

CONTROL ALLOCATION AGAINST ACTUATOR FAILURES
IN OVERACTUATED SMALL SATELLITES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ÖZGÜR KAHRAMAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
AEROSPACE ENGINEERING

NOVEMBER 2007

Approval of the thesis:

**CONTROL ALLOCATION AGAINST ACTUATOR FAILURES
IN OVERACTUATED SMALL SATELLITES**

submitted by **ÖZGÜR KAHRAMAN** in partial fulfillment of the requirements for the degree of **Master of Science in Aerospace Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Smail H. Tuncer
Head of Department, **Aerospace Engineering**

Prof. Dr. Ozan Tekinalp
Supervisor, **Aerospace Engineering Dept., METU**

Examining Committee Members:

Assoc. Prof. Dr. Altan Kayran
Aerospace Engineering Dept., METU

Prof. Dr. Ozan Tekinalp
Aerospace Engineering Dept., METU

Prof. Dr. Kemal Özgören
Mechanical Engineering Dept., METU

Dr. Güçlü Seber
Aerospace Engineering Dept., METU

Dr. Volkan Nalbanto lu
ASELSAN

Date:

22.11.2007

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name :

Signature :

ABSTRACT

CONTROL ALLOCATION AGAINST ACTUATOR FAILURES IN OVERACTUATED SMALL SATELLITES

Kahraman, Özgür

M.S, Department of Aerospace Engineering

Supervisor: Prof. Dr. Ozan Tekinalp

November 2007, 91 pages

In this thesis, attitude control of small satellites with dissimilar actuator is studied and the effects of control allocation methods on maneuvering are examined in detail. Magnetorquers and reaction wheels are considered as the actuators of a modeled remote sensing -nadir pointing- small satellite. Matlab[®] Simulink simulation models are developed to model the satellite dynamics and the actuators on the satellite. The simulations are based on conceptual RASAT satellite, and, for verification, orbit data is taken from BILSAT satellite that is operated by TUBITAK Space Research Institute.

Basic satellite control modes are developed and tested to obtain nominal control. Actuator failures are analyzed for different possible cases.

A control allocation method called Blended Inverse that was originally proposed for steering CMGs is applied to select the actuators to avoid actuator saturation and singularity transition. The performance of traditional pseudo inverse method is compared with the blended inverse method and simulation results are given and discussed. The superiority of blended inverse over pseudo inverse is demonstrated.

Keywords: Satellite Attitude Control, Actuator Failure, Control Allocation Method

ÖZ

ARTIK EYLEYİCİLİK KÜÇÜK UYDULARDA EYLEYİCİ ARIZALARINA KARŞI KONTROL DA İTİMİ

Kahraman, Özgür

Yüksek Lisans, Havacılık ve Uzay Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Ozan Tekinalp

Kasım 2007, 91 sayfa

Bu tezde benzer meyden eyleyicilere sahip küçük uyduların yönelim kontrolü çalı ılımlı ve kontrol da ıtım yöntemlerinin manevra üzerine etkileri ayrıntılı olarak incelenmiştir. Ayakucu do rultulu, uzaktan algılamalı bir küçük uydu için, eyleyici olarak tork çubukları ve tepki tekerleri göz önünde bulundurulmuştur. Uydu dinamiğini ve eyleyicileri modellemek için Matlab® Simulink benzetim modelleri geliştirilmiştir. Benzetimler, kavramsal RASAT uydusu baz alınarak yapılmış ve do rulama için yörünge bilgileri, TÜB TAK Uzay Teknolojileri Enstitüsü tarafından iletilen B LSAT uydusundan alınmıştır.

Nominal kontrol sağlamak için, temel uydu kontrol kipleri geliştirilip test edilmiştir. Eyleyici hataları mümkün olan farklı durumlar için analiz edilmiştir.

Bir kontrol da ıtım yöntemi olan, orjinal olarak CMG yöneltimi için önerilmiş harmanlanmış ters alma yöntemi, eyleyici saturasyonunu engellemek ve tekilliklerden kaçınmak amacıyla eyleyiciler arasında kontrol da ıtımı için uygulanmıştır. Geleneksel sanki ters alma yöntemi ile harmanlanmış ters alma yöntemi kıyaslanarak benzetim sonuçları verilmiştir ve tartışılmıştır. Harmanlanmış ters alma yönteminin sanki ters alma yöntemi üzerindeki üstünlüğü gösterilmiştir.

Anahtar Kelimeler: Uydu yönelim kontrolü, eyleyici hatası, kontrol da ıtım yöntemi

To My Parents

ACKNOWLEDGMENTS

I wish to express my deepest gratitude to my supervisor Prof. Dr. Ozan Tekinalp for his suggestions and comments.

I would like to thank Nahit Ertongur for his guidance, advice, criticism, encouragements and insight throughout the research.

Special thanks to my colleagues in TÜB TAK–UZAY Satellite Technologies Group for their great support for the completion of this thesis.

I would also like to express my appreciation to my family, my fiancée Duygu, and my sincere friends for their continuous support and understanding.

TABLE OF CONTENTS

ABSTRACT	iv
ÖZ	v
ACKNOWLEDGMENTS	vii
TABLE OF CONTENTS	viii
LIST OF TABLES	xi
LIST OF FIGURES	xii
LIST OF SYMBOLS	xv

CHAPTER

1. INTRODUCTION	1
2. BUILDING A SATELLITE SIMULATION MODEL	3
2.1. Coordinate Transformations.....	3
2.1.1. Euler Sequence to Direction Cosine Matrix	3
2.1.2. Relation between Euler Sequence and Rotation Quaternions.....	5
2.1.3. Quaternion to Direction Cosine Matrix	6
2.2. General Equations of Motion of a Satellite.....	7
2.2.1. Dynamic Equations.....	7
2.2.2. Kinematic Equations.....	7
2.2.3. Quaternion Error.....	8
2.3. Orbit Models.....	9
2.3.1. Orbit Model: SGP4	9
2.3.2. NORAD Two Line Element Set.....	10
2.3.3. Magnetic Field Model: IGRF.....	12
2.3.4. Disturbances	15
2.3.4.1. Gravity Gradient Torque	15
2.3.4.2. Magnetic Torque	17
2.3.4.3. Other Disturbance Torques.....	17

2.4.	Sensor Modeling.....	18
2.4.1.	Magnetometer Modeling.....	18
2.4.2	Sun Sensor Modeling.....	24
2.4.2.1.	Sun Position Vector Estimation	25
2.4.2.2.	Transformation of Sun Position Vector to Satellite Body Frame.....	25
2.4.2.3.	Eclipse Times Estimation	25
2.4.2.4.	Sun Sensor Simulators.....	25
2.4.2.5.	Installation Matrix of the sensors.....	25
2.4.2.6.	Sensor limits and Field of Views	26
2.4.2.7.	Sensor blocks	26
2.4.2.8.	Simulation Results	27
2.5.	Actuator Modeling.....	30
2.5.1.	Magnetorquer Modeling	30
2.5.2.	Reaction Wheel Modeling	35
2.5.2.1.	Reaction Wheel Configuration	35
2.5.2.2.	Reaction Wheel Model	39
3.	BASIC ATTITUDE CONTROL MODES	42
3.1.	Detumbling Control Mode	42
3.1.1.	B-dot Control	42
3.1.2.	Stability of the B-dot controller	43
3.1.3.	Simulation Model for B-dot Controller	44
3.2.	Nominal Control Mode: 3 Axis Attitude Control	49
3.2.1	Desaturation of Reaction Wheels by Magnetorquers	51
4.	ACTUATOR FAILURE COMPENSATION THROUGH CONTROL ALLOCATION METHODS	57
4.1.	Literature Survey	57
4.2.	Control Allocation Methods.....	60
4.3.	Actuator Failure Cases.....	62
4.3.1.	Attitude Control by 3 Reaction Wheels	62

4.3.2. Slew Maneuver with 2 Magnetorquers and a Y axis Reaction Wheel.....	62
4.3.3. Slew Maneuver with 3 Magnetorquers and Y axis Reaction Wheel	65
4.3.4 3 Magnetorquers and two Reaction Wheel Control Mode	72
5. CONCLUSION	80
REFERENCES.....	82
APPENDIX.....	85

LIST OF TABLES

TABLE

Table 2.1 Two Line Element Set Format.....	10
Table 2.2 Two Line Element Set Format Descriptions	11
Table 2.3 BILSAT Satellite Example TLE Set	12
Table 2.4 Satrec Initiative FSS-05 Sun Sensor Specifications [10]	24
Table 2.5 MTR 10 Magnetometer Specifications [11]	31
Table 3.1 PD Control Strategy	51

LIST OF FIGURES

FIGURES

Figure 2.1 Euler Angles	4
Figure 2.2 Euler Angles to Quaternions	5
Figure 2.3 Earth's Magnetic Field Dipole Model	13
Figure 2.4 Magnitude of the Earth's Magnetic Field	14
Figure 2.5 Torque about the Geometric Center.....	15
Figure 2.6 Sensor Models Simulink Diagram.....	18
Figure 2.7 Magnetometer Model Simulink Model.....	19
Figure 2.8 Magnetic Field Vector wrt Body Fixed Frame (IGRF Model with noise)	20
Figure 2.9 Magnetic Field Vector Estimated By Approximate Model wrt Body Fixed Frame	21
Figure 2.10 Normalized Magnetic Field Vector wrt Body Frame	22
Figure 2.11 Magnetic Field Vector wrt ECEF Frame	23
Figure 2.12 Magnetic Field Vector wrt ECI Frame.....	23
Figure 2.13 Satrec Initiative FSS_05 Analog Sun Sensor	24
Figure 2.14 Sun Sensor Model Simulink Diagram	28
Figure 2.15 Simulation Results of Azimuth and Elevation Angles.....	29
Figure 2.16 Eclipse times during the simulation	29
Figure 2.17 Selected Sun sensor heads for estimation	30
Figure 2.18 MTR 10 Magnetorquer	32
Figure 2.19 Magnetorquer Simulink Model	34
Figure 2.20 Magnetorquer Effect	35
Figure 2.21 Reaction Wheel Configuration	36
Figure 2.22 Side views [2]	36
Figure 2.23 Tetrahedral Geometry [2]	37
Figure 2.24 SIMULINK Reaction Wheel Model.....	40

Figure 2.25 Reaction Wheel Model with Saturation Control	41
Figure 2.26 Model For the Reaction Wheel Configuration	41
Figure 3.1 B-dot Estimator Simulink Diagram	44
Figure 3.2 Attitude rates of the satellite during detumbling mode	46
Figure 3.3 Magnetic moment that is commanded for the magnetorquers	46
Figure 3.4 Torques generated by magnetorquers	47
Figure 3.5 Detumbling Control Simulink Diagram.....	48
Figure 3.6 Attitude Control through Reaction Wheels	49
Figure 3.7 Reaction Wheel PD Controller	50
Figure 3.8 Magnetorquer Control for Reaction Wheel Desaturation	51
Figure 3.9 Control Current Direction Determination	52
Figure 3.10 Magnetorquer PD Controller	53
Figure 3.11 Euler Angles during desaturation mode	54
Figure 3.12 Reaction Wheel Torques in Nominal Mode	54
Figure 3.13 Magnetic Torques in Nominal Mode	55
Figure 3.14 Nominal Control Mode Simulink Diagram.....	56
Figure 4.1 Euler angles time histories after 20^0 commanded roll angle using blended inverse in singular zone with two magnetorquers and a reaction wheel	63
Figure 4.2 Magnetic Moments for 2 Magnetorquers and Y Wheel Mode	64
Figure 4.3 Magnetic Torques for 2 Magnetorquers and Y Wheel Mode	64
Figure 4.4 Reaction Wheel Torques for 2 Magnetorquers and Y Wheel Mode	65
Figure 4.5 A 20^0 roll maneuver with pseudo inverse at a singular zone	66
Figure 4.6 Magnetic moments, Y axis reaction wheel torque and the singularity measure during 20^0 roll maneuver using pseudo inverse algorithm in a singular zone	67
Figure 4.7 Euler angles during 20^0 Roll maneuver with blended inverse algorithm in a singular zone.....	68
Figure 4.8 Magnetic moments, Y axis reaction wheel torque and the singularity measure during 20^0 roll maneuver using blended inverse algorithm in a singular zone	69
Figure 4.9 Euler angles time histories after 20^0 commanded roll, pitch and yaw angles using pseudo inverse in a singular zone	70

Figure 4.10 Magnetic moments, Y axis reaction wheel torque and the singularity measure during 20 ⁰ RPY maneuver using pseudo inverse algorithm in a singular zone	71
Figure 4.11 Euler angles time histories after 20 ⁰ commanded roll, pitch and yaw angles using blended inverse in singular zone	71
Figure 4.12 Magnetic moments, Y axis reaction wheel torque and the singularity measure during 20 ⁰ RPY maneuver using blended inverse algorithm in a singular zone	72
Figure 4.13 Desired Torques for 2 Reaction Wheels and 3 Magnetorquers Mode ..	73
Figure 4.14 RPY for 2 Reaction Wheels and 3 Magnetorquers Mode	73
Figure 4.15 Angular Velocity for 2 Reaction Wheels and 3 Magnetorquers Mode ..	74
Figure 4.16 Magnetic Moments for 2 Reaction Wheels and 3 Magnetorquers Mode	74
Figure 4.17 Magnetic Torques for 2 Reaction Wheels and 3 Magnetorquers Mode ..	75
Figure 4.18 RW Torques for 2 Reaction Wheels and 3 Magnetorquers Mode	76
Figure 4.19 Singularity Check for Blended Inverse	77
Figure 4.20 Magnetic Moments in Blended Inverse Mode	78
Figure 4.21 Magnetic Torques in Blended Inverse Mode	78
Figure 4.22 Reaction Wheels Torques in Blended Inverse Mode	79
Figure 4.23 Reaction Wheel Torques in RW Frame	79

LIST OF SYMBOLS

ϕ	Roll angle with respect to satellite body frame
θ	Pitch angle with respect to satellite body frame
ψ	Yaw angle with respect to satellite body frame
R_x	Rotation matrix around X axis
R_y	Rotation matrix around Y axis
R_z	Rotation matrix around Z axis
μ	Unit vector
ω_{BI}	Spacecraft angular rates with respect to satellite body frame
I	Inertia matrix
$T_{magnetic}$	Magnetic control torques
T_{wheel}	Reaction wheel torques
$T_{disturbance}$	Disturbance torques
ω_B^O	Body angular rate referenced to orbital frame
q_e	Attitude Quaternion Error
q_c	Quaternion Command
\vec{B}	Geomagnetic induction vector
V	Scalar potential
dF	Gravitational force
\hat{N}_{GG}	Gravity gradient torque
M	Magnetic moment
I_D	Magnetorquer current
S	Scaling factor for magnetorquer
\vec{N}_M	Effectuated magnetic torque during the orbit
MT_{real}^t	Applied magnetic moment with delay
$MT_{commanded}^t$	Commanded magnetic moment
T	Time coefficient of delay function

L	Magnetic inductance
H	Hamiltonian
h	Total angular momentum
TF_{RW}	Laplace transform of the transfer function of a reaction wheel
K_d	Positive controller derivative gain
$\dot{\tau}$	Time derivative of rotational kinetic energy
ω_{err}	Difference between commanded angular velocity and estimated angular velocity wrt body fixed frame of the satellite.
$T_{control}$	Control torque
U_{pseudo}	Control input matrix with Pseudo inverse
J	Jacobian Matrix
$U_{desired}$	Desired Control Input Matrix
β	Smoothing Gain

CHAPTER 1

INTRODUCTION

Attitude Determination and Control Systems (ADCS) have critical role in Low Earth Orbit (LEO) small satellite missions. The success of the mission depends on the capability of these systems. The ADCS systems consist of two main parts: ADCS hardware and ADCS software.

ADCS hardware is formed by sensors and actuators. The sensors obtain the attitude measurements for the estimation of attitude knowledge such as satellite Euler angles, quaternions and the angular rates. Most fundamental sensors that are used in the satellites are the magnetometers for magnetic field measurements, Sun sensors for Sun reference, star trackers for star catalog matching and the rate gyros for the angular measurements. All these sensors provide an attitude reference for the attitude estimation algorithms of the ADCS software. The main purpose of the determination algorithms is to combine the attitude measurements and estimate the attitude through estimation algorithms such as a Kalman filter. Estimated angles and the rates are used as feedback signals for the control system.

The actuators form the other half of the ADCS hardware. The main goal is achieving the control objectives during the operational life such as detumbling to point the satellite to the nadir direction and the three axis control for all the nominal operations such as maneuvering. The hardware is composed of reaction wheels that are used as the main control hardware in nominal mode and the magnetorquers that are the only actuators which have the control capability during detumbling mode. Magnetorquers are also used for desaturation of reaction wheels when the wheels are about to exceed their torque and momentum limits. Although control moment gyros are used for fast maneuvers, in many satellites, they are not considered in this thesis.

Actuator failures cause unexpected control problems during operational life of the satellite. In this thesis, reaction wheels are considered as positioned in tetrahedral configuration for the nominal three axis control mode and all the failure modes are composed of reaction wheel failures. The failure detection is not evaluated in this work and it is assumed that the failures of such components can be detected by checking the telemetry data to be sure of the physical failure. Thus the controller configuration is assumed to be changed manually when the new ADCS software is being uploaded from the ground station.

The underactuated conditions are covered in the literature by investigating both controllability and stability issues. However in this thesis, overactuated conditions are examined for the control allocation problem such as the control mode with one reaction wheel and three magnetorquers and the case with two reaction wheels and three magnetorquers. The stability conditions are proven in the literature for these scenarios thus the basic PD control scheme is applied to achieve the attitude control of the satellite.

The main objective of this thesis is to examine the utilization of the steering algorithms to overcome the control allocation problem in over actuated small satellites. In this thesis, the steering law called the Blended Inverse, which is proposed by Tekinalp and Yavuzoglu [5] to avoid from singularities or quick transition from the singularity zone, is implemented. Basic pseudo inverse method is compared to Blended inverse method for different scenarios through simulations.

The Matlab Simulink simulation is used as a simulation environment and the S functions are used to implement the codes that are written in both C and Matlab programming languages. All the satellite components and the environment are modeled to simulate the maneuvers.

In the following chapters first the satellite simulation model is given, followed by the basic attitude control modes and their implementation. Chapter 4 is devoted to the steering of overactuated small satellites with magnetorquers and reaction wheels. Simulation results are given and discussed. Final section contains same concluding remarks and future work.

CHAPTER 2

BUILDING A SATELLITE SIMULATION MODEL

In this section, satellite simulation model is presented. The model includes attitude and orbital motion models. Earth's magnetic field, sensor and actuator models are also given.

2.1. Coordinate Transformations

Coordinate transformations between the reference frames are frequently used for the satellite simulations. In this work, Euler angles, direction cosine matrix and quaternions representations are used basically.

2.1.1. Euler Sequence to Direction Cosine Matrix

In this thesis, the user inputs to the model for attitude requirements are expected as Euler rotation angles sequence. Although Euler angles are not employed in dynamic equations due their singularities, the choice of representation by Euler sequence stems from their relatively easy visualization. 2-1-3 Euler sequence is used which represents first a rotation around pitch axis (θ), a second rotation around roll axis (ϕ) and a final rotation around yaw axis (ψ) respectively in classical sense. Although 3-2-1 notation is used in the flight mechanics, especially for the geosynchronous satellites, 2-1-3 notation is used in LEO satellites. The reason for using 2-1-3 notation is that has a physical meaning for the satellite dynamics. Pitch axis of the satellite is perpendicular to the orbital plane and the rotation around that axis presents the azimuth changes and the rotation around the roll axis presents the elevation changes which are used to define the swath width of the satellite.

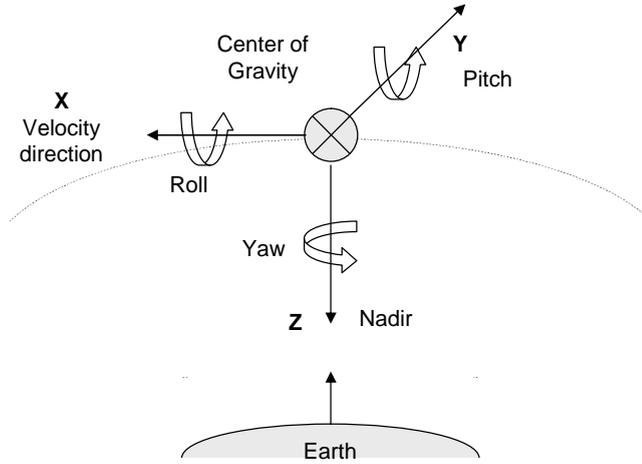


Figure 2.1 Euler Angles

Conversion from this sequence to a single 3-by-3 rotation matrix or Direction Cosine Matrix (DCM) is as follows;

Rotation matrices for each rotation are,

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \quad R_y = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$R_z = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

The corresponding DCM for Euler 2-1-3 sequence will be,

$$DCM_{213} = R_z \cdot R_x \cdot R_y \quad (2.2)$$

Substituting respective rotation matrices, we get;

$$DCM_{213} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \quad (2.3)$$

$$DCM_{213} = \begin{bmatrix} \cos\psi \cos\theta + \sin\psi \sin\phi \sin\theta & \sin\psi \cos\phi & -\cos\psi \sin\theta + \sin\psi \sin\phi \cos\theta \\ -\sin\psi \cos\theta + \cos\psi \sin\phi \sin\theta & \cos\psi \cos\phi & \sin\psi \sin\theta + \cos\psi \sin\phi \cos\theta \\ \sin\theta \cos\phi & -\sin\phi & \cos\theta \cos\phi \end{bmatrix} \quad (2.4)$$

2.1.2. Relation between Euler Sequence and Rotation Quaternions

An alternative way to represent a rotation is using quaternions. A quaternion basically stands for a vector and an angle which together correspond to a rotation about a unit vector $[\mu_x \ \mu_y \ \mu_z]$ through an angle θ .

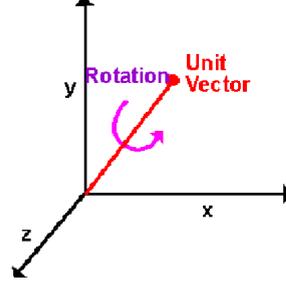


Figure 2.2 Euler Angles to Quaternions

A unit quaternion has unit magnitude, and can be written in the following vector format.

$$q = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos(\theta/2) \\ \sin(\theta/2)\mu_x \\ \sin(\theta/2)\mu_y \\ \sin(\theta/2)\mu_z \end{bmatrix} \quad (2.5)$$

An alternative representation of a quaternion is as a complex number,

$$q = q_0 + iq_1 + jq_2 + kq_3 \quad (2.6)$$

Again quaternions are commonly used to compensate the singularities introduced by Euler angles. The corresponding quaternion for the Euler 2-1-3 sequence can be calculated as [1];

$$q = q_\theta q_\phi q_\psi = [\cos(\theta/2) - \vec{j} \sin(\theta/2)] \otimes [\cos(\phi/2) - \vec{i} \sin(\phi/2)] \otimes [\cos(\psi/2) - \vec{k} \sin(\psi/2)] \quad (2.7)$$

Where,

$$\begin{aligned} \vec{i} \otimes \vec{j} &= -\vec{j} \otimes \vec{i} = \vec{k} \\ \vec{j} \otimes \vec{k} &= -\vec{k} \otimes \vec{j} = \vec{i} \\ \vec{k} \otimes \vec{i} &= -\vec{i} \otimes \vec{k} = \vec{j} \end{aligned} \quad (2.8)$$

The quaternion multiplication can be restated as:

$$\vec{i} \otimes \vec{i} = \vec{j} \otimes \vec{j} = \vec{k} \otimes \vec{k} = -1 \quad (2.9)$$

Finally, the four elements of the quaternion are;

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos(\phi/2)\cos(\theta/2)\cos(\psi/2) + \sin(\phi/2)\sin(\theta/2)\sin(\psi/2) \\ \sin(\phi/2)\cos(\theta/2)\cos(\psi/2) - \cos(\phi/2)\sin(\theta/2)\sin(\psi/2) \\ \cos(\phi/2)\sin(\theta/2)\cos(\psi/2) + \sin(\phi/2)\cos(\theta/2)\sin(\psi/2) \\ \cos(\phi/2)\cos(\theta/2)\sin(\psi/2) - \sin(\phi/2)\sin(\theta/2)\cos(\psi/2) \end{bmatrix} \quad (2.10)$$

2.1.3. Quaternion to Direction Cosine Matrix

Rotation of a vector using quaternion algebra is given by the following relationship:

$$\vec{X}' = q \otimes \vec{X} \otimes q^c \quad (2.11)$$

where,

$$\begin{aligned} q &= q_0 + \vec{i}q_1 + \vec{j}q_2 + \vec{k}q_3 \\ q^c &= q_0 - \vec{i}q_1 - \vec{j}q_2 - \vec{k}q_3 \\ \vec{X} &= 0 + \vec{i}X_1 + \vec{j}X_2 + \vec{k}X_3 \end{aligned} \quad (2.12)$$

Rearranging Equation (2.11); \vec{X}' can be written as follows,

$$\vec{X}' = \begin{bmatrix} 0 \\ X_1' \\ X_2' \\ X_3' \end{bmatrix} = \begin{bmatrix} 0 \\ (q_0^2 + q_1^2 - q_2^2 - q_3^2)X_1 + 2(q_1q_2 - q_0q_3)X_2 + 2(q_1q_3 + q_0q_2)X_3 \\ 2(q_0q_3 + q_1q_2)X_1 + 2(q_0^2 - q_1^2 + q_2^2 - q_3^2)X_2 + 2(q_2q_3 - q_0q_1)X_3 \\ 2(q_1q_3 - q_0q_2)X_1 + 2(q_0q_1 + q_2q_3)X_2 + (q_0^2 - q_1^2 - q_2^2 + q_3^2)X_3 \end{bmatrix} \quad (2.13)$$

Then the DCM to realize this rotation can be determined by observing Equation

(2.13):

$$DCM = \begin{bmatrix} (q_0^2 + q_1^2 - q_2^2 - q_3^2) & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_0q_3 + q_1q_2) & 2(q_0^2 - q_1^2 + q_2^2 - q_3^2) & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_0q_1 + q_2q_3) & (q_0^2 - q_1^2 - q_2^2 + q_3^2) \end{bmatrix} \quad (2.14)$$

2.2. General Equations of Motion of a Satellite

2.2.1. Dynamic Equations

The attitude dynamics of the satellite can be expressed as the following formula;

$$I\dot{\omega}_{BI} = -\omega_{BI} \times I\omega_{BI} + T_{magnetic} + T_{wheel} + T_{disturbance} \quad (2.15)$$

where ω_{BI} is spacecraft angular rates with respect to satellite body frame, I is the inertia matrix that is taken from BILSAT satellite, $T_{magnetic}$ is the magnetic control torques that are generated by Magnetorquers, T_{wheel} is the reaction wheel torques and the $T_{disturbance}$ is the disturbance torques.

According to Wertz [1], dynamic and kinematic equations must be solved simultaneously that is in general, applied torques depends on the spacecraft attitude.

2.2.2. Kinematic Equations

By using quaternion representation we can obtain the attitude kinematic equations of the satellite [4];

$$\dot{q} = \frac{1}{2}\Omega q = A(q)\omega_B^O \quad (2.16)$$

where;

$$\Omega = \begin{bmatrix} 0 & \omega_{OZ} & -\omega_{OY} & \omega_{OX} \\ -\omega_{OZ} & 0 & \omega_{OX} & \omega_{OY} \\ \omega_{OY} & -\omega_{OX} & 0 & \omega_{OZ} \\ -\omega_{OX} & -\omega_{OY} & -\omega_{OZ} & 0 \end{bmatrix} \quad (2.17)$$

and,

$\omega_B^O = [\omega_{OX} \quad \omega_{OY} \quad \omega_{OZ}]^T$: Body angular rate referenced to orbital frame. Hence $A(q)$ becomes;

$$A(q) = \begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ q_1 & -q_2 & -q_3 \end{bmatrix} \quad (2.18)$$

For an Earth pointing (nadir pointing) satellite, the reference angular rate of the satellite is orbital rate vector,

$$\omega_o = [0 \quad -\omega_o \quad 0]^T \quad (2.19)$$

Therefore body angular rate referenced to orbital frame, w_B^O can be given by the difference between body angular rates referenced to inertial frame and the orbital rate in body frame;

$$\omega_B^O = \omega_B^I - A(q)\omega_o \quad (2.20)$$

The angular body rates in Equation (2.19) become,

$$\begin{aligned} \begin{bmatrix} \omega_{OX} \\ \omega_{OY} \\ \omega_{OZ} \end{bmatrix}_B^O &= \begin{bmatrix} \omega_X \\ \omega_Y \\ \omega_Z \end{bmatrix}_B^I - \hat{A} \begin{bmatrix} 0 \\ -\omega_o(t) \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} \omega_X \\ \omega_Y \\ \omega_Z \end{bmatrix}_B^I - \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \begin{bmatrix} 0 \\ -\omega_o(t) \\ 0 \end{bmatrix} = \begin{bmatrix} \omega_X \\ \omega_Y \\ \omega_Z \end{bmatrix}_B^I - \begin{bmatrix} -A_{12}\omega_o(t) \\ -A_{22}\omega_o(t) \\ -A_{32}\omega_o(t) \end{bmatrix} \end{aligned} \quad (2.21)$$

Finally the kinematic equations in terms of quaternions are;

$$\begin{aligned} \dot{q}_1 &= 0.5(\omega_{OZ} \cdot q_2 - \omega_{OY} \cdot q_3 + \omega_{OX} \cdot q_4) \\ \dot{q}_2 &= 0.5(-\omega_{OZ} \cdot q_1 + \omega_{OX} \cdot q_3 + \omega_{OY} \cdot q_4) \\ \dot{q}_3 &= 0.5(\omega_{OY} \cdot q_1 - \omega_{OX} \cdot q_2 + \omega_{OZ} \cdot q_4) \\ \dot{q}_4 &= 0.5(-\omega_{OX} \cdot q_1 - \omega_{OY} \cdot q_2 + \omega_{OZ} \cdot q_3) \end{aligned} \quad (2.22)$$

where, the angular rates in body frame are;

$$\begin{aligned} \omega_{OX} &= \omega_X + A_{12}\omega_o(t) \\ \omega_{OY} &= \omega_Y + A_{22}\omega_o(t) \\ \omega_{OZ} &= \omega_Z + A_{32}\omega_o(t) \end{aligned} \quad (2.23)$$

2.2.3. Quaternion Error

We have to define an attitude quaternion error that is used inside the control algorithms. The quaternion error will be the quaternion difference between the quaternion of the satellite at a particular time and the commanded quaternion. That is [6];

$$\begin{bmatrix} q_{1e} \\ q_{2e} \\ q_{3e} \\ q_{4e} \end{bmatrix} = \begin{bmatrix} q_{4c} & q_{3c} & -q_{2c} & -q_{1c} \\ -q_{3c} & q_{4c} & q_{1c} & -q_{2c} \\ q_{2c} & -q_{1c} & q_{4c} & -q_{3c} \\ q_{1c} & q_{2c} & q_{3c} & q_{4c} \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \quad (2.24)$$

where;

$$q_e = [q_{1e} \quad q_{2e} \quad q_{3e} \quad q_{4e}]^T \quad : \text{Attitude Quaternion Error}$$

$$q_c = [q_{1c} \quad q_{2c} \quad q_{3c} \quad q_{4c}]^T \quad : \text{Quaternion Command}$$

2.3. Orbit Models

The orbit models are required to simulate the satellite position and the velocity during the orbital motion. The SGP4 model is used in this work to generate position information for all sensor and actuator models. Thus the position of the Sun for Sun sensor simulation, the magnetic field model location data for the magnetometer model and the disturbance torques are estimated from that position and velocity value of the SGP4 propagator.

2.3.1. Orbit Model: SGP4

Our simulation model requires orbital information as an input to sub-models that deal with the Earth's magnetic field and gravity gradient effects on the satellite dynamics. Additionally satellite's orbital angular velocity is necessary information to keep track of the nadir direction for pointing requirements. In order to fulfill these requirements an orbit propagator needs to be implemented in the model.

There are a great number of orbit propagator models developed with various accuracy and complexity characteristics. However, most respected propagators are the ones developed by North American Air Defense (NORAD). Derived from the area of interest of NORAD, these propagator models are classified as near-Earth (with periods less than 225 minutes) and deep-space models, and there are five accepted mathematical models in record. [7]

The first of these models is the Simplified General Perturbations (SGP) developed in 1966, which uses a quadratic variation of mean anomaly with time and models the eccentric drag effect with a constant perigee height.

The next model SGP4, developed in 1970, which takes a more extensive theory on the gravitational model and atmospheric model into account. These two models were developed for near-Earth space. There is also SDP4 model for deep-space applications, near-Earth space model SGP8 which uses a different technique for differential equations and its counterpart SDP8 for deep-space applications. All these five models are compatible with NORAD two line element set. [9]

In this study, SGP4 propagator model was implemented, giving credit to its wide spread acceptance. Accordingly, the FORTRAN implementation of the model presented by Kelso was translated into C-code and embedded into an s-function SIMULINK block (Appendix). The input of this model is the initial NORAD two line element set which describes satellite position and velocity together with orbital geometry and time information.

2.3.2. NORAD Two Line Element Set

NORAD two line element set data for a satellite consists of three lines in the following format [9]:

Table 2.1 Two Line Element Set Format

<pre> AAAAAAAAAAAAAAAAAAAAAAAAA 1 NNNNU NNNNAAA NNNN.NNNNNNNN +.NNNNNNNN +NNNNN-N +NNNNN-N N NNNNN 2 NNNNN NNN.NNNN NNN.NNNN NNNNNNNN NNN.NNNN NNN.NNNN NN.NNNNNNNNNNNNNN </pre>
--

Line 0 is a twenty-four character name, and Lines 1 and 2 are the standard Two-Line Orbital Element Set Format used by NORAD and NASA. The format description is:

Table 2.2 Two Line Element Set Format Descriptions

Line 1	
Column	Description
01	Line Number of Element Data
03-07	Satellite Number
08	Classification (U=Unclassified)
10-11	International Designator (Last two digits of launch year)
12-14	International Designator (Launch number of the year)
15-17	International Designator (Piece of the launch)
19-20	Epoch Year (Last two digits of year)
21-32	Epoch (Day of the year and fractional portion of the day)
34-43	First Time Derivative of the Mean Motion
45-52	Second Time Derivative of Mean Motion (decimal point assumed)
54-61	BSTAR drag term (decimal point assumed)
63	Ephemeris type
65-68	Element number
69	Checksum (Modulo 10)
Line 2	
Column	Description
01	Line Number of Element Data
03-07	Satellite Number
09-16	Inclination [Degrees]
18-25	Right Ascension of the Ascending Node [Degrees]
27-33	Eccentricity (decimal point assumed)
35-42	Argument of Perigee [Degrees]
44-51	Mean Anomaly [Degrees]
53-63	Mean Motion [Revs per day]
64-68	Revolution number at epoch [Revs]
69	Checksum (Modulo 10)

For the model BILSAT's orbital parameters for a particular time were used as the initial inputs to the SGP4 model. The two line element set for BILSAT on 23 May 2005, 06:30:55 is as follows;

Table 2.3 BILSAT Satellite Example TLE Set

BILSAT 1	
1	27943U 03042E 05143.27147421 .00000100 00000-0 28805-4 0 7980
2	27943 098.1351 034.3744 0012522 125.8067 234.4294 14.62716601 88299

In this respect, the SGP4 model inputs are listed below;

- BStarDragTerm=0.28805e-4 (Ballistic Coefficient)
- Inclination=098.1351
- RAAN=034.3744
- Eccentricity=0.0012522
- ArgumentOfPerigee=125.8067
- MeanAnomaly=234.4294
- RevolutionPerDay=14.62716601

2.3.3. Magnetic Field Model: IGRF

Earth's magnetic field information is required to determine the magnetic torques (disturbances) affecting the satellite and more importantly for the simulation of the magnetorquers. As explained in the following parts, the order of the magnetic disturbances on the satellite is negligible.

The magnetic field must be measured by sensors or estimated by employing a field model in order to control the magnetorquers and model magnetorquer dynamics. Magnetic field vector can be measured in real time by a magnetometer. However, sensor requirements and dynamics are not considered for the control modes.

As a basic approximation, Earth's magnetic field can be modeled with the magnetic field of a sphere uniformly magnetized in the direction of a dipole axis. In this simple model, the dipole axis goes through the center of the Earth, and is offset from the rotational axis by 11.3°. This simple approximation can lead to errors as great as 30% in some locations. On the other hand, the error can be reduced to 10% by displacing the dipole axis about 400 km towards the western Pacific from the center of the Earth [1]. For the dipole model, the geomagnetic induction vector is;

$$\vec{B} = \frac{\mu_m}{r^3} [\hat{e}_M - 3\langle \hat{e}_M, \hat{e}_R \rangle \hat{e}_R] \quad (2.25)$$

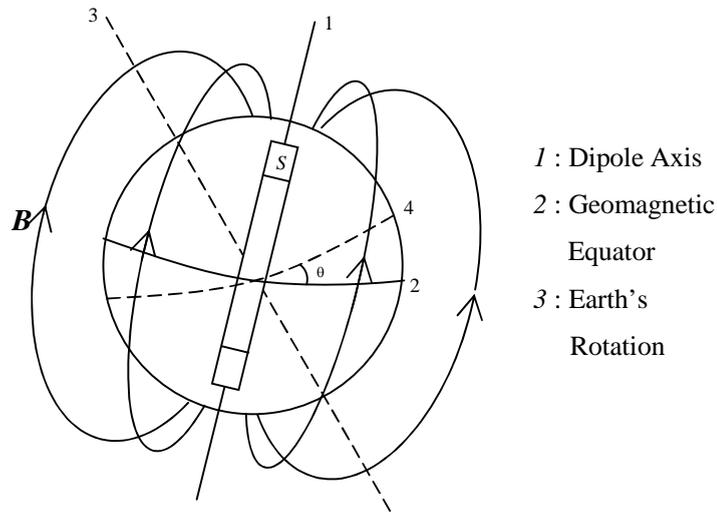


Figure 2.3 Earth's Magnetic Field Dipole Model

On the other hand, the magnetic field shows a variance over the Earth surface as depicted in Figure 2.4. At the sea level, the field is horizontal and the field strength is about 30 μT at the equator, while it becomes vertical with field strength of 60 μT around the poles. The magnetic field also varies over time in an unpredictable manner due to inner dynamics of Earth's core.

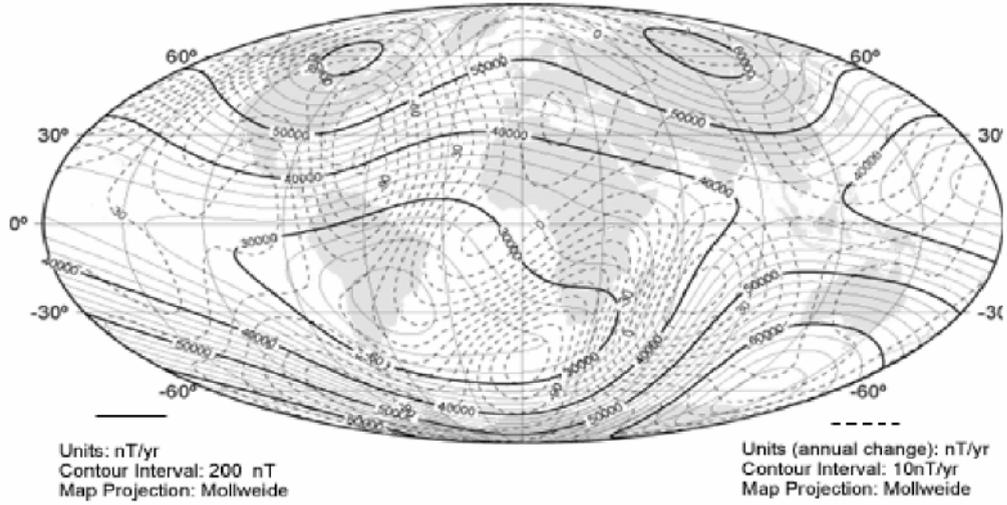


Figure 2.4 Magnitude of the Earth's Magnetic Field

The International Geomagnetic Reference Field (IGRF) model is developed by the International Association of Geomagnetism and Aeronomy (IAGA) [3]. The IGRF model gives useful approximations on the Earth's surface and up to a certain altitude where the Earth's magnetic field is still the dominant field. The IGRF is a series of mathematical models of the Earth's main field and its annual rate of change. At any one time, the IGRF specifies the numerical coefficients of a truncated spherical harmonic series. At present the truncation is at $n=13$, so there are 195 coefficients. The IGRF model is specified every 5 years, for epochs 1900, 1905 etc. The latest IGRF model specified is thus the IGRF 2005, which is implemented in this study.

The magnetic field is the negative gradient of a scalar potential V which can be represented by the truncated series expansion:

$$V(r, \theta, \lambda, t) = R \sum_{n=1}^{n_{\max}} \left(\frac{R}{r} \right) \sum_{m=0}^m (g_n^m(t) \cos m\lambda + h_n^m(t) \sin m\lambda) P_m^n(\theta) \quad (2.26)$$

where r , θ , λ are geocentric coordinates (r is the distance from the centre of the Earth, θ is the colatitude, i.e. $90^\circ - \text{latitude}$, and λ is the longitude), R is a reference radius (6371.2 km); $g_n^m(t)$ and $h_n^m(t)$ are the coefficients at time t and $P_m^n(\theta)$ are the Schmidt semi-normalized associated Legendre functions of degree n and order m . The main field coefficients are functions of time and for the IGRF the

change is assumed to be linear over five-year intervals. The Matlab code for the model which provides the field direction and strength for a given time and location is presented in Appendix.

2.3.4. Disturbances

2.3.4.1. Gravity Gradient Torque

Earth's gravitational force over a satellite varies during the orbit and presents disturbing moments on the satellite. If we consider the general situation, satellite can be assumed as a nonsymmetrical object of finite dimensions. In reality there is no uniform gravitational field and the gravitational torque varies during the orbit. However, for the sake of simplicity a spherical Earth is assumed. In addition, the moment of inertia tensor is known for an arbitrary reference frame. For a unit mass element of satellite dm_i , which is located in R_i distance from the geometric center, the acting gravitational force dF_i can be defined as; [1]

$$dF_i = \frac{-\mu R_i dm_i}{R_i^3}, \quad (2.27)$$

Where $\mu = GM = 3.986005 \times 10^{14} \text{ m}^3/\text{s}^2$.

The torque about the geometric center can also be defined at r_i distance from the geocenter as;

$$dN_i = r_i \times dF_i = (\rho + r'_i) \times dF_i \quad (2.28)$$

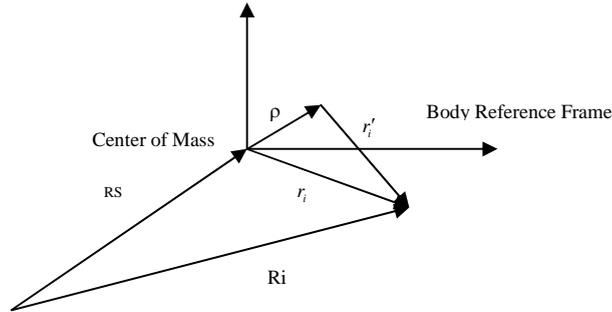


Figure 2.5 Torque about the Geometric Center

By integrating (2.28), the gravity gradient torque can be obtained as;

$$\hat{N}_{GG} = \int r_i \times dF_i = \int (\rho + r_i') \times \frac{-\mu R_i dm_i}{R_i^3} \quad (2.29)$$

The relation between R_i and R_s is;

$$R_i = R_s + r_i = R_s + \rho + r_i' \quad (2.30)$$

For an artificial satellite $R_i = R_s + \rho + r_i' \gg \rho + r_i'$. Therefore;

$$R_i^{-3} = (R_s + \rho + r_i')^{-3} = \left\{ R_s^2 \left[1 + \frac{2R_s \cdot (\rho + r_i')}{R_s^2} + \frac{(\rho + r_i')^2}{R_s^2} \right] \right\}^{-\frac{3}{2}} \quad (2.31)$$

By Using Binomial Expansion;

$$R_i^{-3} \approx R_s^{-3} \left[1 - \frac{3R_s(\rho + r_i')}{R_s^2} \right] \quad (2.32)$$

If we define M as the total mass of the satellite the gravity gradient torque can be rewritten as;

$$\hat{N}_{GG} = \frac{\mu M}{R_s^2} (\hat{R}_s \times \rho) + \frac{3\mu}{R_s^3} \int (r_i \times \hat{R}_s)(r_i \cdot \hat{R}_s) dm_i \quad (2.33)$$

By the definition of the center of the mass $\int r_i' dm_i = 0$. Assuming the geometric center and the center of mass are at the same point, ρ will be equal to zero, $\rho=0$. By these assumptions the gravity gradient torque can finally be written with respect to moment of inertia tensor as [1];

$$\hat{N}_{GG} = \frac{3\mu}{R_{EARTH}^3} [\hat{R}_s \times (I \cdot \hat{R}_s)] \quad (2.34)$$

\hat{R}_s vector is always in nadir direction and instead of \hat{R}_s we can use \hat{z}_0 nadir unit vector in body coordinates;

$$\hat{z}_0 = [A_{13} \quad A_{23} \quad A_{33}]^T \quad I = \begin{bmatrix} I_{XX} & I_{XY} & I_{XZ} \\ I_{YX} & I_{YY} & I_{YZ} \\ I_{ZX} & I_{ZY} & I_{ZZ} \end{bmatrix}$$

$$\hat{N}_{GG} = \frac{3\mu}{R_{EARTH}^3} [\hat{z}_0 \times (I \cdot \hat{z}_0)]$$

$$\begin{aligned}
&= \frac{3\mu}{R_{EARTH}^3} \begin{bmatrix} A_{13} \\ A_{23} \\ A_{33} \end{bmatrix} \times \left(\begin{bmatrix} I_{XX} & I_{XY} & I_{XZ} \\ I_{YX} & I_{YY} & I_{YZ} \\ I_{ZX} & I_{ZY} & I_{ZZ} \end{bmatrix} \begin{bmatrix} A_{13} \\ A_{23} \\ A_{33} \end{bmatrix} \right) \\
\hat{N}_{GG} &= \frac{3\mu}{R_{EARTH}^3} \begin{bmatrix} A_{13} \\ A_{23} \\ A_{33} \end{bmatrix} \times \begin{bmatrix} I_{XX} A_{13} + I_{XY} A_{23} + I_{XZ} A_{33} \\ I_{YX} A_{13} + I_{YY} A_{23} + I_{YZ} A_{33} \\ I_{ZX} A_{13} + I_{ZY} A_{23} + I_{ZZ} A_{33} \end{bmatrix} \quad (2.35)
\end{aligned}$$

For cross product we can use the Dyad notation;

$$\hat{z}_0 = [A_{13} \quad A_{23} \quad A_{33}]^T \quad \tilde{z}_0 = \begin{bmatrix} 0 & -A_{33} & A_{23} \\ -A_{33} & 0 & -A_{13} \\ -A_{23} & A_{13} & 0 \end{bmatrix} \quad (2.36)$$

Finally the gravity gradient torque acting on the satellite in matrix form is;

$$\hat{N}_{GG} = \frac{3\mu}{R_{EARTH}^3} \begin{bmatrix} 0 & -A_{33} & A_{23} \\ -A_{33} & 0 & -A_{13} \\ -A_{23} & A_{13} & 0 \end{bmatrix} \begin{bmatrix} I_{XX} A_{13} + I_{XY} A_{23} + I_{XZ} A_{33} \\ I_{YX} A_{13} + I_{YY} A_{23} + I_{YZ} A_{33} \\ I_{ZX} A_{13} + I_{ZY} A_{23} + I_{ZZ} A_{33} \end{bmatrix} \quad (2.37)$$

2.3.4.2. Magnetic Torque

The magnetic torque acting on the satellite is simply the cross product of the satellites magnetic field and the Earth's magnetic field. The satellites magnetic field is composed of the magnetic dipole vectors of the control magnetorquers and other magnetic disturbances in the satellite. The magnetic disturbance torques is ignored, as they are negligibly small with respect to the other major torques affecting the satellite.

2.3.4.3. Other Disturbance Torques

There is also a number of other disturbance torques acting on the satellite. Following are the relatively important disturbances; however they are negligible for applications similar to the one considered here and they are all ignored in our model.

- The tidal forces created by the earth-moon system creating gravity torques,
- Torques due to solar wind and electromagnetic particles, thus solar pressure,

- Atmospheric drag for LEO satellites,
- Magnetic toques generated by the electric components on board the satellite.

2.4. Sensor Modeling

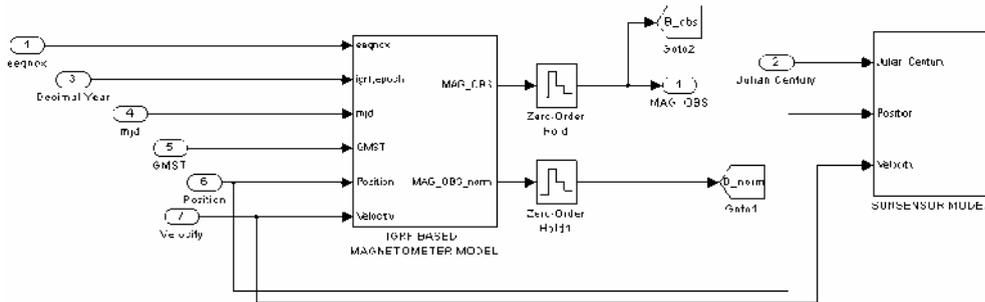


Figure 2.6 Sensor Models Simulink Diagram

The attitude information can be obtained from sensors such as magnetometers, Sun sensors and star trackers. In this thesis, magnetometer model which use the IGRF 2005 magnetic field model is modeled to acquire the attitude rates of the satellite during the detumbling mode. Magnetometer measurements are used for the rate filter block during the detumbling phase. Sun sensors are modeled to obtain the azimuth and elevation angles and transformed into attitude angles. However, the measurement data that is generated by the sensor models are not used for the rate estimation during the control modes and the theoretical angular rate that is produced from the satellite dynamics simulator block is used for the maneuvers.

2.4.1. Magnetometer Modeling

The IGRF model that is examined section 2.3.3 is used for generating the magnetometer outputs. The approximate magnetic field model that is a simplified dipole model of Earth Magnetic field is also usable to compare the results with the IGRF model. The Figure 2.7 shows the results of dipole model that use just one harmonic coefficient. As it seen from the Figure 2.8, IGRF model use Greenwich Mean Sidereal Time value, the radius vector of the satellite and the IGRF model epoch time. Normalized values of magnetic field are also used for the usage of controllers as it seen from Figure 2.10. The noise is added to the magnetic field

vector that is estimated in ECI frame to simulate the real magnetometer data for the convenience. The magnetic field components in ECI and ECEF frames can be seen from the Figures 2.11 to 2.12. $3 \mu T$ noise is considered for the magnetometers.

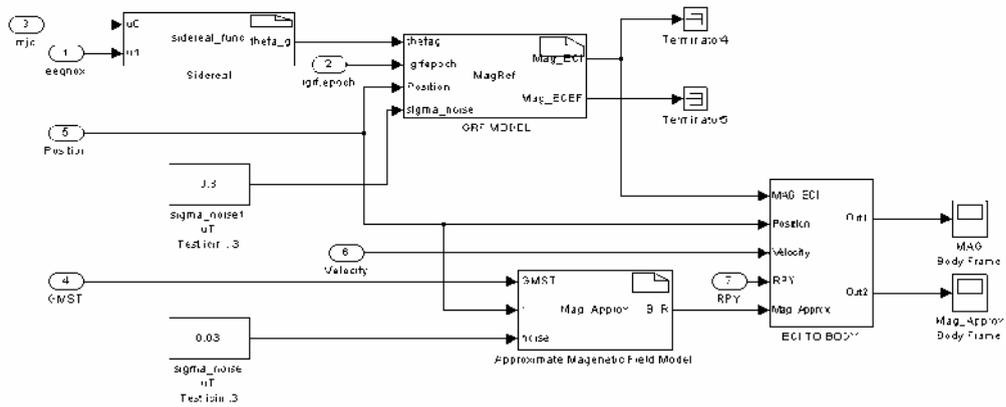


Figure 2.7 Magnetometer Model Simulink Model

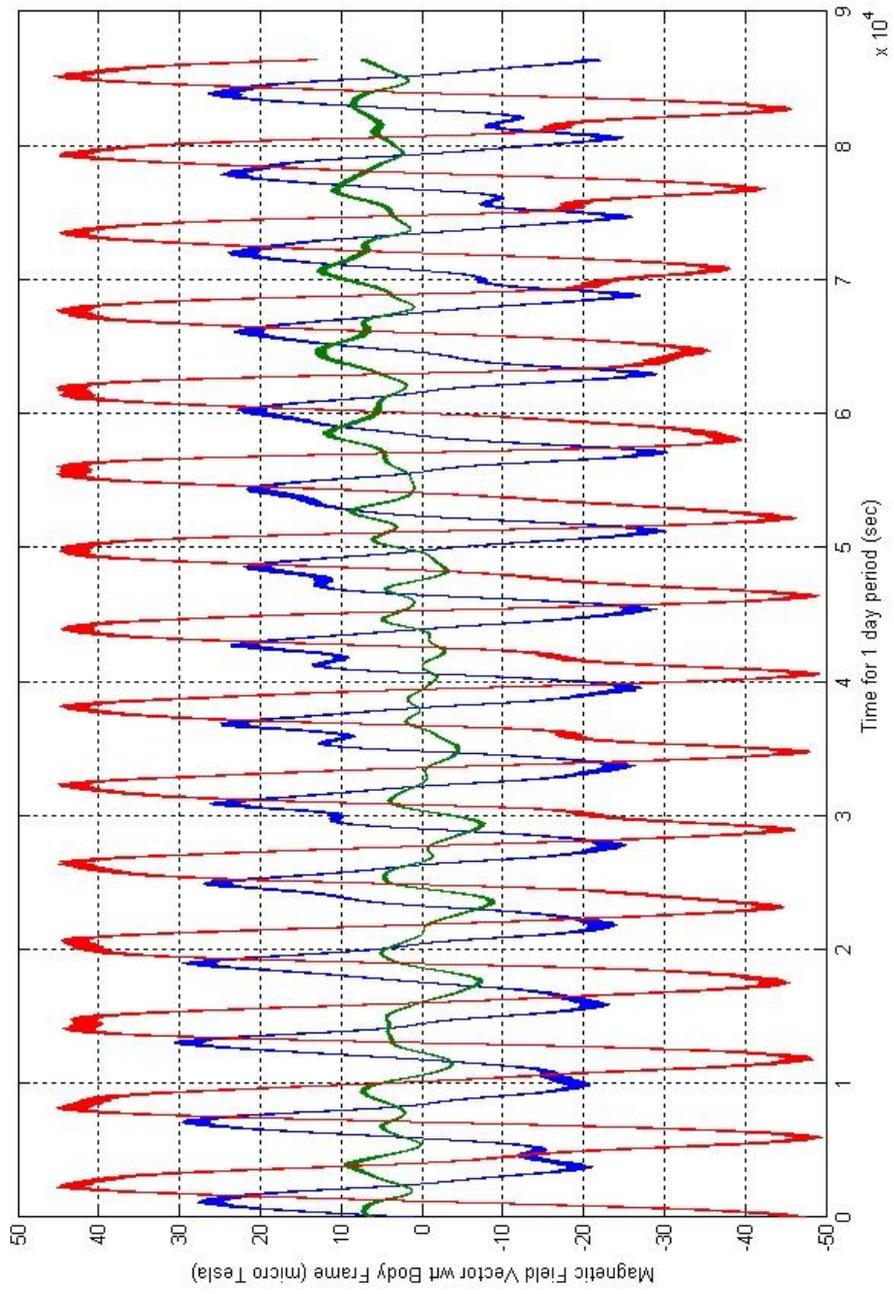


Figure 2.8 Magnetic Field Vector wrt Body Fixed Frame (IGRF Model with noise)

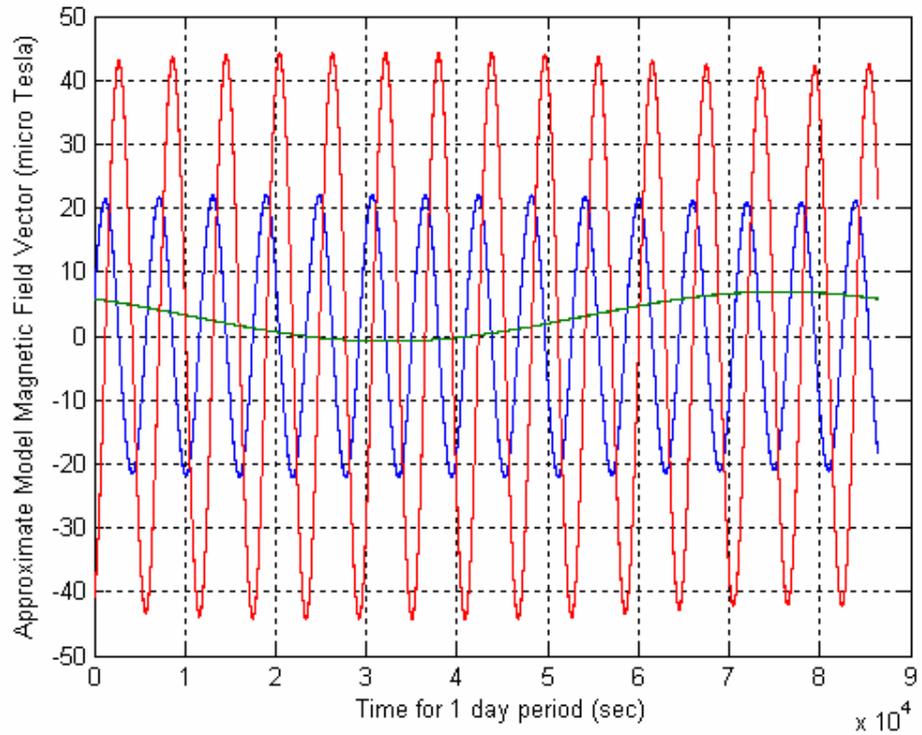


Figure 2.9 Magnetic Field Vector Estimated By Approximate Model wrt Body Fixed Frame

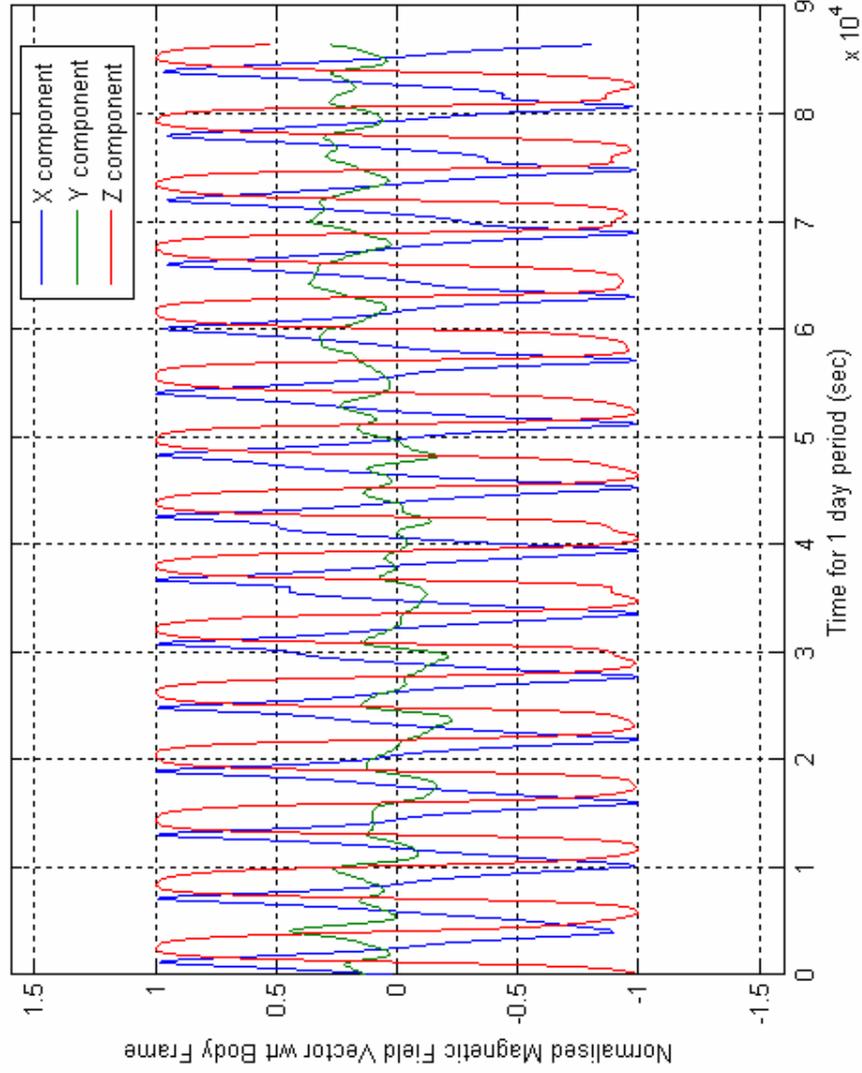


Figure 2.10 Normalized Magnetic Field Vector wrt Body Frame

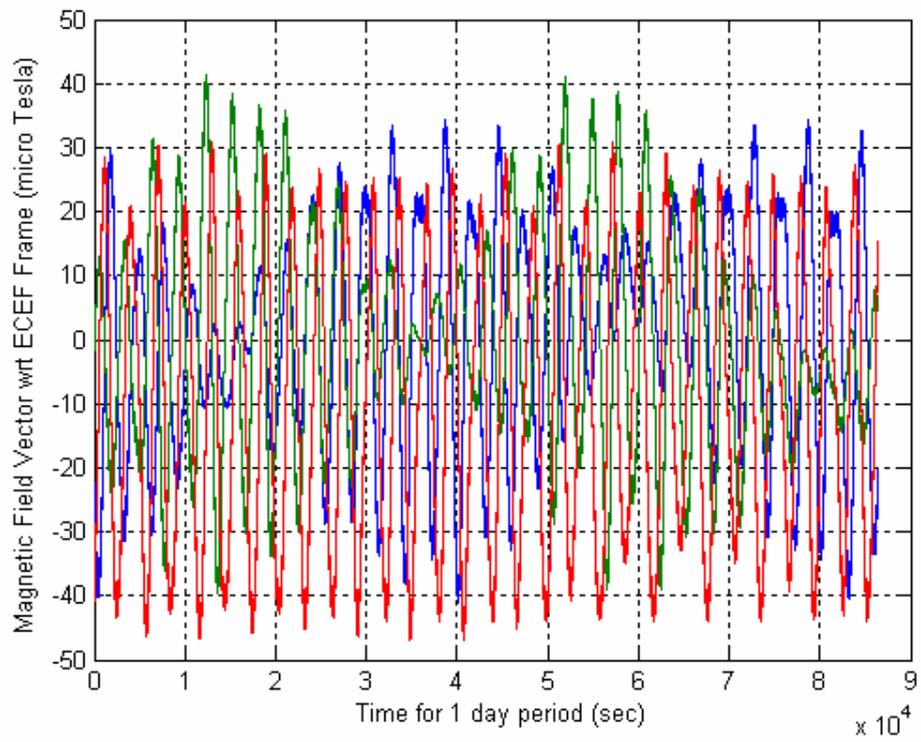


Figure 2.11 Magnetic Field Vector wrt ECEF Frame

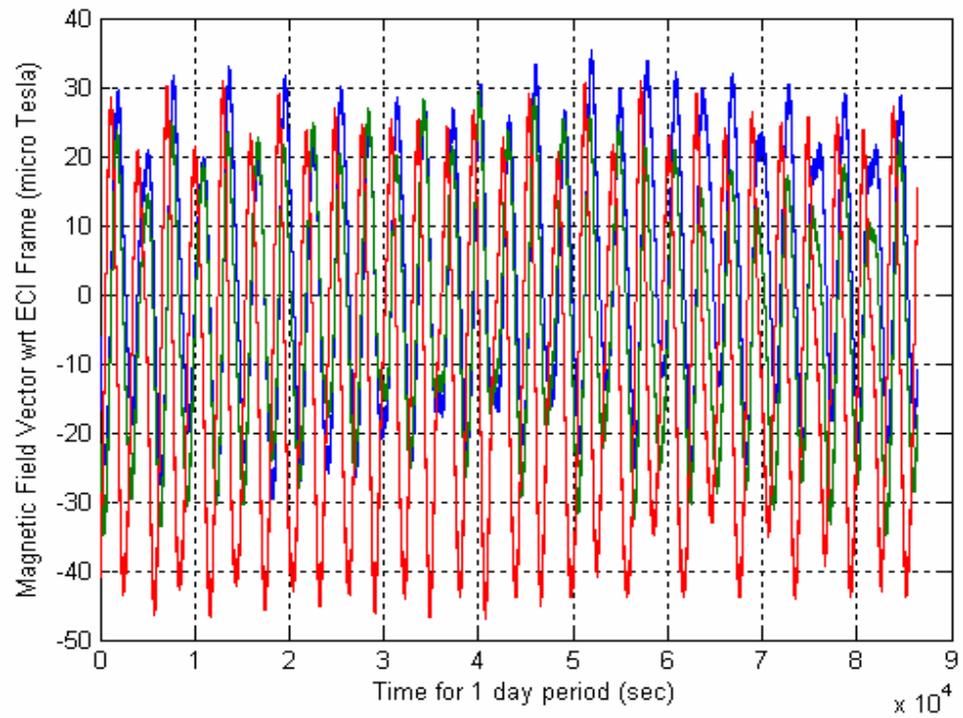


Figure 2.12 Magnetic Field Vector wrt ECI Frame

2.4.2. Sun Sensor Modeling



Figure 2.13 Satrec Initiative FSS_05 Analog Sun Sensor

In this thesis, Satrec Initiative Sun Sensor FSS-05 is based for modeling. The main characteristics of these sensors are given in Table 2.4. Sun Sensors provide more accurate attitude measurements than the magnetometers. The only period that it can not be used is the eclipse phase. In most cases, the Star Trackers are not used apart from the imaging periods and the Sun Sensors are the most accurate sensors for approximately 1 hour of the 97 minutes Sun synchronous orbit. In this work, Sun sensor simulation blocks generate azimuth and elevation signals of the Sun wrt their own reference frame just like the real case. [10]

Table 2.4 Satrec Initiative FSS-05 Sun Sensor Specifications [10]

Sensor Type	2 axis Analog Sun Sensor
Field of view	$\pm 60^\circ$ (each axis)
Accuracy	0.5° (after calibration)

2.4.2.1. Sun Position Vector Estimation

Sun position vector is estimated by using Julian Century input, wrt ECI (Earth Centered Inertial Frame). For the simulation, “Simulation Manager” block provides all the necessary data for all other blocks such as Julian Century. After the beginning of simulation from a definite epoch time, the time is propagated and Julian Century is estimated wrt this information. Vallado’s algorithm is used for estimation of Sun position vector. [8]. The results that are obtained from that algorithm are checked by the STK[®] software.

2.4.2.2. Transformation of Sun Position Vector to Satellite Body Frame

The position vector of the Sun is needed in satellite body frame for convenience. After the estimation of the Sun vector wrt ECI (Earth Centered Inertial Frame), this vector is transformed to the ECEF (Earth Centered Earth Fixed Frame). The ECEF components are then transformed to Satellite Body frame. The final step is to transform the position vector into sensor frames so that it is possible to simulate each Sun sensor signal separately.

2.4.2.3. Eclipse Times Estimation

The eclipse times are estimated by using Satellite’s position vector and Sun Position vector. Sun sensor measurements are simulated as “not available” during eclipse period. Apart from the satellite eclipse, the regions where the Sun Vector is not available for some sensors is simulated in Sun Sensor blocks. The eclipse flag which defines the eclipse period is given in Figure 2.16.

2.4.2.4. Sun Sensor Simulators

Sun Sensor simulator blocks simply generate the azimuth and elevation signals from the necessary inputs. They contain all the characteristics of the Sun sensors such as the orientation information, physical structure, field of view limits and noise levels.

2.4.2.5. Installation Matrix of the sensors

Installation matrix consist of the angles of direction cosine matrix which defines a transformation between Satellite body frame and each sensor frames. There

are two conditions for the installation matrix: the ideal case in which angles are desired installation values and the real case which involves the integration errors. The angles of the real case are determined by the sensitive measurement apparatus.

The coordinate system of each sensor is designated from this definition: The axis which is perpendicular to the measurement plane is +X, the axis which lies from azimuth slit's long edge to the elevation slit is +Y and the axis which could also be found from the right hand rule and lies in azimuth slit long edge to the down of the sensor is defined as +Z axis.

2.4.2.6. Sensor limits and Field of Views

There exist four Sun sensor on the Satellite. Sun sensor limit parameter defines the field of view of each sensor for the limit values. Unfortunately these values are not accurate enough in near the boundary regions. Because of that situation, the sensor limits are defined with field of view parameter. For instance the limit values of Bilsat's Sun sensor are given as 60 degrees but effectively these sensors obtain 55 degrees field of view.

2.4.2.7. Sensor blocks

Every Sun sensor model is evaluated from the same algorithm. The only difference between each sensor is their installation matrices. In Matlab simulation, the algorithm that exists in S function form, transforms the Sun position vector components into sensor frame components by using installation matrices with the order of "3-2-1" sequence. By using the Sun vector components azimuth and elevation values are estimated from the equation 2.38.

$$\begin{aligned} Azimuth &= (180 \setminus \Pi) \cdot a \tan(R_Y \setminus R_X) \\ Elevation &= (180 \setminus \Pi) \cdot a \tan(-R_Z \setminus R_X) \end{aligned} \quad (2.38)$$

For the next step, the estimated values are checked whether they are inside the sensor field of views or not. The reason for that is, even the satellite is not in eclipse phase some sensors may not be inside the Sun's field of view because of the geometric constraints of the sensors.

In real case, all the analog sensors on the satellite are connected to the ADCS board. ADCS board communicates with the OBC (Onboard Computer). All Sun sensors on BilSAT provide 4 measurement signals. Sensors are physically composed of two diagonally symmetric triangle CCDs for each slit. The 2 CCDs under the azimuth slit generates the “Azimuth A” and “Azimuth B” channel signals. Likewise, 2 CCDs under the elevation slit generates “Elevation A” and “Elevation B” channel signals. In simulation mode, azimuth and elevation signals are converted into azimuth A&B and elevation A&B signals for creating a real case condition. Thus, the signal that is obtained from the Sun sensors in real case and the simulation case can be compared for the convenience.

2.4.2.8. Simulation Results

The Simulink block of the Sun sensor simulation is shown in Figure 2.14. The signals obtained from the Sun sensor simulator blocks are compared with the real sensor data and the simulation is verified. However, the sensor data does not give a clear physical understanding for simulation purposes. Thus, the results of Flight code simulator that are the elevation and azimuth angles are used. Flight code simulator consists of a Sun sensor head selection algorithm so that azimuth and elevation angle of the Sun is estimated from the appropriate Sun sensor by considering the field of view of each sensor. The azimuth and the elevation angles that is obtained from the beginning of the simulation epoch, is given in Figure 2.15. Selected Sun sensor heads is shown in Figure 2.17.

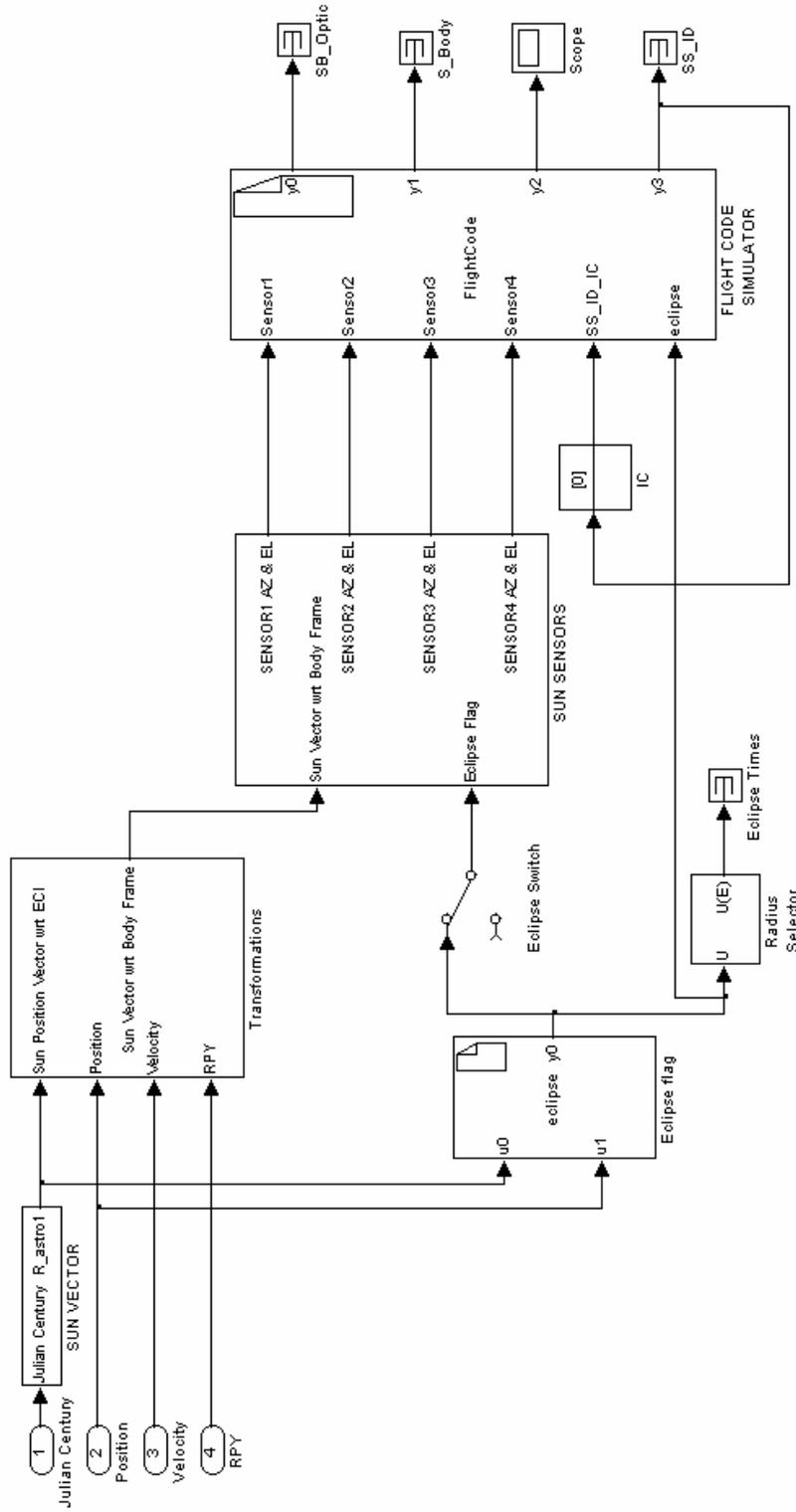


Figure 2.14 Sun Sensor Model Simulink Diagram

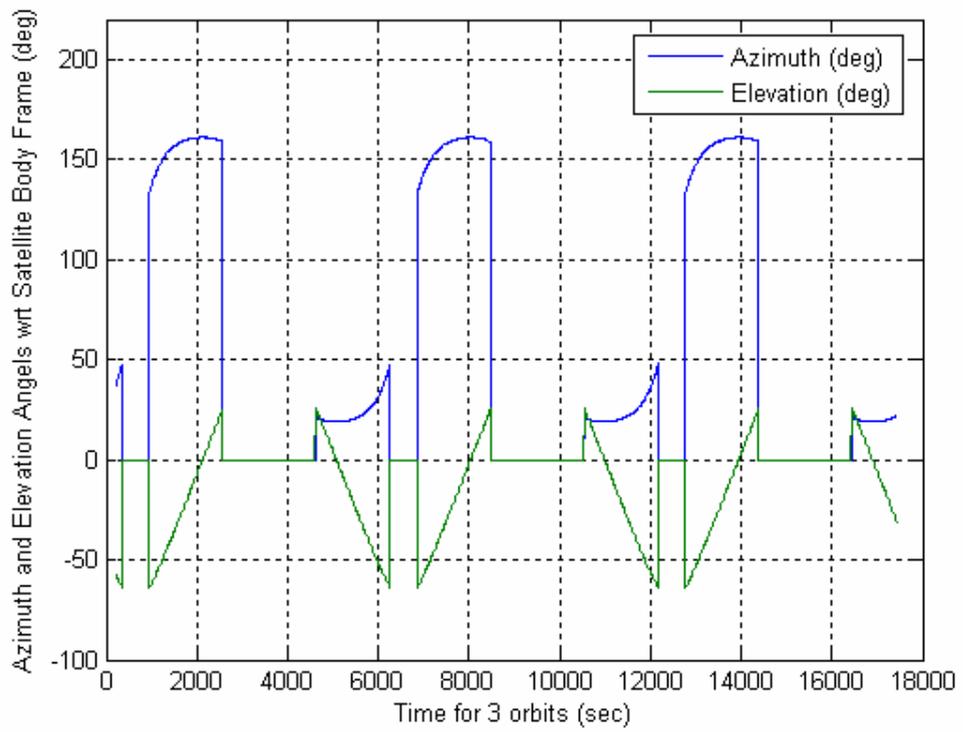


Figure 2.15 Simulation Results of Azimuth and Elevation Angles

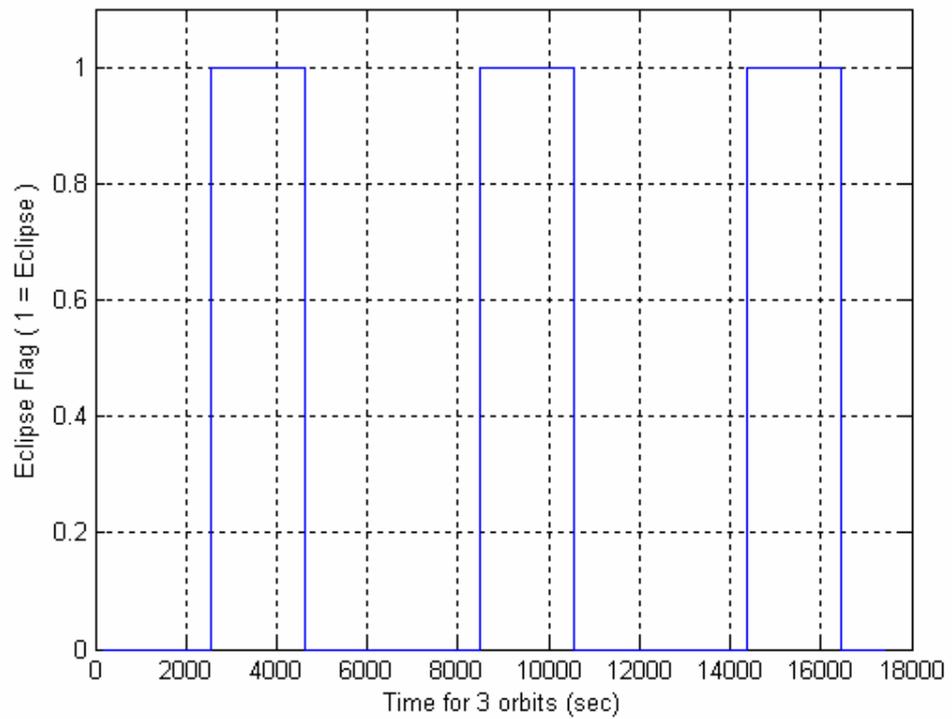


Figure 2.16 Eclipse times during the simulation

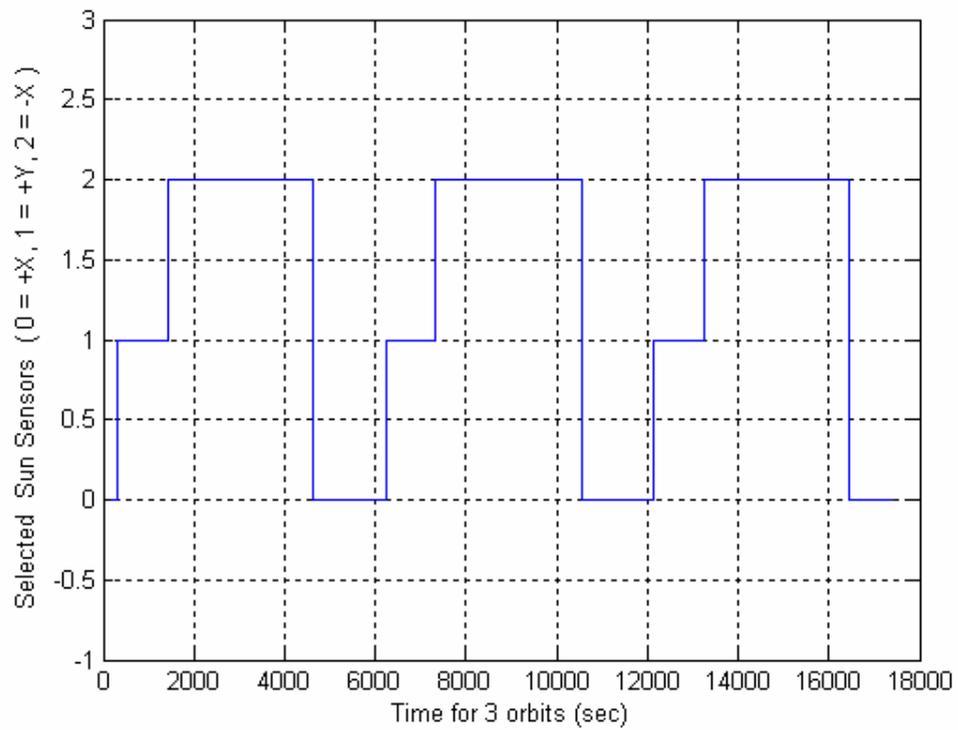


Figure 2.17 Selected Sun sensor heads for estimation

2.5. Actuator Modeling

2.5.1. Magnetorquer Modeling

In this thesis, SSTL magnetorquer MTR-10 is based for modeling. The main characteristics of the magnetorquer are given in Table 2.5 [11]. Magnetorquer consist of two coils per each which could be energized together or alone. The main purposes of this are redundancy and extension of the generated torque level.

Table 2.5 MTR 10 Magnetometer Specifications [11]

Magnetic Moment	$\pm 6.24\text{Am}^2 @ 100\text{mA} \pm 3\frac{1}{2} \% (2 \text{ coils active })$
Scale Factor	$0.062.4 \text{ Am}^2/\text{mA} \pm 1\%$
Linearity	$\pm 5\% (\text{ Range : } \pm 130\text{mA})$
Residual Magnetic Moment	$< \pm 0.1\text{Am}^2$
10 Am ² Drive Current	$100\text{mA} \pm 1.5\% @ 5\text{Am}^2$
Temperature	$>50 \Omega @ -30^\circ \text{C}$ $<80 \Omega @ +25^\circ \text{C}$
Inductance	$< 1.6 \text{ mH} @ 25^\circ \text{C}$
Isolation	$> 10^7 \Omega @ 50 \text{ Volts}$
Weight	$500 \pm 10 \text{ grams}$

In design process using magnetorquers, the current or generally speaking power consumption is the key parameter from ADCS hardware point of view. Related with the current, the resistances of the coils are chosen according to desired torque. In dynamic perspective, magnetic moment parameter describes the torque potential of the magnetorquer and it represents the control capacity. The magnetic moment is calculated from the current and the scale factor parameters as in equation 2.39.

$$M = I_D \times S \quad (2.39)$$



Figure 2.18 MTR 10 Magnetorquer

As it seen from equation 2.39, the magnetic moment is proportional to the control torque of the magnetorquer. This linear relation is valid up to 6.2 Am^2 . The torque that is obtained from the magnetorquer can be increased beyond this range but a nonlinear relation exists.

For a detailed model, temperature effect to the coil resistance and the drive voltage may be taken into account when calculating the exact value of magnetic moment. [11]

$$M = \min \left[(I \times S), \left(\frac{V \times S \times 1000}{R + (T + 20) \times \frac{\Delta R}{\Delta T}} \right) \right] \quad (2.40)$$

M	magnetic moment (Am^2)
I	current (mA)
S	scale factor (Am^2 / mA)
V	voltage (V)
R	resistance @ 20°C
T	temperature of the coil ($^\circ\text{C}$)
$\Delta R/\Delta T$	resistance change rate wrt temperature change rate ($\Omega/^\circ\text{C}$)

The torque that is generated by the magnetorquer is dependant of Earth's Geomagnetic Field which changes over the orbit of the satellite. In this thesis, IGRF model is used to simulate the magnetic field of the Earth during the orbit. From the average values of the magnetic field vector the maximum expected magnetic field value, in the Sun synchronous orbit of 686 km, is $48 \mu T$. The generated, IGRF based magnetic field components for the nominal mode of the satellite is given in Figure 2.8. The maximum magnetic moment that can be generated by the magnetorquer is limited to 5 Am^2 . The torque during the orbit is then calculated as in equation 2.41.

$$\vec{N}_M = \vec{m} \times \vec{B} \quad (2.41)$$

$$[\text{Nm}] = [\text{A.m}^2].[\text{Tesla}] = [\text{A.m}^2].[\text{weber} / \text{m}^2] = [\text{A}].[\text{V.s}] = [\text{Watt.s}] = [\text{N.m}]$$

The maximum torque that can be obtained from the magnetorquer is $2.4 \cdot 10^{-4} \text{ Nm}$ over the poles at a 98° inclined orbit.

In real case the magnetorquer is a magnetic moment generator. The generated magnetic moment is proportional to the current. In pulse width modulation the driver voltage and the applied current are constant. The only parameter that is changed by the ADCS hardware is the time fraction. Magnetic moment is calculated by the ADCS software and corresponding time fraction value is sent to the ADCS hardware. The symmetric pulse width modulation is fulfilled by the software that is embedded into ADCS module and the current direction is defined by that software.

In simulation mode, the magnetorquer is modeled as a magnetic moment generator which forms first order delay by using an exponential function. The magnetorquer controllers that are used in the simulation generate continuous magnetic moment signals. In real case the delay of current is shown in Figure 2.20 and the delay function of the current is given in equation 2.42.

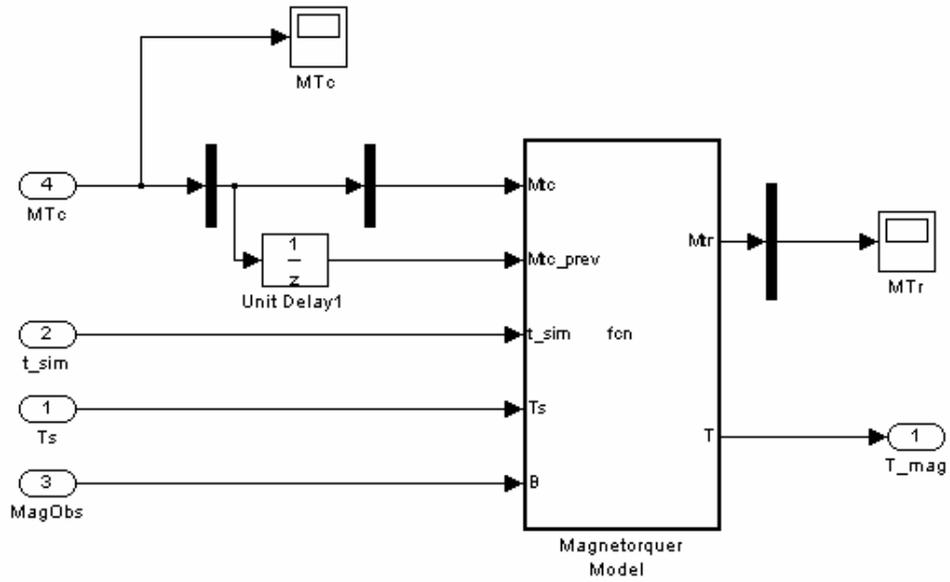


Figure 2.19 Magnetorquer Simulink Model

Same approach can be used for the magnetic moment that is proportional with the current and the magnetorquer forms an exponential delay in magnetic moment. The delay function is given in equation 2.42.

$$MT_{real}^{t+1} = MT_{real}^t + \left[MT_{commanded}^{t+1} - MT_{commanded}^t \right] \left[1 - e^{-\frac{[tsim-Tsample]}{T}} \right] \quad (2.42)$$

The time constant of the exponential function is calculated from the physical system properties that are defined as inductance over resistance of the coil. For the simulation model inductance is used as 1.64mH and the resistance is 50.6 ohm [11].

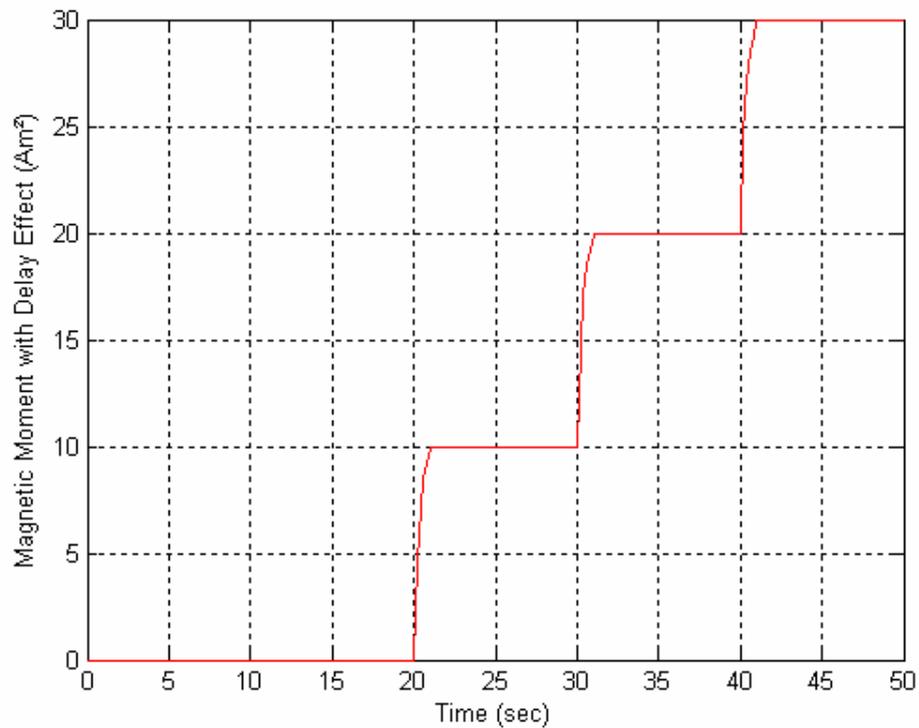


Figure 2.20 Magnetorquer Effect

$$T = \frac{L}{R} = \frac{1.64mH}{50.6\Omega} = 0.32s \quad (2.43)$$

The figure shows the time delay effect of commanded step signal so the magnetic moment values are not the values used in real simulation. The figure only represents the exponential delay to model the magnetorquer response.

2.5.2. Reaction Wheel Modeling

2.5.2.1. Reaction Wheel Configuration

In order to compose a simple control system, it is possible to use three reaction wheels orthogonally located in the satellite where each rotation axis is aligned with one of the satellite body axis. Although this simple scheme is good enough to control the satellite, in a condition such as a single wheel malfunction, 3-

axis control of the satellite will not be available. For this reason a fourth wheel is usually added to the system in a tetrahedral configuration. Thus redundancy is obtained by using the fourth wheel. [12]

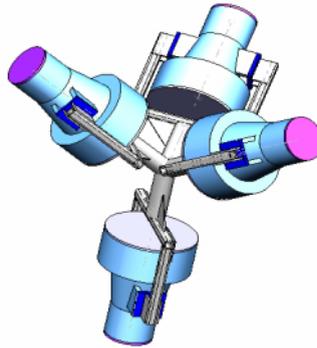


Figure 2.21 Reaction Wheel Configuration

Another advantage of the tetrahedral configuration is the chance to produce in a direction normal to any face of the tetrahedral, twice the amount of torque that can be produced in orthogonal configuration. This is achieved by increasing the spin rate of three of the wheels in the same direction while decelerating the fourth wheel (or accelerating in the opposite direction and vice versa). From the geometry, the torque generated by three wheels will be in the same direction with the fourth wheel axis.

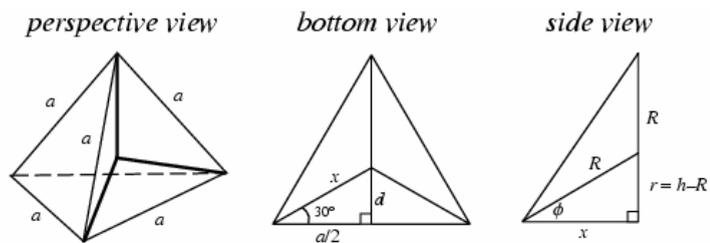
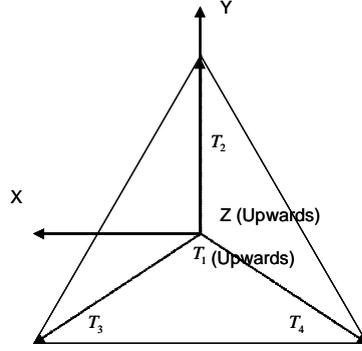


Figure 2.22 Side views [2]



TOP VIEW

Figure 2.23 Tetrahedral Geometry [2]

The angular momentum of the wheels and the torque generated must be projected from the tetrahedral frame to the orthogonal frame of the satellite in order to be included in the equations. As it is seen in Figure 2.22, from the different perspectives of views there are two main characteristic angles that defines the tetrahedron. Firstly, if we check the bottom view, the projection is equilateral triangle and angle between any side and the direction that connects the center with a corner of this side is 30° . Angle between bottom plane and the center of tetrahedron is $\phi = 19.4712^\circ$. If we assume an orthogonal reference frame center of which coincide with the tetrahedral reference frame, we can define a transformation matrix between the tetrahedral axes and the orthogonal axes by using the angles between them.

$$\begin{aligned}
 \hat{T}_{Axis-X} &= 0T_1 + 0T_2 + 0.8165T_3 - 0.8165T_4 \\
 \hat{T}_{Axis-Y} &= T_1 - 0.3333T_2 - 0.3333T_3 - 0.3333T_4 \\
 \hat{T}_{Axis-Z} &= 0T_1 + 0.9428T_2 - 0.4714T_3 - 0.4714T_4
 \end{aligned} \tag{2.44}$$

$$\begin{bmatrix} \hat{T}_{Axis-X} \\ \hat{T}_{Axis-Y} \\ \hat{T}_{Axis-Z} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0.8165 & -0.8165 \\ 1 & -0.3333 & -0.3333 & -0.3333 \\ 0 & 0.9428 & -0.4714 & -0.4714 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} \tag{2.45}$$

Here, the transformation matrix between the tetrahedral frame and orthogonal frame, A_W , is not a square matrix and can not be inverted using classical methods. Therefore, an optimization criterion must be assumed to construct the inverse matrix. If we define the Hamiltonian H as the norm of the T vector according to Wertz [1];

$$H = \sum_{i=1}^4 T_i^2 \quad (2.46)$$

Also ;

$$\begin{aligned} \hat{T}_{Axis-X} &= T_1 - T_2 \\ \hat{T}_{Axis-Y} &= T_2 - T_4 \\ \hat{T}_{Axis-Z} &= T_1 + T_2 + T_3 + T_4 \end{aligned} \quad (2.47)$$

defining g for simplification;

$$\begin{aligned} g_1 &= T_1 - T_3 - \hat{T}_{Axis-X} \\ g_2 &= T_2 - T_4 - \hat{T}_{Axis-Y} \\ g_3 &= T_1 + T_2 + T_3 + T_4 - \hat{T}_{Axis-Z} \end{aligned} \quad (2.48)$$

so that the Lagrangian is;

$$L = H + \lambda_1 g_1 + \lambda_2 g_2 + \lambda_3 g_3 + \lambda_4 g_4 \quad (2.49)$$

The conditions for minimizing the H are;

$$\begin{aligned} \frac{\partial L}{\partial T_1} &= 2T_1 + \lambda_1 + \lambda_3 = 0 \\ \frac{\partial L}{\partial T_2} &= 2T_2 + \lambda_2 + \lambda_3 = 0 \\ \frac{\partial L}{\partial T_3} &= 2T_3 - \lambda_1 + \lambda_3 = 0 \\ \frac{\partial L}{\partial T_4} &= 2T_4 - \lambda_2 + \lambda_3 = 0 \end{aligned} \quad (2.50)$$

By the derivation of the final situation:

$$\Delta T = T_1 - T_2 + T_3 - T_4 \quad (2.51)$$

After the optimization the new transformation matrix is;

$$\begin{bmatrix} \hat{T}_{Axis-X} \\ \hat{T}_{Axis-Y} \\ \hat{T}_{Axis-Z} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0.8165 & -0.8165 \\ 1 & -0.3333 & -0.3333 & -0.3333 \\ 0 & 0.9428 & -0.4714 & -0.4714 \\ -1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} \quad (2.52)$$

So the inverse matrix could be found easily as;

$$\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & 1.499 & 0 & 0.499 \\ 0 & 0.5 & 0.7071 & 0.5 \\ 0.6124 & 0.5 & -0.3536 & 0.5 \\ -0.6124 & 0.5 & -0.3536 & 0.5 \end{bmatrix} \begin{bmatrix} \hat{T}_{Axis-X} \\ \hat{T}_{Axis-Y} \\ \hat{T}_{Axis-Z} \\ 0 \end{bmatrix} \quad (2.53)$$

Pseudo Inverse may also be used to obtain the same result :

$$\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} = \begin{bmatrix} 0 & 0.75 & 0 \\ 0 & -0.25 & 0.7071 \\ 0.6124 & -0.25 & -0.3536 \\ -0.6124 & -0.25 & -0.3536 \end{bmatrix} \begin{bmatrix} \hat{T}_{Axis-X} \\ \hat{T}_{Axis-Y} \\ \hat{T}_{Axis-Z} \end{bmatrix}$$

2.5.2.2. Reaction Wheel Model

The angular momentum, h , and relevant torque, \dot{h} , of a rigid body with an inertia tensor of I , spinning at a rate ω , is given by;

$$h = I\omega \quad \dot{h} = I\dot{\omega} \quad (2.54)$$

A reaction wheel is basically a rigid disc spun by an electric motor. Thus the angular acceleration of a reaction wheel is proportional to the current, i , driving the electric motor. Then the relevant torque of the reaction wheel can be determined as:

$\dot{\omega} \propto i$ Then,

$$\dot{h} \propto I \cdot i \quad \text{or} \quad \dot{h} \propto I \cdot u(t) \quad (2.55)$$

Where $u(t)$ is the input.

Integrating the relation in Equation (2.55) and taking Laplace transform, the transfer function of a reaction wheel will be in the form of,

$$TF_{RW} = L\left(\int I \cdot u(t)\right) = \frac{I}{s} u(s) \quad (2.56)$$

Furthermore, the response of the electric motor, $R_{motor}(s)$, together with internal friction torques must be taken into account:

$$TF_{RW} = R_{motor}(s) \cdot \left(\frac{I}{s} u(s) \right) \quad (2.57)$$

The response of the electric motor, $R_{motor}(s)$, was obtained using the discrete model the developed for the reaction wheels onboard BILSAT which gives the torque generated, to a given angular momentum as input;

$$TF_{h \rightarrow h} = \frac{2.4z - 2.1}{z - 1} \quad (2.58)$$

Finally, taking angular momentum as input to Equation (2.57) and, including Equation (2.58), the model for a single reaction wheel is given in Figure 2.24. The reaction wheels are assumed to be controlled with angular momentum inputs.

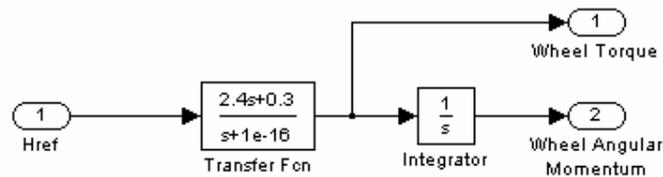


Figure 2.24 SIMULINK Reaction Wheel Model

Next step is to control the momentum and torque of the wheel so as to avoid momentum and torque saturation to generate torque continuously. For this purpose, two saturation blocks were added to the model. Additionally, the difference between the torque required and the torque available to the wheel, and the difference between the angular momentum required and the angular momentum available to the wheel are fed back in a Proportional-Derivative (PD) control manner.

CHAPTER 3

BASIC ATTITUDE CONTROL MODES

In this section two basic control modes are presented. The first one is the detumbling control using magnetorquers. The other one is the nominal three-axis control through reaction wheels. The desaturation of the reaction wheels is also presented.

3.1. Detumbling Control Mode

After the satellite is separated from the launch capsule, the satellite is almost always a tumbling object in its particular orbit. First, the flight computer boots up and all the necessary functions such as ADCS task begins to work and satellite is brought under control. The first control mode that is activated in the satellite is the detumbling control mode that tries to stop the tumbling motion. In this mode only the magnetometer measurements are used for the attitude rates and magnetorquers are used as actuators. The main purpose is to put the satellite in locked rotation around the Earth. The attitude estimation is not very accurate because of the inaccuracy of the IGRF model and the magnetometer measurements.

Although the satellite has 3 magnetorquers, only Y axis magnetorquer is used to damp the angular rate around X and Z axis. After the detumbling mode the satellite Y axis spin rate is controlled for the nadir pointing attitude.

3.1.1. B-dot Control

B-dot controller is the most common control algorithms used to stabilize spacecraft with magnetorquers [13]. The main goal is to activate a magnetorquer to damp out the other orthogonal axes angular velocities (X and Z axes in this case). The law can be described as:

$$M_i = -K_d \cdot \dot{B}_i \quad (3.1)$$

Where K_d a positive controller gain is \dot{B}_i is magnetic field i-component vector in the body coordinate system of the satellite. According to Yoshi [13], “ \dot{B}_i stands change rate of the i^{th} component of magnetic field vector with the time and with respect to body axis coordinate system.” The negative sign is to generate a rotation in the opposite direction. In this thesis the same approximation is made for the \dot{B}_i as the Yoshi [13] which defines the \dot{B}_i as the difference between the magnetic field vector component in firing magnetorquer axis over time interval:

$$\dot{B}_{i,k} \approx \frac{B_{i,k} - B_{i,k-1}}{\Delta t} \quad (3.2)$$

Where Δt is the sampling period. For more exact approximation following smoothing may be applied[13].

$$\dot{\bar{B}}_{i,k} = (1-s)\dot{B}_{i,k} + s\dot{\bar{B}}_{i,k-1} \quad (3.3)$$

Where $\dot{\bar{B}}$ is smoothed measurement and s is in the range of $0 \leq s \leq 1$. In our simulation s is taken as “0.9”.

Wisniewski et al [14], describe B-dot controller principle as minimizing the derivative of the magnetic field vector measured by a magnetometer. The controller generates a dipole moment that is perpendicular to the magnetic field. Without using any cross product, B-dot controller tries to minimize changes in the magnetic field that is caused by the spacecraft rotation and the reason is that the derivative of the magnetic field is perpendicular to the field vector. [13]

3.1.2. Stability of the B-dot controller

Yoshi [13] proves the Lyapunov stability of the B-dot controller by assuming the time derivative of magnetic field vector as a negligible quantity or satellite tumbling rate is “larger than the rotation rate of the Earth magnetic field vector in some inertial frame”. This assumption can be asserted as:

$$\left(\frac{d\bar{B}}{dt} \right)_B = \frac{d\bar{B}}{dt} - \bar{\omega}_{BI} \times \bar{B} \approx -\bar{\omega}_{BI} \times \bar{B} \quad (3.4)$$

The change in rotational kinetic energy derivative may be written as [13]

$$\dot{\tau} = I\dot{\bar{\omega}}_{BI} \cdot \bar{\omega}_{BI} \quad (3.5)$$

By using cross product control law derivative of rotational kinetic energy can be rephrased as;

$$\dot{T} = \left[\bar{N} - \bar{\omega}_{BI} \times I\bar{\omega}_{BI} \right] \cdot \bar{\omega}_{BI} = \bar{N} \cdot \bar{\omega}_{BI} = (\bar{M} \times \bar{B}) \cdot \bar{\omega}_{BI} = (\bar{B} \times \bar{\omega}_{BI}) \cdot \bar{m} = \dot{\bar{B}} \cdot \bar{M} \quad (3.6)$$

If the control input is chosen as $T_{control} = \dot{\bar{M}} = -K\dot{\bar{B}}$ then the time derivative of rotational kinetic energy is given as [13]:

$$\dot{\tau} = -K|\dot{\bar{B}}|^2 \leq 0 \quad (3.7)$$

Since, the equation 3.7 is negative definite, Wisniewski et al [14], the energy is dissipated from the system in detumbling mode. The rate of energy dissipation is determined by the gain, K.

3.1.3. Simulation Model for B-dot Controller

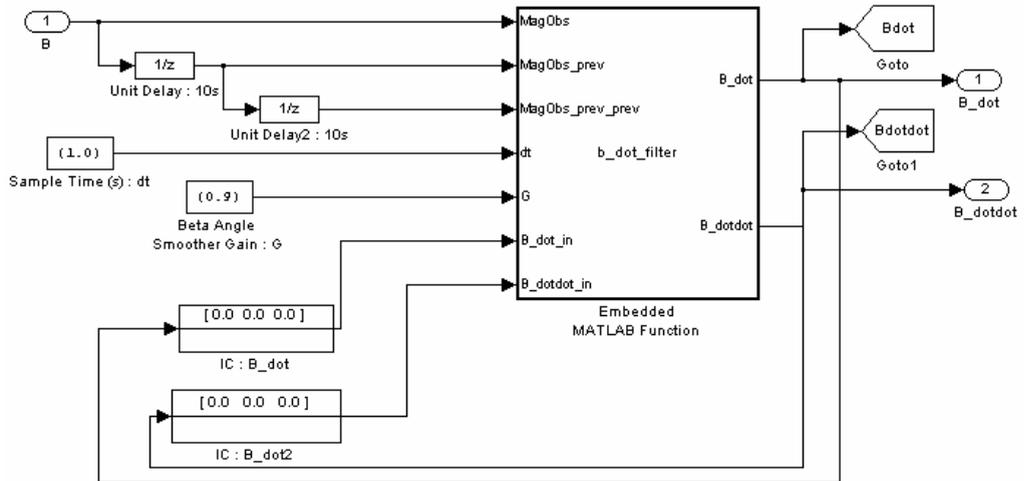


Figure 3.1 B-dot Estimator Simulink Diagram

In Figure 3.1 it is seen that, the detumbling object is being controlled to desired attitude after one day period. The time is constrained with the capacity of magnetorquers and in this case BILSAT magnetorquer which has 6.2 Am^2 magnetic moment capacity is used in the simulation. The magnetic moment that is commanded by the controller for the magnetorquer model is shown in Figure 3.2. The Simulink diagram of the B-dot controller is given in Figure 3.5.

The initial value of the satellite tumbling rate is 0.1 radian per second in each axis. The initial attitude is taken as “0” degrees in each axis also. The Euler rates wrt satellite body frame are less than 0.005 deg/sec. As it seen from the Figure 3.2, Y magnetorquer is capable of controlling the satellite tumbling rates. Because of the cross coupling elements of the inertia matrix, the body angular rates in all three axes are reduced. The Y magnetorquer can only damp out the X and Z axis so a Y spin controller is needed for the desired nadir point up (i.e. Locked Rotation).

The magnetic torque generated by the magnetorquers is obtained by the interaction of magnetorquer dipole with the Earth’s magnetic field. By implementing the basic cross product law the torque in satellite body coordinate frame can be seen in Figure 3.3. K_D Control gain is taken as 10.

The magnetic moments that are generated from magnetorquers and the torques applied to the satellite are shown in Figure 3.3 and 3.4.

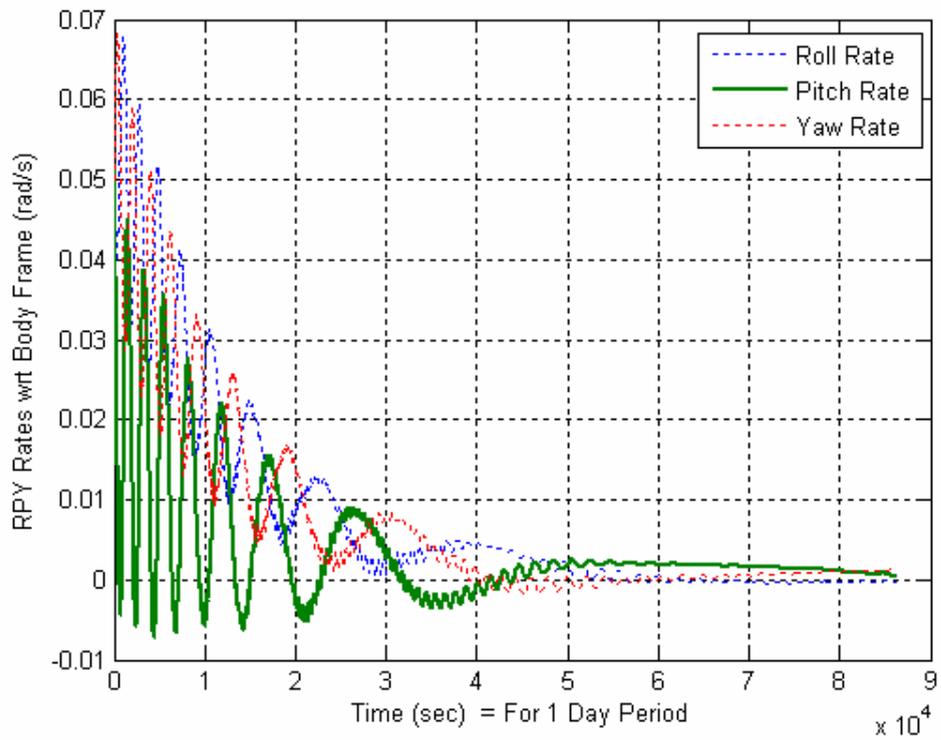


Figure 3.2 Attitude rates of the satellite during detumbling mode

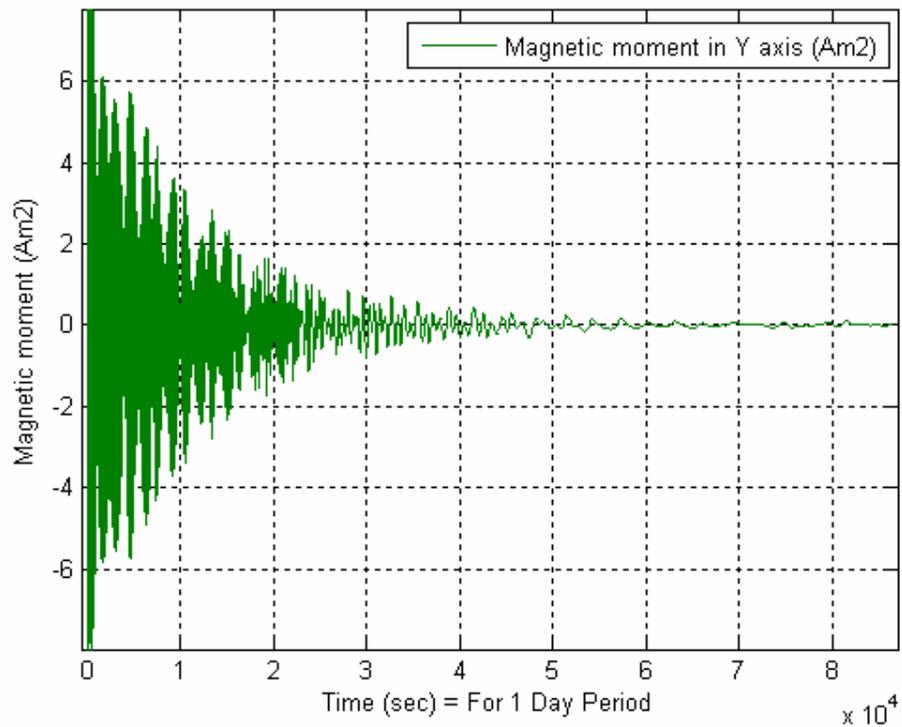


Figure 3.3 Magnetic moment that is commanded for the magnetorquers

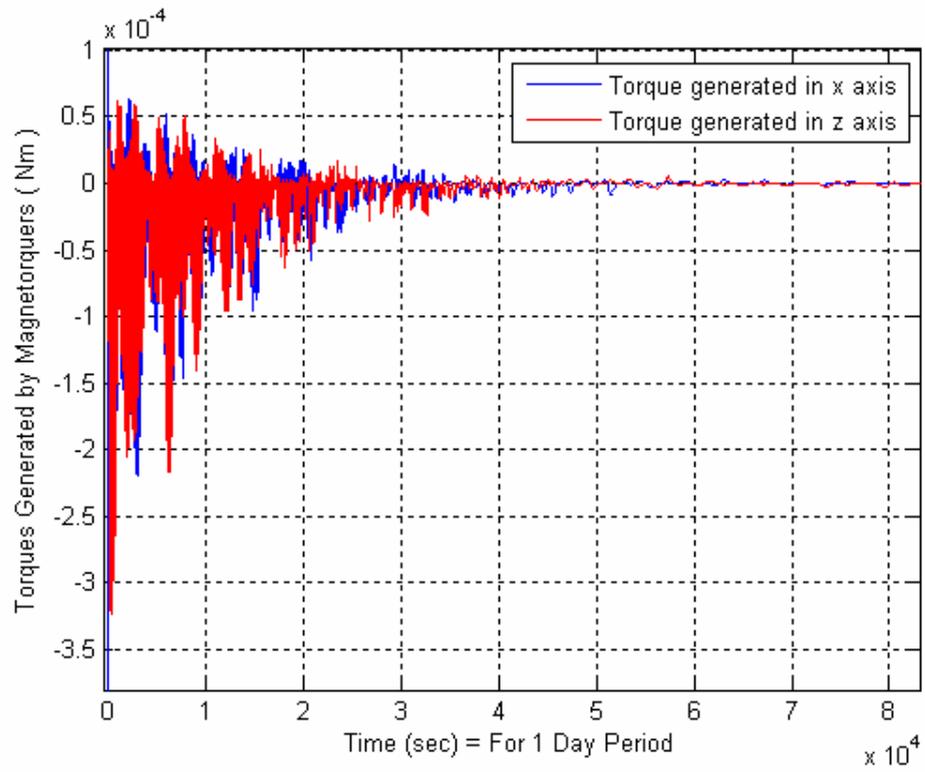


Figure 3.4 Torques generated by magnetorquers

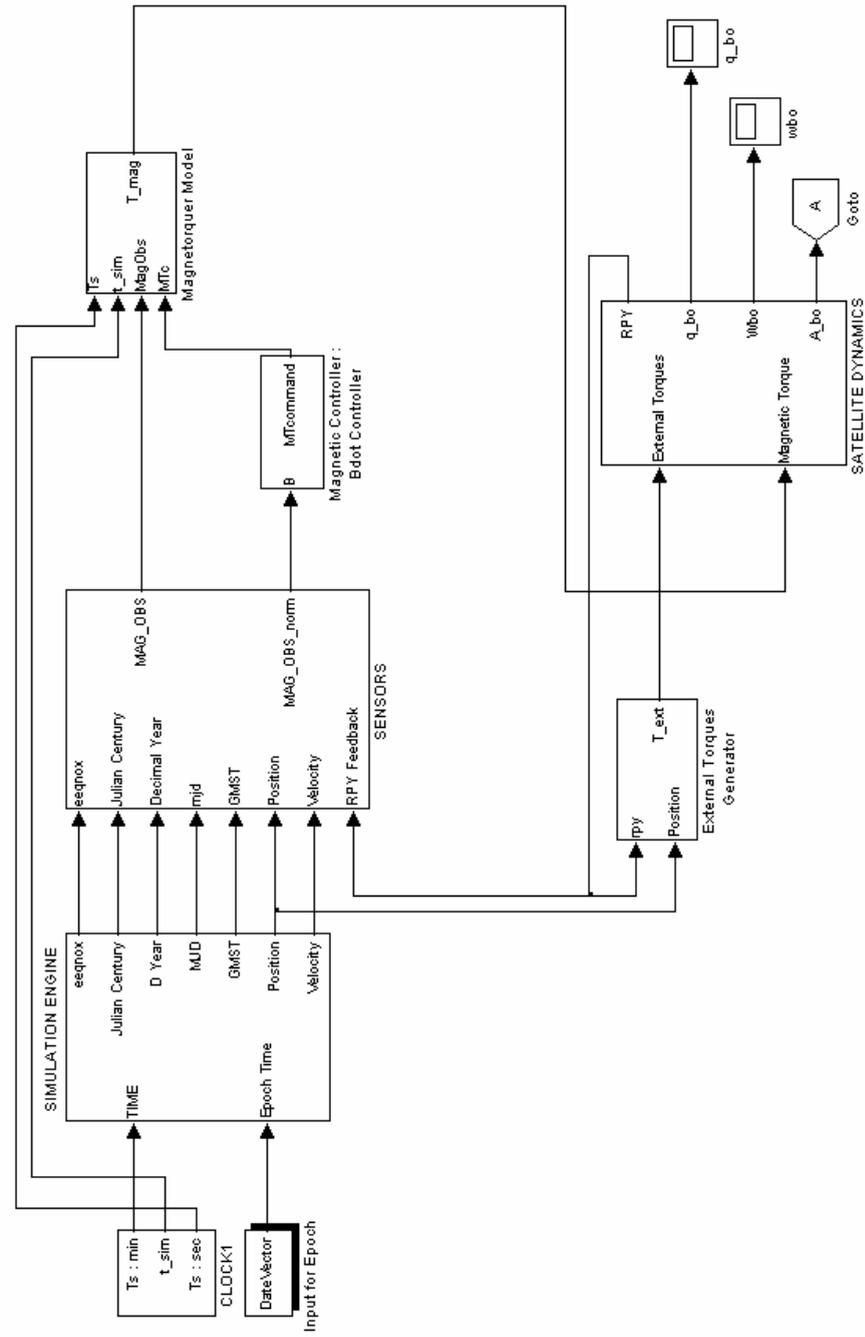


Figure 3.5 Detumbling Control Simulink Diagram

3.2. Nominal Control Mode: 3 Axis Attitude Control

As illustrated in Figure 3.1, the attitude control problem requires the control of the reaction wheels to steer the satellite to the desired attitude.

Given the fact that quaternions have no geometric singularity, the satellite kinematics is commonly implemented in quaternions and also used in this thesis. The attitude error is defined as the difference between the desired attitude and the current attitude, again with respect to orbital frame due to mission requirements.

Reference [15] suggests three quaternion feedback control laws, which are proved in terms of closed loop stability for 3-axis control using the Lyapunov stability theorem. These control laws are;

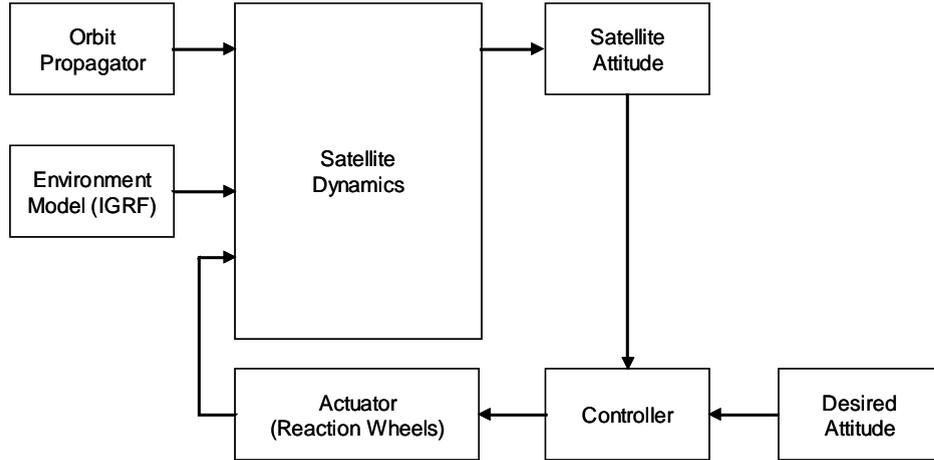


Figure 3.6 Attitude Control through Reaction Wheels

- $\vec{T}_{control} = -K_p \vec{q}_{vec} - K_d \vec{\omega}_{err}$
- $\vec{T}_{control} = -K_p \frac{\vec{q}_{vec}}{q_{4e}^3} - K_d \vec{\omega}_{err}$
- $\vec{T}_{control} = -sign(q_{4e}) K_p \vec{q}_{vec} - K_d \vec{\omega}_{err}$ (3.8)

Where:

K_p and K_d : the appropriate gains of the control laws

ω_{err} : the difference between commanded angular velocity and estimated angular velocity wrt body fixed frame of the satellite.

And \vec{q}_{err} and \vec{q}_{vec} are:

$$\vec{q}_{err} = [q_{1e} \quad q_{2e} \quad q_{3e} \quad q_{4e}] \quad (3.9)$$

$$\vec{q}_{vec} = [q_{1e} \quad q_{2e} \quad q_{3e}] \quad (3.10)$$

In this model the control law given in Equation (3.8) is employed for its simplicity, which is simply a proportional plus derivative (PD) control since \dot{q} and ω are related. Angular momentum output of the controller is;

$$T_{control} = -K_p \vec{q}_{vec} - K_D \vec{\omega}_{err} \quad (3.11)$$

The resulting SIMULINK Block for the reaction wheel controller is presented in Figure 3.7.

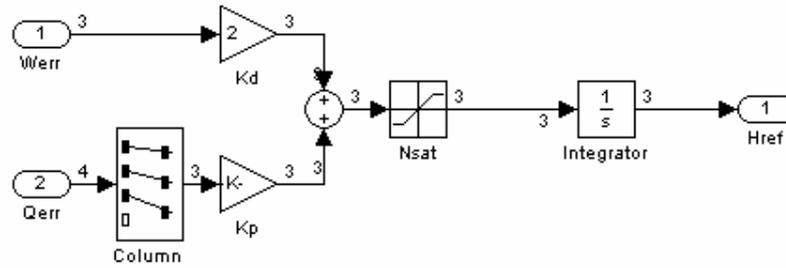


Figure 3.7 Reaction Wheel PD Controller

If we examine the two parts of a PD controller; a proportional controller (K_p) has the effect of reducing the rise time and will reduce but never eliminate, the steady-state error. A derivative control (K_D) has the effect of increasing the stability of the system, reducing the overshoot, and improving the transient response. Effects of each of controllers K_p , and K_D on a closed-loop system are summarized in Table (3.1), however these coefficients are coupled in the controller and change in one coefficient will also alter the other coefficients effects.

Table 3.1 PD Control Strategy

System Response	Rise Time	Overshoot	Settling Time	Steady State Error
Kp	Decrease	Increase	Small Change	Decrease
Kd	Small Change	Decrease	Decrease	Small Change

3.2.1. Desaturation of Reaction Wheels by Magnetorquers

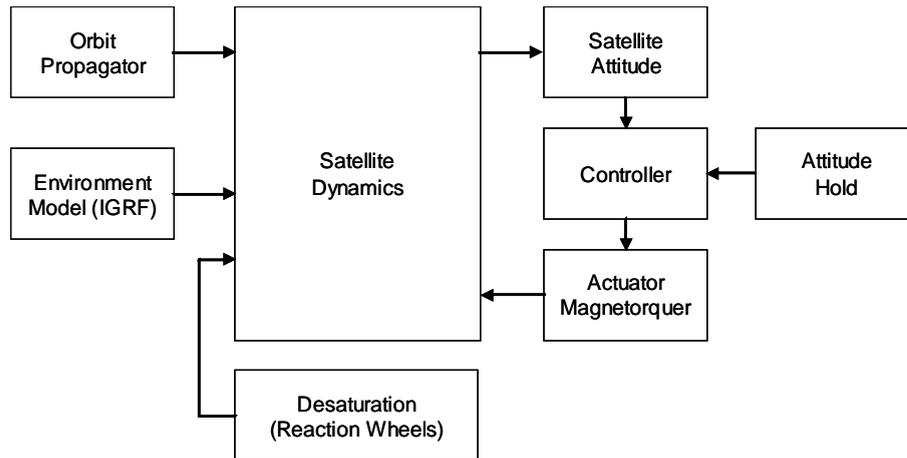


Figure 3.8 Magnetorquer Control for Reaction Wheel Desaturation

The desaturation of reaction wheels process means simply slowing down or speeding up of the reaction wheels to their nominal rotational rates. However, these speed changes will generate torques, affecting the satellites attitude. For this reason, magnetorquers are used to hold the satellite in its current attitude while the wheels speeds regulated. In the example below, the attitude is hold (zero Euler angles) during the control scheme.

The torque generated by the magnetorquer coils are given by;

$$\vec{T}_{Control} = \vec{M}_{Satellite} \times \vec{B}_{Earth} \quad (3.12)$$

Where,

$$\vec{M} = n \cdot \vec{i}_c \cdot A \quad (3.13)$$

$$\vec{i} = [i_x \quad i_y \quad i_z] \quad (3.14)$$

At first glance, it looks like the required control input direction, in terms of the currents for each magnetorquer, can be determined from Equation (3.12). However, cross product does not have an inverse operator, cross product of any vector which lies in magnetorquer and magnetic field plane with the magnetic field vector will be in the same direction with the required control input. In the case of cross multiplication of $A \times B$, there are many possible directions for vector B that will give the same value for

$A \times B$, in fact, any direction in the plane that is normal to $A \times B$ (except a vector pointing in the same direction as A).

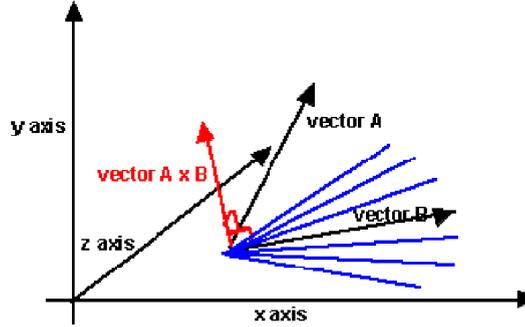


Figure 3.9 Control Current Direction Determination

From this, the cross product of the required current with the magnetic field will lie in $\vec{M}_{Satellite} - \vec{B}_{Earth}$ plane and can be used for the current direction. The required current direction can be determined as;

$$\frac{\vec{I}}{|\vec{I}|} = \frac{\vec{T}_{Control} \times \vec{B}_{Earth}}{|\vec{B}_{Earth}|} \quad (3.15)$$

Now, a controller must be designed to control the magnitude of the currents. Similar PD control law used for RW control is also implemented for the magnetorquer control. On the other hand, the simulation results are nearly the same as the way it was expected. This is due to the fact that, current Earth's magnetic field direction determines the available torque direction, which is, of course, not always in the direction required in order to compensate the torques introduced by the reaction wheels during desaturation. To test the success of this mode a simulation is carried out. During simulation controller gains $K_D = -2$ and $K_P = -0.3$ as shown in Figure 3.10 is used.

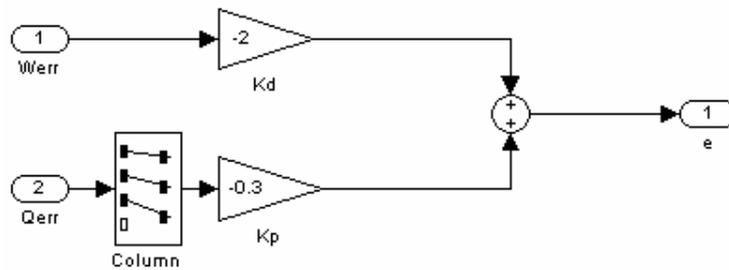


Figure 3.10 Magnetorquer PD Controller

Desaturation mode simulation results are given below from the Figure 3.11 to Figure 3.14. In this simulation the roll attitude is commanded to 10^0 degrees while yaw and pitch attitudes were kept at zero degrees. The resulting attitude during the operation may be observed from the Figure 3.11

Figure 3.12 gives the reaction wheel torques. These are due to the torques generated while bringing the wheel speeds to their nominal values. Figure 3.13 gives the magnetic torques applied to the magnetorquers. Finally Figure 3.14 shows the reaction wheel speed. From this figure it may be observed that the wheel speeds have reached their desired nominal velocities.

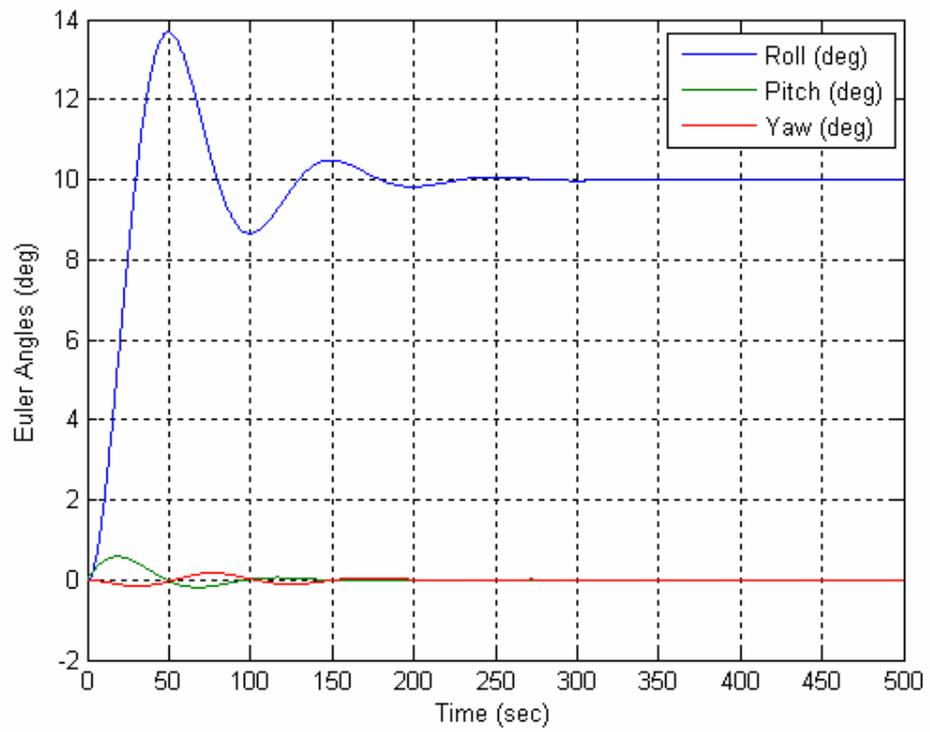


Figure 3.11 Euler Angles during desaturation mode

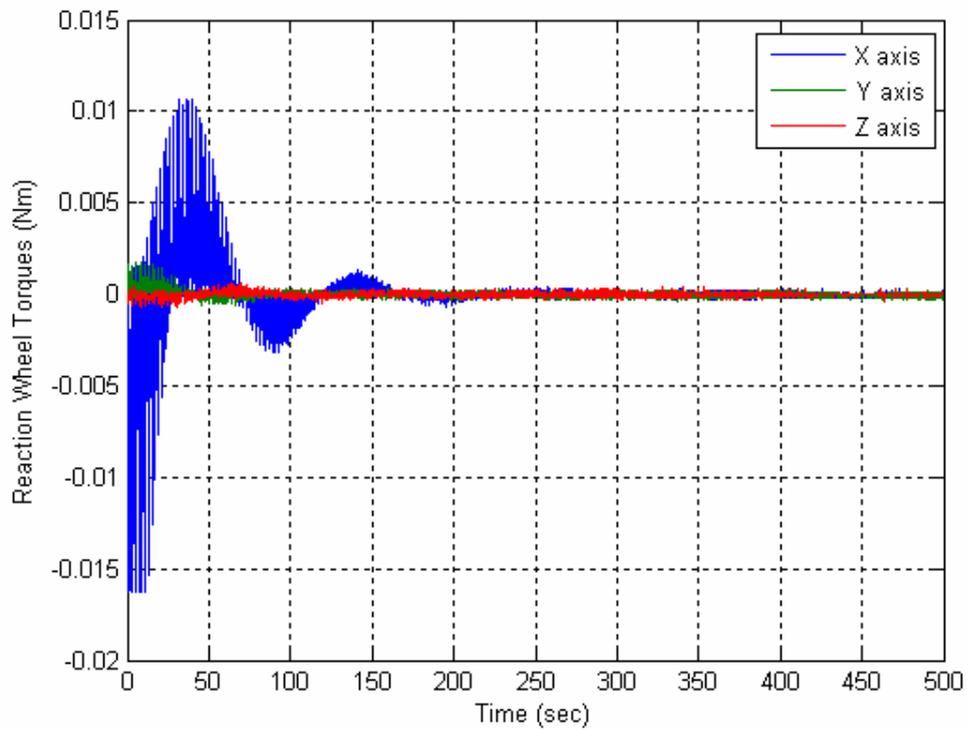


Figure 3.12 Reaction Wheel Torques in Nominal Mode

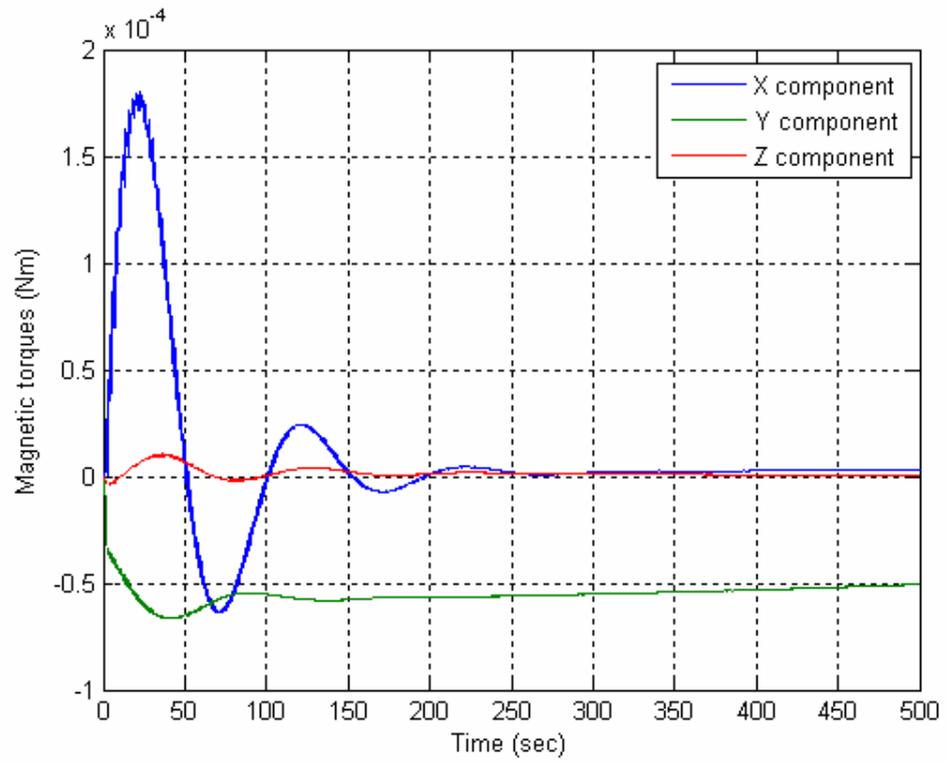


Figure 3.13 Magnetic Torques in Nominal Mode

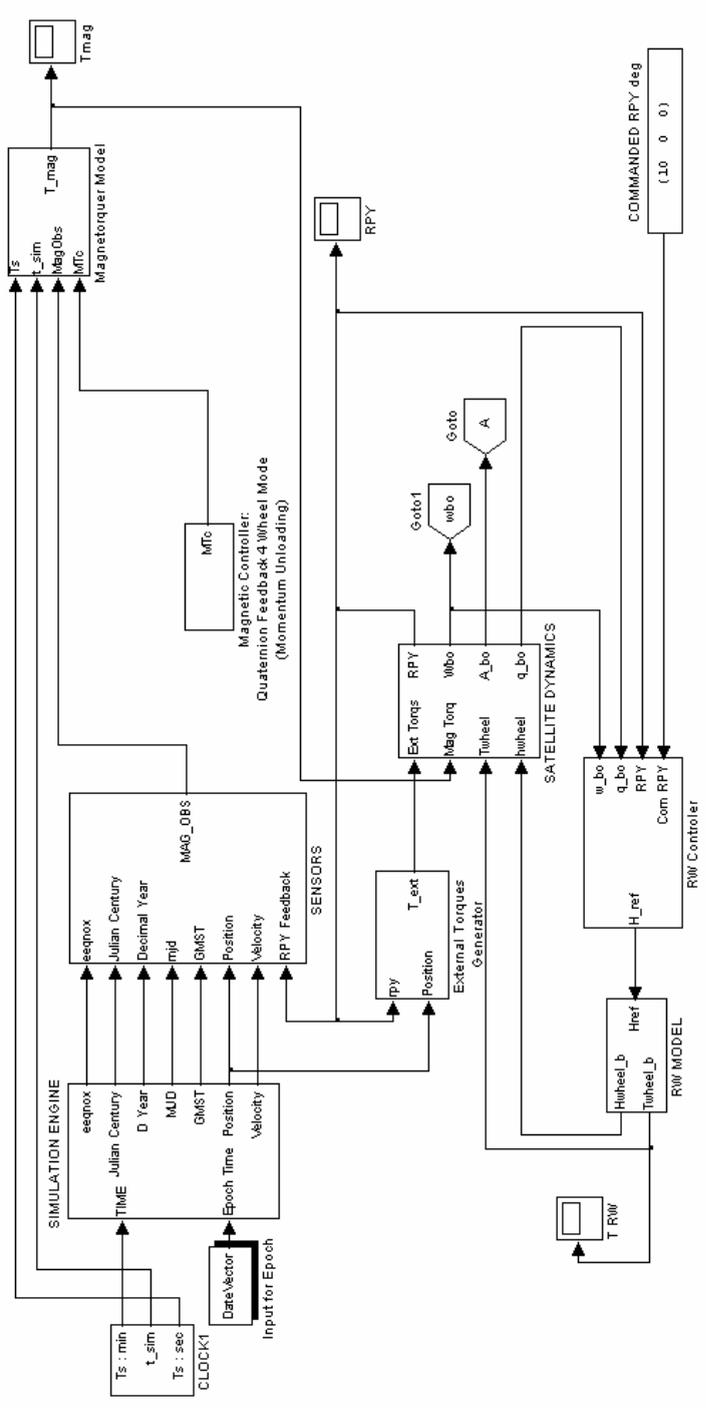


Figure 3.14 Nominal Control Mode Simulink Diagram

CHAPTER 4

ACTUATOR FAILURE COMPENSATION THROUGH CONTROL ALLOCATION METHODS

4.1. Literature Survey

Actuator failures in LEO satellites play an important role for the success of the missions. In recent years, literature on the issues related to the actuator failures, has increased. Although the controllability and stability issues for providing sufficient control with one, two and three independent control torques are studied in the literature, the actuator failure compensation through control allocation is not very common.

Tsiotras and Doumtchenko [17] considered one actuator failure case to solve the problem of detumbling with simultaneous attitude stabilization about the unactuated axis and focused on the stabilization problems. Time-varying and time-invariant feedback controllers are designed for the stabilization of rigid spacecrafts during the actuator failures. Gas jet actuators are used for the stabilization of the spacecraft. Feasible trajectory generation is also covered for the attitude tracking problems.

Li and Zhang [18] investigated the failure of one reaction wheel of total four and proposed a new reconfiguration structure for the controller using a predictive filter. In this thesis, it is assumed that the actuator failures are diagnosed by the operators and the controllers are reconfigured according to this basic knowledge and the flight code is uploaded from the groundstation. Li and Zhang suggested a fault diagnosis condition to identify the actuator failures in real time and reconfigured the controller.

Roberts et al.[19] has shown that three axis control is achievable with two out of four reaction wheels and three magnetic torque bars. A new orthogonal coordinate system which decouples reaction wheel and magnetic control torques from each other, was proposed. Control torques generated by the magnetorquers are reconfigured by modifying the control algorithms for momentum unloading of the reaction wheels and the control modes and also the PWM cycle is reduced to a 3 seconds time interval from 10 seconds range. Finally the predictive techniques are applied for the cases that the available torque is not sufficient for the pointing control.

In the recent work of Sakai et al [20], three magnetic torquers and one momentum wheel are used as actuators for the three-axis attitude control. Magnetic torquers are used to generate control torque in two axes and the control of the third axis is assured with a momentum wheel. The singularity case is defined as the point that one of the magnetic field components is zero during a period in satellite orbit. Pseudo inverse and singularity robust inverse are used to solve the control allocation problem and to apply the linear feedback controller. In addition, some anti-windup methods are applied to cover the unstable behaviors and a new feedback controller is designed.

Tekinalp and Yavuzoglu [5] have proposed a new inverse kinematics algorithm called Blended inverse, which provides singularity avoidance and quick transition through CMG clusters internal singularities during attitude maneuvers. The maneuvers are either carried out by planning the gimbal trajectories in advance or by disturbing the gimbals at a singularity.

Ge et al. [21] presented a new approach by using genetic algorithm for the optimal control of the nonlinear satellite systems using only two of the total three reaction wheels after a reaction wheel failure. Ge applied the steering problem solution to a drift free control system when the total angular momentum of the system is zero. Steering of the spacecraft is obtained by genetic algorithm.

Boskovic et al [22], has shown the failure detection and identification and adaptive reconfigurable control stability in the presence of actuator failures. The spacecraft is modeled as a rigid body and six thrusters are assumed as the torque sources in three axes. The failure modes are modeled to compensate for the effects of

the failure of the each thruster. The types of the failures are described according to some assumptions that control objectives are accomplished and the global stability of the proposed techniques is proved.

Barnes et al [23], proposed attitude control system software which is “integrated with a control mode manager that dictates which software components are currently active. Control mode changes are performed by that mode manager component or by a manual command. Actuator failure is checked with Failure Detection and Handling algorithm that monitors the health of Y-axis reaction wheel.

Large angle maneuvers using two reaction wheels in case of Z-axis reaction wheel failure is examined by Hodgart et al [24]. Nonlinear, time invariant, and discontinuous control approach is proposed to accomplish the stability and slewing in 3 axes is performed without transient oscillations.

Pseudo-Inverse method for the reconfigurable controller systems is investigated by Gao et al [25]. By using Modified Pseudo-Inverse, “closed-loop stability is maintained while recovering the performance as much as possible” and all the stability conditions are analyzed.

Jayaram et al [26], presented a model based actuator fault detection. The stability-based measure to diagnose an actuator failure is proposed by using Lyapunov functions as a robust control method. In an actuator failure case, “Kalman filter is used to compute the state matrix.”

Bogh et al [27], designed “Failure Mode and Effect Analysis” for all sensor and actuator systems in the satellite. Potential faults of the magnetic coils are examined and coil driver malfunction problem is considered for the simulation of detumbling control of the satellite.

In this thesis pseudo inverse (Moore–Penrose generalized inverse) and blended inverse algorithms are used and compared with each other as control allocation methods.

4.2. Control Allocation Methods

Inverse of a rectangular control input matrix with Pseudo inverse can be defined as:

$$U_{pseudo} = J^T [JJ^T]^{-1} T_{desired} \quad (4.1)$$

Where $T_{control} = J.U_{pseudo}$ and J is the Jacobian matrix.

This equation becomes singular when J loses rank and the inverse does not exist. The Jacobian matrix for three magnetorquers aligned in three axes of the satellite body frame and four reaction wheels that are mounted in a tetrahedral configuration may be written as follows:

$$J = \begin{bmatrix} 0 & -B_z & -B_y & 0 & 0 & -0.8165 & 0.8165 \\ -B_z & 0 & B_x & 1 & -0.333 & -0.333 & -0.333 \\ B_y & -B_x & 0 & 0 & 0.9428 & -0.4740 & -0.4740 \end{bmatrix} \quad (4.2)$$

As it seen from the equation 4.2, J matrix has orbital location dependent terms, that are magnetic field components. Every column in J matrix is associated by an actuator. When one of the actuators fail or not, the new Jacobian matrix may be found by removing the associated column. Some components of J matrix becomes zero during the zero crossings of magnetic field components.

The pseudo inverse given in the above equation gives the minimum norm solution of the U_{pseudo} matrix. Since the components related to the magnetorquers and those related to the reaction wheels have different magnitudes, a scaling is necessary.

A mathematical expression of singularity conditions is necessary. Tekinalp [5] proposes a singularity measure which is given in equation 4.3.

$$m = \det(JJ^T) \quad (4.3)$$

This measure is taken in the simulations given below to identify the singularities.

Blended inverse technique is proposed by Yavuzoglu and Tekinalp [5] for CMG control allocation problems. Although Tekinalp et al [5] proposed the algorithm for

singularity avoidance and or transition for CMG steering, the algorithm may also be used for weighting the desired actuators against the undesired ones during maneuver. This additional flexibility is also used in this study. Note that no singularity is encountered when both magnetorquers and reaction wheels are used. Saturation type singularities occur when the reaction wheels reach their maximum torque and angular momentum limits or when some magnetic field components become zero around pole passages.

Blended inverse algorithm can be given as:

$$U_{blended} = [I_{n \times n} - \beta \cdot J^T (I_{n \times n} + \beta \cdot J J^T)^{-1} J] \times (U_{desired} + \beta J^T T_{desired}) \quad (4.4)$$

Where $U_{blended}$ is the control input matrix that is calculated by Blended inverse algorithm, I is identity matrix, β is smoothing gain for blended inverse that is chosen as 0.01, J is Jacobian matrix, $U_{desired}$ is desired control inputs which allows balancing between magnetorquers and reaction wheels. Moreover, it is possible to decide which actuator will be used dominantly during the maneuver. In this thesis, it is observed that the torque capacity of reaction wheels can not be used optimally during the maneuvers and it causes unstable conditions around singularities. Thus the magnetorquer gains in $U_{desired}$ vector are chosen greater than the reaction wheel components. The $U_{desired}$ vector used for the simulations with one reaction wheel and three magnetorquers is given as an example in equation 4.5.

$$U_{desired} = [3.1 \quad 3.1 \quad 3.1 \quad 0.005] \quad (4.5)$$

As it seen from the equation 4.5, the desired magnetorquer values are chosen 3.1, as half of the maximum magnetic moment limit of the magnetorquers. However the desired reaction wheel component is chosen relatively very small.

4.3. Actuator Failure Cases

4.3.1. Attitude Control by 3 Reaction Wheels

Tetrahedral geometry has some benefits for reaction wheel control. Each normal vector of a tetrahedron surface represents a torque direction of a reaction wheel. In most cases all reaction wheels locate in the same surface over the satellite. Every wheel has wedges to form tetrahedral geometry. There are two main reasons to choose the tetrahedral geometry.

In tetrahedral configuration, maximum torque output can be increased to twice the maximum torque limit of a reaction wheel. Geometric projection of a unit torque in each of three axes contributes a unit torque to that axis. Thus, using that geometric property of the cluster, it is possible to generate more torque on each axis than possible by a simple reaction wheel.

Another important reason to use tetrahedral configuration is for redundancy. Basically, three reaction wheels that are mounted orthogonally are enough to generate torques in 3 axes. The fourth reaction gives an overactuated system. Thus, one wheel fail attitude control is not so different from the nominal 4 wheel control mode.

4.3.2. Slew Maneuver with 2 Magnetorquers and a Y axis Reaction Wheel

In this mode 2 magnetorquers and a Y-axis reaction wheel are used as actuators. Magnetorquers generate torques in two axes and Y wheel generates torque in pitch axis only. First direct inversion is used to find the controls. The main purpose of this simulation is to see the torque usage level of each actuator. The J matrix for this mode can be given as:

$$J = \begin{bmatrix} 0 & 0 & -B_y \\ -B_z & 1 & B_x \\ B_y & 0 & 0 \end{bmatrix} \quad (4.7)$$

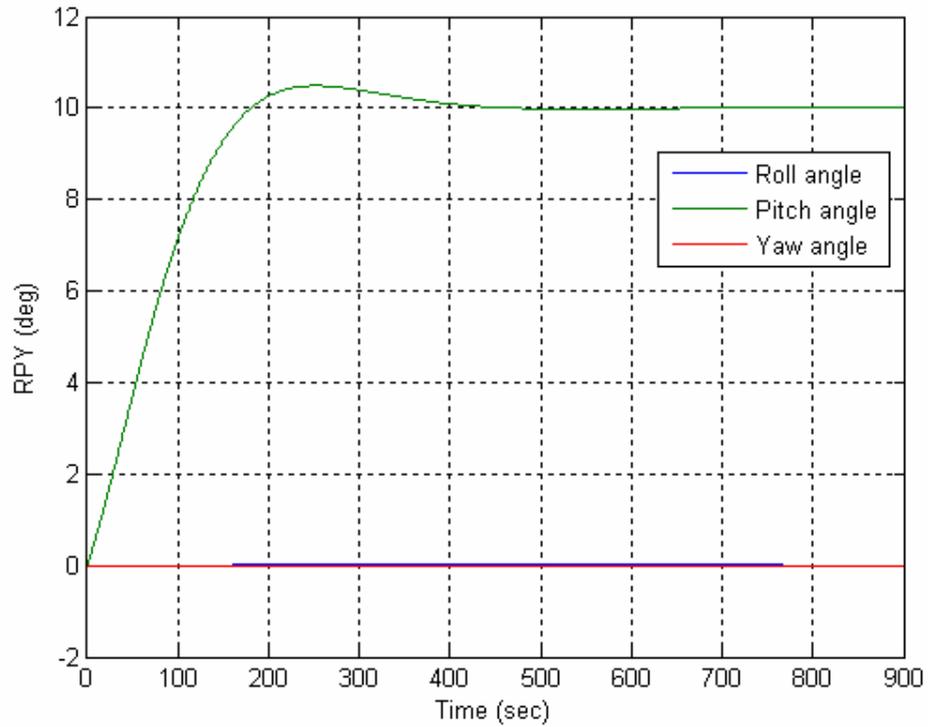


Figure 4.1 Euler angles time histories after 20° commanded roll angle using blended inverse in singular zone with two magnetorquers and a reaction wheel

The simulation results may be examined from Figure 4.1 to 4.4. Mathematically there is no difficulty in taking the inverse of a square matrix. Again the singularity is dependent on the magnetic field components. In this mode pitch axis maneuver is achieved as it seen from the Figure 4.1.

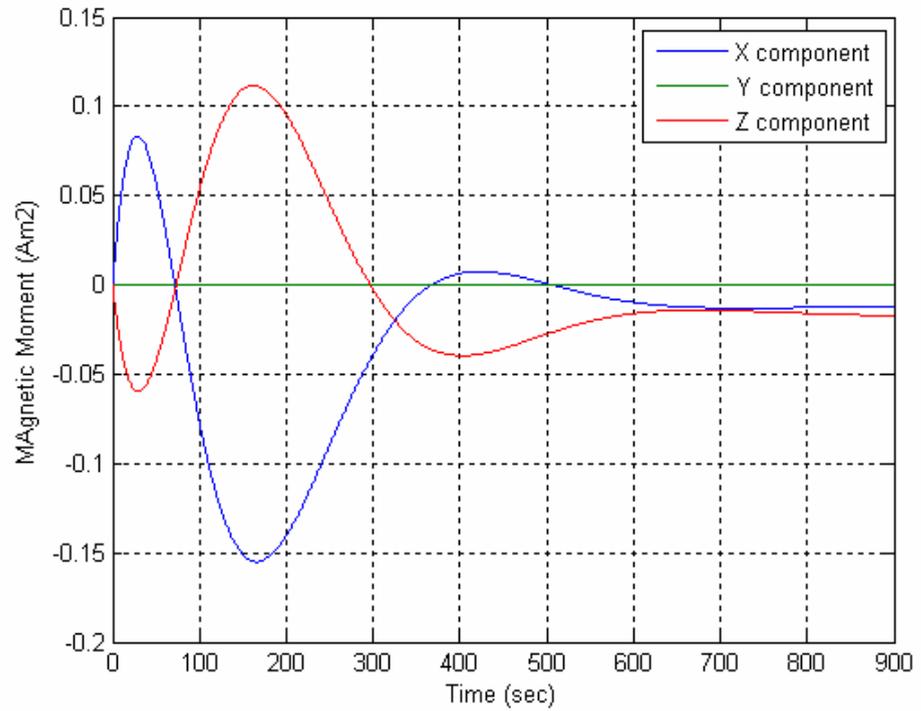


Figure 4.2 Magnetic Moments for 2 Magnetorquers and Y Wheel Mode

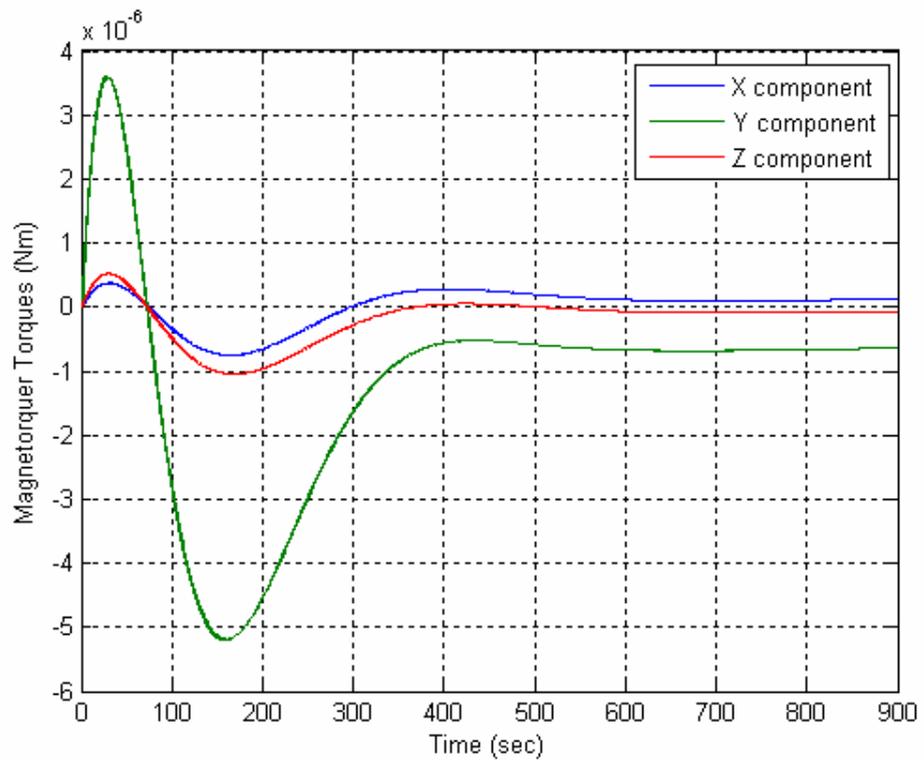


Figure 4.3 Magnetic Torques for 2 Magnetorquers and Y Wheel Mode

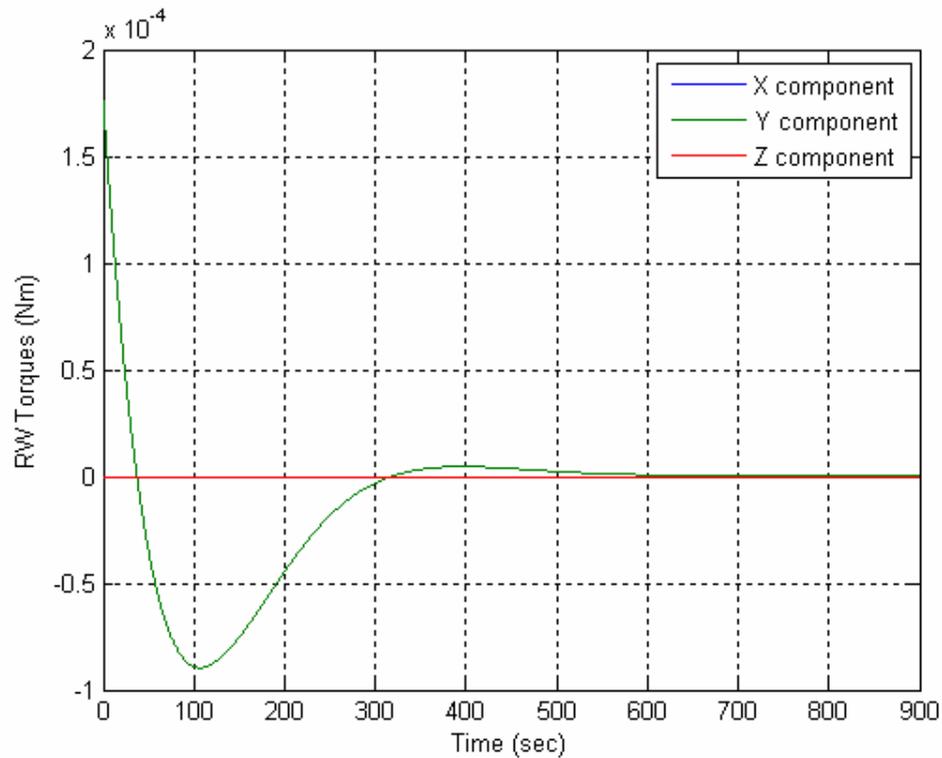


Figure 4.4 Reaction Wheel Torques for 2 Magnetorquers and Y Wheel Mode

4.3.3. Slew Maneuver with 3 Magnetorquers and Y axis Reaction Wheel

In this mode, Y axis reaction wheel and 3 magnetorquers are used as actuators. Y axis wheel can be considered as the most critical wheel for the operational safety and control capabilities. The main reason is that the Y wheel generates the pitch torque necessary for stereoscopic imagery maneuvers. Also the pitch rate control that is necessary for nadir pointing is also assured by Y reaction wheel.

Available torque capacity in Y axis is sufficient for a pitch maneuver. However, necessary torque will be increased during a roll or yaw axis maneuver due to gyroscopic moments. Because of the insufficient torque capacity in that axis, the control torques generated by actuators will be compensated by the Magnetorquers. Thus, the torques will be depended on varying magnetic field components. This dependency may cause some singularities during the maneuvers.

In this thesis the singularity term is used for some different meanings. Mathematically it is the singularity of the Jacobian matrix. Saturation singularity term is also used when the actuators reach their available torque limits. During the maneuvers, the points where the Jacobian matrix is singular (i.e. loses rank) are called singular zones. This is possible when Earth's magnetic field direction changes at the Polar Regions.

For example in Figure 2.9, magnetic field Y component is zero at certain locations. If a maneuver is planned at those locations, Jacobian matrix will definitely be singular. Through pseudo inverse it is not possible to transit these singular zones. These arguments are demonstrated in the following simulation. A roll maneuver of 20° which is perpendicular to Y axis is carried out to compare the pseudo inverse and blended inverse.

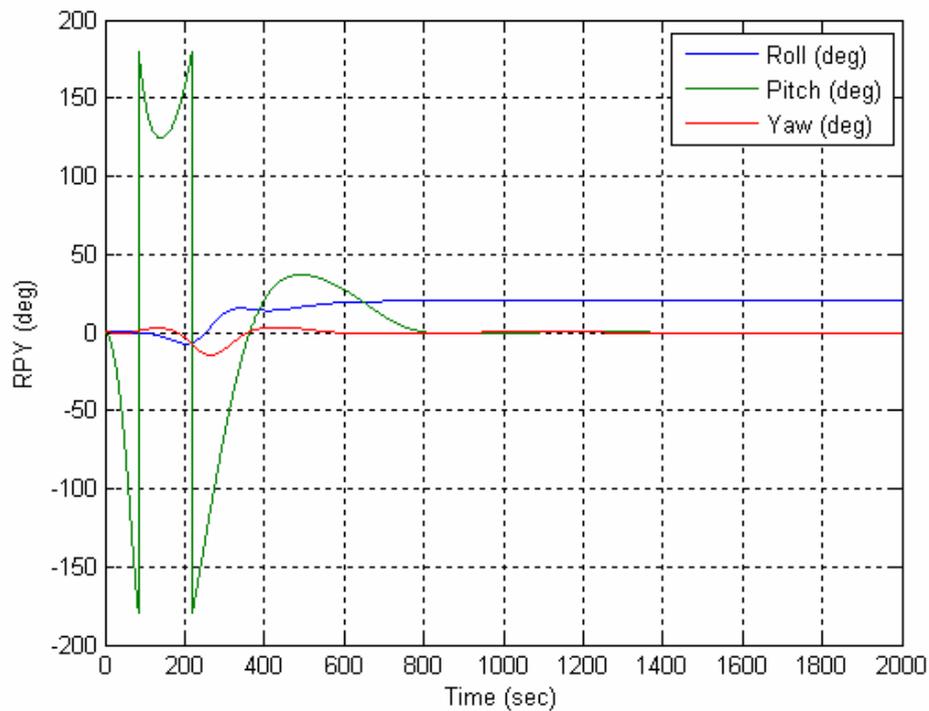


Figure 4.5 A 20° roll maneuver with pseudo inverse at a singular zone

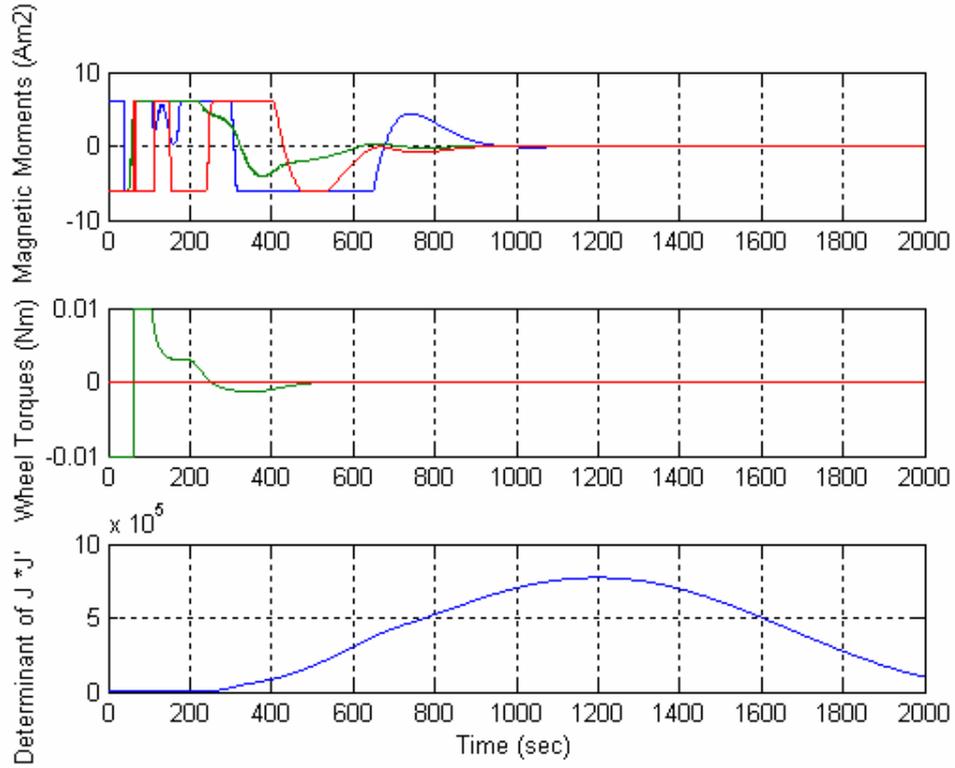


Figure 4.6 Magnetic moments, Y axis reaction wheel torque and the singularity measure during 20^0 roll maneuver using pseudo inverse algorithm in a singular zone

Three magnetorquers and Y-axis reaction wheel are used as actuators. In this case the Jacobian becomes:

$$J = \begin{bmatrix} 0 & B_z & -B_y & 0 \\ -B_z & 0 & B_x & 1 \\ B_y & -B_x & 0 & 0 \end{bmatrix} \quad (4.6)$$

First a 20^0 roll maneuver simulation is carried out using pseudo inverse. As it observed from the Figure 4.5 pseudo inverse causes excessive rolls over 200 degrees. Moreover it is clear in Figure 4.6 that the Y reaction wheel reaches its saturation limit for a while. The reason for this erratic behavior may easily be deduced if the singularity measure given in Figure 4.6 is examined. At the beginning of the maneuver the system is singular (i.e., $m=0$). In Figure 4.7 blended inverse simulation results are given. It is clear from this simulation that the blended inverse method provides a quick transition from the singularity zone despite the relatively

high overshoot. For Figure 4.8 it may be observed that the singularity measure $J * J'$ is zero for 200 seconds but the blended inverse algorithm avoids the saturation of reaction wheel (Figure 4.8). This smooth transition can be seen from Figure 4.7 and Figure 4.8.

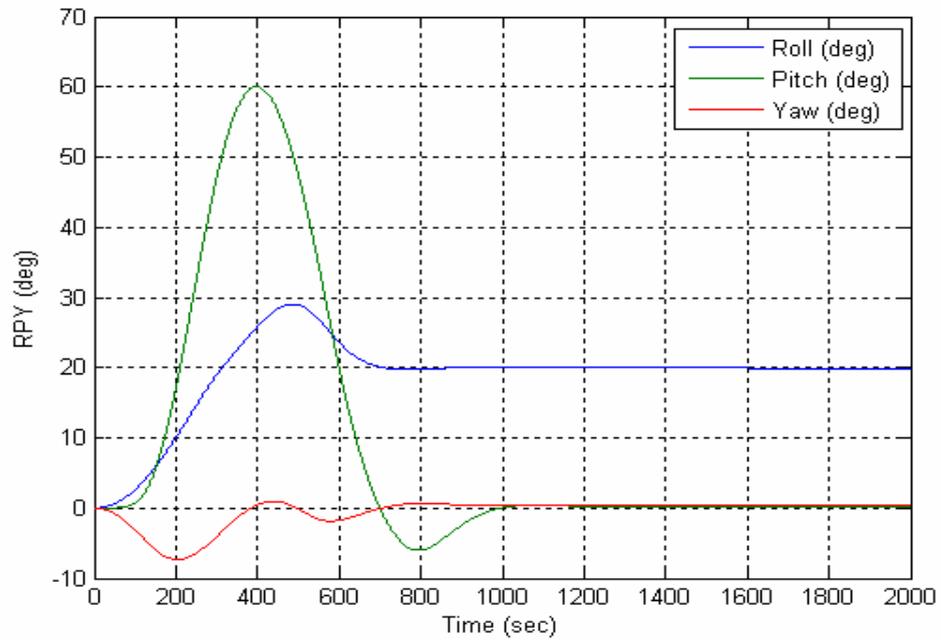


Figure 4.7 Euler angles during 20° Roll maneuver with blended inverse algorithm in a singular zone

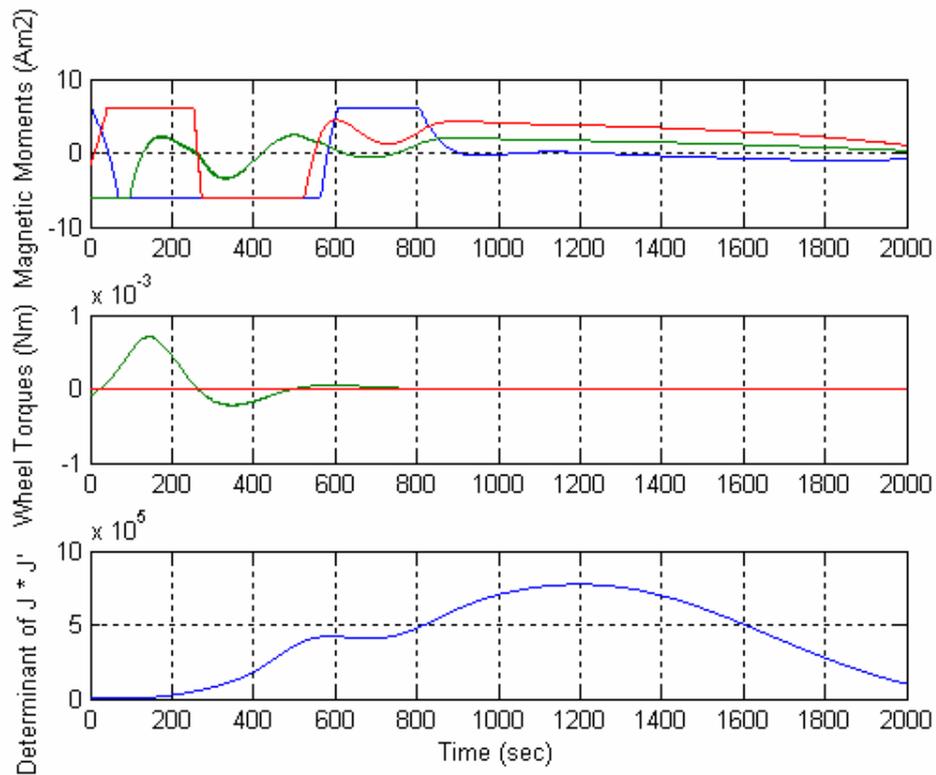


Figure 4.8 Magnetic moments, Y axis reaction wheel torque and the singularity measure during 20° roll maneuver using blended inverse algorithm in a singular zone

The above simulation is repeated this time commanding 20° degrees maneuvers in all roll, pitch and yaw axis in the singularity zone as well. The pseudo inverse algorithm results are presented in Figure 4.9 and 4.10. The erratic behavior of the satellite is clear in these figures.

The simulation results of the same maneuver using the blended inverse algorithm are presented in Figure 4.11 and 4.12. It may be observed from these figures that the blended inverse algorithm is very successful. The Euler angles reach the commanded values quickly and the reaction wheel is not saturated.

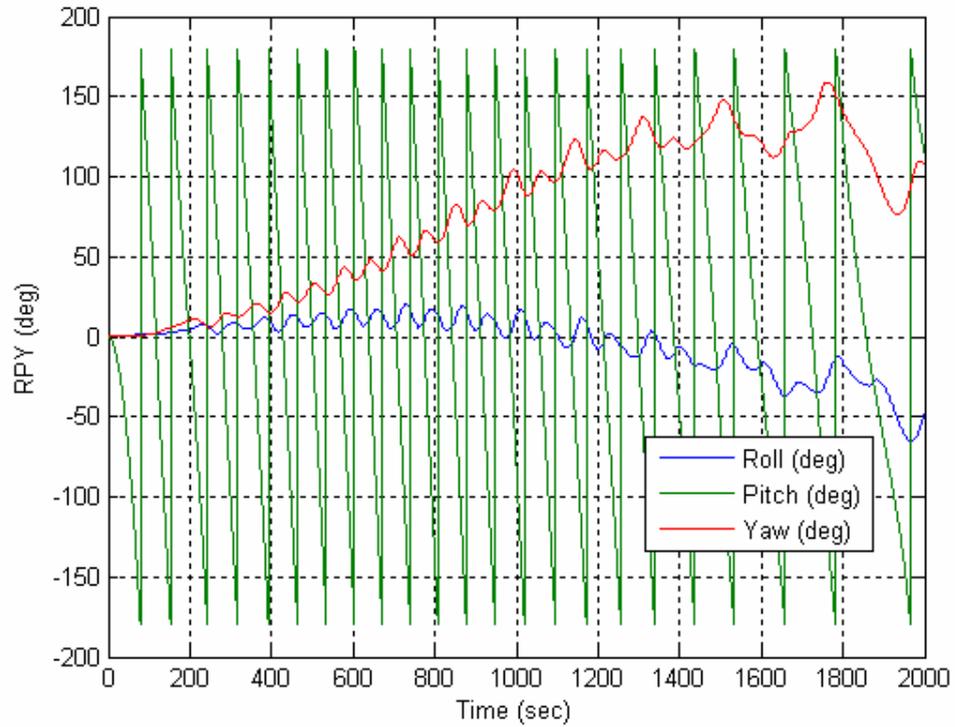


Figure 4.9 Euler angles time histories after 20^0 commanded roll, pitch and yaw angles using pseudo inverse in a singular zone

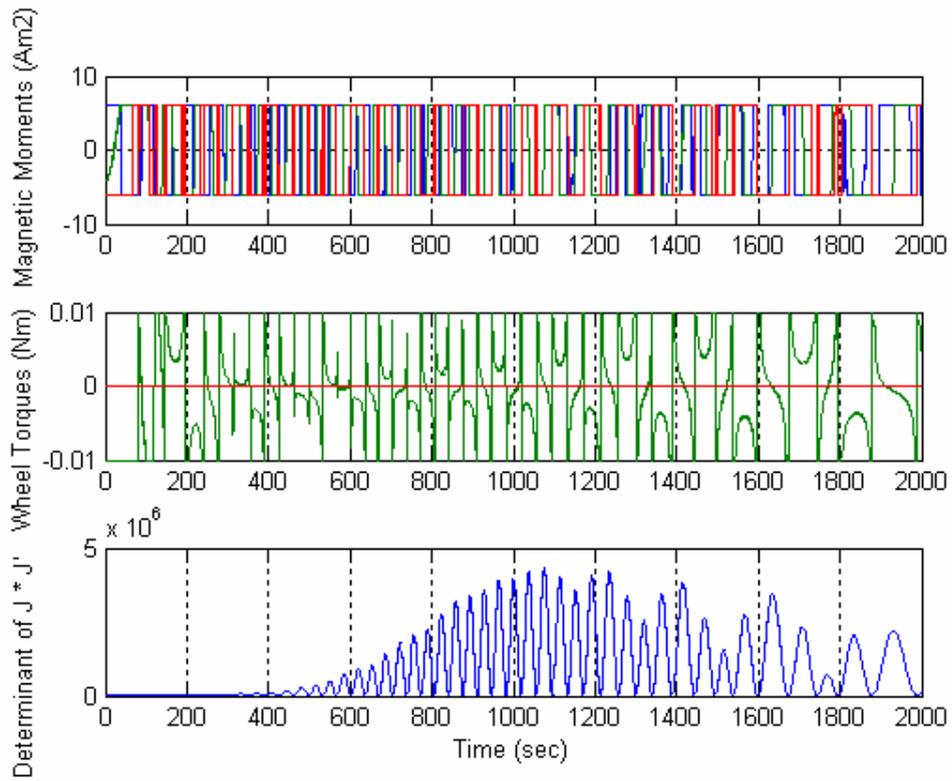


Figure 4.10 Magnetic moments, Y axis reaction wheel torque and the singularity measure during 20^0 RPY maneuver using pseudo inverse algorithm in a singular zone

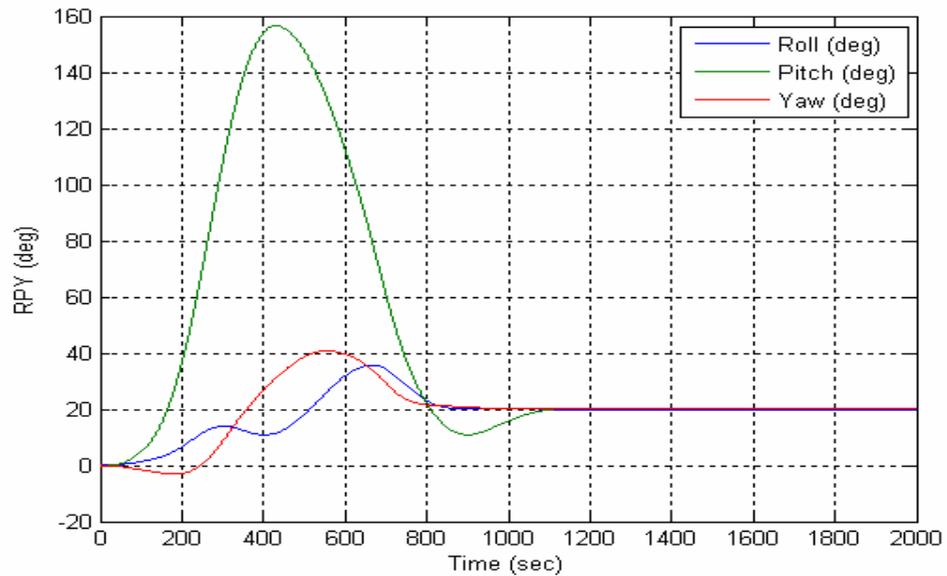


Figure 4.11 Euler angles time histories after 20^0 commanded roll, pitch and yaw angles using blended inverse in singular zone

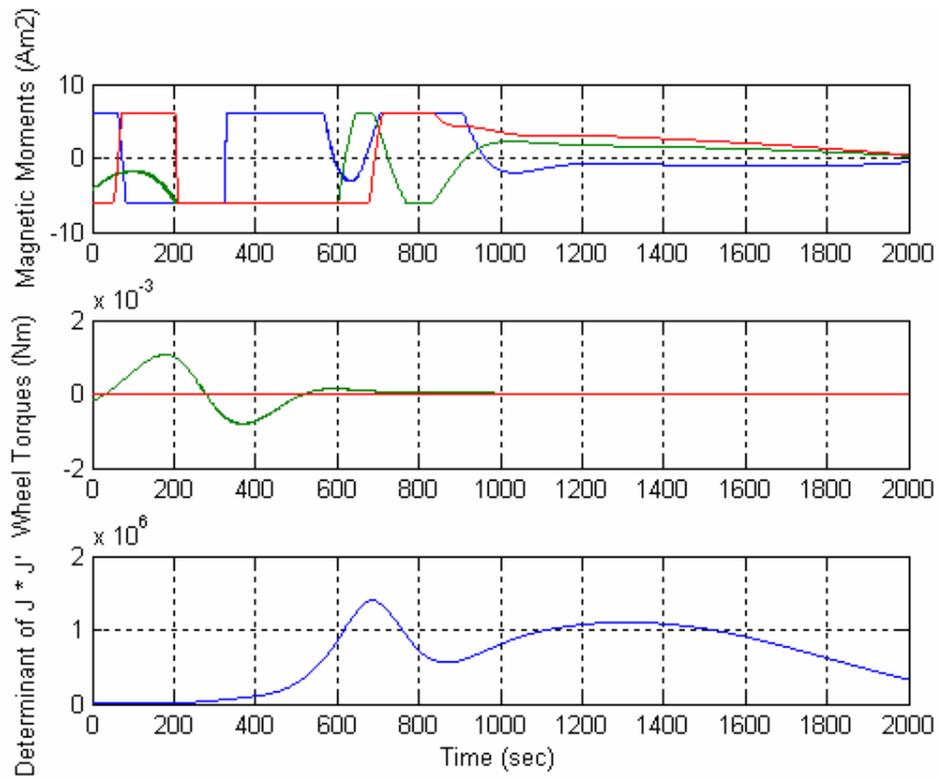


Figure 4.12 Magnetic moments, Y axis reaction wheel torque and the singularity measure during 20^0 RPY maneuver using blended inverse algorithm in a singular zone

4.3.4. 3 Magnetorquers and two Reaction Wheel Control Mode

Again, pseudo inverse and blended inverse methods are used for the simulation. First, a 20^0 roll maneuver is carried out using pseudo inverse algorithm. The results are given in Figure 4.13 to 4.16. Figure 4.13 gives the desired torques for this maneuver. The resulting attitudes are given in Figure 4.14. It may be observed that the roll attitude reaches 20^0 within 400 seconds. The satellite angular velocity, magnetic moments, magnetic torques and reaction wheel torques are presented in the remaining figures.

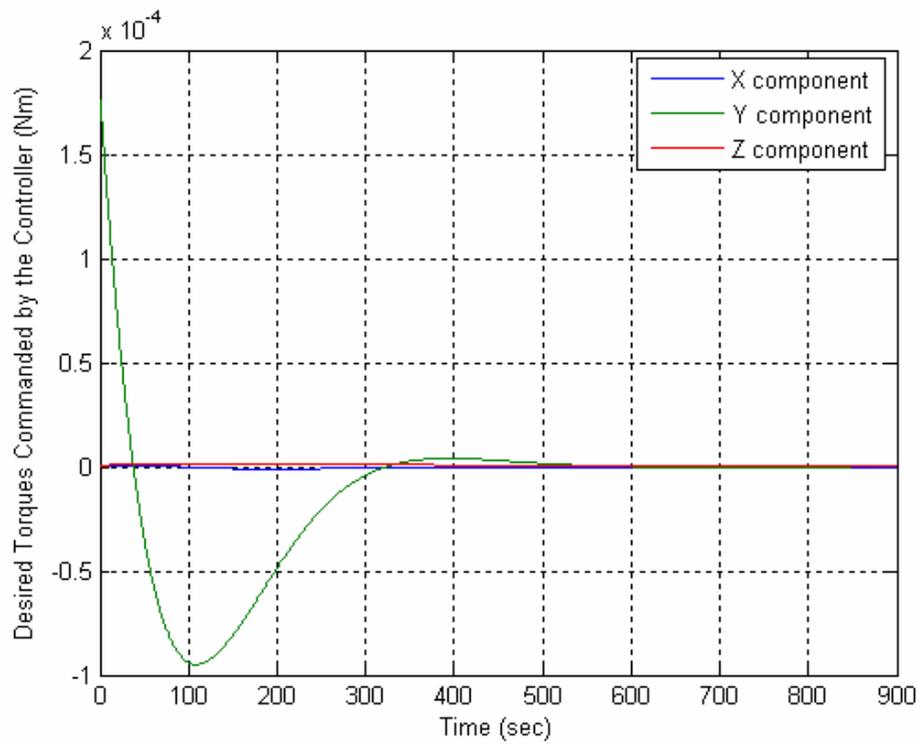


Figure 4.13 Desired Torques for 2 Reaction Wheels and 3 Magnetorquers Mode

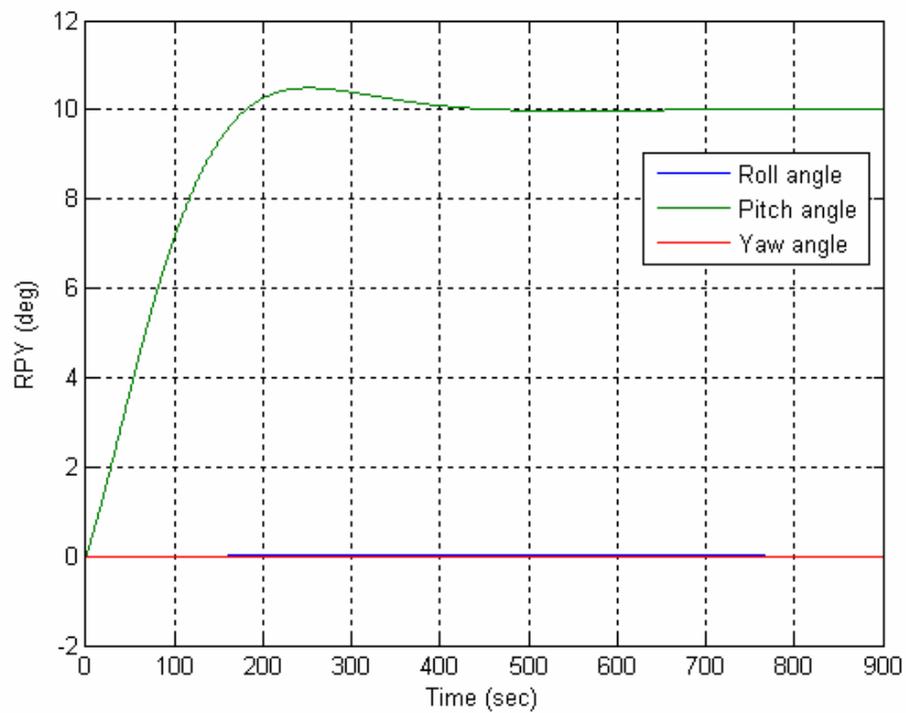


Figure 4.14 RPY for 2 Reaction Wheels and 3 Magnetorquers Mode

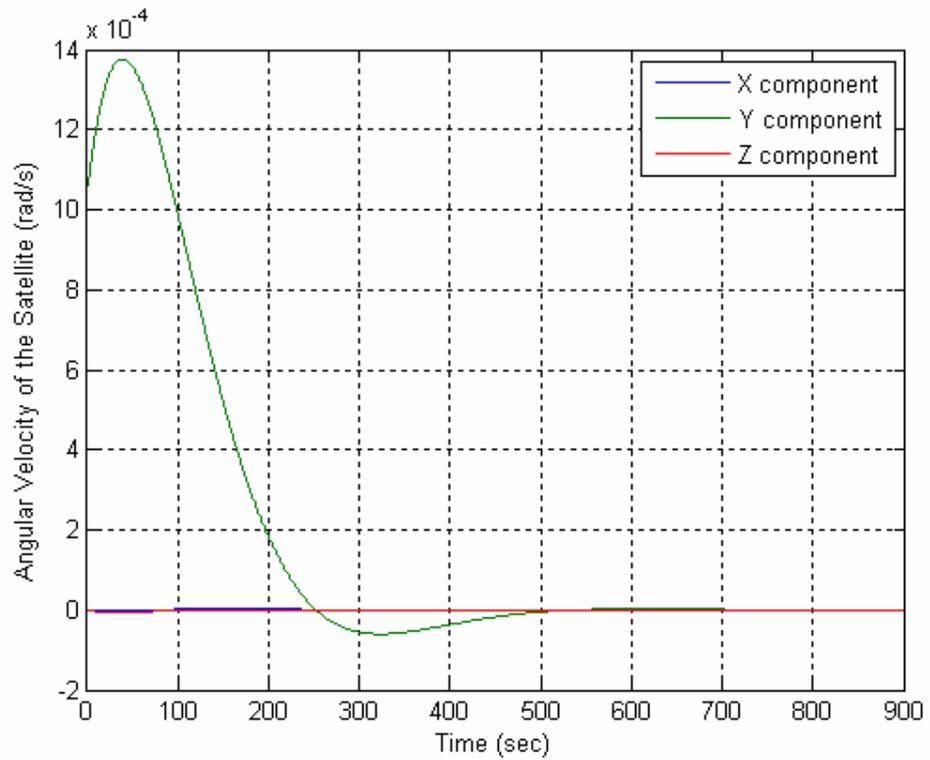


Figure 4.15 Angular Velocity for 2 Reaction Wheels and 3 Magnetorquers Mode

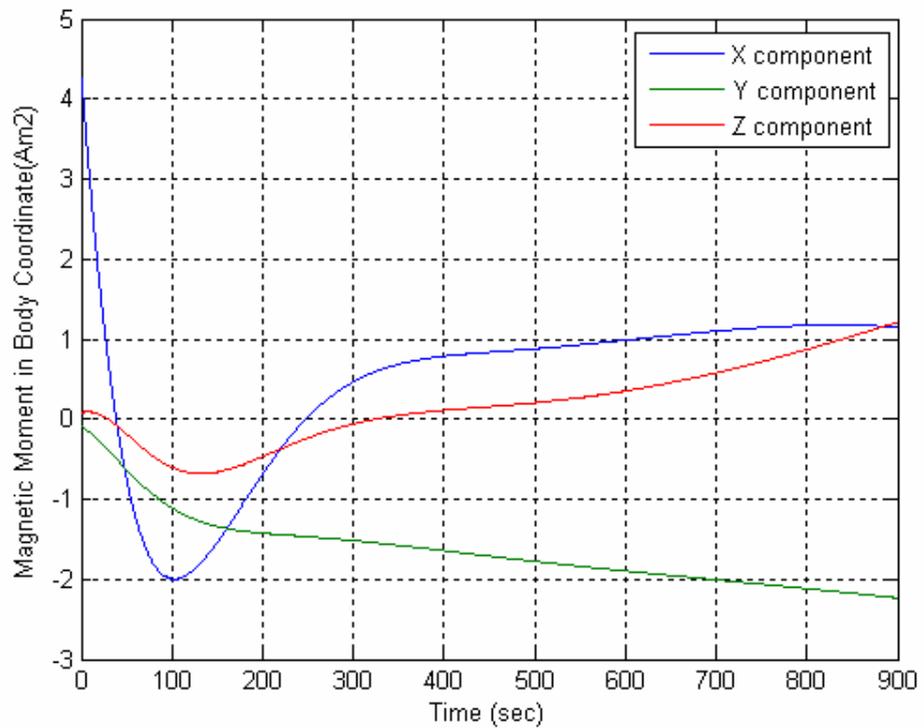


Figure 4.16 Magnetic Moments for 2 Reaction Wheels and 3 Magnetorquers Mode

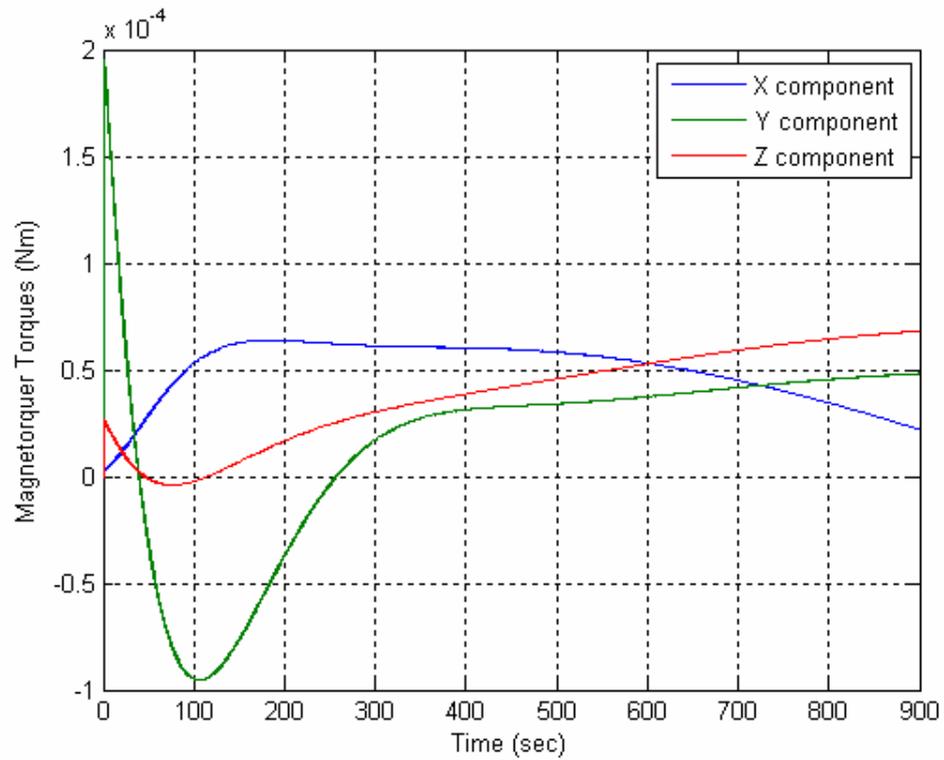


Figure 4.17 Magnetic Torques for 2 Reaction Wheels and 3 Magnetorquers Mode

From the pseudo inverse simulation results, it is seen that the reaction wheels are not used effectively during the maneuver. For this reason blended inverse algorithm is also tested. Again the $U_{desired}$ vector is chosen as follows:

$$U_{desired} = [3.1 \quad 3.1 \quad 3.1 \quad 0.005 \quad 0.005] \quad (4.8)$$

In the simulation $\beta = 0.001$ is used. The value of β is tuned by simulation results and the optimal value is chosen.

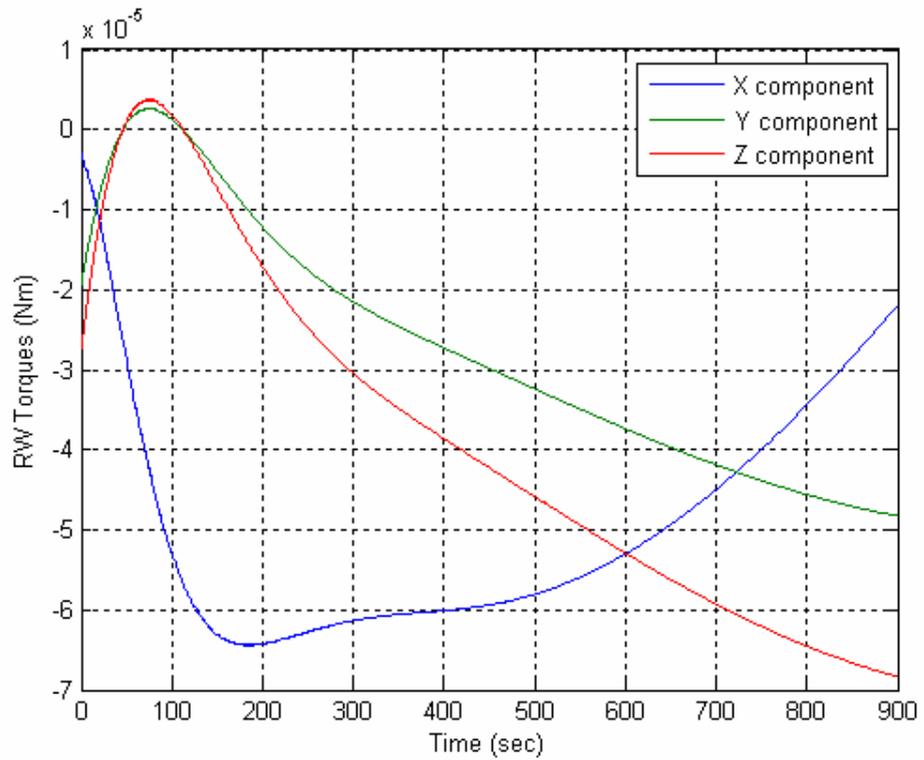


Figure 4.18 RW Torques for 2 Reaction Wheels and 3 Magnetorquers Mode

The Jacobian matrix in this case may be given as:

$$J_{Blended} = \begin{bmatrix} 0 & B_z & -B_y & -0.8165 & 0.8165 \\ -B_z & 0 & B_x & -0.3333 & -0.333 \\ B_y & -B_x & 0 & -0.4714 & -0.4714 \end{bmatrix} \quad (4.9)$$

As it seen from the equation, Jacobian matrix entries related to the magnetic field changes throughout the orbit.

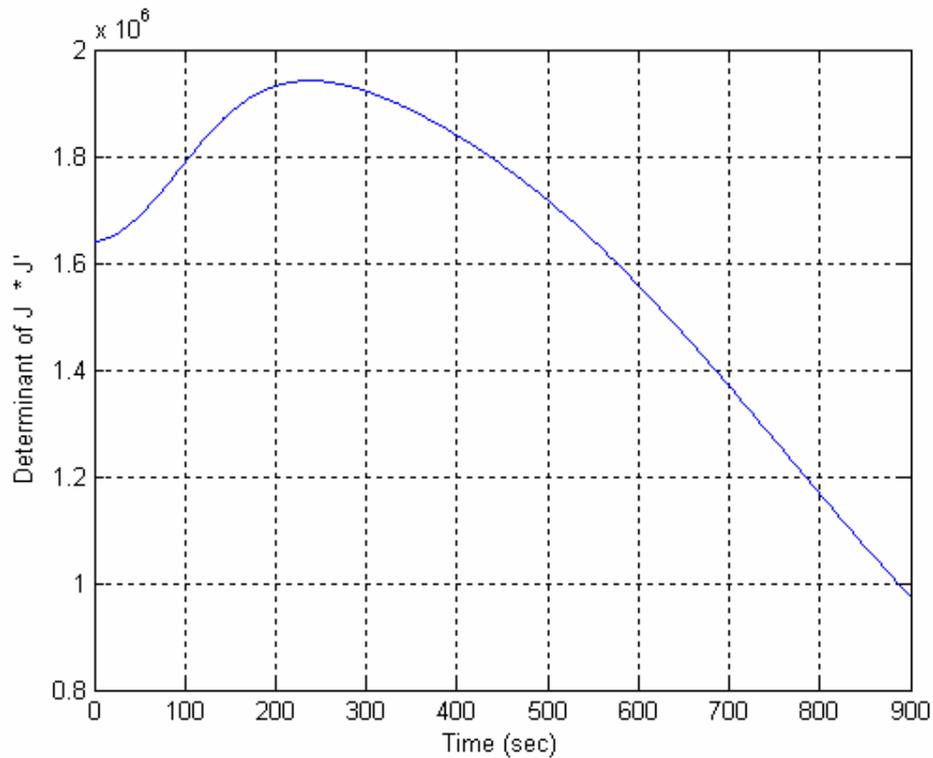


Figure 4.19 Singularity Check for Blended Inverse

Figure 4.19 gives the singularity condition of blended inverse. As may be observed from this figure, the satellite attitude is controlled quite effectively that no singularity is occurred and the maneuver is completed in time. When reaction wheel torque of Figure 4.18, and Figure 4.22 are compared, it may be observed that the torques go to zero at the end of the maneuver, with blended inverse, while pseudo inverse continues to generate torques after the completion of the maneuver. The torques of each reaction wheel in their own frame is seen from Figure 4.23.

A close examination of the magnetorquer and reaction wheel torques (Figure 4.16 and 4.17) shows that the torques generated by the reaction wheels is balanced by the magnetorquer torques. However, with blended inverse such a case is not encountered. This case, among the others, shows the strength of blended inverse control allocation approach over pseudo inverse.

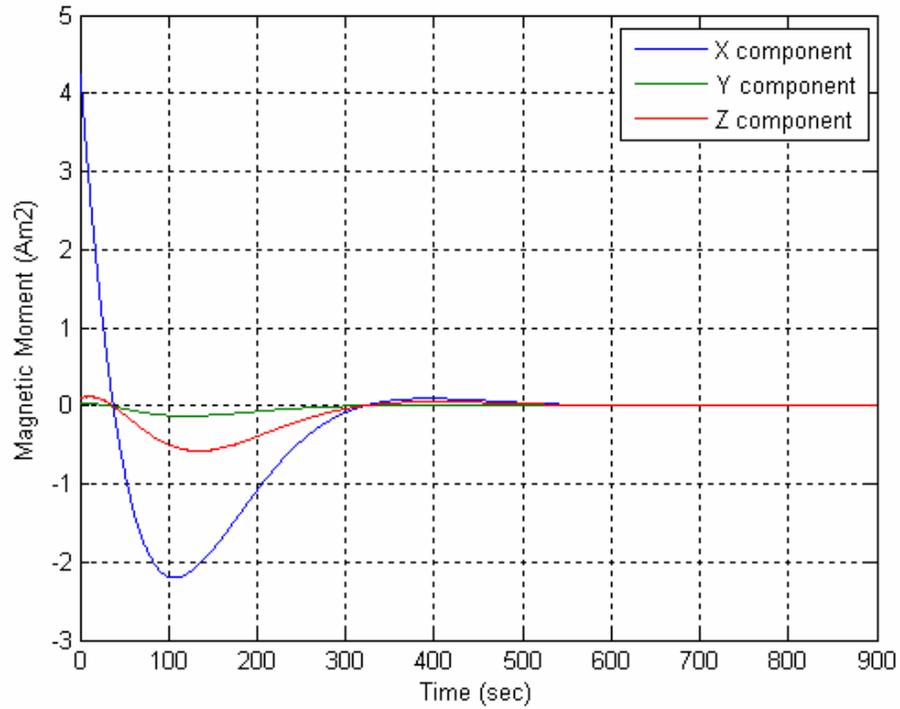


Figure 4.20 Magnetic Moments in Blended Inverse Mode

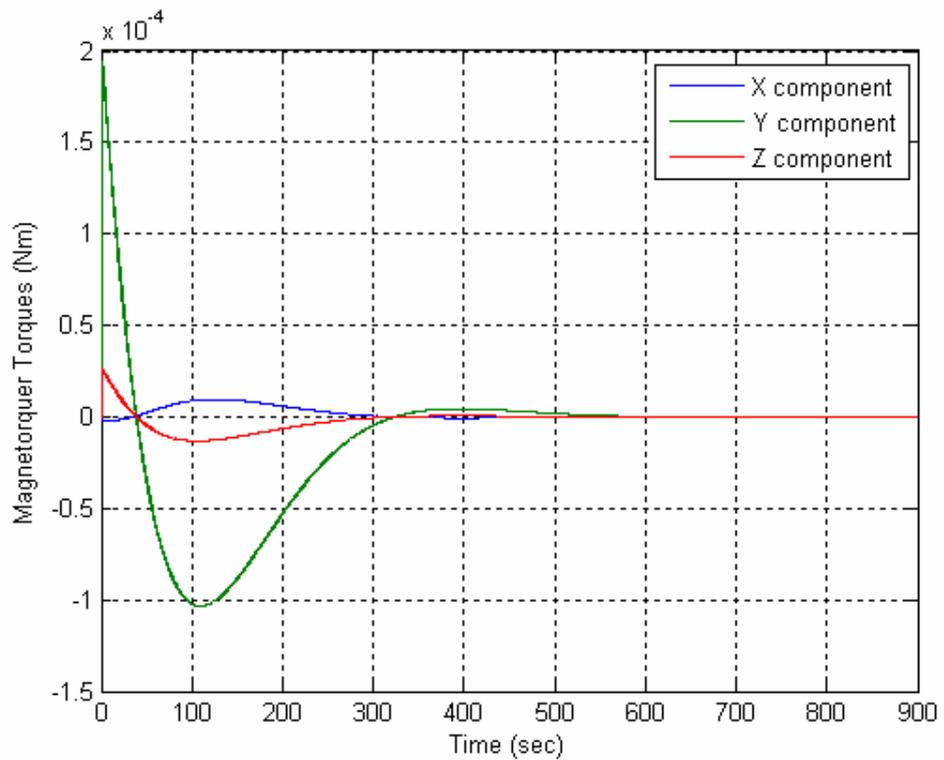


Figure 4.21 Magnetic Torques in Blended Inverse Mode

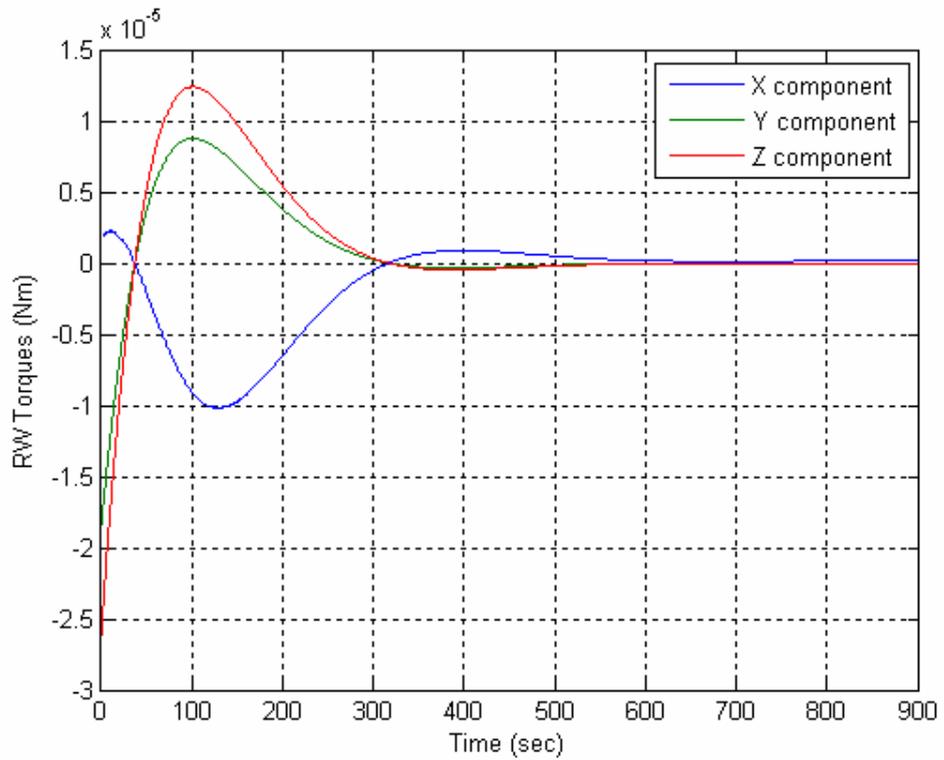


Figure 4.22 Reaction Wheels Torques in Blended Inverse Mode

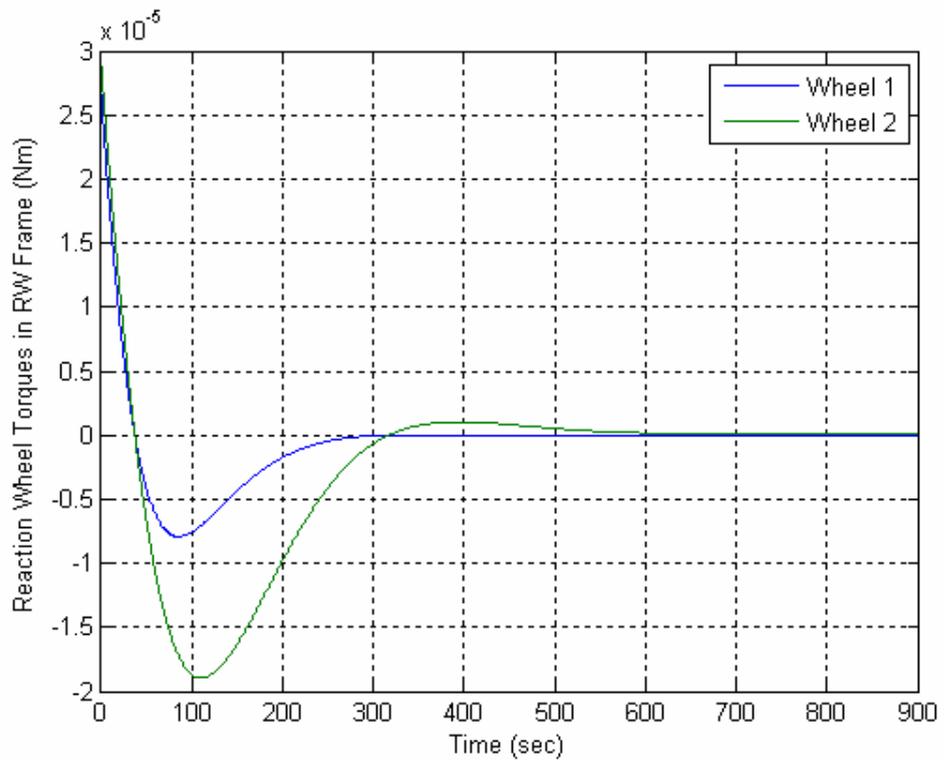


Figure 4.23 Reaction Wheel Torques in RW Frame

CHAPTER 5

CONCLUSION

In this thesis, control allocation methods to be used in the case of the actuator failures for overactuated satellites are investigated and these methods are applied to the small satellites with dissimilar actuators.

A simulation is developed to model all sensors and actuators such as magnetometers, sun sensors, magnetorquers and reaction wheels by using Matlab[®] Simulink environment. Moreover all satellite dynamics and controllers are also simulated. The mathematical background to cover satellite simulation is given in detail and the results are discussed.

The basic attitude control modes are examined and simulated. The detumbling mode and three-axis nominal control modes are designed and simulations are carried out by using magnetorquers and reaction wheels. Then the actuator failures are investigated for different cases under some special circumstances such as singularities. The control allocation method called blended inverse is applied to the satellite with dissimilar actuators.

Blended Inverse algorithm has demonstrated its ability to make a trade off between magnetorquers and reaction wheels that the other control allocation methods do not have. The simulations are made with both pseudo inverse and blended inverse and the results are compared. The results have shown that pseudo inverse method may generate unstable conditions during the singular cases of Jacobian matrix; however Blended inverse method generates quick transition ability on those points of the orbit. It is shown that this method can be used especially in actuator failure modes with overactuated satellites.

In conclusion, Blended inverse algorithms may be improved to design new configurable controllers to be used in small satellites even during the singularity zone passes.

REFERENCES

- [1] Wertz, J., R., “Spacecraft Attitude Determination and Control”, Kluwer Academic Publishers, 1978.
- [2] Kahraman, O., San, H., “Attitude Control of an Earth Pointing LEO Small Satellite” Term Project, METU, 2005.
- [3] The International Association of Geomagnetism and Aeronomy, <http://www.ngdc.noaa.gov/IGA/vmod/igrf.html>., Last accessed date November 2007.
- [4] Steyn, W.H “A multi-mode Attitude Determination and Control System for Small Satellites” Doctorate Thesis,1995.
- [5] Tekinalp, O., Altay, A., and E. Yavuzoglu, “Control Moment Gyros for Energy Storage and Attitude Control in Small Satellite Missions” Proceedings of the RTO Specialists' Meeting on Emerging and Future Technologies for Space Based Operations Support to NATO Military Operations, RTB-SPSM-001, pp 19(1-13), Bucharest, Romania, 6-7 September, 2006.
- [6] Lappas, V.J. “A Control Moment Gyro Based Attitude Control System For Agile Small Satellites” Doctorate Thesis, 2002.
- [7] Hoots, F.R., Roehrich, R.L., “Spacetrack Report No:3” NORAD, 1988.
- [8] Vallado D.A., “Fundamentals of Astrodynamics and Applications “Kluwer Academic Publishers, 2001.
- [9] T.S. Kelso www.celestrak.com., Last accessed date November 2007.

- [10] Satrec Initiative Sun Sensor ICD Documents, 2007.
- [11] Satrec Initiative Magnetometer ICD Documents, 2007.
- [12] Sidi, M.J., "Spacecraft Dynamics and Control" Cambridge University Press, 1997.
- [14] Wisniewski, R., "Project : Attitude Control System for AAUSAT-II" Aalborg University, Institute of Electronic Systems, 2004.
- [15] Wie, B., and Barba, P.M., "Quaternion Feedback for Spacecraft Large Angle Maneuvers," *Journal of Guidance*, Vol. 8, No. 3, May-June 1985.
- [16] De Marchi, E., Rocco, L. D., Morea, G., & Lovera, M. (1999). "Optimal magnetic momentum control for inertially pointing spacecraft." In *Proceedings of the 4th ESA international conference on spacecraft guidance, navigation and control systems*, ESTEC, Noordwijk, The Netherlands.
- [17] Tsiotras, P., Doumtchenko, V., "Control of Spacecraft Subject to Actuator Failures: State-Of-Art and Open Problems" *R. H. Battin Astrodynamics Symposium* , Texas A&M, College Station, TX, March 20-21, 2000. AAS paper 00-264.
- [18] Li, J., Zhang, H.Y., "Controller Reconfiguration against Reaction Wheel Failure Based on Predictive Filters "1st International Symposium on Systems and Control in Aerospace and Astronautics (ISSCAA), 2006.
- [19] Roberts, B.A., Kruk, J.W., Ake, T.B., Englar., T.S., "Three-axis Attitude Control with Two Reaction Wheels and Magnetic Torquer Bars" *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 16-19 August 2004, Providence, Rhode Island.

- [20] Sakai, S.I., Fukushima, Y., Saito, H., “Studies on Magnetic Attitude Control System of the REMEI Microsatellite” AIAA Guidance, Navigation, and Control Conference and Exhibit 21-24 August 2006, Keystone, Colorado.
- [21] Ge, X.S., Chen, L.Q., “Attitude Control of a rigid spacecraft with two momentum wheel actuators using genetic algorithm” *Acta Astronautica*, 55 (1), p.3-8, Jul 2004.
- [22] Boskovic, J.D., Li, S.M., Mehra, R.K., “Intelligent Control of Spacecraft in the Presence of Actuator Failures” Proceedings of the 38th Conference on Decision & Control Phoenix, Arizona USA, December 1999.
- [23] Barnes, K.C., Melhorn, C.M., Phillips, T., “The software Design for the Wide-Field Infrared Explorer Attitude Control System” 12th AIAA/USU Conference on Small Satellites, Utah, 1998.
- [24] Horri, N.M., Hodgart, S., “Large angle Manoeuvre of An Underactuated Small Satellites Using Two Wheels” 4th IAA Symposium on Small satellites for Earth Observation, April 7-11, 2003, Berlin, Germany.
- [25] Gao, Z., Antsaklis, P.J., “On the Stability of Pseudo-Inverse Method for Reconfigurable Control Systems” IEEE, 1989.
- [26] Jayaram, S., Johnson, R.W., Prasad, G., “Near Real-Time Autonomous Health Monitoring of Actuators: Fault Detection and Reconfiguration” Florida Conference on the Recent Advances in Robotics, 2004.
- [27] Bogh, S.A., Blanke, M. (1997) “Fault-tolerant control-a case study of the Orsted satellite.” *Fault Diagnosis in Process Systems (Digest No: 1997/174)*, IEE Colloquium on 21 April 1997 Page(s):11/1 – 1113.

APPENDIX

SGP4 CODE

```
double SGP_XKE, n, SGP_TWOTHIRD, SGP_CK2,pi, SGP_PI, SGP_TWOPI,
SGP_PIO2, SGP_X3PIO2;
double
Inclination,Eccentricity,a1,delta1,e2,theta,theta2,a0,delta0,n0dp,a0dp,SGP_XKMPE
R,perigee,SGP_AE,s4,SGP_S,q0ms24,SGP_QOMST2T;
double
tsi,eta,BStarDragTerm,sini0,beta0,beta02,A30,SGP_XJ3,ArgumentOfPerigee,C1,C2
,C3,C4,C5,D2,D3,D4;
double T1;
double
MDF,MeanAnomaly,tSince,wDF,OmegaDF,RAAN,deltaw,deltaM,Mp,w,Omega,e,a
,IL,beta,axN,ayNL,ILL,ILT,ayN,SGP_CK4;
double U,y,z;
double Epw,deltaEpw;
double
ecosE,esinE,eL,pL,r,rdot,rfdot,cosu,sinu,u,deltar,deltau,deltaOmega,deltai,deltardot,
deltarfdot,rk,uk,Omegak,ik,rdotk,rfdotk;
double sOmegak,cOmegak,sinuk,cosuk,SGP_XMNPDA;
double vMx,vMy,vMz,vNx,vNy,vNz,vUx,vUy,vUz,vVx,vVy,vVz;
double tleN;

double w0,pi2,cmm,mm0,Period,RevolutionPerDay;

int SGP4Initialized = 0;
int simpleFlag = 0;
int i;
tSince=u0[0]/60.0;
i=1;
pi=4*atan(1);
SGP_TWOPI = 2 * pi;
SGP_PIO2 = pi / 2.0;
SGP_X3PIO2 = 3 * pi / 2.0;
SGP_PI = pi;
```

```
/******
```

```

Inclination=98.1514*(pi/180);
Eccentricity=0.0011880;
BStarDragTerm=.23767e-4;
ArgumentOfPerigee=160.6162*(pi/180);
MeanAnomaly=199.5494*(pi/180);
RAAN=269.8731*(pi/180);
tleN=0.063820820213934;
/*****
SGP_XKE=0.74366916e-1;
SGP_TWOTHIRD=0.66666667;
SGP_CK2=5.413080e-4;
SGP_XKMPER=6378.135;
SGP_AE=1.0;
SGP_S=1.01222928;
SGP_QOMST2T=1.88027916e-9;
SGP_XJ3=-0.253881e-5;
SGP_CK4=0.62098875e-6;
SGP_XMNPDA=1440.0 ;

if (SGP4Initialized == 0) {
    SGP4Initialized = 1 ;

    theta      = cos(Inclination);
    theta2     = theta * theta;
    sini0      = sin(Inclination);
    e2         = Eccentricity*Eccentricity;
    beta02     = 1 - e2;
    beta0      = sqrt(beta02);
    A30        = -SGP_XJ3 *SGP_AE * SGP_AE * SGP_AE ;
    a1         = pow(SGP_XKE / tleN, SGP_TWOTHIRD);
    delta1     = 1.5 * SGP_CK2 * (3*theta2 - 1) / pow(1 - e2, 1.5) / a1 / a1;
    a0         = a1 * ((1 - delta1/3.0 - delta1*delta1) -
(134.0*(delta1*delta1*delta1)/81.0));
    delta0     = (1.5 * SGP_CK2 * (3*theta2 - 1)) / pow(1 - e2, 1.5) / a0 /
a0;
    n0dp       = tleN / (1 + delta0);
    a0dp       = a0 / (1-delta0);
    perigee    = (a0dp * (1-Eccentricity) - SGP_AE) * SGP_XKMPER;
}

if (perigee < 220.0)
    simpleFlag = 1;

s4          = SGP_S;
q0ms24     = SGP_QOMST2T;

if (perigee<156)
{

```

```

if (perigee <= 98.0)
    s4 = 20.0;
else
    s4 = perigee - 78;

q0ms24 = pow((120-s4) * SGP_AE / SGP_XKMPER, 4);
s4 = s4/SGP_XKMPER + SGP_AE;
}

tsi = 1.0 / (a0dp - s4);
eta = a0dp * Eccentricity * tsi;
T1=q0ms24;
C2=q0ms24 * pow(tsi, 4) * n0dp/pow((1-eta*eta), 3.5) *
(
    a0dp*(1+1.5*eta*eta+4.0*Eccentricity*eta+Eccentricity*eta*eta*eta) +
    1.5*SGP_CK2*tsi*(-0.5+1.5*theta2)*(8+24.0*eta*eta+3.0*pow(eta,4.0))/(1-
eta*eta)
);
C1 = BStarDragTerm * C2;
C3 = q0ms24*pow(tsi,5.0)*A30*n0dp*SGP_AE*sini0/SGP_CK2/Eccentricity;
C4 = 2.0*n0dp*q0ms24*pow(tsi,4.0)*a0dp*beta02/pow(1-eta*eta,3.5) *
(
    (2.0*eta*(1+Eccentricity*eta)+.5*Eccentricity+.5*eta*eta*eta) -
2*SGP_CK2*tsi/a0dp/(1-eta*eta)*
(
    3.0*(1-3.0*theta2)*(1+1.5*eta*eta-2.0*Eccentricity*eta-
.5*Eccentricity*eta*eta*eta)+
    0.75*(1-theta2)*(2.0*eta*eta-Eccentricity*eta-
Eccentricity*eta*eta*eta)*cos(2.0*ArgumentOfPerigee)
)
);
C5 = 2.0*q0ms24*pow(tsi,4.0)*a0dp*beta02/pow(1-
eta*eta,3.5)*(1+2.75*eta*(eta+Eccentricity)+Eccentricity*eta*eta*eta);

D2 = 4.0*a0dp*tsi*C1*C1;
D3 = 4.0/3.0*a0dp*tsi*tsi*(17.0*a0dp+s4)*C1*C1*C1;
D4 = SGP_TWOTHIRD*a0dp*tsi*tsi*tsi*(221.0*a0dp+31.0*s4)*pow(C1,4.0);

MDF = MeanAnomaly +
(
    1+
    1.5*SGP_CK2*(-1+3*theta2)/(a0dp*a0dp*beta02*beta0)+
    0.1875*SGP_CK2*SGP_CK2*(7-
78.0*theta2+137.0*theta2*theta2)/(pow(a0dp,4.0)*pow(beta0,7.0))
) * n0dp * tSince;

wDF      = ArgumentOfPerigee+

```

```

(
-1.5*SGP_CK2*(1-5.0*theta2)/pow(a0dp*beta02, 2.0)+
0.1875*SGP_CK2*SGP_CK2*(13-
114.0*theta2+395.0*theta2*theta2)/pow(a0dp*beta02, 4.0)+
1.25*SGP_CK4*(3-36.0*theta2+49.0*theta2*theta2)/pow(a0dp*beta02, 4.0)
)*n0dp*tSince;

OmegaDF = RAAN+
(
-3.0*SGP_CK2*theta/pow(a0dp*beta02, 2.0)+
1.5*SGP_CK2*SGP_CK2*(4.0*theta-19.0*theta*theta2)/pow(a0dp*beta02,
4.0)+
2.5*SGP_CK4*theta*(3-7.0*theta2)/pow(a0dp*beta02, 4.0)
)*n0dp*tSince;

deltaw = BStarDragTerm*C3*cos(ArgumentOfPerigee)*tSince;
deltaM = -
SGP_TWOTHIRD*q0ms24*BStarDragTerm*pow(tsi,4.0)*SGP_AE/(Eccentricity*e
ta)*
(
pow((1+eta*cos(MDF)),3.0)-
pow((1+eta*cos(MeanAnomaly)),3.0)
);

Mp = MDF + deltax + deltaM;
w = wDF - deltax - deltaM;
Omega = OmegaDF -
10.5*n0dp*SGP_CK2*theta*C1*tSince*tSince/(a0dp*a0dp*beta02);
e = Eccentricity - BStarDragTerm*C4*tSince -
BStarDragTerm*C5*(sin(Mp)-sin(MeanAnomaly));
a = a0dp*
pow(
1-
C1*tSince-
D2*tSince*tSince-
D3*pow(tSince,3.0)-
D4*pow(tSince,4.0)
,2.0);
IL = Mp+w+Omega+n0dp*
(
1.5*C1*tSince*tSince+
(D2+2*C1*C1)*pow(tSince,3.0)+
0.25*(3.0*D3+12.0*C1*D2+10.0*pow(C1,3.0))*pow(tSince,4.0)+
0.2*(3.0*D4+12.0*C1*D3+6.0*D2*D2+30.0*C1*C1*D2+15.0*pow(C1,4.0))*pow(
tSince,5.0)
);

```

```

beta      = sqrt(1-e*e);
n         = SGP_XKE / pow(a, 1.5);

axN       = e*cos(w);
ayNL      = A30*sin(Inclination)/(4.0*SGP_CK2*a*beta*beta);
ILL       = ayNL/2.0*axN*(3+5.0*theta)/(1+theta);
ILT       = IL + ILL;
ayN       = e*sin(w)+ayNL;

U=(ILT - Omega);
while ((U<-SGP_TWOPI) || (U>SGP_TWOPI)) {
    if (U<-SGP_TWOPI)
        U += 2*SGP_TWOPI;
    else if (U>SGP_TWOPI)
        U -= 2*SGP_TWOPI;
}
if (U < 0)
    U += SGP_TWOPI;

Epw       = U;
deltaEpw  = 1;
for (i = 0; i < 10;i++)
{
    deltaEpw = (U-ayN*cos(Epw)+axN*sin(Epw)-Epw)/(-ayN*sin(Epw)-
axN*cos(Epw)+1);
    if (abs(deltaEpw) < 1e-6)
        break;
    Epw      += deltaEpw;
}

ecosE     = axN*cos(Epw) + ayN*sin(Epw);
esinE     = axN*sin(Epw) - ayN*cos(Epw);

eL        = sqrt(axN*axN+ayN*ayN);
pL        = a*(1-eL*eL);
r         = a*(1-ecosE);
rdot      = SGP_XKE*sqrt(a)*esinE/r;
rfdot     = SGP_XKE*sqrt(pL)/r;
cosu      = a/r*(cos(Epw) - axN + ayN*esinE/(1+sqrt(1-eL*eL)));
sinu      = a/r*(sin(Epw) - ayN - axN*esinE/(1+sqrt(1-eL*eL)));

/*****
**/
//u       = atan(sinu/cosu);
{

```

```

if(cosu == 0) {
    if(sinu > 0)
        u = (SGP_PIO2);
    else
        u = (SGP_X3PIO2);
}
else
{
    if(cosu > 0) {
        if(sinu > 0)
            u = atan(sinu/cosu);
        else
            u = SGP_TWOPI + atan(sinu/cosu);
    }
    else
        u = SGP_PI + atan(sinu/cosu);
}
} /*Function actan*/

/*****
***/

deltar    = SGP_CK2/(2.0*pL)*(1-theta2)*cos(2.0*u);
deltau    = -SGP_CK2/(4.0*pL*pL)*(7.0*theta2-1)*sin(2.0*u);
deltaOmega = 1.5*SGP_CK2*theta*sin(2.0*u)/pL/pL;
deltai    = 1.5*SGP_CK2*theta/pL/pL*sin(Inclination)*cos(2.0*u);
deltardot = -SGP_CK2*n/pL*(1-theta2)*sin(2.0*u);
deltarfdot = SGP_CK2*n/pL*((1-theta2)*cos(2.0*u)-1.5*(1-3.0*theta2));

rk        = r*(1-1.5*SGP_CK2*sqrt(1-eL*eL)*(3.0*theta2-1)/pL/pL) + deltar;
uk        = u + deltau;
Omegak    = Omega + deltaOmega;
ik        = Inclination + deltai;
rdotk     = rdot + deltardot;
rfdotk    = rfdot + deltarfdot;

sOmegak   = sin(Omegak);
cOmegak   = cos(Omegak);
sinuk     = sin(uk);
cosuk     = cos(uk);

vMx       = -sOmegak*cos(ik);
vMy       = cOmegak*cos(ik);
vMz       = sin(ik);

vNx       = cOmegak;

```

```

vNy      = sOmegak;
vNz      = 0.0;

vUx      = vMx*sinuk + vNx*cosuk;
vUy      = vMy*sinuk + vNy*cosuk;
vUz      = vMz*sinuk + vNz*cosuk;

vVx      = vMx*cosuk - vNx*sinuk;
vVy      = vMy*cosuk - vNy*sinuk;
vVz      = vMz*cosuk - vNz*sinuk;

```

```

/* CALCULATION OF W0 */

```

```

RevolutionPerDay=14.62665459;
pi2=2.0*pi;
cmm=pi2/1440.0;
mm0=RevolutionPerDay*cmm;
Period=60.0*pi2;
w0=(pi2/Period);

```

```

y0[0]= rk * vUx * SGP_XKMPER * 1000.0;
y1[0]= rk * vUy * SGP_XKMPER * 1000.0;
y2[0]= rk * vUz * SGP_XKMPER * 1000.0;

```

```

y3[0]= (rdotk * vUx +
rfdotk*vVx)*SGP_XKMPER*SGP_XMNPDA/(86.4)/SGP_AE;
y4[0]= (rdotk * vUy +
rfdotk*vVy)*SGP_XKMPER*SGP_XMNPDA/(86.4)/SGP_AE;
y5[0]=(rdotk * vUz +
rfdotk*vVz)*SGP_XKMPER*SGP_XMNPDA/(86.4)/SGP_AE;

```

```

y6[0]=sqrt((y0[0]*y0[0])+(y1[0]*y1[0])+(y2[0]*y2[0]));

```

```

y7[0]=w0;

```