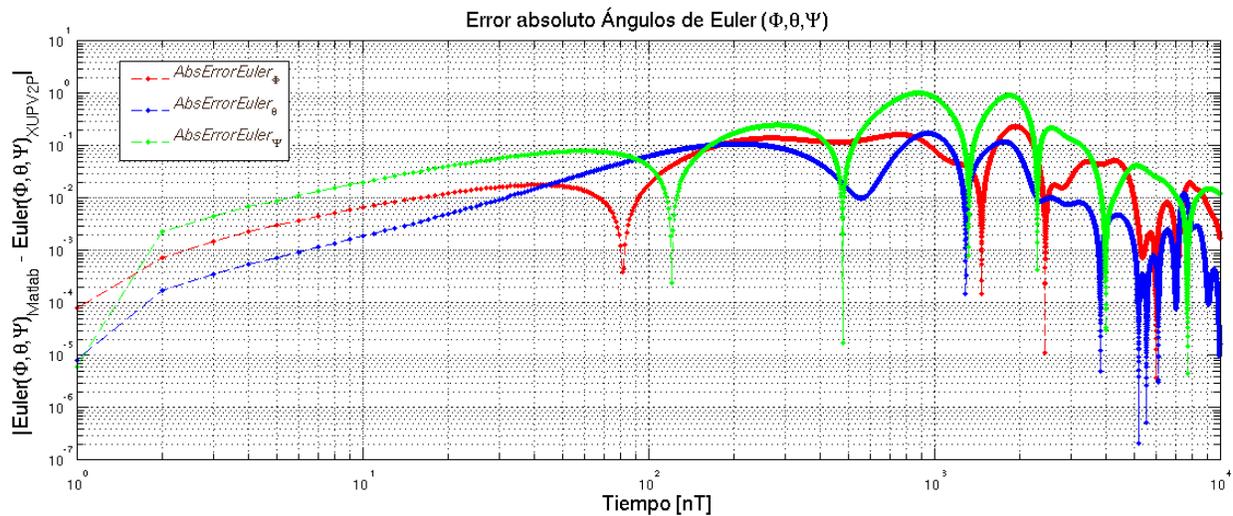


Figura 3.46: Gráfica en escala logarítmica del error absoluto entre Matlab y XUPV2P de Ángulos de Euler para valores iniciales de $\phi = 12.825$, $\theta = 15.987$, $\psi = -8.197$.

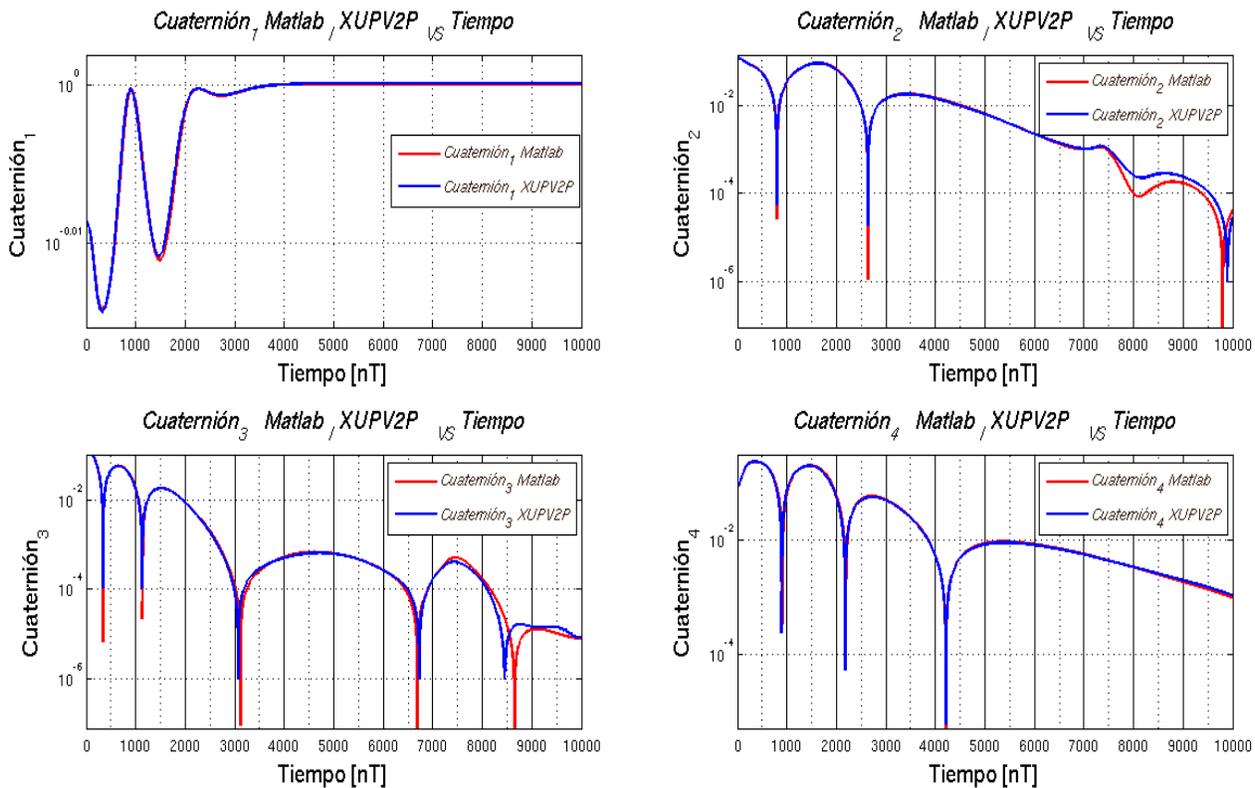


Figura 3.47: Gráficas comparativas en escala logarítmica entre Matlab y XUPV2P de Cuaterniones Unitarios para valores iniciales de $\phi = 12.825$, $\theta = 15.987$, $\psi = -8.197$.

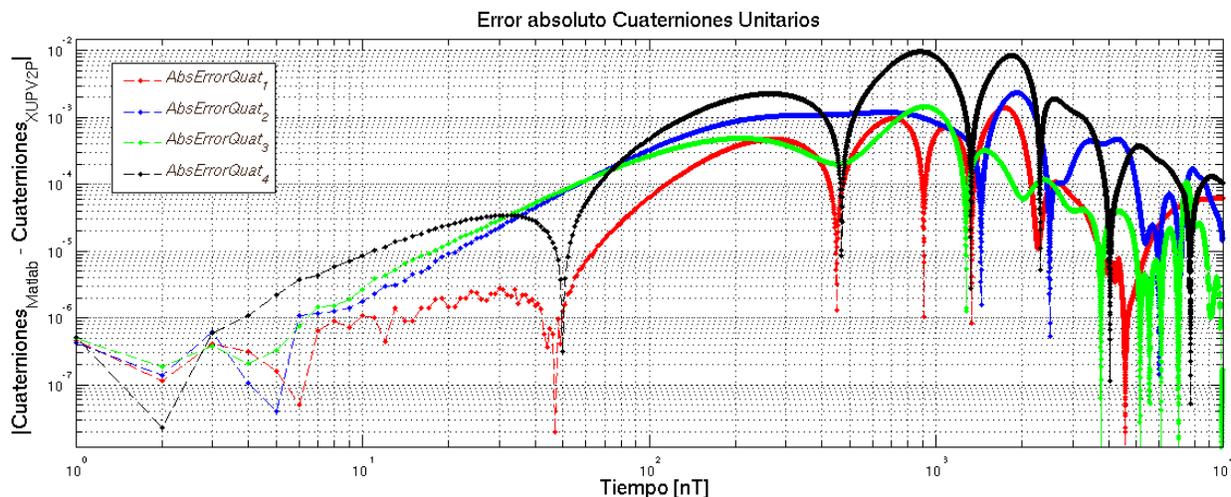


Figura 3.48: Gráfica en escala logarítmica del error absoluto entre Matlab y XUPV2P de Cuaterniones Unitarios para valores iniciales de $\phi = 12.825$, $\theta = 15.987$, $\psi = -8.197$.

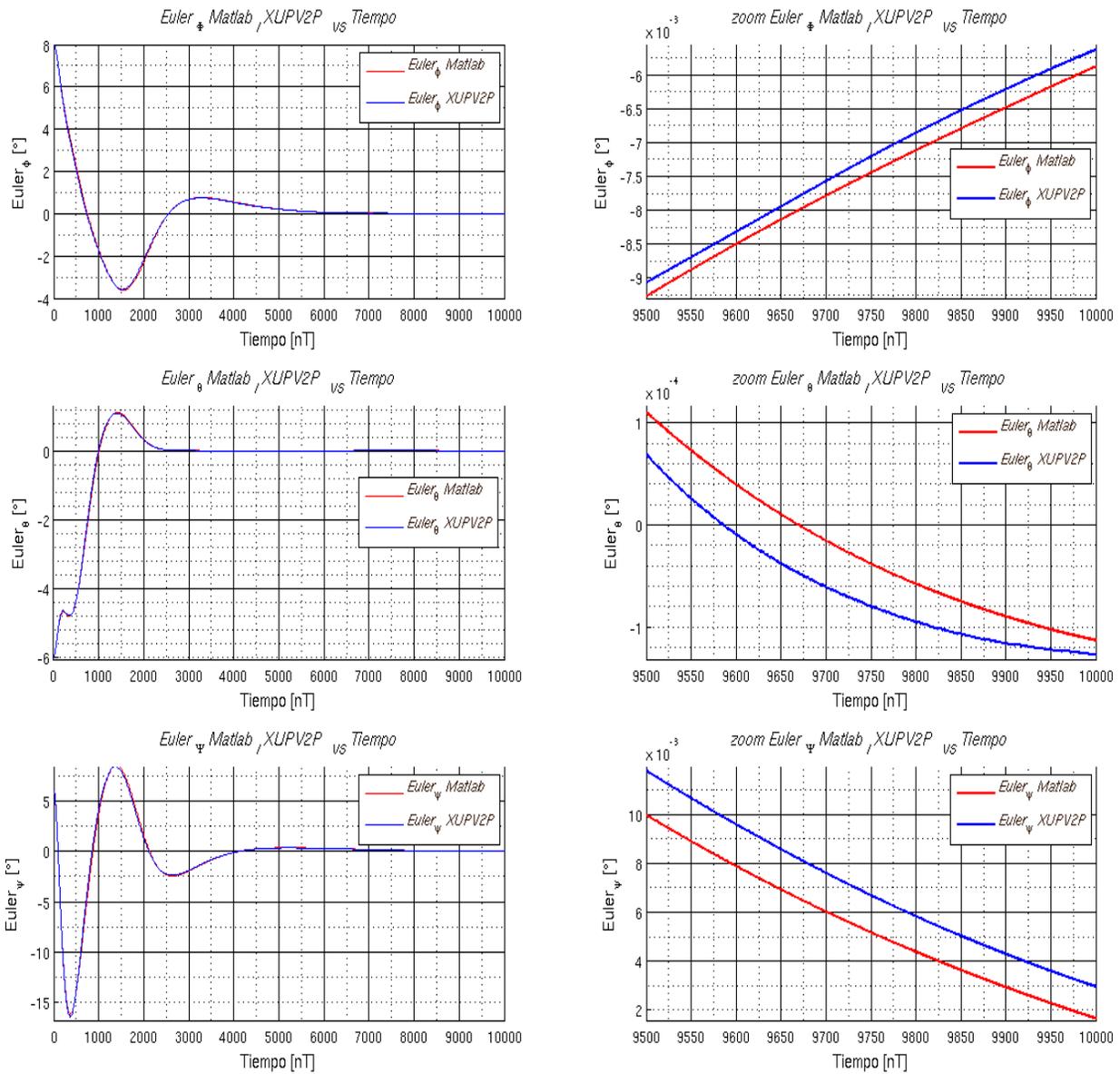


Figura 3.49: Gráficas comparativas entre Matlab y XUPV2P de ángulos de Euler para valores iniciales de $\phi = 7.99$, $\theta = -5.987$, $\psi = 6.197$. Gráficas completas y con un zoom en los últimos 500 puntos.

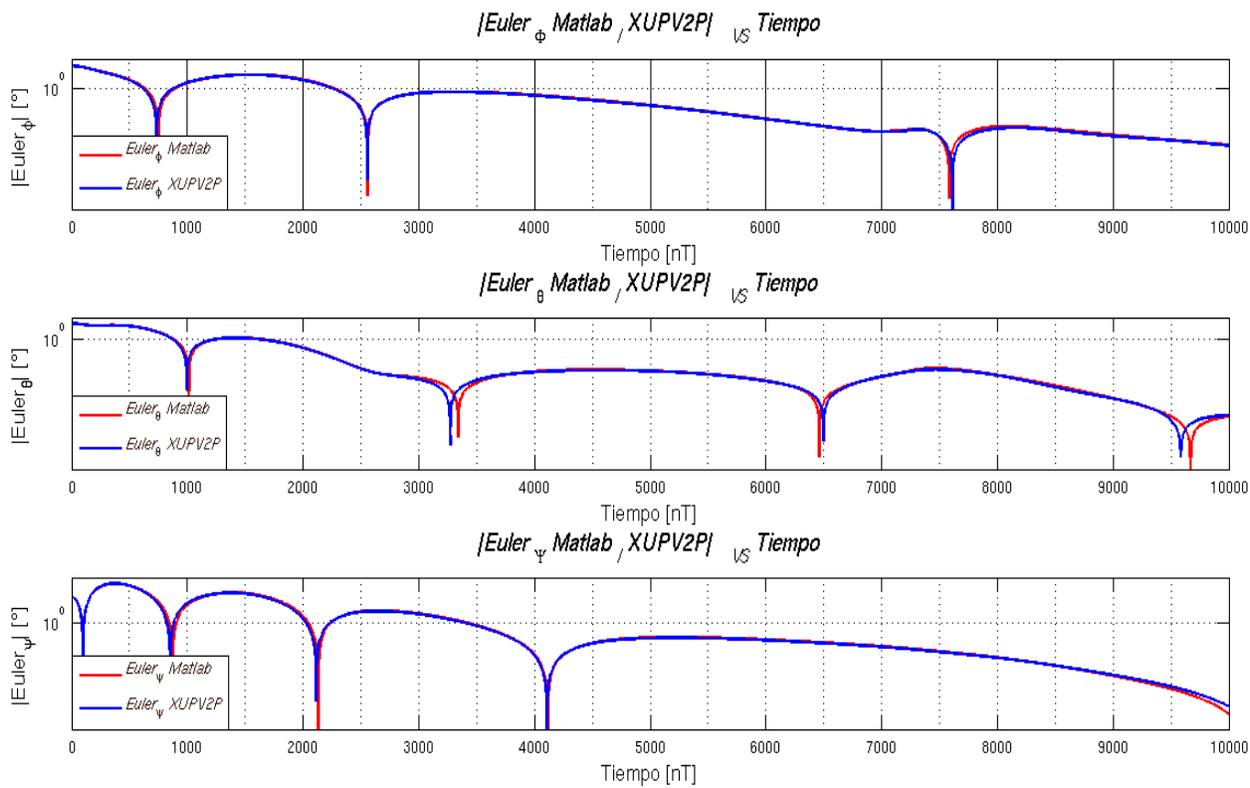


Figura 3.50: Gráficas comparativas en escala logarítmica entre Matlab y XUPV2P de Ángulos de Euler de para valores iniciales de $\phi = 7.99$, $\theta = -5.987$, $\psi = 6.197$.

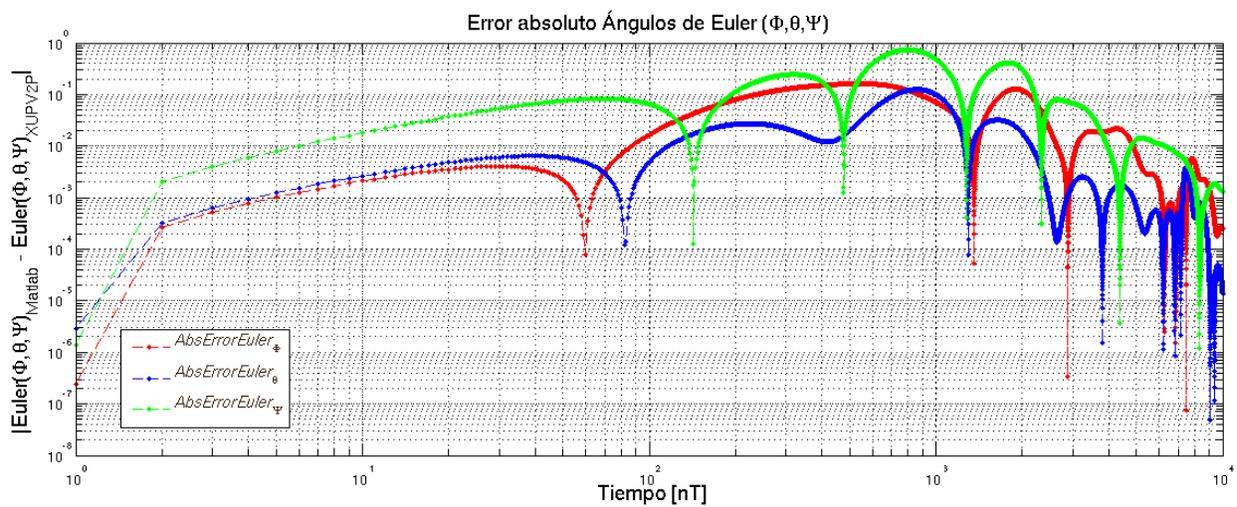


Figura 3.51: Gráfica en escala logarítmica del error absoluto entre Matlab y XUPV2P de Ángulos de Euler para valores iniciales de $\phi = 7.99$, $\theta = -5.987$, $\psi = 6.197$.

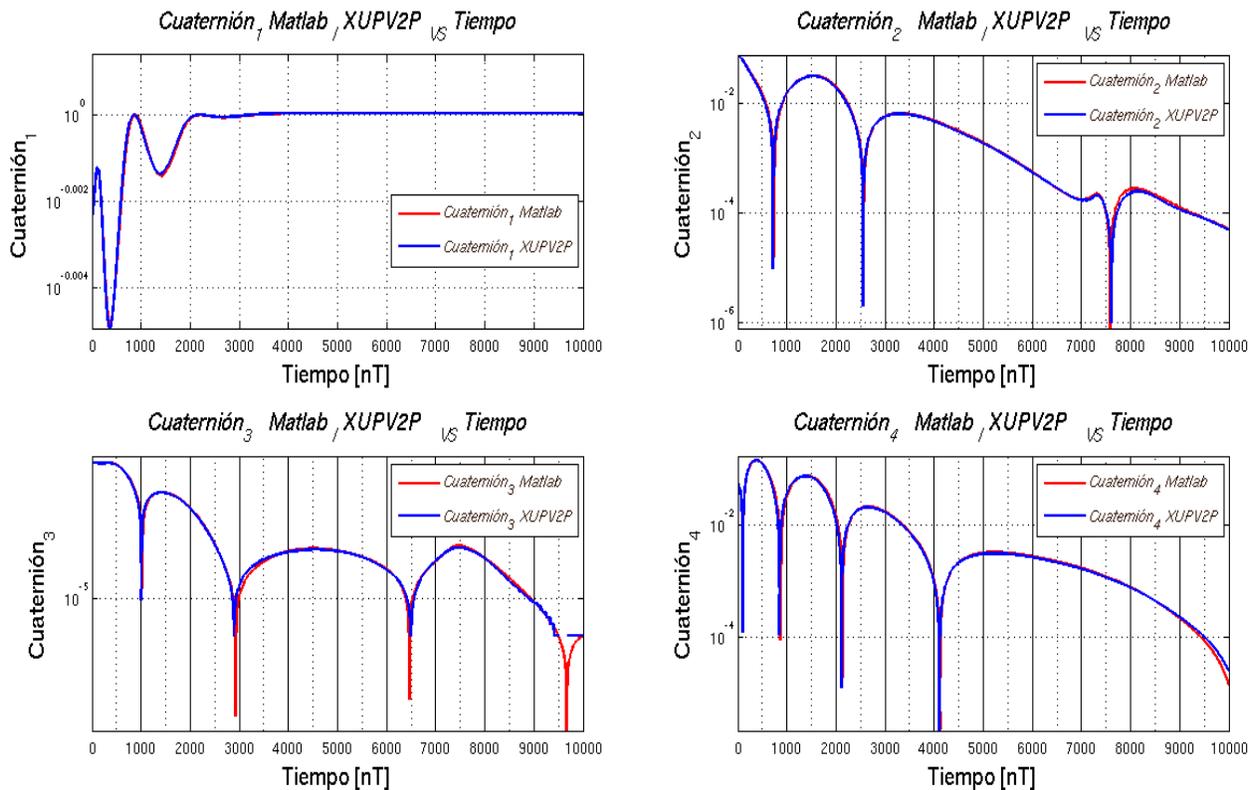


Figura 3.52: Gráficas comparativas en escala logarítmica entre Matlab y XUPV2P de Cuaterniones Unitarios para valores iniciales de $\phi = 7.99$, $\theta = -5.987$, $\psi = 6.197$.

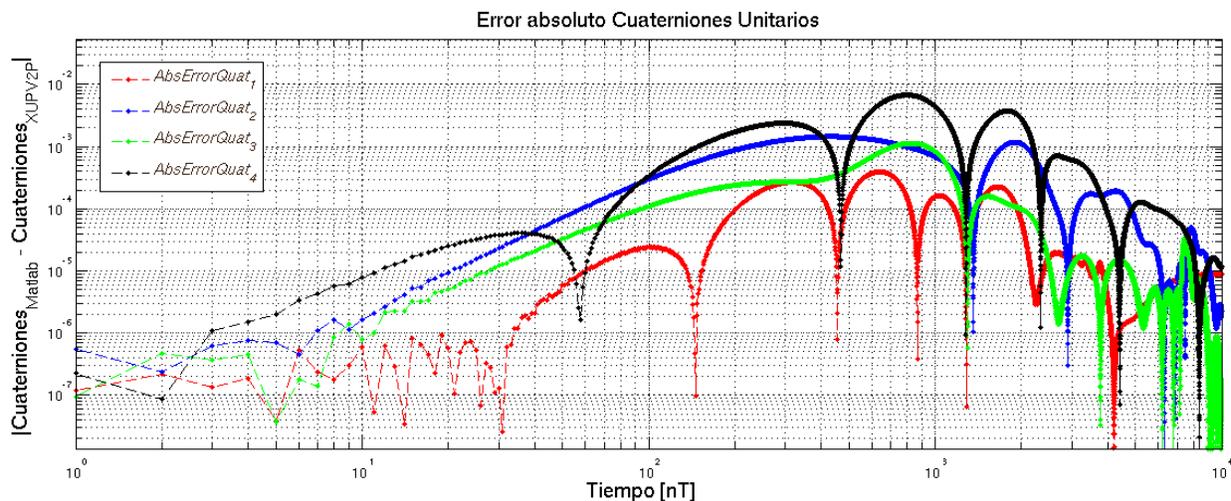


Figura 3.53: Gráfica en escala logarítmica del error absoluto entre Matlab y XUPV2P de Cuaterniones Unitarios para valores iniciales de $\phi = 7.99$, $\theta = -5.987$, $\psi = 6.197$.

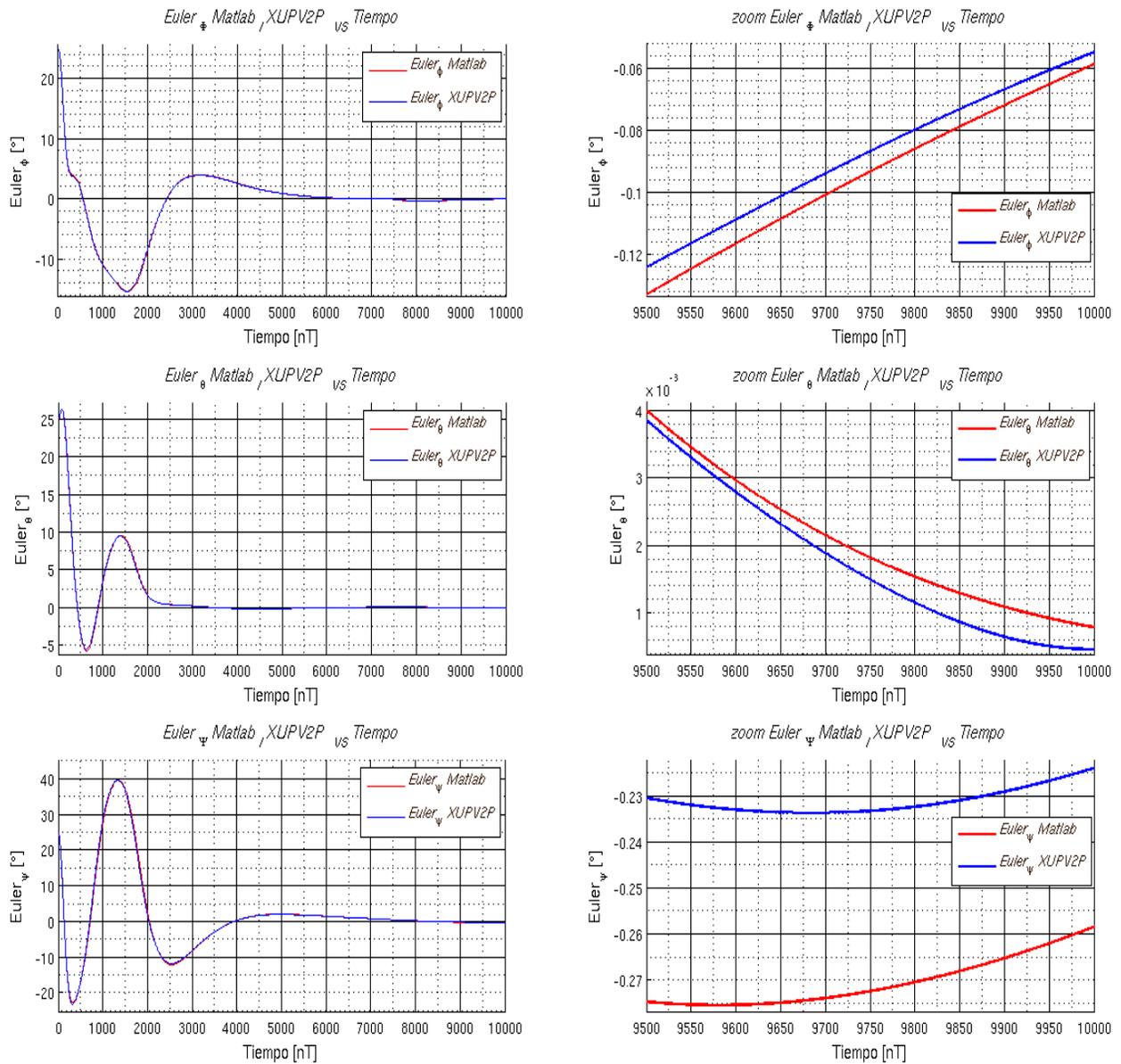


Figura 3.54: Gráficas comparativas entre Matlab y XUPV2P de ángulos de Euler para valores iniciales de $\phi = 25$, $\theta = 25$, $\psi = 25$. Gráficas completas y con un zoom en los últimos 500 puntos.

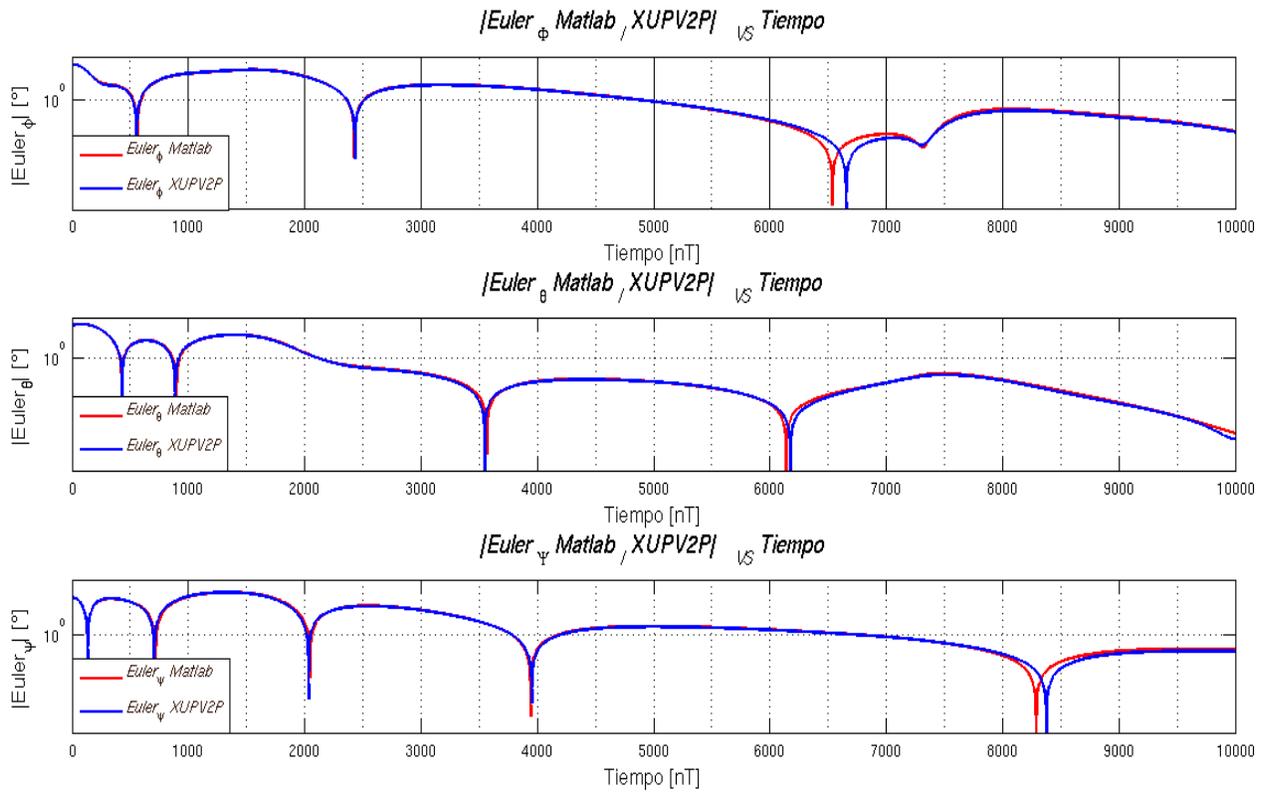


Figura 3.55: Gráficas comparativas en escala logarítmica entre Matlab y XUPV2P de Ángulos de Euler de para valores iniciales de $\phi = 25$, $\theta = 25$, $\psi = 25$.

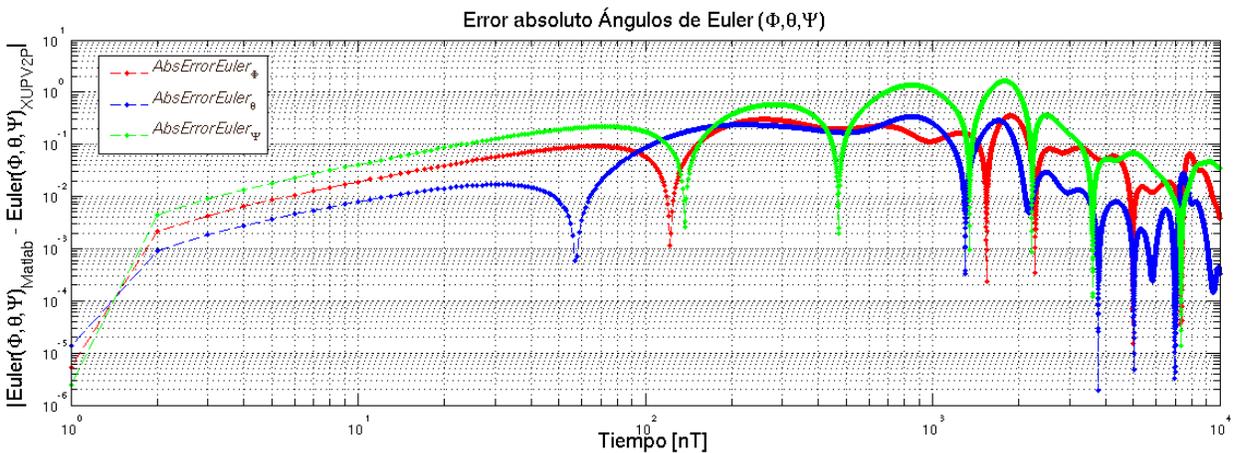


Figura 3.56: Gráfica en escala logarítmica del error absoluto entre Matlab y XUPV2P de Ángulos de Euler para valores iniciales de $\phi = 25$, $\theta = 25$, $\psi = 25$.

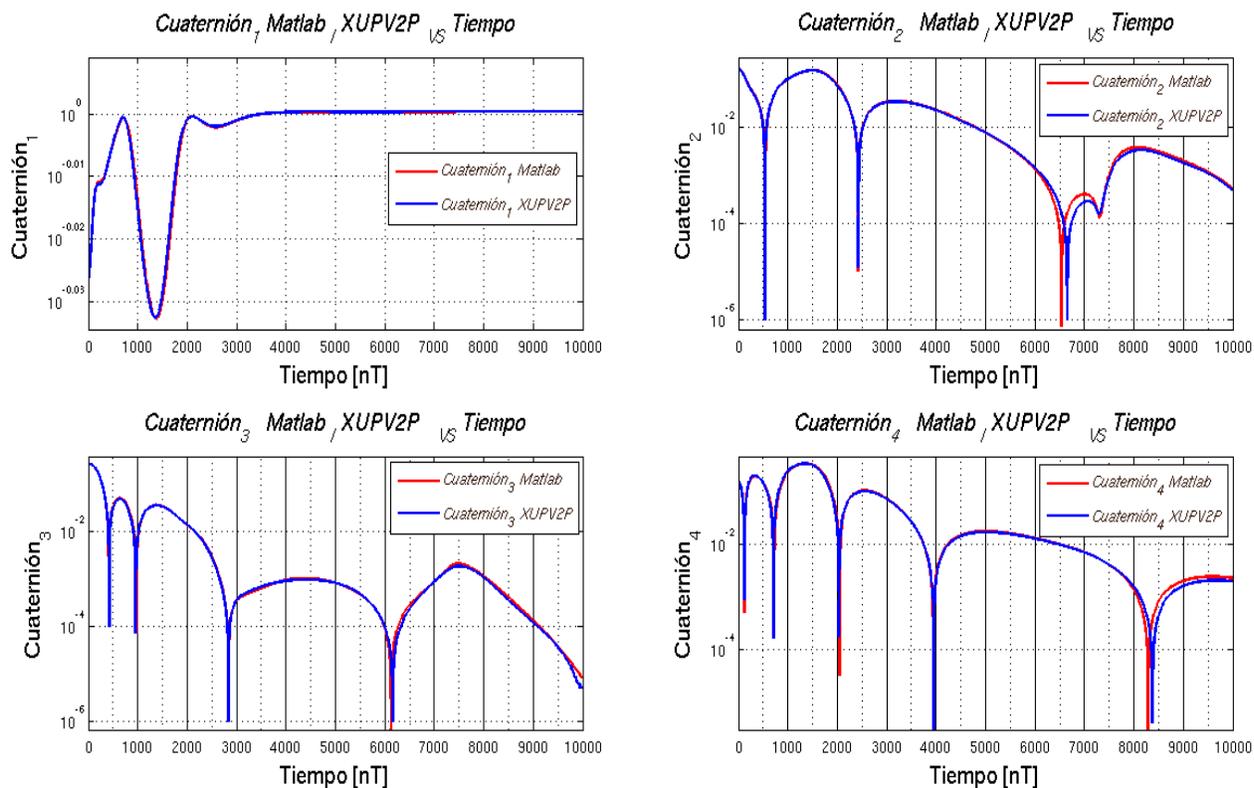


Figura 3.57: Gráficas comparativas en escala logarítmica entre Matlab y XUPV2P de Cuaterniones Unitarios para valores iniciales de $\phi = 25$, $\theta = 25$, $\psi = 25$.

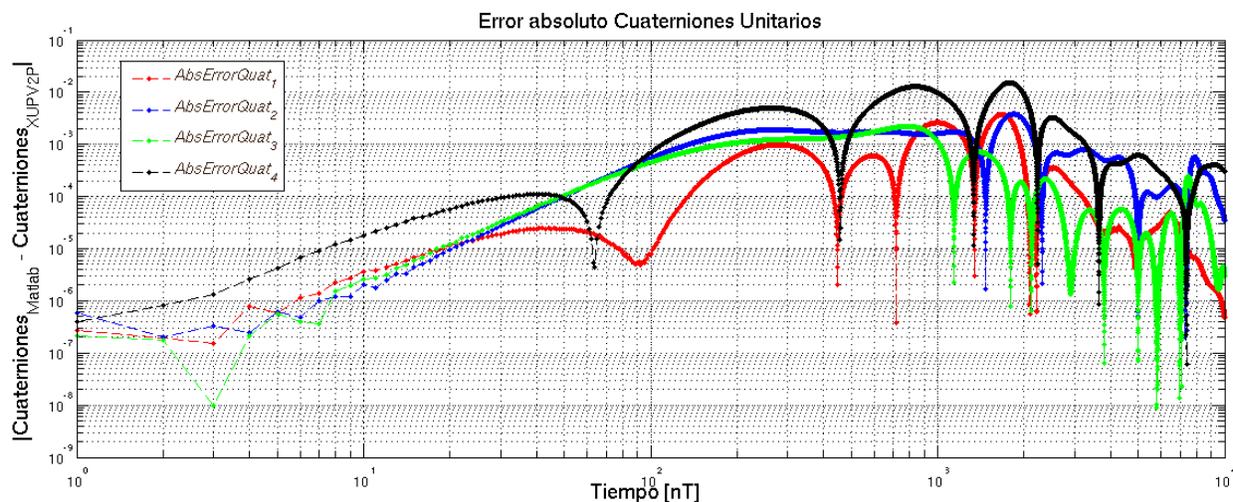


Figura 3.58: Gráfica en escala logarítmica del error absoluto entre Matlab y XUPV2P de Cuaterniones Unitarios para valores iniciales de $\phi = 25$, $\theta = 25$, $\psi = 25$.

Capítulo 4

Conclusiones

Mediante la comparación entre los resultados numéricos obtenidos por XUPV2P y la simulación en Matlab se demostró la viabilidad de uso de un dispositivo FPGA como base de procesamiento para implementar algoritmos de control para un satélite y para el desarrollo del Sistema de Determinación de Posición y Control. En base a los resultados numéricos se logró una estimación de posición menor a 0.2 grados respecto a la simulación. De igual forma se alcanzó un tiempo de estabilización aproximado de 10 minutos. La capacidad de procesamiento generada por el sistema permite estimar y controlar la posición con una frecuencia de 8 Hz, superando el valor sugerido por [3] de 5 Hz, y los requerimientos propuestos para CASTOR en los que se tiene una frecuencia de estimación y control de 1 Hz. Los resultados son concordantes a los requerimientos de precisión planteados por la misión CASTOR.

La utilización de un procesador embebido en el FPGA permitió la implementación de una solución basada en software, con todas las ventajas que presenta, sin limitar el uso de aceleración por hardware al poder incluir diseños basados exclusivamente en hardware. Teniendo en cuenta esta dualidad es fácil expandir el diseño al incluir módulos de procesamiento de sensores magnéticos, inerciales y GPS, así como de actuadores para migrar este trabajo al siguiente nivel de prototipo enfocado a los requerimientos propios de la misión Quetzal UNAM-MIT, logrando esto sin disminuir el rendimiento del sistema.

Dada la capacidad y flexibilidad del dispositivo XC2VP30 COTS de Xilinx que incluye dos núcleos PowerPC, y permite tanto la implementación de varios microprocesadores embebidos como de diseños específicos, se tiene la oportunidad de implementar arquitecturas de procesamiento en paralelo, así como la aplicación de técnicas de tolerancia a fallas y reconfiguración. Estas capacidades hacen al dispositivo XC2VP30 COTS de Xilinx ideal para su uso en aplicaciones espaciales en misiones micro que requieren alta capacidad de procesamiento y el uso de componentes de tipo COTS sin comprometer confiabilidad.

El trabajo realizado se puede tomar como base para el diseño e implementación del ADCS que será usado en el satélite Quetzal.

4.1. Trabajo a futuro

Como trabajo futuro se plantea la optimización del software dada su naturaleza de sistema embebido. De igual manera se plantea el uso de aceleración por hardware para el cálculo de funciones trascendentes mediante la implementación de un módulo de cálculo de algoritmo CORDIC.

Se plantea también como parte del trabajo a futuro la implementación de un sistema operativo en tiempo real (RTOS) en un microprocesador embebido en el FPGA, teniendo así la capacidad de uso de librerías de algebra lineal y sistemas de control como Armadillo C++ Linear Algebra Library, Ch Control System Toolkit, Computer-Aided Control System Design (CACSD) Tools para GNU Octave, y Python Control Systems Library (python-control) como base de la implementación de una solución en tiempo real de la ARE haciendo más robusto el controlador LQR.

Adicional a el uso de RTOS, actualmente se trabaja en el aprovechamiento del potencial de implementación de arquitecturas en paralelo del dispositivo XC2VP30 para generar varios núcleos de procesamiento independientes, y en base a los requerimientos de confiabilidad, proponer un esquema de implementación de técnicas de tolerancia a fallas con el objetivo de generar un prototipo capaz de soportar las condiciones de ambiente espacial brindando la confiabilidad requerida mediante el uso de un dispositivo COTS.

Generar publicaciones en revistas internacionales.

Bibliografía

- [1] James R. Wertz, David F. Everett, and Jeffery J. Puschell. *Space Mission Engineering: The New SMAD*. Microcosm Press, Hawthorne, California, 1st edition, 2011.
- [2] Carlos Romo Fuentes, Paulo Lozano, Alvar Saenz Otero, and J.A.F Perez. Satellite Project Quetzal UNAM-MIT. *2013 6th International Conference on Recent Advances in Space Technologies (RAST)*, pages 903–907, 2013.
- [3] K. L. Bedingfield. Spacecraft System Failures and Anomalies Attributed to the Natural Space Environment. Technical report, National Aeronautics and Space Administration, 1996.
- [4] Jean-Pierre Deschamps. *Synthesis of Arithmetic Circuits. FPGA, ASIC and Embedded Systems*. Wiley, 2006.
- [5] Benjamin F. Harding and R.C. Cofer. *Rapid System Prototyping with FPGAs*. Elsevier, Oxford, UK, 2006.
- [6] Inc. Xilinx. PowerPC 405 Processor Block Reference Guide Embedded Development Kit. Technical report, Xilinx, Inc., 2003.
- [7] Inc. Xilinx. Microblaze Processor Reference Guide: Embedded Development kit EDK 10.1. Technical report, Xilinx, Inc., 2008.
- [8] Inc. Xilinx. Xilinx University Program Virtex-II Pro Development System Hardware Reference Manual. Technical report, Xilinx, Inc., 2009.
- [9] J. Larson Wiley and James R. Wertz. *Space Mission Analysis and Design*. Microcosm Press, El Segundo, California, 3rd edition, 2006.
- [10] Castor Satellite Design Document. Technical report, MIT, 2010.
- [11] Sinclair Interplanetary. SS-411 Sun Sensor Interface Control Document. Electronic Document, February 2010. Sinclair Interplanetary.
- [12] PNI Corporation. MicroMag3 3-Axis Magnetic Sensor Module. Electronic Document, April 2005.

- [13] Inc Analog Devices. Six Degrees of Freedom Inertial Sensor ADIS16360/ADIS16365. Electronic Document, 2012.
- [14] Sinclair Interplanetary. Microsatellite Reaction Wheels (-0.060-). Electronic Document, 2013.
- [15] Anil K Maini and Varsh Agrawal. *Satellite Technology Principles and Applications*. Willey, 2nd edition, 2011.
- [16] Ted Huffmire, Cynthia Irvine, Thuy D. Nguyen, Timothy Levin, Ryan Kastner, and Timothy Sherwood. *Handbook of FPGA Design Security*. Springer, California, 2010.
- [17] Frank Jr. Hall Schmidt. Fault Tolerant Design Implementation on Radiation Hardened by Design SRAM-Based FPGAs. Master’s thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2013.
- [18] Inc. Xilinx. SPACE-GRADE RAD-HARD VIRTEX-5QV FPGA. Technical report, Xilinx, Inc., 2011.
- [19] David Ratter. FPGAs on Mars. *Xcell Journal*, pages 1–4, 2004.
- [20] Peter Bergsman. Xilinx FPGA Blasted into Orbit. *Xcell Journal*, (46):86–88, 2003.
- [21] H. Quinn, P. Graham, K. Morgan, Z. Baker, M. Caffrey, D. Smith, and R. Bell. On-Orbit Results for the Xilinx Virtex-4 FPGA. *2012 IEEE Radiation Effects Data Workshop (REDW)*, pages 1–8, 2012.
- [22] Military and Aerospace Electronics, F-35 Joint Strike Fighter uses Actel FPGA for engine electronics. *Military and Aerospace Electronics*, 2004.
- [23] Design and Reuse Magazine, TTP controller IP in Altera’s low-cost cyclone FPGA families for Aerospace Applications. *Design and Reuse Magazine*, 2007.
- [24] Military and Aerospace Electronics, FPGA processors keep Mars Rovers moving. *Military and Aerospace Electronics*, 2005.
- [25] Rainer Sandau, Hans-Peter Roeser, and Arnoldo Valenzuela. *Small Satellites for Earth Observation Selected Contributions*. Springer, Berlin, Germany, 2008.
- [26] Saúl Daniel Santillán Gutierrez, Carlos Romo Fuentes, Saúl de la Rosa Nieves, Jorge Alfredo Ferrer Pérez, and Mendieta Jimenez Francisco Javier. Developing a Space Program for Mexico Challenges and opportunities. *IEEE*, 2013.
- [27] Schwank James R., Shaneyfelt Marty R., and Dodd Paul E. Radiation Hardness Assurance Testing of Microelectronic Devices and Integrated Circuits: Test Guideline for Proton and Heavy Ion Single-Event Effects. Technical report, Sandia National Laboratories, 2008.

- [28] Schwank James R. and Shaneyfelt Marty R. Radiation Hardness Assurance Testing of Microelectronic Devices and Integrated Circuits: Radiation Environments, Physical Mechanisms, and Foundations for Hardness Assurance. Technical report, Sandia National Laboratories, 2008.
- [29] Fred S. Singer. *Torques and Attitude Sensing in Earth Satellites*. Google eBook, New York, NY, 2012.
- [30] James R. Wertz. *Spacecraft Attitude Determination and Control*. 1978.
- [31] Norton Maxwell. *Space Navigation and Guidance*. Springer, London, 1st edition, 1998.
- [32] Larson Wiley J. and Wertz James R. *Space Mission Analysis and Design*. Microcosm Press and Kluwer Academic Publishers, El Segundo, California, 3rd edition, 1999.
- [33] Ceren Kaplan. LEO Satellites: Attitude Determination and Control Components; Some Linear Attitude Control Techniques. Master's thesis, Middle East Technical University, ankaya Ankara, Turkey, 2006.
- [34] Yang Y. Spacecraft Attitude Determination and Control: Quaternion based method. *Annual Reviews in Control*, 36(2):198–219, 2012.
- [35] Oliver L. de Weck. Space Systems Product Development (ADCS). Electronic Source.
- [36] Ulrik Jorgensen. Synchronization and Control of Attitude for Spacecrafts: Design, Analysis and Experiments. Master's thesis, Norwegian University of Science and Technology, Trondheim, Norway, 2010.
- [37] James Diebel. Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors. Electronic Document, 2006.
- [38] Kristian Svartveit. Attitude Determination of the NCUBE satellite. Master's thesis, Norwegian University of Science and Technology, Trondheim, Norway, 2003.
- [39] Bevilacqua R., T. Lehmann, and M. Romano. Development and Experimentation of LQR/APF Guidance and Control for Autonomous Proximity Maneuvers of Multiple Spacecraft. *Acta Astronautica*, 68(7-8):1260–1275, 2011.
- [40] Rogers L. C. Fluid Models in Queueing Theory and Wiener-Hopf factorization of Markov Chanis. 1994.
- [41] Jonq Juang. Global Existence and Stability of Solutions of Matrix Riccati Equations. *Journal of Mathematical Analysis and Applications*, 258(1):1–12, 2001.
- [42] Abou-Kandil H. and Freiling G. *Matrix Riccati Equations. Systems and Control: Foundations and Applications*. 2003.

- [43] J.C. Wu, J.N. Yang, and W.E. Schmitendorf. Reduced-order H LQR Control for Wind-excited tall buildings. *Engineering Structures*, 20(3):222–236, 1998.
- [44] Jiang Jun, Hua Liuwen, and Zhong Liu. Trajectory Control and Optimization of Supercavitating Torpedo. *ISSCAA 2008. 2nd International Symposium on Systems and Control in Aerospace and Astronautics, 2008*, pages 1–5, 2008.
- [45] Volker L. Mehrmann. *The Autonomous Linear Quadratic Control Problem: Theory and Numerical Solution*. Springer, Berlin, Germany, 7th edition, 1991.
- [46] Vasile Sima. *Algorithms for Linear-Quadratic Optimization*. Marcel Dekker, New York, NY, 1996.
- [47] Tian Xianga, Tao Mengb, Hao Wangb, Ke Hana, and Zhong-He Jinb. Design and on-orbit Performance of the Attitude Determination and Control System for the ZDPS-1A pico-satellite. *Acta Astronautica*, 77:182–192, 2012.
- [48] ECE684. Algorithms for solving the Algebraic Riccati Equation. Electronic Document, December 2009. Notes of ECE684 Indiana University-Purdue University Indianapolis.
- [49] Jinghong Kang. *The Computational Kleinman-Newton Method in Solving Nonlinear Nonquadratic Control Problems*. PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 1998.
- [50] P.K. Menon, T. Lam, L.S. Crawford, and V.H.L. Cheng. Real-time Computational Methods for SDRE Nonlinear Control of Missiles. *Proceedings of the 2002 American Control Conference*, 1:232–237, 2002.
- [51] Kirsten Morris and Carmeliza Navasca. Solution of Algebraic Riccati Equations Arising in Control of Partial Differential Equations. 2001.
- [52] Alexander Larzon, Yantao Feng, B.D.O. Anderson, and M. Rotkowitz. Computing the Positive Stabilizing Solution to Algebraic Riccati Equations with an Indefinite Quadratic Term via a Recursive Method. *IEEE Transactions on Automatic Control*, 53(10):2280–2291, 2008.
- [53] Hiroaki Mukaidani, Hua Xu, and Kiochi Mizukami. A revised Kleinman Algorithm to solve Algebraic Riccati Equation of Singularity Perturbed Systems. *Elsevier Science Ltd. Automatica*, 38(2002):553–558, 2001.
- [54] Ron Sass and Andrew G. Schmidt. *Embedded Systems Design with Platform FPGAs Principles and Practices*. Morgan Kaufmann-Elsevier, Burlington, Massachusetts, 2010.
- [55] Richard S. Sandige and Michael L. Sandige. *Fundamentals of Digital and Computer Design with VHDL*. McGraw-Hill, New York, 2012.

- [56] Peter J. Ashenden. *The Designer's Guide to VHDL*. Systems on Silicon, 3rd edition, 1996.
- [57] Volnei A. Pedroni. *Circuit Design with VHDL*. MIT Press, Cambridge, Massachusetts, 2004.
- [58] Steve Kilts. *Advanced FPGA Design: Architecture, Implementation and Optimization*. Wiley-Interscience, IEEE, Minneapolis, Minnesota, 2007.
- [59] Jari Nurmi. *Processor Design System-on-Chip Computing for ASICs and FPGAs*. Springer, Finland, 2007.
- [60] Gulati Kanupriya and Khatri Sunil P. *Hardware Acceleration of EDA Algorithms Custom ICs, FPGAs and GPUs*. Springer, Coppel, Texas, 2010.
- [61] Gokul Govindu, Ronald Scrofano, and Viktor K. Prasanna. A Library of Parameterizable Floating Point Cores for FPGA and their Application to Scientific Computing. *In Proc. of International Conference on Engineering Reconfigurable Systems and Algorithms*, pages 137–148, 2005.
- [62] David Bishop. VHDL IEEE-2008 Floating Point Package User's guide. Technical report, VHDL.
- [63] Felix Huber, Peter Behr, Hans-Peter Roser, and Samuel Pletner. FPGA based On-Board Computer System for the Flying Laptop Micro-Satellite. 2005.
- [64] Inc. Xilinx. Virtex-II Pro and Virtex-II Pro X Platform FPGAs: Complete Data Sheet. Technical report, Xilinx, Inc., 2011.
- [65] Kemin Zhou, John Doyle, and Keith Glover. *Robust and Optimal Control*. Prentice Hall, Upper Saddle River, New Jersey, 1996.

Apéndice A

Derivación de algunas expresiones

A.1. Representación numérica en punto fijo y punto flotante

El formato de punto fijo permite representar un, relativamente, reducido rango de valores numéricos con una *precisión absoluta* constante. Por otro lado, el formato de punto flotante permite la representación de un rango más amplio de valores numéricos, con una *precisión relativa* constante [4].

Definiciones. Formato numérico de punto fijo.

1. En un *sistema de numeración de punto fijo*, el número representado en la formato

$$x_{n-p-1}x_{n-p-2} \dots x_1x_0.x_{-1}x_{-2} \dots x_{-p} \quad (\text{A.1})$$

es x/B^p , donde x es el entero representado por por la misma secuencia de dígitos sin punto.

2. Siendo x_{min} y x_{max} los enteros mínimos y máximos que pueden ser representados con n dígitos, esto es, $x_{min} = 1 - B^{n-1}$ y $x_{max} = B^{n-1} - 1$ en representación de magnitud signada, y $x_{min} = -B^n/2$ y $x_{max} = -B^n/2 - 1$ en complemento a B o representación de exceso $B^n/2$. Entonces, cualquier número real x perteneciente al intervalo

$$B^{-p}.x_{min} \leq x \leq B^{-p}.x_{max} \quad (\text{A.2})$$

puede ser representado en la forma (A.1) con algun error igual al valor absoluto de la diferencia entre x y su representación.

3. La *distancia* d entre números representados exactamente es igual a la unidad en la posición menos significativa (ulp), esto es, B^{-p} , de tal modo que el *error máximo* es igual a

$$ulp/2 = B^{-p}/2 \quad (\text{A.3})$$

4. El *error relativo máximo* es igual a $ulp/(2 \cdot |x|) = 1/(2 \cdot |x| \cdot B^p)$. Si $x \neq 0$ entonces $|x| \geq B^{-p}$, así que el error relativo máximo es menor o igual a $1/2$.

Definiciones. Formato numérico de punto flotante.

1. En un *sistema de numeración de punto flotante*, la representación consiste de dos números: un número de punto fijo (el *significante*) $+s$ o $-s$, donde s es un número no negativo, y un entero (el *exponente*) e . El número correspondiente es $\pm s \cdot b^e$, donde b es la base elegida (no necesariamente igual a B).

2. Siendo s_{min} , s_{max} , e_{min} y e_{max} los valores mínimos y máximos de s y e , respectivamente. El rango de números representados es

$$-s_{max} \cdot b^{e_{max}} \leq x \leq s_{max} \cdot b^{e_{max}} \quad (\text{A.4})$$

y el valor mínimo absoluto de un número representado es

$$|x| \geq s_{min} \cdot b^{e_{min}} \quad (\text{A.5})$$

3. Siendo ulp la unidad en la posición menos significativa del significante. Entonces la *distancia* D entre números representados exactamente es $D = d \cdot b^e$, donde $d = ulp$ es la distancia entre dos valores sucesivos del significante. Así el valor de D depende del exponente e . El *error máximo* es igual a

$$D_{max}/2 = ulp \cdot b^{e_{max}}/2 \quad (\text{A.6})$$

4. El *error relativo máximo* es igual a $D/(2 \cdot |x|) = ulp \cdot b^e / (2 \cdot s \cdot b^e) = ulp/2 \cdot s$. El error relativo máximo es menor o igual a $1/2$.

A.2. Polinomio Schmidt de Legendre

Los polinomios de Legendre son soluciones a la ecuación diferencial asociada de Legendre:

$$\frac{d}{dx} \left[(1-x^2) \frac{dy}{dx} \right] + \left[l(l+1) - \frac{m^2}{1-x^2} \right] y = 0 \quad (\text{A.7})$$

Donde l es un número entero positivo y $m = 0, \dots, l$.

El polinomio Schmidt asociado de Legendre se define de la siguiente forma:

$$P_n^m(x) = (-1)^m (a-x^2)^{m/2} \frac{d^m}{dx^m} \frac{1}{2^n n!} \left[\frac{d^n}{dx^n} (x^2-1)^n \right] \quad (\text{A.8})$$

Apéndice B

Códigos *main* de la Simulación e Implementación

B.1. Función *mainSoftware.m* de Simulación en Matlab

```
1 % Simulacion algoritmo ADCS de un satelite
2 % micro usando Bobinas Magneticas como Actuador.
3 % Simulacion realizada por Angel Mancilla Carrasco
4 % FI UNAM 2014
5 clear all
6 clc
7 close all
8 fprintf('————— \n')
9 fprintf('— ADCS \n');
10 fprintf('— AMC 303084206 \n');
11 fprintf('— Starting variables...\n');
12 Ix = 9.097;
13 Iy = 9.987;
14 Iz = 3.667;
15 InertialVector = [Ix Iy Iz];
16 InertialMatrix = [Ix 0 0; 0 Iy 0; 0 0 Iz]/10;
17 IM = InertialMatrix;
18 InverseIM = InvMatrix3x3(IM);
19 tau_G = [0 0 0]';
20 tau_Gv2 = [0 0 0]';
21 tau_M = [0 0 0]';
22 B_b = [0 0 0];
23 B_o = [0 0 0];
24 B_bv2 = [0 0 0];
25 B_ov2 = [0 0 0];
```

```

26 Heigh = 600e3;
27 Mass_earth = 5.9742e24;
28 G = 6.67*10^-11;
29 Heighfromcenter = 6.378e6 + Heigh;
30 omega_0 = sqrt(G*Mass_earth/(Heighfromcenter^3));
31 %phi = -45.125;
32 %theta = -45.987;
33 %psi = 48.978;
34 % phi = 12.825;
35 % theta = 15.987;
36 % psi = -8.197;
37 % phi = 7.99;
38 % theta = -5.987;
39 % psi = 6.197;
40 phi = 25;
41 theta = 25;
42 psi = 25;
43 deg_0 = [phi; theta; psi]*pi/180;
44 tau = [0 0 0]';
45 quat_0 = euler2quat(deg_0);
46 quat = quat_0;
47 quat_dot = 0;
48 [R_B_O] = RotationMatrix(quat_0);
49 c1 = R_B_O(:,1);
50 w_B_OB = [0 0 0]';
51 w_B_IB_0 = w_B_OB+omega_0*c1;
52 w_B_IB = w_B_IB_0;
53 tm = 10000;
54 t = 1:tm;
55 euler = deg_0;
56 for time = 1:tm
57 fprintf('euler initial\n');
58 fprintf('%f %f %f \n', phi, theta, psi);
59 [w_B_IB, w_B_IB_dot] = dynamics(tau ,IM, InverseIM, w_B_IB);
60 [quat, quat_dot, R_B_O, w_B_OB] = kinematics(w_B_IB, quat, omega_0);
61 xvector = [quat(2) quat_dot(2) quat(3) quat_dot(3) quat(4) quat_dot(4) ];
62 [tau_G, B_b, B_o] = environmentmodel(R_B_O, omega_0, time, InertialMatrix, ...
    Heigh);
63 [m,lqrrr] = lqrcontrol(xvector, B_o, omega_0, InertialVector,time);
64 tau_M = cross(m, B_b);
65 tau = tau_M + tau_G;
66 euler = qua2eul(quat);
67 end
68 }

```

B.2. Funciones utilizadas por *mainSoftware.m* de Simulación

```
1 function [q] = euler2quat(A)
2 function [eul] = qua2eul(qua)
3 function [m, lqrr] = lqrcontrol(xvector, b_o, omega_o, InertialVec, tiempo)
4 function [q_new, q_dot_new, r_b_o, w_b_ob] = kinematics(w_b_ib, q_old, w_0)
5 function [tau_g, b_b, b_o] = environmentmodel(r_b_o, ...
        omega_o, timein, imatrix, heigh)
6 function [w_B_IB, w_B_IB_dot] = dynamics(tau, IM, InverseIM, w_B_IB_old)
```

B.3. Función *plotData.m* para graficar datos de XUPV2P

```
1 function plotData(archivo, xlabe, xunid, ylabe, yunid, legenda, decimar)
2 % Funcion disenada por Angel Mancilla Carrasco
3 % FI UNAM 2014
4 % Funcion para graficar datos de archivo.dat
5 function plotData(archivo, xlabe, xunid, ylabe, yunid, legenda, decimar)
6 dato = ...
7     strcat('/home/elN/Dropbox/MATLAB/CodingTesis/PlotingData/',archivo, '.dat');
8 titulo = strcat(ylabe, ' ', '_V_S ', ' ', xlabe);
9 tituloz = strcat('zoom', ' ', titulo);
10 colorvec = ['—or'; '—xb'; '—^g'; '—.k'];
11 load (dato)
12 [mm nn] = size(eval(archivo));
13 figure
14 for n = 1:nn
15     eval([archivo, int2str(n), '=', archivo '(:,n);']);
16     limitex = length(decimate(eval([archivo, int2str(n)]),decimar))
17     plot(decimate(eval([archivo, ...
18         int2str(n)]),decimar),colorvec(n,:), 'MarkerSize',7); hold on; % ...
19         xlim([0 limitex]);
20 end
21 title(titulo, 'fontSize',16, 'FontAngle', 'italic'); xlabel(strcat(xlabe,xunid), 'fontSize',16);
22     ylabel(strcat(ylabe,yunid), 'fontSize',16); grid minor; grid on;
23 hleg = legend(legenda); ...
24     set(hleg, 'FontAngle', 'italic', 'fontSize',16, 'TextColor', [.2,.1,.05]);
25 set(gca, 'XTickLabel', {'0', '125', '250', '375', '500', '625', '750', '875', '1000', '1125', '1250'});
26
27 figure; title(titulo, 'fontSize',16, 'FontAngle', 'italic'); grid minor; hold on;
28 for m = 1:nn
29     if m == 1
30         subplot(nn,1,m); xlabel(strcat(xlabe,xunid), 'fontSize',16); ...
31             ylabel(legenda(m,:), 'fontSize',16); grid minor; grid on; hold on;
32         title(titulo, 'fontSize',16, 'FontAngle', 'italic'); grid minor; hold on;
33     end
34     subplot(nn,1,m); xlabel(strcat(xlabe, ' [s]'), 'fontSize',16); ...
35         ylabel(strcat(legenda(m,:),yunid), 'fontSize',16); grid minor; grid on; ...
36         hold on;
37     plot(decimate(eval([archivo, int2str(m)]),decimar), ...
38         colorvec(m,:), 'MarkerSize',12); hleg = legend(legenda(m,:)); ...
39         set(hleg, 'FontAngle', 'italic', 'fontSize',16, 'TextColor', [.2,.1,.05]); ...
40         %xlim([0 limitex]);
41 set(gca, 'XTickLabel', {'0', '125', '250', '375', '500', '625', '750', '875', '1000', '1125', '1250'});
42 end
43 figure
44 for n = 1:nn
```

```

35 eval([archivo, int2str(n), '=', archivo '(:,n);']);
36 limitex = length(decimate(eval([archivo, int2str(n)]),decimar))
37 plot(decimate(eval([archivo, ...
    int2str(n)]),decimar),colorvec(n,:), 'MarkerSize',14); hold on; ...
    xlim([(limitex-limitex/5) limitex]);
38 end
39 title(titulo, 'fontSize',16, 'FontAngle', 'italic');xlabel(strcat(xlabe,xunid), 'fontSize',16);
    ylabel(strcat(ylabe,yunid), 'fontSize',16); grid minor; grid on;
40 hleg = legend(legenda); ...
    set(hleg, 'FontAngle', 'italic', 'fontSize',16, 'TextColor', [.2,.1,.05]);
41 set(gca, 'XTickLabel', {'1000', '1050', '1100', '1150', '1200', '1250'});
42 end

```

B.4. Código *mainadcs.c* implementado en XUPV2P

```
1 #include "xparameters.h"
2 #include "mb_interface.h"
3 #include "stdio.h"
4 #include "math.h"
5 #include "xbasic_types.h"
6 #include <stdlib.h> /* abs */
7 ///////////////////////////////////////////////////////////////////
8 //// Algoritmo ADCS con bobinas magneticas como actuador
9 //// XUPV2P
10 //// Diseñado y programado por Angel Mancilla Carrasco
11 //// Facultad de Ingenieria
12 //// Universidad Nacional Autonoma de Mexico
13 //// 2014
14 /////////////////////////////////////////////////////////////////// functions
15 void prnt_flo(float *a, char b);
16 void VecCross(float A[3], float B[3], float C[3]);
17 void Matrix3x3Sum(float A[3][3], float B[3][3], float C[3][3]);
18 void MatrixMulSc(float A[3][3], float B, float C[3][3]);
19 void MatrixMul3x3(float A[3][3], float B[3][3], float C[3][3]);
20 void MatrixMul3x1(float A[3][3], float B[3], float C[3]);
21 void MatrixMul6x1(float A[3][6], float B[6], float C[3]);
22 void MatrixTranspose(float A[3][3], float B[3][3]);
23 void InvMatrix3x3(float A[3][3], float InverseA[3][3]);
24 void printMatrix(float A[3][3], char B[15]);
25 void printMatrix3x6(float A[3][6]);
26 void printVect(float a[], int tamano);
27 int Massimo(float arr[], int tamano, float *massimo);
28 int Minnimo(float arr[], int tamano, float *minnimo);
29 void skewsymmetric(float vec[3], float r[3][3]);
30 void RotMatrix(float q[4], float rob[3][3]);
31 void e2q(float eul[3], float qu[4]);
32 void q2e(float rbo[3][3], float eu[3]);
33 void polinomialF(float covect[], int tiempo, float *element, int orden, ...
    char f);
34 void dynamics(float tau[3], float IM[3][3], float invIM[3][3], float ...
    w_B_IB_old[3], float w_B_IB[3], float w_B_IB_dot[3]);
35 void kinematics(float wbib[3], float qold[4], float q[4], float qdot[4], ...
    float rbo[3][3]);
36 void statevector(float x[4], float xd[4], float estados[6]);
37 void envModel(float rbo[3][3], float omega_o, int tiempo, float ...
    imatrix[3][3], float taue[3], float bb[3]);
38 void lqr(int tiempo, float xvec[6], float eme[3]);
39 void fourier(float covect[], int tiempo, float *element, int orden);
40 /////////////////////////////////////////////////////////////////// functions
41 /////////////////////////////////////////////////////////////////// variables
42 float pi = 3.14159265358979;
43 int main (void){
```