A Multi-mode Attitude Determination and Control System for Small Satellites

by W.H. Steyn

Dissertation presented for the Degree of Doctor of Philosophy
(Engineering)
at the University of Stellenbosch

December 1995

Promoter: Prof. A. Schoonwinkel

Declaration:

I the undersigned hereby declare that the work contained in this dissertation is my own original work and has not previously in its entirety or in part been submitted at any university for a degree.

Signature Date

ABSTRACT

New advanced control techniques for attitude determination and control of small (micro) satellites are presented. The attitude sensors and actuators on small satellites are limited in accuracy and performance due to physical limitations, e.g. volume, mass and power. To enhance the application of sophisticated payloads such as high resolution imagers within these confinements, a multi-mode control approach is proposed, whereby various optimized controller functions are utilized during the orbital life of the satellite.

To keep the satellite's imager and antennas earth pointing with the minimum amount of control effort, a passive gravity gradient boom, active magnetic torquers and a magnetometer are used. A "cross-product" detumbling controller and a robust Kalman filter angular rate estimator are presented for the preboom deployment phase. A fuzzy controller and magnetometer full state extended Kalman filter are presented for libration damping and Z-spin rate control during inactive imager periods.

During imaging, when high performance is required, additional fine resolution earth horizon, sun and star sensors plus 3-axis reaction wheels are employed. Full state attitude, rate and disturbance estimation is obtained from a horizon/sun extended Kalman filter. A quaternion feedback reaction wheel controller is presented to point or track a reference attitude during imaging. A near-minimum time, eigenaxis rotational reaction wheel controller for large angular maneuvers.

Optimal linear quadratic and minimum energy algorithms to do momentum dumping using magnetic torquers, are presented. A new recursive magnetometer calibration method is designed to enhance the magnetic in-flight measurements. Finally, a software structure is proposed for the future onboard implementation of the multi-mode attitude control system.

SAMEVATTING

Nuwe gevorderde beheertegnieke vir die oriëntasiebepaling en -beheer van klein (mikro-) satelliete word behandel. Die oriëntasiesensors en -aktueerders op klein satelliete het 'n beperkte akkuraatheid en werkverrigting as gevolg van fisiese volume, massa en kragleweringbeperkings. Om gesofistikeerde loonvragte soos hoë resolusie kameras binne hierdie tekortkominge te kan hanteer, word 'n multimode beheerbenadering voorgestel. Hiermee kan 'n verskeidenheid van optimale beheerfunksies gedurende die wentelleeftyd van die satelliet gebruik word.

Om die satellietkamera en -antennas aardwysend te rig met 'n minimale beheerpoging, word 'n passiewe graviteitsgradiëntstang, aktiewe magneetspoele en 'n magnetometer gebruik. 'n "Kruisproduk" onttuimellings beheerder en 'n robuuste hoektempo Kalmanfilter afskatter is ontwikkel vir die periode voordat die graviteitsgradiëntstang ontplooi word. 'n Wasige beheerder en 'n volledige toestand, uitgebreide Kalmanfilter afskatter is ontwikkel om librasie-demping en Z-rotasietempo beheer te doen gedurende tydperke wanneer die kamera onaktief is.

Gedurende kamera-opnames word hoë werkverrigting verlang. Fyn resolusie aardhorison, son en stersensors met 3-as reaksiewiele kan dan gebruik word. 'n Volledige oriëntasie, hoek-tempo en steurdraaimoment Kalmanfilter afskatter wat inligting van bogenoemde sensors gebruik, is ontwikkel. 'n "Quaternion" reaksiewiel terugvoerbeheerder waarmee die satelliet na verwysings orientasiehoeke gerig kan word of waarmee oriëntasiehoektempos gevolg kan word, word behandel. 'n Naby minimumtyd, "eigen"-as reaksiewielbeheerder vir groothoek rotasies is ontwikkel.

Optimale algoritmes om momentumontlading van reaksiewiele met lineêre kwadratiese en minimumenergie metodes te doen, word afgelei en aangebied. 'n Nuwe rekursiewe kalibrasie-tegniek waarmee 'n magnetometer outomaties gedurende vlug ingestel kan word, is ontwikkel. Ten slotte, word 'n programstruktuur voorgestel vir aanboord implementering van die nuwe multimode beheerstelsel.

Table of Contents

1. Introduction

1.1 Overview 1-1
1.2 Background 1-2
1.3 Attitude Control Subsystems 1-4
1.3.1 ACTUATORS 1-4
1.3.2 SENSORS 1-5
1.3.3 ATTITUDE CONTROL PROCESSORS 1-6
1.3.4 CONTROL SYSTEM SOFTWARE 1-7
1.4 Attitude Definitions and Conventions 1-9
1.5 Equations of Motion 1-12
1.5.1 DYNAMIC EQUATIONS 1-12
1.5.2 KINEMATIC EQUATIONS 1-13
1.5.3 SMALL ANGULAR MOTIONS 1-14
1.5.4 ROLL BIAS ANGLE 1-15
1.6 External Disturbance Torques 1-17
1.6.1 GRAVITY GRADIENT 1-17
1.6.2 AERODYNAMIC 1-18
1.6.3 SOLAR RADIATION 1-23
1.7 References 1-24

2. Magnetic Torquer Control

2.1 Introduction 2-1
2.1.1 MAGNETORQUING 2-1
2.1.2 GEOMAGNETIC FIELD 2-2
2.1.3 LITERATURE SURVEY 2-3
2.2 Boom Deployment 2-6
2.2.1 THE ALGORITHM 2-6
2.2.2 PULSE WIDTH MODULATION 2-7
2.2.3 SIMULATION RESULTS 2-10
2.3 Libration Damping 2-13
2.3.1 PRELIMINARIES 2-13
2.3.2 LIBRATION MODEL 2-14
2.3.3 FUZZY CONTROLLER 2-15
2.3.4 FUZZY CONTROLLER ANALYSIS 2-20
2.3.5 SIMULATION RESULTS 2-24
2.3.5.1 Disturbance damping performance 2-25

2.3.5.2 Libration damping/Spin rate control performance 2-26
2.3.5.3 Interpretation of results 2-28
2.4 Conclusions 2-31
2.5 References 2-32

3. Reaction Wheel Control

3.1 Introduction 3-1
3.2 Pointing/Tracking Control 3-2
3.2.1 CURRENT METHODS 3-2
3.2.2 SUNSAT IMPLEMENTATION 3-6
3.3 Large Angular Slew Maneuvers 3-10
3.3.1 INTRODUCTION 3-10
3.3.2 EIGENAXIS MANEUVER 3-12
3.3.3 EIGENAXIS CONVERGENCE 3-17
3.3.4 SIMULATION RESULTS 3-19
3.4 Conclusions 3-24
3.5 References 3-25

4. Momentum Dumping

4.1 Introduction 4-1
4.2 Preliminaries 4-2
4.3 Optimal Controllers 4-3
4.3.1 LQR CONTROLLER 4-3
4.3.2 MINIMUM-ENERGY CONTROLLER 4-4
4.4 Simulation Results 4-7
4.5 Conclusions 4-11
4.6 References 4-11

5. Attitude Determination

5.1 Introduction 5-1
5.2 Robust Angular Rate Estimation 5-3
5.2.1 MOTIVATION 5-3
5.2.2 KALMAN FILTER RATE ESTIMATOR 5-4
5.2.3 SIMULATION RESULTS 5-6
5.3 Full Satellite State Determination 5-9
5.3.1 BACKGROUND 5-9

5.3.2 NEW EXTENDED KALMAN FILTER 5-11
5.3.3 MODIFIED PSIAKI EXTENDED KALMAN FILTER 5-15
5.3.4 VECTOR COMPUTATIONS FOR EKF 5-18
5.3.4.1 Magnetometer EKF 5-19
5.3.4.2 Horizon/Sun sensor EKF 5-20
5.3.5 EKF PROPAGATION OF THE SATELLITE STATE 5-23
5.3.6 EKF SIMULATION RESULTS 5-29
5.4 Conclusions 5-39
5.5 References 5-40

6. Magnetometer Calibration

6.1 Introduction 6-1
6.2 LMS Self-Calibration Algorithm 6-3
6.3 RMS Self-Calibration Algorithm 6-4
6.4 Simulation Results 6-6
6.5 Conclusions 6-12
6.6 References 6-12

7. Conclusion

7.1 Summary of Contributions 7-1
7.1.1 MAGNETIC TORQUER CONTROL 7-1
7.1.2 REACTION WHEEL CONTROL 7-2
7.1.3 MOMENTUM DUMPING 7-2
7.1.4 ATTITUDE DETERMINATION 7-3
7.1.5 MAGNETOMETER CALIBRATION 7-4
7.2 Onboard Software Structure 7-4

Appendix A. IGRF Magnetic Field Modelling

Appendix B. Magnetorquer Controllers
B.1 Pre-boom Deployment Controller B-1
B.2 Libration Damping Fuzzy Controller B-3
B.3 Libration Damping Cross-Product Controller B-9

Appendix C. Reaction Wheel Control

Appendix D. Momentum Dumping

Appendix E. Attitude Determination

E.1 Robust Angular Rate Estimator E-1

E.2 Full Satellite State Estimators E-3

Appendix F. Magnetometer Calibration

F.1 LMS Algorithm F-1

F.2 RLS Algorithm F-2

Appendix G. ADCS Software Specification

G.1 Tasks triggered by asynchronous events G-1

G.2 Tasks driven by timers G-2

G.3 Data (message) driven tasks G-3

List of Acronyms

ADCS	Attitude Determination and Control System
ACP	Attitude Control Processor
CCD	Charge Couple Device
DCM	Direction Cosine Matrix
EKF	Extended Kalman Filter
GG	Gravity Gradient
ICP	Interface Control Processor
IGRF	International Geomagnetic Reference Field
IRAS	Infrared Astronomical Satellite
KITSAT	Korean Institute of Technology Satellite
LEO	Low Earth Orbit
LMS	Least Mean Square
LQR	Linear Quadratic Regulator
MEDL	Minimum Energy Desaturation Law
MISO	Multi-Input Single Output
MOI	Moment of Inertia
MO	Multi-Output
NASA	National Aeronautics and Space Administration
OBC	Onboard Computer
PD	Proportional and Derivative
PWM	Pulse Width Modulation
QUEST	Quaternion Estimator
RLS	Recursive Least Square
RMS	Root Mean Square
rpo	revolutions per orbit
SO	Single Output
SUNSAT	Stellenbosch University Satellite
UART	Universal Asynchronous Receiver and Transmitter
UoSAT	University of Surrey Satellite
XPROD	Cross Product
ZOH	Zero Order Hold circuit

List of Symbols

Coordinate frames:

X, Y, Z	Satellite body coordinate axes
$\vec{\mathbf{x}}, \vec{\mathbf{y}}, \vec{\mathbf{z}}$	Satellite body unit vectors
X_o, Y_o, Z_o	Circular orbit coordinate axes
$\vec{\mathbf{x}}_o, \vec{\mathbf{y}}_o, \vec{\mathbf{z}}_o$	Orbit referenced unit vectors
X_i, Y_i, Z_i	Inertially fixed coordinate axes
$\vec{\mathbf{x}}_i, \vec{\mathbf{y}}_i, \vec{\mathbf{z}}_i$	Inertial referenced unit vectors

Orbit and space environment:

ν	True anomaly
M	Mean anomaly
e	Orbit eccentricity
i	Orbit inclination
α	Orbit angle as measured from the ascending node
ω_o	Mean orbital angular rate
$\tilde{\omega}_o$	True eccentric orbital angular rate
$GM_{\oplus}$	Earth gravitational constant
a	Earth radius at the equator
f	Earth flattening factor
λ	Earth latitude
R_e	Earth radius at latitude λ
R_s	Geocentric orbital radius
$\vec{\mathbf{R}}$	Unit geocentric position vector
$h_{satellite}$	Height of satellite's orbit
$h_{perigee}$	Height of orbit's perigee
T	Orbital period
ρ_a	Atmospheric density
V	Magnitude of spacecraft's velocity vector
$\vec{\mathbf{V}}$	Unit velocity vector
A_p	Total projected area of spacecraft
$\mathbf{c}_p$	Vector between centre of mass and centre of pressure
d_o	Solar radiation constant
c	Velocity of light

Operators:

$\otimes$	Quaternion multiplication
∇	Vector gradient operator
$N\{\ \}$	Noise system properties
$E\{\ \}$	Expected value

$\langle \mathbf{F} \vert \mathbf{G} \rangle$	Sampled covariance matrix of two vectors
$\langle \mathbf{F} \rangle$	Sampled mean of a vector

Satellite's attitude plus rate parameters:

ϕ	Roll angle
θ	Pitch angle
ψ	Yaw angle
ω_{ϕ}	Roll libration rate
ω_{θ}	Pitch libration rate
ω_{λ}	Body nutation rate
$\mathbf{A}$	Attitude transformation matrix (DCM)
A_{ij}	Component of matrix A at row i and column j
$\mathbf{q}$	Attitude quaternion vector (orbit referenced)
$\delta\mathbf{q}$	Quaternion perturbation vector
q_1, q_2, q_3, q_4	Quaternion components (orbit referenced)
q_{ie}	Error quaternion component
$\mathbf{q}_{vec}$	Vector part of error quaternion
$\mathbf{q}_c$	Commanded quaternion vector (orbit referenced)
q_{ic}	Commanded quaternion component (orbit referenced)
Φ	Euler rotation angle
e_x, e_y, e_z	Euler axis components in orbit referenced coordinates
E_1, E_2, E_3	Euler axis components in orbit referenced coordinates
$\boldsymbol{\omega}_B^I$	Inertially reference body angular rate vector
$\delta\boldsymbol{\omega}$	Angular rate perturbation vector
$\omega_x, \omega_y, \omega_z$	Inertialy referenced body rate components
$\boldsymbol{\omega}_B^O$	Orbit reference body angular rate vector
$\omega_{ox}, \omega_{oy}, \omega_{oz}$	Orbit referenced body rate components
$\boldsymbol{\omega}_{ref}$	Commanded angular rate reference vector
$\omega_{oi\text{-}ref}$	Commanded angular rate reference component (orbit referenced)
$\mathbf{\Omega}$	Angular rate matrix
$\mathbf{\Omega}_c$	Commanded angular rate matrix
$\mathbf{L}$	Total satellite angular momentum vector

Satellite MOI values:

$\mathbf{1}$	Identity matrix
$\mathbf{I}$	Moment of inertia tensor
I_{xx}, I_{yy}, I_{zz}	Principal satellite body axis moment of inertias
I_T	Transverse moment of inertia
I_{ij}	Product of inertia of satellite
$\Delta\mathbf{I}$	Moment of inertia tensor uncertainty
$\Delta I_X, \Delta I_Y, \Delta I_Z$	Principal satellite body axis moment of inertia uncertainties

Satellite torques:

$\mathbf{N}_{GG}$	Gravity gradient torque vector
$\mathbf{N}_{M}$	Magnetic torque vector
N_{mx}, N_{my}, N_{mz}	Magnetic torque vector components
n_{mx}, n_{my}, n_{mz}	Normalized magnetic torque vector components
$\mathbf{N}_{D}$	External disturbance torque vector
N_{dx}, N_{dy}, N_{dz}	Disturbance torque vector components in body coordinates
n_{doy}	Disturbance torque around the orbital Y_o-axis
$\mathbf{N}_{EXT}$	Total (plus GG) external disturbance torque vector
$\mathbf{N}_{AERO}$	Aerodynamic torque vector
N_{SOLAR}	Magnitude of torque caused by solar radiation pressure
$\mathbf{N}_{slew}$	Slew torque vector
$\boldsymbol{\eta}_{slew}$	Normalized slew torque vector
$\mathbf{N}_{add}$	Additional non-linear torques
$\mathbf{N}_{comp}$	Compensation torque vector
$\mathbf{N}_{control}$	Actuator control torque vector

Reaction wheel parameters:

$\mathbf{h}$	Reaction wheel angular momentum vector
h_x, h_y, h_z	Reaction wheel angular momentum components
$h_{wheel\text{-}max}$	Maximum reaction wheel angular momentum
I_w	Reaction wheel moment of inertia
$\boldsymbol{\omega}_w$	Reaction wheel angular rate vector
ω_{wx}, ω_{wy}, ω_{wz}	Reaction wheel angular rate components
$\mathbf{N}_{wheel}$	Reaction wheel torque vector
$N_{wheel\text{-}i}$	Reaction wheel i torque
$N_{sat\text{-}i}$	Reaction wheel i saturation torque
N_{wx}, N_{wy}, N_{wz}	Reaction wheel torque vector components

Magnetorquer/Magnetometer and magnetic field parameters:

$\mathbf{M}$	Magnetic dipole vector of magnetorquers
M_x, M_y, M_z	Magnetic dipole components of magnetorquers
$t_{on\text{-}i}$	On-time for magnetorquer i
$M_{coil\text{-}max}$	Maximum magnetic dipole moment of magnetorquer
$\mathbf{M}_e$	Vector geomagnetic strength
$\mathbf{B}_o$	Geomagnetic field strength orbit referenced
B_{ox}, B_{oy}, B_{oz}	Geomagnetic field components orbit referenced
$\mathbf{B}$	Geomagnetic field strength body referenced
B_x, B_y, B_z	Geomagnetic field components body referenced
B_{mi}	Magnetometer measurement for axis-i
$\boldsymbol{\Psi}$	Geomagnetic field strength matrix
$\mathbf{S}$	Scale factor/misalignment matrix for magnetometer
$\mathbf{G}$	Magnetometer gain calibration matrix

g_{ij}	Scale factor gain component in row i and column j
$\mathbf{b}$	Magnetometer bias vector
b_i	Bias component i
μ	Step size for LMS algorithm during magnetometer calibration
e	Parameter calibration error
$\boldsymbol{\varphi}$	Regression vector for RLS calibration
$\boldsymbol{\theta}$	Parameter vector for RLS calibration
λ	Forgetting factor in RLS calibration algorithm

Control system plus estimator parameters:

T_s	Discrete sampling period
h	Numerical integration step size
$\mathbf{e}$	Error vector (attitude or rate)
$\mathbf{K}_i$	Control gain matrix i
$\mathbf{C}$	Compensation feedback gain matrix
$\mathbf{K}$	Error quaternion gain matrix
k_i	Angular control gain i
$\mathbf{D}$	Error angular rate gain matrix
d_i	Rate control gain i
x_i	Fuzzy input variable i
χ_i	Normalized fuzzy input i
$m_I\,(x_i)$	Fuzzy membership function I
R^i	Fuzzy rule i
$\mu\,(\mathrm{R}^i)$	Fuzzy rule i consequent (truth value)
$y\,(\mathrm{R}^i)$	Fuzzy rule i output
y_{out}	Fuzzy rule base output
ω_n	Natural undamped oscillation frequency of a second order system
ζ	Damping factor of a second order system
J	Cost function
t_0, t_f	Initial and final time
ε	Arbitrary small constant
$\mathbf{Q}$	State weighting matrix or System noise covariance matrix
$\mathbf{R}$	Control weighting matrix or Measurement noise covariance matrix
$\mathbf{K}(t)$	Ricatti time solution matrix
$\mathbf{K}_{\infty}$	Static Ricatti solution matrix
$\mathbf{F}_{LQR}$	LQR feedback gain matrix
$\mathbf{p}$	Costate vector
$\mathbf{H}$	System Hamiltonian
$\Xi(t)$	Costate solution matrix
$\mathbf{P}_i$	State i influence function matrix
$\mathbf{v}$	Measurement vector
$\mathbf{v}_{meas}$	Sensor measurement vector in body coordinates
$\mathbf{v}_{body}$	Modelled measurement vector in body coordinates

$\mathbf{v}_{orb}$	Modelled measurement vector in orbit coordinates
$\delta\mathbf{v}$	Measurement vector difference
$\mathbf{s}$	System noise vector
$\mathbf{m}$	Measurement noise vector
$\mathbf{f}(\mathbf{x}(t),t)$	Non-linear continuous system model
$\mathbf{f}_k(\mathbf{x}_k,k)$	Non-linear discrete system model
$\mathbf{F}(\mathbf{x}(t),t)$	Linearized perturbation state matrix
$\mathbf{\Phi}$	Discrete system matrix
$\mathbf{\Gamma}$	Discrete control input matrix
$\mathbf{x}_k$	Discrete state vector at sample k
$\delta\mathbf{x}_k$	Discrete state perturbation vector at sample k
$\hat{\mathbf{x}}_k$	Estimated state vector at sample k
$\mathbf{P}_k$	Discrete state covariance matrix at sample k
$\mathbf{K}_k$	Discrete innovation gain matrix at sample k
$\mathbf{h}(\mathbf{x}(t),t)$	Non-linear continuous output model
$\mathbf{H}_k$	Discrete output measurement matrix at sample k
$\mathbf{e}_k$	Discrete innovation error vector at sample k

Sensor angles:

θ_{hi}	i-Horizon sensor measurement angle
θ_{sun}	Sun sensor measurement angle
Azim	Azimuth angle
Elev	Elevation angle
Ψ	Azimuth angle of horizon vector in the X_oY_o-plane, measured from east

List of Figures

Fig. 1.1 Typical micro satellite 1-3
Fig. 1.2 Blockdiagram of SUNSAT attitude control subsystem 1-8
Fig. 1.3 Coordinate systems 1-9
Fig. 1.4 Definition of 1-2-3 Euler angle rotations 1-10
Fig. 1.5 Typical GG induced libration for an elliptical orbit 1-18
Fig. 1.6 Simplified geometrical structure of SUNSAT 1-21
Fig. 1.7 GG and average aerodynamic influence on the pitch dynamics 1-21
Fig. 1.8 Average aerodynamic disturbance torque on SUNSAT structure 1-22
Fig. 1.9 GG and maximum aerodynamic influence on the pitch dynamics 1-22
Fig. 1.10 Maximum aerodynamic disturbance torque on SUNSAT structure 1-23

Fig. 2.1 PWM of magnetic moment control signal 2-8
Fig. 2.2 Angular rates during the boom deployment sequence 2-11
Fig. 2.3 Pitch angle and transverse MOI during the boom deployment sequence 2-12
Fig. 2.4 Magnetic coil switching polarities pre-boom deployment 2-12
Fig. 2.5 Block diagram of the full fuzzy controller 2-16
Fig. 2.6 Fuzzy variable membership functions 2-17
Fig. 2.7 Positive-positive control surface for the non-overlapping controller 2-23
Fig. 2.8 Positive-positive control surface for the overlapped controller 2-24
Fig. 2.9 Disturbance damping using the cross-product controller 2-26
Fig. 2.10 Disturbance damping using the SO fuzzy controller 2-27
Fig. 2.11 Cross-product libration damping/Z-spin control ($e = 0.03$) 2-29
Fig. 2.12 SO fuzzy libration damping/Z-spin control ($e = 0.03$) 2-29
Fig. 2.13 Cross-product libration damping/Z-spin control ($e = 0$) 2-30
Fig. 2.14 Overlap fuzzy libration damping/Z-spin control ($e = 0$) 2-30

Fig. 3.1 Attitude control of quaternion feedback controller 1 3-8
Fig. 3.2 Attitude control of quaternion feedback controller 2 3-8
Fig. 3.3 Reaction wheel angular momentum of controller 1 and 2 3-9
Fig. 3.4 Attitude performance of SUNSAT tracking controller 3-9
Fig. 3.5 Reaction wheel angular momentum for the tracking controller 3-10
Fig. 3.6 Angular rate during eigenaxis rotation without wheel speed limiting 3-13
Fig. 3.7 Angular rate during eigenaxis rotation with wheel speed limiting 3-13
Fig. 3.8 Euler angles for a large angular slew (New and Q-regulator) 3-21
Fig. 3.9 Large slew reaction wheel angular momentum (New and Q-regulator) 3-22
Fig. 3.10 Large slew reaction wheel torque (New and Q-regulator) 3-22
Fig. 3.11 Large slew quaternion error plots (New and Q-regulator) 3-23
Fig. 3.12 Euler angles during a small angular slew (New and Q-regulator) 3-23
Fig. 3.13 Euler angles during a large slew with 10% MOI mismatch 3-24

Fig. 4.1 Desaturation using an optimal LQR controller 4-8
Fig. 4.2 Desaturation using an infinite time LQR controller 4-8
Fig. 4.3 Desaturation using a 0.25 orbit MEDL controller 4-9
Fig. 4.4 Desaturation using a 0.5 orbit MEDL controller 4-9
Fig. 4.5 Desaturation using the "Cross-product law" controller 4-10
Fig. 4.6 Comparison of a quasi-static and an optimal LQR feedback gain term 4-10

Fig. 5.1 A typical performance of the new angular rate Kalman filter 5-8
Fig. 5.2 The estimated angular rate errors of the new angular rate Kalman filter 5-8
Fig. 5.3 Placement of the horizon and sum sensors on SUNSAT 5-20
Fig. 5.4 Definition of the azimuth&elevation angles within the orbit coordinates 5-22
Fig. 5.5 Various integration step size results using the new single step method 5-27
Fig. 5.6 Various integration step size results using 4-stage Runge-Kutta method 5-28
Fig. 5.7 Pointing angle estimation of New Magnetometer EKF 5-32
Fig. 5.8 Pointing angle estimation of modified Psiaki EKF 5-32
Fig. 5.9 Z-spin estimation of New Magnetometer EKF 5-33
Fig. 5.10 Z-spin estimation of modified Psiaki EKF 5-33
Fig. 5.11 Y_o-axis disturbance torque estimation of New Magnetometer EKF 5-34
Fig. 5.12 Y_o-axis disturbance torque estimation of modified Psiaki EKF 5-34
Fig. 5.13 Estimation RMS pointing error of New Magnetometer EKF 5-35
Fig. 5.14 Estimation RMS pointing accuracy of modified Psiaki EKF 5-35
Fig. 5.15 Convergence performance of New Magnetometer EKF 5-36
Fig. 5.16 Convergence performance of modified Psiaki EKF 5-36
Fig. 5.17 New Magnetometer EKF tracking performance during magnetorquing 5-37
Fig. 5.18 New Magnetometer EKF RMS pointing error during magnetorquing 5-37
Fig. 5.19 Typical pointing angle estimation of the New Horizon/Sun EKF 5-38
Fig. 5.20 Estimation RMS pointing accuracy of the New Horizon/Sun EKF 5-38
Fig. 5.21 Z-spin & disturbance torque estimation of the New Horizon/Sun EKF 5-39

Fig. 6.1 Geomagnetic field magnitude of the uncalibrated magnetometer 6-8
Fig. 6.2 Geomagnetic field magnitude for the LMS calibrated magnetometer 6-8
Fig. 6.3 Scaling factor convergence for the LMS calibrated magnetometer 6-9
Fig. 6.4 Bias value convergence for the LMS calibrated magnetometer 6-9
Fig. 6.5 Convergence errors for the non-orthogonal axes during LMS calib. 6-10
Fig. 6.6 Geomagnetic field magnitude for the RLS calibrated magnetometer 6-10
Fig. 6.7 Scaling factor convergence for the RLS calibrated magnetometer 6-11
Fig. 6.8 Bias value convergence for the RLS calibrated magnetometer 6-11

Fig. 7.1 Software structure of the Attitude Control Processor 7-5

Fig. A.1 North/East to X_o/Y_o coordinate transformation A-6

List of Tables

Table 1.1 Typical atmospheric density values ρ_a 1-19
Table 1.2 Coefficients for the aerodynamic disturbance torque models 1-20

Table 2.1 Rule table for M_x fuzzy controller 2-18
Table 2.2 Rule table for M_y fuzzy controller 2-18
Table 2.3 Rule table for M_z fuzzy controller 2-18
Table 2.4 Controller comparison for disturbance damping 2-26
Table 2.5 Controller comparison for libration damping/spin rate control 2-27
Table 2.6 Controller comparison for libration damping/spin rate control 2-28

Table 3.1 Comparison between the new algorithm and a quaternion regulator 3-20

Table 4.1 Comparison of the various desaturation controllers 4-7

Table 6.1 Calibration Errors 6-6

Table A.1 Eighth order IGRF Gaussian Coefficients for EPOCH 1990-1995 A-3

1. INTRODUCTION

1.1 Overview

This thesis documents newly contributed research results from a study undertaken to investigate the implementation of advanced attitude control and estimation techniques for nadir pointing micro satellites. These techniques are essential when stringent attitude pointing and maneuver requirements are needed on micro satellites, for example, to point payloads such as high resolution earth imagers. The various algorithms developed, are compared to current state of the art methods where possible. The comparison and testing cf all the newly proposed techniques have been done through computer simulations. A satellite, SUNSAT [Milne, 1993], to serve as a test-bed and to help with in-flight experience of the various new attitude control methods will be launched early 1997.

The scope of this thesis document covers the following topics:

- The rest of the introductory chapter will present some background to the attitude control subsystem for the SUNSAT micro satellite. The attitude description and conventions used in the this document will be defined. The dynamic and kinematic models used for the spacecraft will be given and the various external disturbance torques will be explained.
- Chapter 2 presents and compares various control algorithms to do magnetic torquer actuation of the spacecraft attitude. Initial detumbling and boom deployment methods will be shown. Libration damping and spin rate controllers, used during normal flight conditions when only coarse attitude control is needed, are developed.
- Chapter 3 will disclose a new method to do large angular slew maneuvers in near minimum time using the reaction wheels. Accurate tracking and pointing techniques using the reaction wheels during imaging, will also be presented.
- Chapter 4 introduces new optimal momentum dumping control laws to desaturate the reaction wheels
- In Chapter 5 methods are discussed to determine the attitude of the satellite from sensor vector observations For example, 1) a robust angular rate estimation filter using magnetometer measurements only, and 2) a newly designed full state extended Kalman filter for coarse attitude determination from magnetometer measurements and fine attitude determination from sun, horizon and star sensor measurements, are presented
- Chapter 6 presents and compares two in-flight magnetometer calibration methods. One of the methods is a novel application of the RLS technique for parameter identification to calibration problems

- Chapter 7 summarizes and reflects on the results presented in the thesis. A structure for a possible onboard software system is also outlined.

1.2 Background

The attitude control algorithms described in this thesis can be used on the emerging class of micro satellites in need of more sophisticated attitude controllers [Milne 1993]. Due to mass and volume constraints on micro satellites, sensors and actuators have to be kept light-weight and small. Therefore, to increase the performance of the attitude controllers on such missions, it will be best to try and meet the requirements by an increase in the software capability. To clarify this point further, the capabilities of existing hardware may be enhanced by making use of more sophisticated control algorithms. It must be stressed at this point that software algorithms alone can not cure all the problems of inadequate hardware.

Most of the newly developed control algorithms were designed for a low-earth orbiting and nadir pointing satellite. The orbit is assumed to be circular or slightly elliptical. This configuration is mostly used by earth observation satellites where the need for stable and accurate pointing of instruments is crucial. As this study was undertaken with the SUNSAT mission in mind, a short introduction to the satellite and specifically the attitude control requirements will be given. The next paragraph reviews the basic attitude control subsystems for SUNSAT.

SUNSAT is a 60 kg almost cubical (sides of approximately 45 cm) micro satellite scheduled for launch in early 1997 as a piggy-back payload on a Delta mission. The orbit configuration will be polar and slightly elliptical with a perigee at 400 km and an apogee at 840 km. The approximate orbital period will be 97 minutes, and the orbital plane will have a nodal drift of 0.77°/day (not sun-synchronous).

The main payload on SUNSAT will be a push broom imager capable of stereo imaging. The 3-colour, 3456 pixel imager will have a sub-satellite ground pixel spacing of 15 meter (51.8 km image swath width) at an altitude of 600 km. To point the imager accurately, the following specifications must be satisfied at the sub-satellite point by the attitude control subsystem:

⇒ To determine the imager boresight position to better than 1 km close to the sub-satellite point, pitch and roll attitude measurement errors must be less than 1.2 mrad, and yaw errors less than 2.4 mrad.

⇒ To ensure less than 5 km image overlap between imaging sessions, pitch and roll attitude control accuracy must be less than 3 mrad, and yaw accuracy less than 6 mrad.

⇒ To reduce the geometric distortion of images to below 1 %, the maximum allowed pitch and roll rates must be less than 0.08 mrad/s, and the yaw rate less than 0.16 mrad/s.

SUNSAT will be an earth pointing satellite (body Z-axis towards nadir) to keep the imager in a nominal direction for usage and to permit acceptable antenna gain. A gravity gradient boom and tip mass will be deployed from the top of the satellite to earth-stabilize it using the minimum amount of control energy. The satellite will be kept in a slow Z-spin during normal operation (not during imaging), for improved solar thermal distribution. The four solar panels on the X/Y facets (see Figure 1.1) will thereby receive an even solar illumination, resulting in an improved life span of the solar cells. A simple momentum transfer to a Z-axis reaction wheel will despin the satellite before imaging sessions.

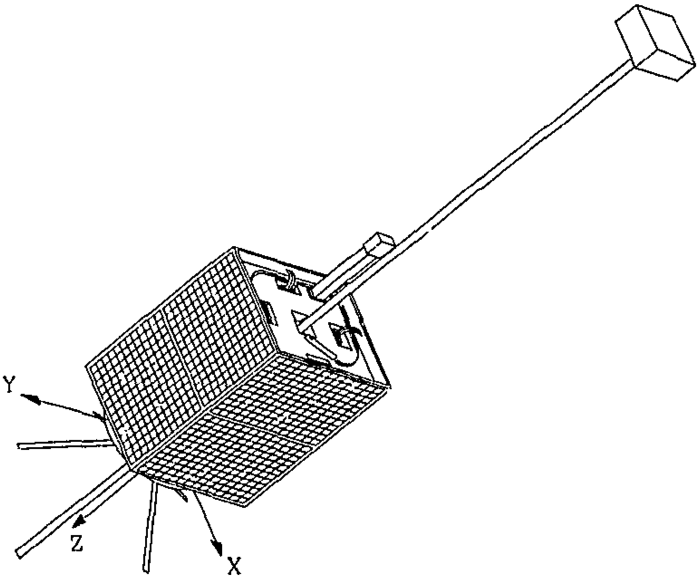

Figure 1.1 Typical micro satellite

1.3 Attitude Control Subsystems

1.3.1 ACTUATORS

3-Axis Reaction Wheels

Four small servomotor-driven reaction wheels will provide an accurate, continuous and fast attitude control capability. A wheel is used for each of the body axes and an extra wheel is added to add redundancy to the Z-control axis. The SUNSAT reaction wheels with their limited operational life in vacuum, is a concerning factor. They will therefore be used for pointing maneuvers and stabilization during imaging only. Tacho feedback is used during wheel angular rate control. Digital counters clocked by optical sensors provide wheel angular position feedback.

The maximum reaction torque per wheel is 4 × 10^{-3} Nm. The maximum wheel speed is 4800 rpm giving a wheel angular momentum of 0.25 Nms. This permits, for example, a 180 degree slew around the Z-axis within 90 seconds. The angular feedback optical sensors provides a resolution of 22.5° on the wheel diameter. A 4000:1 ratio of the body Z-axis moment of inertia (MOI) and the wheel MOI will then makes it possible to do open-loop yaw position control to an accuracy of 0.1 mrad.

3-Axis Magnetic Torquer Coils

Air core coils are wound into channels around the X/Y solar panels and around the Z facets. They ensure a high reliability (lack of moving parts), long life (no expendables) and are digitally switched on/off with dual polarity. Magnetic torquing is used as the primary active stabilization method to do libration damping, Z-spin rate control and momentum dumping of the reaction wheels. Detumbling and attitude capture can also be done initially from simple and space proven control rules for magnetic torquers [Hodgart, 1987]. Control torque is generated using a pulse width modulation method, the direction being dependent on the geomagnetic vector.

Libration damping for a circular orbit can be reduced to within 1° with control algorithms making use of full attitude information [Steyn, 1990; Pal, 1993]. Pitch and roll librations are mainly caused by gravity gradient and aerodynamic disturbance torques. The magnetic torquers have a magnetic dipole moment of 20 Am^2 per axis when powered in parallel. The maximum torque obtainable from the SUNSAT magnetic torquers will then be approximately 1 × 10^{-3} Nm at the polar region, mainly used for pitch and roll control and 5 × 10^{-4} Nm at the equatorial region, mainly for yaw control.

Gravity Gradient Boom

A 2.3 meter long boom with a tip mass of 6.0 kg will be deployed for passive attitude control. Gyroscopic torque will make it possible to create a constant roll offset angle (e.g. for imaging) by controlling the Z-axis angular momentum through Z-spin control. The tip mass will contain an accurate magnetometer for scientific measurements and a star sensor. To prevent any wires impeding boom deployment, the tip mass is preferred to be a self contained unit.

1.3.2 SENSORS

3-Axis Fluxgate Magnetometer

The magnetometer is used to measure the strength and direction of the geomagnetic field vector. This information is used to calculate the magnetic torques and to obtain full attitude data by comparing the measurements to geomagnetic field models. An advantage of using a magnetometer for attitude information, is the availability of data throughout the orbit of the satellite. A disadvantage is the inaccuracy of the data due to errors in the geomagnetic field models - these models can not account for the influence of solar activity on the field vector. With Kalman filtering of the magnetometer data, total RMS errors in attitude of less than 1.0° per axis has been reported [Psiaki,1990]. Simple control rules using short term variations in the magnetometer measurements can be used for initial attitude capture. The magnetometer has a dynamic range of ± 64 μT and a resolution of 32 nT. With a power consumption of only 100 mW it is suitable for continuous operation.

2-Axis Horizon Sensor

Two orthogonal linear CCD and lens assemblies look below the local horizontal level, to obtain orthogonal measurements of the sunlit earth horizon. A ± 15° view on each 2048 element CCD is used to obtain pitch and roll attitude angles to an accuracy of 0.5 mrad [Steyn, 1990]. This sensor is currently flown onboard the UoSAT-5 satellite. Both sensors consumes approximately 2 W when active and will only be used when the CCD's see a valid sunlit horizon, and accurate attitude information is needed. The rest of the time they will be inactive.

Fine Sun Sensor

Similar linear CCD technology is used to obtain a sun azimuth measurement within a ± 60° view with an angular resolution of 1 mrad. The sensor head consist of a slit aperture perpendicular to the CCD array. During imaging this sensor will face the sun and accurate yaw attitude information will be available. When the satellite is spinning, sensor data will be available on average for only 20% of each orbit. The active consumption of 1 Watt can therefore be reduced substantially by switching off the sensor when not needed.

Coarse Sun Sensor

Six cosine-law solar cells mounted on each spacecraft facet are used to obtain full attitude information to within ± 5 with the aid of a sun and satellite orbital model. The short circuit currents from each cell are compared to obtain the sun vector direction with respect to the satellite body. The surface temperature of each cell is also measured to make the necessary temperature sensitivity corrections to all the measurements. The sensor consumes almost no spacecraft power and is very simple and reliable. It can be used directly after launch to determine the non-stabilized satellite's attitude during the sunlit part of the orbit.

Star Sensor

A 10°×10° star image is projected onto a 376×291 pixel matrix CCD sensor, providing accurate 3-axis backup attitude information during earth imaging sessions. With a sensitivity of 4 mlux on the CCD pixels, V-6 magnitude stars can be detected. At least 3 separated stars must then be detectable within the sensor's field of view to enable an algorithm using a star catalogue [Van Bezooijen, 1989] to calculate the attitude. The roll and yaw angular resolution will be at least 0.5 mrad and the pitch resolution will depend on the star separation distance. For example, 1° in separation will give a resolution of at least 15 mrad. During earth imaging the star sensor will always be pointing towards the orbit anti-normal, therefore, only a small part of a full star catalogue has to be present onboard the satellite.

1.3.3 ATTITUDE CONTROL PROCESSORS

Due to the high processing load required by the attitude determination and control algorithms presented in the rest of this document, it was decided to dedicate a micro processor (T800) to this task. To keep the system redundant, most ADCS functions can be implemented on one of the onboard computers (OBC) on SUNSAT, whenever the dedicated ADCS processor might fail. Under normal conditions low level interfacing to the sensors and actuators will be done by an interface processor (80C31). This adds another level of redundancy, as some simpler autonomous control actions will also be possible when the interface processor is utilized on its own. For flexibility reasons all the processors are designed to be reloadable with new software updates during the satellite's mission.

Interface Control Processor (ICP)

A 80C31 based micro controller is used to interface directly to all the actuators and the sensors. It supplies the reaction wheel speed reference signals to the analogue wheel speed control system and switching pulses to the magnetic torquers. The control command updates are received every 1 second from the attitude control processor (ACP). The ICP must also control the sensor hardware, for example, select the illumination time on the horizon and star sensors, low pass filter the magnetometer, horizon and sun sensors, and read out positions of detected stars form the star sensor. All the sensor data has to be send to the ACP at 1 second intervals.

Attitude Control Processor (ACP)
A T800 transputer will be used to implement all the control system software. The ACP will take over most of the critical functions of the ICP if the 80C31 fails. If the T800 fails, a 80386EX OBC can implement most ACP functions. All these possibilities are selected using a multiplexor switch as shown in the attitude control system block diagram in Figure 1.2. The ACP communicates to the ICP via a bi-directional UART and to the OBCs via its links and link adaptors. The links to the OBC are used mainly for communication of attitude reference and mode commands. The transputer can also reduce its own clock speed to save power. This feature will be used to dynamically adjust processing speed to changing work load conditions.

1.3.4 CONTROL SYSTEM SOFTWARE

The control system software is implemented as tasks on the ACP which are scheduled by a hard real-time kernel. This will ensure that all asynchronous events (e.g. communication with ICP and OBC), timer driven events (e.g. discrete sampling periods) and message passing between tasks is done in an orderly manner. The tasks consist mainly of environmental models, sensor calibration, measured attitude computations, attitude estimators and controller algorithms. For example, 2 extended Kalman filters (EKF) are used to extract full attitude data. The first EKF uses data from the continuous magnetometer measurements for coarse attitude determination. The second EKF uses data from all the accurate but intermittent attitude sensor measurements such as the horizon, fine sun and star sensors. The output from the first filter will be limited by the accuracy of the geomagnetic model. The output from the second filter will be much more accurate. A 1 mrad attitude error is expected during the sunlit parts of the orbit when accurate sensor data becomes available from the horizon, sun and star sensors. Fortunately it is also during this period of an orbit when imaging occurs and accurate 3-axis stabilization and pointing requirements are needed.

A few controller modes will be available using either the magnetic torquers and/or the reaction wheels. After the initial detumbling phase, the boom will be deployed and gravity gradient lock achieved. The following modes will be used during normal satellite operation:

1. A normal mode using only the magnetic torquers for libration damping and Z-spin control. Momentum dumping can also be done whenever this mode is entered with the reaction wheels running.
2. A set-up mode to orientate the imager to the correct 3-axis stabilized attitude before imaging commences. The orientation maneuver will be done in a time-optimal manner using the reaction wheels.

3. A 3-axis stabilization mode, using the reaction wheels, to keep the push broom imager steady while scanning. Slow angular tracking will also be possible in this mode, for example to compensate for the earth's rotation.
4. A reset mode to return to the satellite's condition before the preparations for imaging had commenced (set-up mode). This can be achieved by doing the inverse orientation maneuvers of the set-up mode. Any reaction wheel momentum built-up due to the influence of external torques during the imaging session, will be dumped during the normal mode.

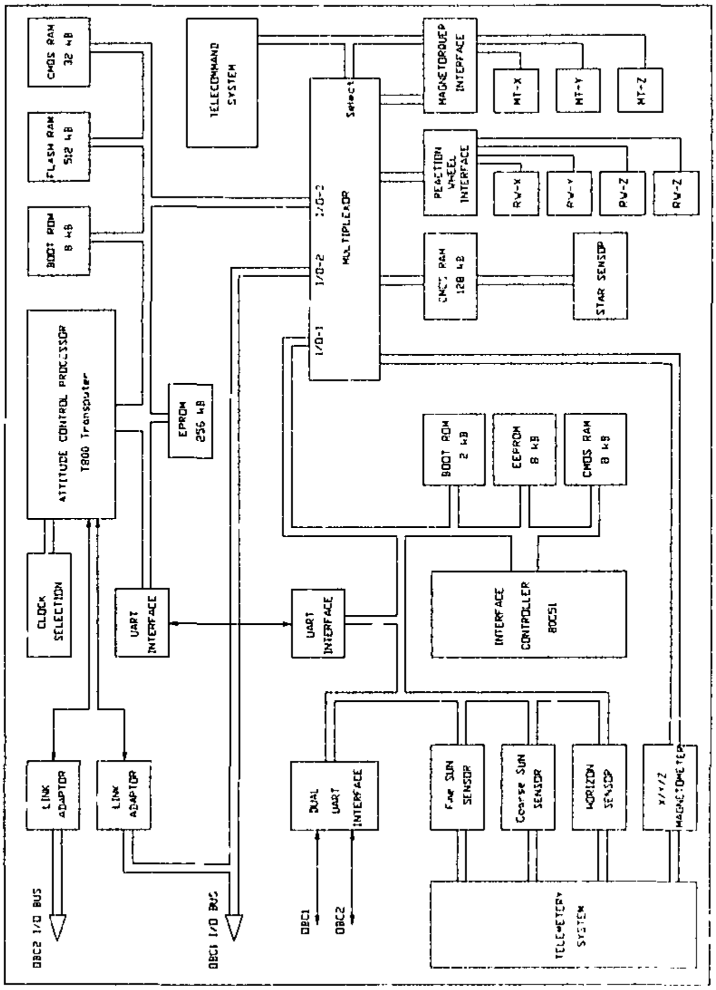

Figure 1.2 Blockdiagram of SUNSAT attitude control subsystem

1.4 Attitude Definitions and Conventions

The conventions used throughout this thesis corresponds fully to the NASA standard as also defined in Wertz [1986]. The relevant definitions and conventions will be repeated in this paragraph to clarify the work in the rest of this document.

Three coordinate systems are used, the first being the spacecraft's body coordinates, the second the orbital coordinates and the third the inertial coordinates. The body X, Y and Z axes are defined as shown in Figure 1.1. The body Z-axis are parallel to the boom's deployment direction but taken as positive opposite to this direction. The X and Y axes are perpendicular to two of the side solar panels. The orbital Z_o-axis are in the nadir direction, the Y_o-axis in the orbit anti-normal direction (see Figure 1.3) and the X_o-axis complete the orthogonal set. The X_o-axis will be in the orbit velocity direction for a circular orbit.

The inertial Y_i-axis is defined as the orbit anti-normal direction, similar to Y_o. This direction is not exactly inertial due to a slow presession of the orbital plane (0.77° per day for the SUNSAT orbit). This rotation is slow enough to have a negligible effect on the satellites's dynamics. The inertial Z_i-axis is in the same direction as the earth's geometric north pole and the X_i-axis complete the orthogonal set. The orbital coordinates therefore rotates once per orbit and is related to the inertial coordinates by the true anomaly $v(t)$, as given in Equation 1.1.

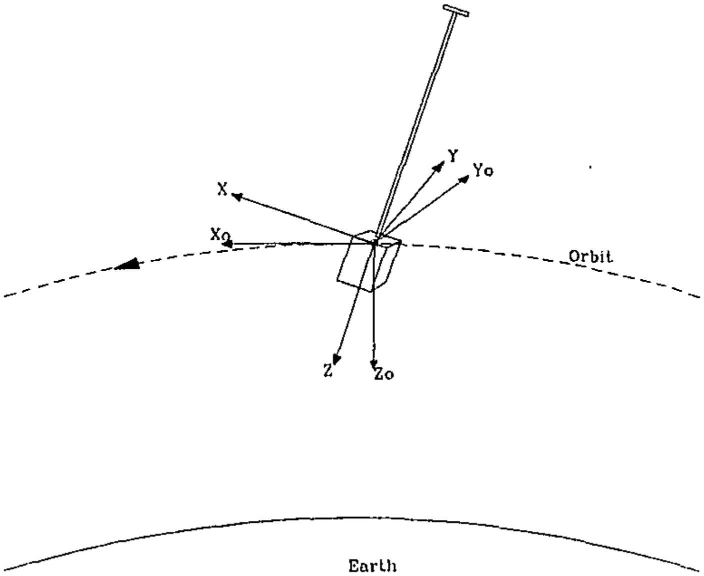

Figure 1.3 Coordinate systems

If the orthogonal axes are presented by a triad of unit vectors, the transformation from inertial to orbital coordinates can be written as:

$$\begin{bmatrix} \vec{\mathbf{x}}_o \\ \vec{\mathbf{y}}_o \\ \vec{\mathbf{z}}_o \end{bmatrix} = \begin{bmatrix} \cos\nu(t) & 0 & \sin\nu(t) \\ 0 & 1 & 0 \\ -\sin\nu(t) & 0 & \cos\nu(t) \end{bmatrix} \begin{bmatrix} \vec{\mathbf{x}}_i \\ \vec{\mathbf{y}}_i \\ \vec{\mathbf{z}}_i \end{bmatrix} \tag{1.1}$$

where,

$\nu(t) \approx M(t) + 2e\sin M(t)$ = the true anomaly for small eccentricities, e

with,

$M(t) = \omega_o t$ = the mean anomaly

and,

$\omega_o = \dfrac{2\pi}{T}$ = the mean orbital angular rate with orbital period T

The attitude of the satellite can be defined by Euler angles [Wertz, 1986]. These angles are obtained from an ordered series of right hand positive rotations from the $X_oY_oZ_o$ set of orbital axes to the XYZ set of body axes, see Figure 1.4. The 1-2-3 sequence of rotations is used in this document. The first rotation is a roll around the orbit X_o-axis, this defines a roll angle ϕ. The next rotation is a pitch around the intermediate Y'-axis, this defines a pitch angle θ. The last rotation is a yaw around the body Z-axis, this defines a yaw angle ψ. The attitude transformation matrix **A** to transform a vector from the reference orbital to the body coordinates, also called the direction cosine matrix (DCM), is given by:

$$\mathbf{A} = \begin{bmatrix} C\psi C\theta & C\psi S\theta S\phi + S\psi C\phi & -C\psi S\theta C\phi + S\psi S\phi \\ -S\psi C\theta & -S\psi S\theta S\phi + C\psi C\phi & S\psi S\theta C\phi + C\psi S\phi \\ S\theta & -C\theta S\phi & C\theta C\phi \end{bmatrix} \tag{1.2}$$

with,

C = cosine function
S = sine function

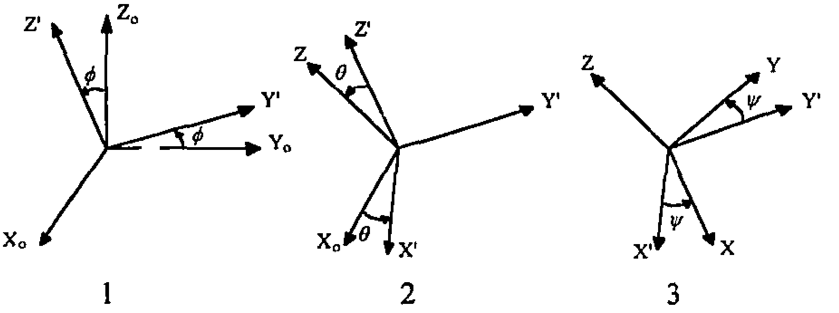

Figure 1.4 Definition of 1-2-3 Euler angle rotations

Although the Euler angle representation has a clear physical interpretation in the roll, pitch and yaw angles, it unfortunately suffers from singularities in the pitch angle θ. A better representation that has no singularities and no trigonometric functions in the transformation matrix, will be the Euler symmetric parameters (quaternion) [Wertz, 1986]. This representation will be used in the control and estimation algorithms in the next few chapters. Due to a lack of obvious physical interpretation of the quaternion, the Euler angles will normally be used to present the attitude during simulation tests. The parameterization of the quaternion is done as:

$$q_1 \equiv e_x \sin\frac{\Phi}{2},\quad q_2 \equiv e_y \sin\frac{\Phi}{2},\quad q_3 \equiv e_z \sin\frac{\Phi}{2},\quad q_4 \equiv \cos\frac{\Phi}{2} \qquad (1.3)$$

where,

e_x , e_y , e_z = components of the unit Euler axis vector in orbit referenced coordinates

Φ = rotation angle around the Euler axis

The quaternion components therefore satisfy the following constraint:

$$q_1{}^2 + q_2{}^2 + q_3{}^2 + q_4{}^2 = 1$$

The transformation matrix to transform any vector from the reference orbital to body coordinates can be written in quaternion format as:

$$\mathbf{A} = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \qquad (1.4)$$

From Equations (1.2) and (1.4) it is now possible to establish a relation between the two representations. If the quaternion representation is used, the respective roll, pitch and yaw angles can be calculated as: (Note: arctan4 is a four quadrant function)

$$\phi = \arctan 4\left\{\frac{-A_{32}}{A_{33}}\right\},\quad \theta = \arcsin\left\{A_{31}\right\},\quad \psi = \arctan 4\left\{\frac{-A_{21}}{A_{11}}\right\} \qquad (1.5)$$

If the Euler angle representation is known, the DCM of Equation (1.2) can be used to calculate the quaternion parameters:

$$q_4 = \frac{1}{2}\left[1 + A_{11} + A_{22} + A_{33}\right]^{0.5} \qquad \text{then,}$$

$$q_1 = \frac{1}{4q_4}\left[A_{23} - A_{32}\right],\quad q_2 = \frac{1}{4q_4}\left[A_{31} - A_{13}\right],\quad q_3 = \frac{1}{4q_4}\left[A_{12} - A_{21}\right] \qquad (1.6)$$

1.5 Equations of Motion

The spacecraft's attitude motion in orbit can be represented by the Euler dynamic and kinematic sets of equations when the quaternion representation is used.

1.5.1 DYNAMIC EQUATIONS

The dynamics of the spacecraft in inertial space will be governed by Euler's equations of motion. With the added influence of the gravity gradient boom and reaction wheel angular momentum, the equation in vector form can be expressed as:

$$\mathbf{I}\dot{\boldsymbol{\omega}}_B^I = \mathbf{N}_{GG} + \mathbf{N}_M + \mathbf{N}_D - \boldsymbol{\omega}_B^I \times \left(\mathbf{I}\boldsymbol{\omega}_B^I + \mathbf{h}\right) - \dot{\mathbf{h}} \tag{1.7}$$

where,

$$\mathbf{I} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} = \text{moment of inertia (MOI) tensor}$$

and,

$$\boldsymbol{\omega}_B^I = \left[\omega_x \quad \omega_y \quad \omega_z\right]^T = \text{inertially referenced body angular rate vector}$$

and, [Wertz, 1986]

$$\mathbf{N}_{GG} = \frac{3GM_{\oplus}}{R_s^3}\left[I_{zz} - \frac{I_{xx}+I_{yy}}{2}\right]\left(\bar{\mathbf{z}}_o \cdot \bar{\mathbf{z}}\right)\left(\bar{\mathbf{z}}_o \times \bar{\mathbf{z}}\right)$$

$= \text{gravity gradient torque vector}$

R_s = geocentric position vector length

$GM_{\oplus}$ = earth's gravitational constant (1.8)

$\bar{\mathbf{z}}_o = \left[A_{13} \quad A_{23} \quad A_{33}\right]^T$

= nadir unit vector in body coordinates

$\bar{\mathbf{z}}$ = principal body Z - axis unit vector

and,

$$\mathbf{h} = \left[h_x \quad h_y \quad h_z\right]^T = \text{reaction wheel angular momentum vector}$$

and,

$\mathbf{N}_M$ = magnetic torque vector

$\mathbf{N}_D$ = external disturbance torque vector

For an axially symmetric satellite like SUNSAT with the principal moments of inertia axes along the body axes, the off-diagonal products of inertia elements in the MOI tensor **I** will be zero. The deployed boom along the body Z-axis also increases the MOI elements I_{xx} and I_{yy} to a much larger and equal value. This value is called the transverse MOI, I_T. The complete set of dynamic equations of motion can then be written in component form as:

$$I_T\,\dot{\omega}_x = N_{mx} + N_{dx} - \frac{3GM_\oplus}{R_s^3}\left(I_T - I_{zz}\right)A_{23}A_{33} - \omega_y\left(I_{zz}\,\omega_z + h_z\right) + \omega_z\left(I_T\,\omega_y + h_y\right) - \dot{h}_x \tag{1.9a}$$

$$I_T\,\dot{\omega}_y = N_{my} + N_{dy} + \frac{3GM_\oplus}{R_s^3}\left(I_T - I_{zz}\right)A_{13}A_{33} + \omega_x\left(I_{zz}\,\omega_z + h_z\right) - \omega_z\left(I_T\,\omega_x + h_x\right) - \dot{h}_y \tag{1.9b}$$

$$I_{zz}\,\dot{\omega}_z = N_{mz} + N_{dz} - \omega_x\,h_y + \omega_y\,h_x - \dot{h}_z \tag{1.9c}$$

For a circular orbit (R_s = constant) and Kepler's third law, which relates the cube of the semimajor axis to the square of the orbital period T (we may ignore the mass of the secondary body - the satellite), we can also write,

$$\frac{GM_\oplus}{R_s^3} \approx \frac{4\pi^2}{T^2} = \omega_o^2 \quad \text{(square of the orbit angular rate)} \tag{1.10}$$

The reaction wheel dynamics for identical wheels aligned to the body principal axes, can be written as:

$$\begin{aligned} \dot{h}_x &= I_w\dot{\omega}_{wx} = N_{wx} \\ \dot{h}_y &= I_w\dot{\omega}_{wy} = N_{wy} \\ \dot{h}_z &= I_w\dot{\omega}_{wz} = N_{wz} \end{aligned} \tag{1.11}$$

with,

I_w = wheel moment of inertia
ω_{wi} = wheel-i angular rate
N_{wi} = wheel-i torque

1.5.2 KINEMATIC EQUATIONS

The kinematic (attitude) update of the satellite will be done by using the quaternion representation. The following vector set of differential equations is used:

$$\dot{\mathbf{q}} = \frac{1}{2}\boldsymbol{\Omega}\,\mathbf{q} \tag{1.12}$$

with,

$$\boldsymbol{\Omega} = \begin{bmatrix} 0 & \omega_{oz} & -\omega_{oy} & \omega_{ox} \\ -\omega_{oz} & 0 & \omega_{ox} & \omega_{oy} \\ \omega_{oy} & -\omega_{ox} & 0 & \omega_{oz} \\ -\omega_{ox} & -\omega_{oy} & -\omega_{oz} & 0 \end{bmatrix} \tag{1.13}$$

where,

$\boldsymbol{\omega}_B^O = \left[\omega_{ox} \quad \omega_{oy} \quad \omega_{oz}\right]^T$ = body angular rate referenced to orbit coordinates

The angular body rates referenced to the orbit following coordinates can be obtained from the inertially referenced body rates by using the transformation matrix **A**:

$$\begin{bmatrix} \omega_{ox} \\ \omega_{oy} \\ \omega_{oz} \end{bmatrix}_B^O = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}_B^I - \mathrm{A} \begin{bmatrix} 0 \\ -\tilde{\omega}_o(t) \\ 0 \end{bmatrix} \tag{1.14}$$

where,

$\tilde{\omega}_o(t) \approx \omega_o \{1 + 2e\cos(\omega_o t)\}$ = the true angular rate for small eccentricities, e

When quaternions are used directly in the control algorithms of the next few chapters, it will be convenient to define an error quaternion. The error quaternion will be the quaternion difference between the current quaternion and the commanded quaternion. This is calculated by using the definition of quaternion division:

$$\begin{bmatrix} q_{1e} \\ q_{2e} \\ q_{3e} \\ q_{4e} \end{bmatrix} = \begin{bmatrix} q_{4c} & q_{3c} & -q_{2c} & -q_{1c} \\ -q_{3c} & q_{4c} & q_{1c} & -q_{2c} \\ q_{2c} & -q_{1c} & q_{4c} & -q_{3c} \\ q_{1c} & q_{2c} & q_{3c} & q_{4c} \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{1.15}$$

where,

q_{ie} = components of the error quaternion
q_{ic} = components of the commanded quaternion

1.5.3 SMALL ANGULAR MOTIONS

A gravity gradient stabilized satellite will passively tend to keep the long axis (axis of smallest MOI) nadir pointing. Small disturbances will however induce librations of this axis around the nadir direction. Due to a lack of any passive damping mechanism in space, these librations will persist unless actively damped by controlled actuation. The nature of these librations can be investigated by analysing Euler's dynamic equations.

In a circular or near circular orbit with the satellite nominally nadir pointing with only small pitch and roll angular deviations and a zero yaw angle, Equations (1.9a) to (1.9c) become, (assume zero reaction wheel momentum and zero applied external or internal torque)

X'-axis: (near roll axis)

$$I_T\dot{\omega}_x \approx -3\omega_o^2(I_T - I_{zz})\phi + \omega_y\omega_z(I_T - I_{zz}) \tag{1.16a}$$

Y'-axis: (pitch axis)

$$I_T\dot{\omega}_y \approx -3\omega_o^2(I_T - I_{zz})\theta - \omega_x\omega_z(I_T - I_{zz}) \tag{1.16b}$$

Z-axis: (yaw axis)

$$I_{zz}\dot{\omega}_z = 0 \tag{1.16c}$$

From Equations (1.2) and (1.14) with small pitch and roll angles and zero yaw angle,

$$\omega_x \approx \dot{\phi}, \qquad \omega_y \approx -\omega_o + \dot{\theta}, \qquad \omega_z \approx \omega_o \phi \approx 0 \tag{1.17}$$

Using Equation (1.17), assuming a deployed boom ($I_T >> I_{zz}$) and ignoring all the small cross coupling terms, the roll and pitch axis equations (1.16a) and (1.16b) can be approximated further,

Roll libration:

$$\ddot{\phi} + \omega_\phi^2 \phi = 0 \qquad \text{with} \qquad \omega_\phi = 2\omega_o \tag{1.18a}$$

Pitch libration:

$$\ddot{\theta} + \omega_\theta^2 \theta = 0 \qquad \text{with} \qquad \omega_\theta = \sqrt{3}\omega_o \tag{1.18b}$$

Equations (1.18a) and (1.18b) represent simple harmonic oscillators. Small oscillations, called librations, will occur around nadir at twice the orbital rate for roll and $\sqrt{3}$ times the orbital rate for pitch. These results have also been derived by Hodgart [1989] using Newton's laws of rotational motion.

1.5.4 ROLL BIAS ANGLE

Another interesting feature of a gravity gradient stabilized spacecraft which is spinning at a constant rate around the nadir pointed axis (body Z-axis), is that a constant roll bias from the nadir direction will occur. This roll angle bias is due to a gravity gradient torque cancelling of a gyroscopic torque induced by the orbital rotation of the Z-spinning satellite. The roll bias can be obtained not only by spinning the satellite's body around the Z-axis, but also by maintaining an angular momentum on the Z-axis reaction wheel. Both these results will now be derived using Equations (1.2), (1.9a) and (1.14) for a circular or near circular orbit.

Body Z-spin: (Zero reaction wheel angular momentum)

If we assume a zero pitch angle ($\theta = 0$), and a constant roll bias ϕ for a constant body angular rate (orbit referenced) around the Z-axis only,

$$\omega_B^O = \begin{bmatrix} 0 & 0 & \omega_{oz} \end{bmatrix}^T = \begin{bmatrix} 0 & 0 & \dot{\psi} \end{bmatrix}^T \tag{1.19}$$

from (1.14) and (1.19)

$$\begin{aligned} \omega_x &= -\omega_o \sin\psi \cos\phi, \qquad & \dot{\omega}_x &= -\omega_o \cos\phi \cos\psi \, \omega_{oz} \\ \omega_y &= -\omega_o \cos\psi \cos\phi \\ \omega_z &= \omega_o \sin\phi + \omega_{oz} \end{aligned} \tag{1.20}$$

from (1.2), (1.9a) and (1.20),

$$I_T\omega_{oz} = 4\omega_o(I_T - I_{zz})\sin\phi + \omega_{oz}(I_T - I_{zz})$$

or,

$$\sin\phi = \left(\frac{\omega_{oz}}{\omega_o}\right) \Big/ \left(4\frac{I_T}{I_{zz}} - 4\right) \qquad (1.21)$$

An increase in the Z-spin rate relative to the orbital rate will increase the roll bias, whereas an increase in the MOI ratio will decrease the roll bias. A typical value for the roll bias in the SUNSAT case will be,

$$\frac{\omega_{oz}}{\omega_o} = 10, \quad \frac{I_T}{I_{zz}} = 20 \qquad \Rightarrow \quad \phi_{bias} = 7.56^\circ$$

Z-wheel angular momentum: (Zero body spin)
If we assume a zero pitch angle ($\theta = 0$) and a constant roll bias ϕ due to a constant Z-wheel angular momentum,

$$\mathbf{h} = [0 \quad 0 \quad h_z] \qquad (1.22)$$

from (1.14),

$$\omega_x = 0, \quad \omega_y = -\omega_o\cos\phi, \quad \omega_z = \omega_o\sin\phi \qquad (1.23)$$

from (1.2), (1.9a) and (1.23),

$$0 = 3\omega_o^2(I_T - I_{zz})\sin\phi\cos\phi + \omega_o\cos\phi\left[(I_T - I_{zz})\omega_o\sin\phi - h_z\right]$$

or,

$$\sin\phi = \left(\frac{h_z}{I_{zz}\omega_o}\right) \Big/ \left(4\frac{I_T}{I_{zz}} - 4\right) \qquad (1.24)$$

If Equations (1.21) and (1.24) are compared, similar bias values will be found as in the body Z-spin case for equal angular momentum values as expected:

$$h_z = I_{zz}\omega_{oz} \qquad (1.25)$$

It will therefore be easy to transfer momentum from the Z-spinning satellite during normal flight mode to the Z-axis reaction wheel before imaging and keep the roll bias angle unchanged. A specific roll bias angle might be needed for pointing the pushbroom imager to the left or right of the sub-satellite track.

1.6 External Disturbance Torques

The attitude of the satellite in orbit will be influenced by both the controlled actuator torquers, for example the magnetic torquers and reaction wheels, and uncontrolled disturbance torques, for example gravity gradient, aerodynamic and solar pressure. All the significant external disturbance torques which tend to disturb the satellite's attitude will now be introduced.

1.6.1 GRAVITY GRADIENT

The gravity gradient (GG) torque is mainly a disturbance torque in the positive sense of the word, because its influence tends to keep the satellite nadir pointing. The GG torque originates from the so called "dumb bell" effect on a long thin rotating object [Wertz, 1986; Hughes, 1986]. The GG torque is as defined in Equation (1.8) for the SUNSAT geometry. Its instant magnitude is mainly dependent on the angular deviations of the body Z-axis from the nadir direction, it will therefore be influenced by the roll and pitch angles of the satellite. For an elliptical orbit, the magnitude will also be inversely affected by the cube of the distance from the orbital position to the geocentric position. From Equations (1.2) and (1.8) for the near roll and pitch axes (yaw angle $\psi = 0$):

X'-axis: (near roll)

$$\begin{aligned} N_{GGX'} &= -\frac{3GM_{\oplus}}{R_S^3}\left(I_T - I_{zz}\right)\text{c[illegible]}\cos\phi\sin\phi \\ &\approx -\frac{3GM_{\oplus}}{R_S^3}\left(I_T - I_{zz}\right)\phi, \qquad \text{for small pitch and roll angles} \end{aligned} \tag{1.26}$$

Y'-axis: (pitch)

$$\begin{aligned} N_{GGY'} &= -\frac{3GM_{\oplus}}{R_S^3}\left(I_T - I_{zz}\right)\cos\theta\cos^2\phi\sin\theta \\ &\approx -\frac{3GM_{\oplus}}{R_S^3}\left(I_T - I_{zz}\right)\theta, \qquad \text{for small pitch and roll angles} \end{aligned} \tag{1.27}$$

The GG torque for SUNSAT at small angular deviations from nadir, will be: (For I_T = 40 kgm^2 and I_{zz} = 2 kgm^2)

At perigee: (400 km)	$N_{GG} = 2.55 \times 10^{-6}$ Nm/deg
At apogee: (840 km)	$N_{GG} = 2.12 \times 10^{-6}$ Nm/deg

For an elliptical orbit the nadir vector does not rotate at a constant inertial rate within the orbital plane. The GG torque will therefore result in a disturbance torque around the Y_o-axis, leading to a pitch libration. For the SUNSAT orbit, with eccentricity e = 0.03, the minimum GG induced libration magnitude will be as shown in Figure 1.5. The peak-to-peak libration amplitude is approximately 4°. This corresponds well to the pitch behaviour's particular solution in Equation (18.57) from Wertz [1986], where an orbit eccentricity of 0.03 and SUNSAT's MOI values will result in a sinusoidal steady state pitch error of 3.72° peak-to-peak amplitude.

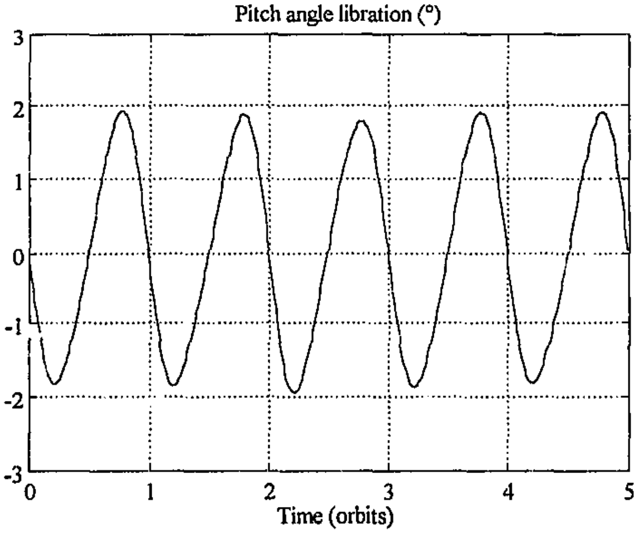

Figure 1.5 Typical GG induced libration for an elliptical orbit.

1.6.2 AERODYNAMIC

At the low earth orbits where micro satellites operate, the total atmospheric density is not totally negligible. Furthermore, at these orbits the spacecraft's velocity is also very high. From Hughes [1986] and Wertz [1986], we can use the following simplified result: Aerodynamic pressure is directly proportional to the air density and the square of the relative air velocity. The major assumption leading to this result is that any surface exposed to the slip stream of the spacecraft, completely absorbs the momentum of the incoming colliding particle. The aerodynamic disturbance torque vector on a spacecraft structure can then be obtained by taking the cross product of the aerodynamic pressure vector on the total projected area, and the vector from the centre of mass to the centre of pressure of the total structure,

$$\mathbf{N}_{AERO} = \rho_a V^2 A_p \left[\mathbf{c}_p \times \bar{\mathbf{V}}\right] \tag{1.28}$$

where,

ρ_a = atmospheric density

V = magnitude of spacecraft's velocity vector

$\bar{\mathbf{V}}$ = unit velocity vector

A_p = total projected area of spacecraft
$\mathbf{c}_p$ = vector between centre of mass and centre of pressure

The atmospheric density is a strong function of altitude, solar activity and whether the sun is visible or not (orbit day or night). Table 1.1 list the expected air density values at orbit day-night during high or low solar activity for the SUNSAT apogee and perigee altitudes. These values were obtained from Figure 8.12, p.259 in Hughes [1986].

Table 1.1 Typical atmospheric density values ρ_a (kg/m^3 × 10^{-15})

	Perigee (400 km)	*Apogee (840 km)*
Low solar activity	1000-500	6-4
High solar activity	50000-20000	800-500
Average [Wertz, 1986]	4000	10

The greatest variation in aerodynamic torque per orbit will therefore be due to the altitude variation of the SUNSAT orbit (× 400 on average), whereas the day-night variation is about 2 orders of magnitude smaller (× 2 on average).

For a spacecraft structure such as SUNSAT, the major components contributing to the total aerodynamic torque, are the main box-like body, the boom element and the tip mass. Due to the symmetry of the satellite, the effective aerodynamic torque will be in the direction of the non-rotating body Y'-axis (the axis that defines the pitch angle). The magnitude of this torque for SUNSAT can approximately be estimated as, (from Figure 1.6 when neglecting a Z-rotation)

$$N_{AERO} = \rho_a V^2 \left(A_1 c_1 + A_2 c_2 + A_3 c_3\right) \tag{1.29}$$

where,

A_1c_1 = -0.057 m^3 (main body)
A_2c_2 = 0.030 m^3 (tip mass)
A_3c_3 = 0.037 m^3 (boom element)
V = 7788 ms^{-1} at perigee, 7313.5 ms^{-1} at apogee

If the average atmospheric density values of Table 1.1 is used,
Perigee: $N_{AERO\text{-}avg} = 2.2 \times 10^{-6}$ Nm
Apogee: $N_{AERO\text{-}avg} = 5.3 \times 10^{-9}$ Nm

If the maximum atmospheric density values of Table 1.1 is used,
Perigee: $N_{AERO\text{-}max} = 3.2 \times 10^{-5}$ Nm
Apogee: $N_{AERO\text{-}max} = 4.2 \times 10^{-7}$ Nm

From these results it is clear that the aerodynamic disturbance torque at perigee will have the greatest influence. Although the torque profile over the full orbit will have some non-regular shape, will not change much from orbit to orbit. It can therefore be modelled as a periodic waveform. Due to the almost logarithmic behaviour of the atmospheric density with altitude, the natural logarithm of the aerodynamic disturbance torque was modelled as a polynomial in a normalized altitude parameter.

$$\ln\left\{N_{AERO}(\bar{x})\right\} = a_0\bar{x}^2 + a_1\bar{x} + a_2 \tag{1.30}$$

where,

$$\bar{x} = \frac{h_{satellite}}{h_{perigee}}$$

The coefficients a_0 to a_2 were chosen to fit, in the least square sense, the computed aerodynamic torque for the SUNSAT structure at a few discrete altitude values. The COSPAR International Reference Atmosphere (CIRA 72) values of Table L-6, p.820, in Wertz [1986] was used for the density values. The respective coefficients for the average and maximum aerodynamic torque models are listed in Table 1.2.

Table 1.2 Coefficients for the aerodynamic disturbance torque models

	a_0	a_1	a_2
Average model	1.5833	-10.5506	-4.0459
Maximum model	-2.0247	1.7812	-10.1063

Figures 1.7 and 1.9 show the combined minimum effect of GG (eccentricity) and aerodynamic disturbance when the average and maximum atmospheric density values are used during simulation. Figures 1.8 and 1.10 show the corresponding aerodynamic disturbance torque during average and maximum atmospheric density conditions.

The maximum peak-to-peak libration amplitude has increased from approximately 4° in Figure 1.7 to a value of 23° in Figure 1.9, during maximum solar activity. When compared to Figure 1.5 (GG disturbance torque only), we conlude that the aerodynamic disturbance torque does not have a large influence on the dynamics during average conditions (normal solar activity) except adding a small constant pitch offset and some distortion to the almost sinusoidal waveform of Figure 1.5. The disturbing effect on the pitch libration can, however, increase 6-fold in magnitude during periods of high solar activity, when the atmospheric density is much higher at perigee.

The magnitude and influence of the aerodynamic disturbance torque can be reduced by improving the geometrical design of the total structure: Equation (1.28) clearly indicates the importance of having the distance between the centre of mass and the centre of pressure of the total spacecraft as small as possible. According to Equation (1.29) we need as much as possible a cancellation of the terms in brackets

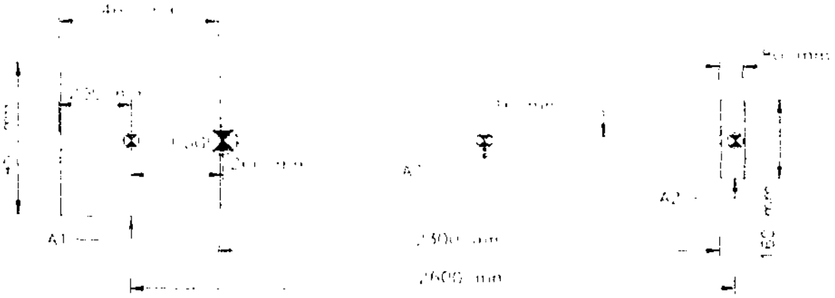

Figure 1.6 Simplified geometrical structure of SUNSAT

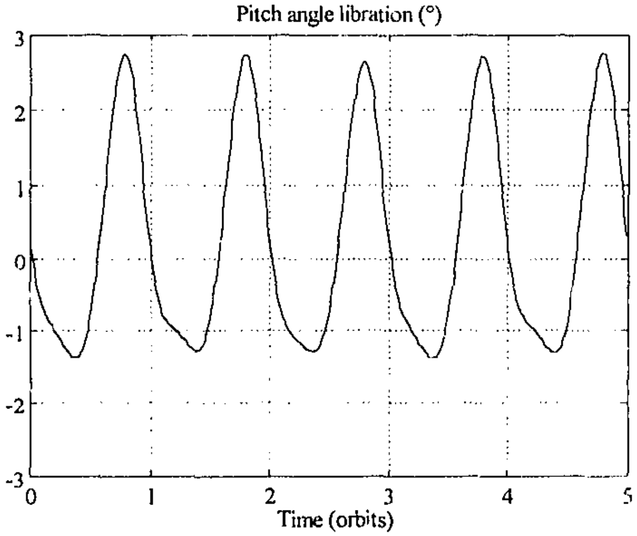

Figure 1.7 GG and average aerodynamic influence on the pitch dynamics

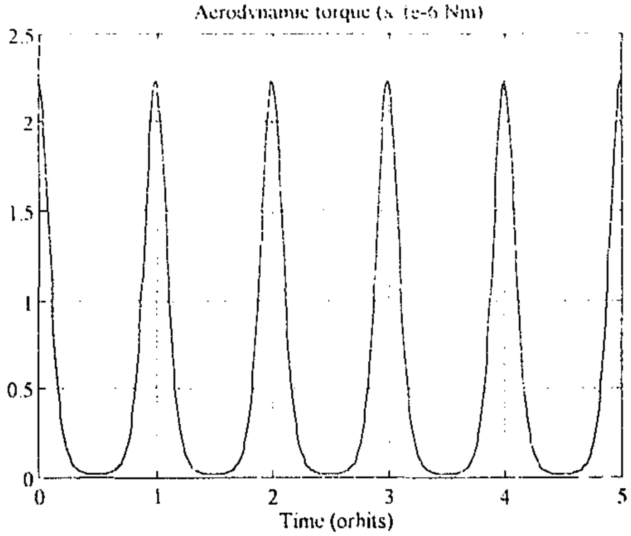

Figure 1.8 Average aerodynamic disturbance torque on SUNSAT structure

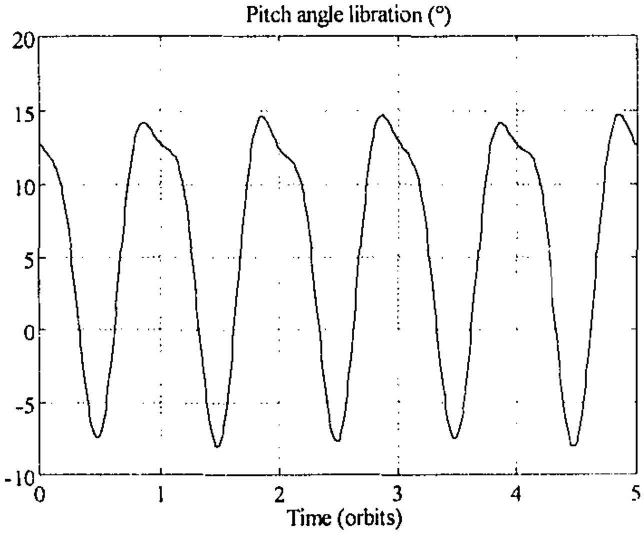

Figure 1.9 GG and maximum aerodynamic influence on the pitch dynamics

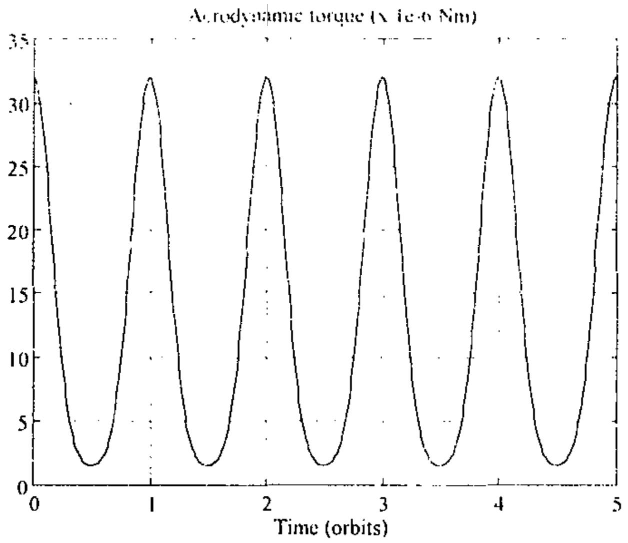

Figure 1.10 Maximum aerodynamic disturbance torque on SUNSAT structure

1.6.3 SOLAR RADIATION

Illumination of the sun on a fully absorbtive surface causes a solar radiation pressure of d_o/c on the surface normal, where d_o = 1358 Wm^{-2} (the average solar radiation constant) and c is the velocity of light (3 × 10^8 ms^{-1}). For general surfaces this effect is much more complicated, see Hughes [1986] and Wertz [1986]. It is normally a function of the absorbtion, specular and diffuse reflection coefficients.

A similar model to Figure 1.6 can be used for the worst case scenario of a maximum incidence angle of the solar radiation on fully absorbtive (black body) surfaces, where:

$$N_{SOLAR} = \frac{d_o}{c}\left(A_1c_1 + A_2c_2 + A_3c_3\right) = 4.5 \times 10^{-8} \text{ Nm} \qquad (1.31)$$

The worst case solar radiation disturbance torque is therefore at least 50 times smaller than the average aerodynamic disturbance torque at perigee, and its influence can be ignored.

1.7 References

Hodgart M.S. and Wright P.S. [1987].
"Attitude determination, control and stabilisation of UoSAT-2", *The Journal of the Institution of Electronic and Radio Engineers*, Supplement to Vol.57, No.5, Sept/Oct 1987, pp. S151-S173

Hodgart M.S. [1989]
"Gravity Gradient and Magnetorquing Attitude Control for Low-Cost Low Earth Orbit Satellites - The UoSAT Experience", *Ph.D. Submission at University of Surrey*, June 1989.

Hughes P.C. [1986].
Spacecraft Attitude Dynamics, John Wiley & Sons, 1986.

Milne G.W. *et.al.* [1993].
"SUNSAT, Stellenbosch University and SA-AMSAT's Remote Sensing and Packet Communications Microsatellite", *Proceedings of the 7th Annual AIAA/USU Conference on Small Satellites*, Utah State University, Sept.1993.

Pal P.K. and Fox S.M. [1993].
"The Effects of Momentum Bias on a Gravity Gradient Stabilized Spacecraft with Active Magnetic Control", *Proceedings of the 7th Annual AIAA/USU Conference on Small Satellites*, Utah State University, Sept.1993.

Psiaki M.L., Martel F., and Pal P.K. [1990].
"Three-Axis Attitude Determination via Kalman Filtering of Magnetometer Data", *Journal of Guidance, Control, and Dynamics*, Vol.13, No 3, May-June 1990, pp. 506-514.

Steyn W.H. [1990].
"Attitude Control Algorithms and Simulation Programs for Low Earth Orbit Spacecraft", *M.Sc University of Surrey*, September 1990.

Van Bezooijen R.H.W. [1989].
"A Star Pattern Recognition Algorithm for Autonomous Attitude Determination", *Proceedings of the 1989 IFAC Conference on Automatic Control in Aerospace*, Tsukuba Japan, pp. 51-58.

Wertz J.R. [1986].
Spacecraft Attitude Determination and Control, D. Reidel Publishing Company Boston U.S.A., Reprint 1986.

2. MAGNETIC TORQUER CONTROL

2.1 Introduction

2.1.1 MAGNETORQUING

The 3-axis magnetic torquer coils will be used primarily for attitude control and momentum dumping of the reaction wheels during periods when imaging are inactive. In this chapter the main focus will be on the attitude control function of the magnetic torquers. The X and Y-axis torquers are air core coils embedded into channels around the edges of the solar panels and the Z-axis coils into channels within the top facet of the satellite. For redundancy and torque control reasons there are two coils per axis and they can be switched on singly or in parallel. The coils can each be fed with a constant current - switched in two directions - to generate a vector magnetic dipole moment **M**. The magnitude M of this moment for a specific coil is given by.

$$M = nIA \tag{2.1}$$

where,

n = number of turns per coil

I = DC current through the coil

A = effective coil area

The vector dipole moment **M** from all coils will interact with the geomagnetic field vector **B** to generate a magnetic torque vector $\mathbf{N}_M$ by taking the vector cross product,

$$\mathbf{N}_M = \mathbf{M} \times \mathbf{B} \tag{2.2}$$

Although the direction and magnitude of **M** can be controlled on average by the correct interleaving of three orthogonal coils, the **B** vector is totally dependent on the orbital location. As a result, the torque $\mathbf{N}_M$ will always be orthogonal to **B** (and **M**) and not favorable in certain regions of the orbit to control the attitude of a specific spacecraft axis. It is also possible that a desirable control torque for a certain attitude axis (pitch, roll or yaw), when a specific coil or combination of magnetic coils are switched, will generate undesired disturbance torques for the other axes. The main objective of the control algorithms in this chapter will be to optimize the control effort maximize the desired influence and minimize the undesired disturbances.

2.1.2 GEOMAGNETIC FIELD

The earth's magnetic field is predominantly a magnetic dipole such as the field produced by a sphere of uniform magnetization or a current loop. The magnetic field can be expressed mathematically by a spherical harmonic model, the so-called IGRF (International Geomagnetic Reference Field) model [Wertz, 1986]. A summary of the derivations and expressions of this model is given in Appendix A. Due to secular drift and magnitude decrease of the geomagnetic field, the coefficients of the IGRF model are updated every 4 years and supplied with secular variation terms. For purposes of simulation and further discussion in this chapter, a first order dipole model [Rodden, 1984] will be used to represent the geomagnetic field vector. This dipole vector can be expressed as,

$$\mathbf{B} = \nabla\left[\frac{\vec{\mathbf{R}}^T\mathbf{M}_e}{R_s^3}\right] = \left[\mathbf{I} - 3\vec{\mathbf{R}}\vec{\mathbf{R}}^T\right]\frac{\mathbf{M}_e}{R_s^3} \qquad (2.3)$$

where,

∇ = vector gradient operator

R_s = geocentric position vector length

$\vec{\mathbf{R}}$ = unit geocentric position vector

$\mathbf{M}_e$ = vector geomagnetic strength of dipole

$\mathbf{I}$ = identity matrix

In orbital coordinates, the model is expressed as,

$$\mathbf{B}_o = \begin{bmatrix} B_{ox} \\ B_{oy} \\ B_{oz} \end{bmatrix} = \frac{M_e}{R_s^3}\begin{bmatrix} \sin i \cdot \cos\alpha \\ -\cos i \\ 2\sin i \cdot \sin\alpha \end{bmatrix} \qquad (2.4)$$

where,

i = orbit inclination

α = orbit angle as measured from the ascending node

From this model it can be calculated that the polar low earth orbit (LEO) of SUNSAT ($i \approx 96°$ and average altitude ≈ 620 km) presents a small constant B_{oy} component of 2.5 μT, a maximum B_{ox} component of 23.2 μT over the equator with B_{oz} zero, and a maximum B_{oz} component of 46.4 μT over the polar region with B_{ox} zero. The geomagnetic vector, therefore, rotates inertially twice per polar orbit, almost within the orbital plane. (Note that the orbital axes rotates inertially once per orbit.)

2.1.3 LITERATURE SURVEY

Most applications, where magnetic torquers are used for attitude control, are dedicated to spin stabilized satellites where the spin axis of the satellite is kept inertially fixed [Ergin, 1965; Goel, 1979; Rodden, 1984; Parkinson, 1990]. These controllers are mostly used for nutation damping and slewi.ıg or pointing of the spin axis. The spin axis is mostly aligned to the normal of the orbital plane or else pointing to a reference object, for example, a distant star or the sun. Some application. substitute the spin moment of the spacecraft's body with a momentum wheel [Chang, 1992; Pal, 1992]. Magnetic torquers are then used to control the momentum wheel speed and to point the wheel spin axis. Applications combining passive gravity gradient torque with active magnetic torquing similar to SUNSAT are mostly used on micro satellites [Martel, 1988; Hodgart, 1987;1989;1994; Baron, 1994; Sebestyen, 1992]. The relevant work from these latter groups will now be overviewed.

Martel, Pal & Psiaki Control System

The only known (to the author) 3-axis stabilization control algorithm for passive gravity g.adient with active magnetic torquing was derived by Martel *et.al.* [1988]. Their algorithm makes use of the cross-product law first presented by Stickler and Alfriend [1974]. This method tends to choose the "most favorable" magnetorquing direction at any control instant by interleaving or simultaneously switching any of the three orthogonal magnetic coils. relying on the current direction of the local geomagnetic field vecto.. Depending on the required torque vector and the given geomagnetic field vector, this most favorable magnetorquing vector unfortunately does not exclude the generation of unfavorable cross disturbances. This will be explained briefly by means of an example.

First their algorithm: The control error on all three axes is expressed as a proportional and derivative (PD) error correction vector **e**:

$$\mathbf{e} = \mathbf{K}_p\,\mathbf{a} + \mathbf{K}_d\,\dot{\mathbf{a}} \qquad (2.5)$$

where,

$\mathbf{K}_p$ = proportional gain diagonal matrix

$\mathbf{K}_d$ = derivative gain diagonal matrix

$\mathbf{a}, \dot{\mathbf{a}}$ = attitude and rate error vectors

The most favorable magnetorquing vector **M** is then expressed as,

$$\mathbf{M} = \frac{\mathbf{e}\cdot\mathbf{B}}{\|\mathbf{B}\|} \qquad (2.6)$$

where,

$\mathbf{B}$ = body geomagnetic field vector from a magnetometer

The actual torque N_M, then applied to the satellite is obtained from Equation (2.2). For example, if the satellite is 3-axis stabilized with zero roll and yaw attitude error but only having a pitch error, and assuming that the local geomagnetic field vector contains components in all three body axis,

then,

$$\mathbf{e} = \theta_{err}\vec{\mathbf{y}} \qquad \text{and} \qquad \mathbf{B} = B_x\,\vec{\mathbf{x}} + B_y\,\vec{\mathbf{y}} + B_z\,\vec{\mathbf{z}}$$

thus,

$$\mathbf{M} = (\theta_{err}b_z)\,\vec{\mathbf{x}} - (\theta_{err}b_x)\,\vec{\mathbf{z}}$$

and,

$$\mathbf{N}_M = \theta_{err}\left[\left(b_xB_y\right)\vec{\mathbf{x}} - \left(b_zB_z + b_xB_x\right)\vec{\mathbf{y}} + \left(b_zB_y\right)\vec{\mathbf{z}}\right]$$

The second term in the previous equation delivers the desired torque to correct the pitch error, but the first term disturbs the roll and the third term the yaw dynamics directly. An approximate stability analysis based on the linearized equations of motion and averaging of the field parameters, has been performed by Martel *et.al.* [1988]. They have also included the effect of aerodynamic disturbance torques in their model. It was found that stable controllers could be implemented for various orbit inclinations and initial conditions for the attitude angles and rates. The reference attitude for their controller is a zero pitch and roll angle and a certain commanded yaw angle.

An automatic deployment sequence for the gravity gradient satellite after launch and injection in the desired orbit, has also been outlined. The spacecraft must first be despun or detumbled. It must then be stabilized in an attitude and with rates suitable for extention of the gravity gradient boom. The residual libration after boom deployment must then be damped. Simple control laws using magnetic torquing is used to achieve all these actions. The various phases and control laws used in Martel *et.al.* [1988] before boom deployment, are:

Despin, detumble and initial orientation

A magnetic torquer along the desired spin axis (Y-axis) is controlled such that,

$$M_y = -k_d\,\dot{B}_y \tag{2.7}$$

where the derivative of the Y-axis mponent of the geomagnetic field is used. This brings the spacecraft momentum vector close to the orbital normal.

Spin rate control

A certain reference Y-spin rate ($\omega_{y\ ref}$) can be controlled by using the X-axis torquer,

$$M_x = k_s\,\mathrm{sgn}\left(\omega_{y\ ref} - \omega_y\right)\mathrm{sgn}\left(B_z\right) \tag{2.8}$$

where ω_y can be estimated from magnetometer measurements

Boom extension

The spacecraft will be ready for boom extension when the Y-spin axis is sufficiently close to the orbit normal and the Y-spin rate is approximately,

$$\omega_y = -\frac{I_{yf}}{I_{yi}}\omega_o \tag{2.9}$$

where,

I_{yf} = final MOI after boom deployment
I_{yi} = initial MOI before boom deployment
ω_o = orbital rate

Hodgart Control Laws

The micro satellites developed by the University of Surrey since 1981 [Sweeting, 1994] have all been stabilized by passive gravity gradient and active magnetorquing [Hodgart, 1987; 1989; 1994]. The performance of these control laws has already been proven in space. The various control laws used, are:

Spin control

A Z-spin (along the deployed boom direction) is normally needed to ensure a uniform temperature distribution within the satellite. A spin period of between 5 and 10 minutes is used. The B_x or B_y measured body components of the geomagnetic field can be used to determine the presence and sign of the spin. A possible spin control formulation can be,

$$M_x = k_s \operatorname{sgn}(\dot{B}_x) \tag{2.10}$$

When the fire time constant k_s is chosen positive, the spin rate will be increased and when negative, decreased.

Detumbling/capture

Before boom deployment the spacecraft is detumbled to ensure gravity gradient capture afterwards. The magnetic field components are measured using a magnetometer and the angle between the body Z-axis and the geomagnetic vector is computed,

$$\beta = 90° - \arctan\frac{B_z}{\sqrt{B_x^2 + B_y^2}} \tag{2.11}$$

The Z-axis magnetorquer is then pulsed at every transition of β through a given angle (90° for maximum torque), according to the sign of the rate of change of β

$$M_z = k_d \operatorname{sgn}(\dot{\beta}) \qquad (2.12)$$

The fire time constant k_d is chosen to have a certain damping rate on the satellite's tumbling motion. This control law will tend to damp any motion of the spacecraft's Z-axis relative to the geomagnetic field vector. Equation (2.12) is also used after boom deployment to reduce the libration and to achieve earth pointing. If the satellite has been captured the wrong side up, an uncapture control law can be used similar to the one above but with a sign change to k_d.

Delibration

A robust method that requires no attitude estimator is used to improve the damping of any libration motion. The basic principle is to compare the rate of change in the magnetometer measured β angle to a reference angle α, calculated from a geomagnetic orbit referenced model:

$$\alpha = 90° - \arctan\frac{B_{oz}}{\sqrt{B_{ox}^2 + B_{oy}^2}} \qquad (2.13)$$

The Z-axis magnetorquer is then pulsed at a polarity and fire time period according to the following control law,

$$M_z = k_l\left|\dot{\beta} - \dot{\alpha}\right|\operatorname{sgn}(\dot{\beta} - \dot{\alpha}) \qquad (2.14)$$

The fire time constant k_l is an empirically determined value for the best rate of delibration. In-flight results on the UoSAT satellites showed peak libration errors of 5° and RMS errors of between 2° and 3° when applying the control law of Equations (2.11), (2.13) and (2.14).

2.2 Boom Deployment

2.2.1 THE ALGORITHM

It is important that any initial detumbling and boom deployment control law make use of rough attitude knowledge. Initially most of the accurate attitude sensors are impractical to use, due their limited field of view. Practical measurements, however, can be obtained from a magnetometer. In combination with a geomagnetic field model and a known position of the satellite in orbit, measurements from the magnetometer can be used to supply innovations to an attitude and angular rate estimator (Chapter 5). A more robust approach will be to estimate only approximate angular body rate values from the magnetometer measurements alone (Chapter 5). The algorithm described in this paragraph uses only estimates of the 3-axis angular rates of the spacecraft

The algorithm is based on the cross-product law [Stickler, 1974] and will choose the magnetic moment vector such that the magnetic torque is as best aligned as possible (depending on the geomagnetic vector) to the negative of the angular rate error vector. The angular rate error vector **e** is defined as the scaled vector difference of the measured body angular rate vector (orbit referenced) and a desired angular rate vector

$$\mathbf{e} = \mathbf{K}_w \left[\boldsymbol{\omega}_B^O - \boldsymbol{\omega}_{ref} \right] = \mathbf{K}_w \begin{bmatrix} \omega_{ox} \\ \omega_{oy} - \omega_{oy\ ref} \\ \omega_{oz} \end{bmatrix} \qquad (2.15)$$

where,

$\mathbf{K}_w$ = diagonal gain matrix

$$\omega_{oy\ ref} = \omega_o \left(1 - \frac{I_{yf}}{I_{yi}} \right)$$

The desired angular rate vector is chosen in such a way as to ensure alignment of the satellite's Y-axis with the orbit normal. Furthermore, the Y-spin rate must conserve the angular momentum before and after boom deployment. All this will happen if the desired X and Z-axis angular rates become zero and the Y-axis rate satisfies the reference value, computed from the Y-axis MOI ratio. The magnetic moment vector **M** is then calculated similar to Equation (2.6). The boom will be deployed as soon as the angular rate error become small enough, at an instant when the +Z facet is pointing approximately towards zenith (Note: Assuming rapid boom extension compared to the Y-spin period). This will ensure the right side up gravity gradient capturing of the satellite and small residual libration after boom deployment.

Before the simulation results are presented on a typical boom deployment sequence, a few practical aspects have to be discussed first.

2.2.2 PULSE WIDTH MODULATION

The required magnetic moment vector **M** can not be supplied as an analogue continuous signal due to the discrete nature of the onboard control processors and the switching action of the magnetic torquer coils. A pulse width modulation (PWM) technique was then adopted to approximate the stepwise continuous signal, normally utilized when a discrete control system plus zero order hold (ZOH) circuit drives an analogue plant. The ZOH is always an integral part of any hybrid control system design. One of the characteristics of a ZOH is that it causes a group delay of half a sample period within the control loop [Franklin, 1990].

The actual plant (spacecraft in this application) will be fed with magnetic control pulses and not a stepwise continuous signal from a ZOH as is normally the case for hybrid control systems. To obtain mathematically similar results at the sampling instances to the ZOH, we have to centre each control pulse within the sample period and ensure that the average of this pulse over the total sampling period equal the constant step value during the sampling period. In other words, due to the low pass characteristic of the satellite's dynamics towards magnetic ɔrquing, the influence of a large pulse centered within the sample period is similar to a smaller, scaled down pulse extending the full period (see Figure 2.1).

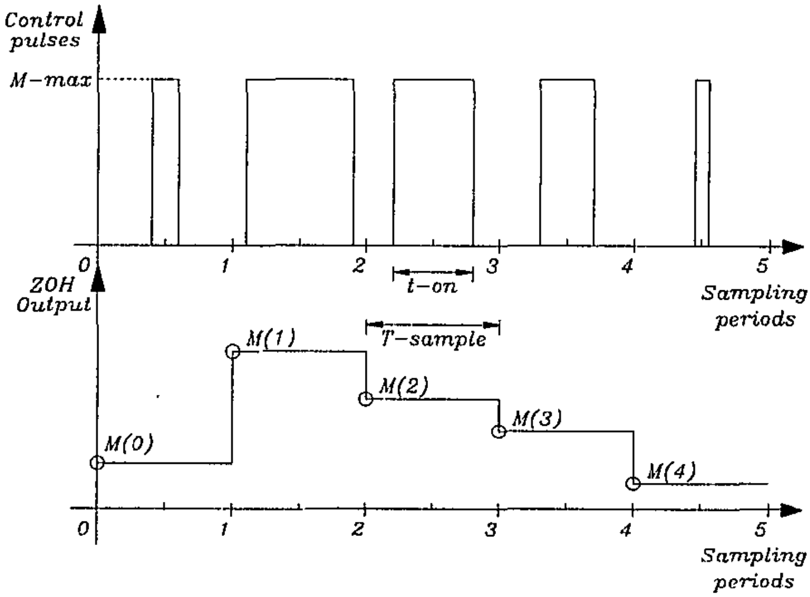

Figure 2.1 PWM of magnetic moment control signal

The switching polarity of each magnetic torquer coil will be determined by the sign of the corresponding magnetic moment. The on-time for coil-*i* will then be determined from,

$$t_{on-i} = T_s \frac{|M_i|}{M_{max-i}} \tag{2.16}$$

where,

T_s = sampling period of the discrete control system
$M_{\max\text{-}i}$ = magnetic moment of magnetic coil-*i* when active

The maximum on-time will be limited to slightly less than the sampling period in practice, to allow for an inactive period between samples to obtain a magnetometer measurement.

The actual magnetic torque vector will be obtained from Equation (2.2). The corresponding portion of the geomagnetic field vector is also centered within the sampling period. Normally the geomagnetic field vector will not change much in magnitude and direction during short sampling periods. However, its corresponding components within the body axes can change significantly during the sampling period for high angular body rates. If control laws such as the cross-product law of Equation (2.6) are used, it will be best to estimate the expected components of the geomagnetic vector at the centre of the sampling period and use these values in the calculations.

If we assume an almost constant body angular rate vector during the sampling period, the Euler rotation axis and angle to the center of the sampling period can be computed at the start of each sampling period,

Euler angle: $$\Phi = \frac{T_s}{2}\sqrt{\omega_{ox}^2 + \omega_{oy}^2 + \omega_{oz}^2} \qquad (2.17a)$$

Euler axis: $$\mathbf{E} = \begin{bmatrix} E_1 \\ E_2 \\ E_3 \end{bmatrix} = \frac{1}{\sqrt{\omega_{ox}^2 + \omega_{oy}^2 + \omega_{oz}^2}}\begin{bmatrix} \omega_{ox} \\ \omega_{oy} \\ \omega_{oz} \end{bmatrix} \qquad (2.17b)$$

The expected geomagnetic body vector can then be computed from the measurement vector,

$$\hat{\mathbf{B}}(kT_s + 0.5T_s) = \mathbf{A}\cdot\mathbf{B}_{meas}(kT_s) \qquad (2.18)$$

where,

$$\mathbf{A} = \begin{bmatrix} \cos\Phi + E_1^2(1-\cos\Phi) & E_1E_2(1-\cos\Phi) + E_3\sin\Phi & E_1E_3(1-\cos\Phi) - E_2\sin\Phi \\ E_1E_2(1-\cos\Phi) - E_3\sin\Phi & \cos\Phi + E_2^2(1-\cos\Phi) & E_2E_3(1-\cos\Phi) + E_1\sin\Phi \\ E_1E_3(1-\cos\Phi) + E_2\sin\Phi & E_2E_3(1-\cos\Phi) - E_1\sin\Phi & \cos\Phi + E_3^2(1-\cos\Phi) \end{bmatrix} \qquad (2.19)$$

2.2.3 SIMULATION RESULTS

A typicai boom deployment sequence for SUNSAT was simulated. The orbit was assumed to be polar, slightly elliptical (e = 0.03) and the orbit period equal to 100 minutes. Initially, before the boom was deployed, the satellite was tumbling at an average, inertially referenced, body angular rate of $\overline{\omega}_B^I = [10 \quad 0 \quad 20]^T$ revolutions per orbit (rpo). The boom extends to its full length within a few seconds. The Y-axis MOI values are, I_{yi} = 2 kgm^2 and $I_{yf} = I_T$ = 40 kgm^2. The diagonal gain matrix as obtained by simulation optimization and used in the cross-product control law of Equation (2.15), is

$$\mathbf{K}_w = diag\begin{bmatrix} \frac{0.2}{\omega_o} & \frac{0.2}{\omega_o} & \frac{0.1}{\omega_o} \end{bmatrix}$$

The active magnetic moment for a magnetic coil is, M_{max} = 20 Am^2 and the maximum on-time is 80% of the sampling period, T_s = 10 seconds. A dipole model has been used to model the geomagnetic field vector.

Figure 2.2 shows the orbit referenced body rate components. The satellite is tumbling at the initial rates until the halfway mark of the first orbit (until 50 minutes). The small oscillation on the rate components is due to a modulation with the orbital rate. At this stage the cross-product controller is enabled and the X and Z-axis angular rates are controlled to zero values (Y-axis aligned to orbit normal). The Y-axis angular rate is controlled to the reference rate of -19 rpo. All this is achieved within half an orbit and at the start of the second orbit (100 minutes) the cross-product controller is disabled and the boom extended. It can be seen that all the orbit referenced angular rate values became close to zero after boom extension, with the result that the satellite is rotating inertially within the orbital frame. Gravity gradient capturing has been achieved and the boom axis is parallel to the nadir vector.

Figure 2.3 shows the pitch angle during the boom deployment sequence. Initially the pitch angle exhibits a clear nutational motion superimposed on a slow orbital rate rotation until the controller is activated. Then the pitch angle begins to linearly sweep from -90° to +90° and then back to -90°. This is due to the definition of the pitch angle in Equation (1.5), only angles for θ in the first and fourth quadrant is possible. The pitch angle is actually indicating a full 360° rotation as it is suppose to do. When the pitch angle decreases from +90° to -90° the boom will be pointing away from the earth and at the instant this angle approaches 0° the boom will be released (approximately at 100 minutes). It can be seen that the transverse MOI increases from 2 kgm^2 to 40 kgm^2, this takes approximately 40 seconds. The pitch angle meanwhile decreased further to about -8° until full deployment is achieved and this is causing a residual pitch libration during the next orbit when no active damping is done. The libration amplitude is not much worse than the normal libration expected due to the ellipticity of the orbit and average aerodynamic disturbance influence (see Chapter 1).

Figure 2.4 presents the polarity status of the X, Y and Z-axis magnetic coils during the active period of the cross-product controller. The X and Z-coil status are displayed with offsets from the nominal +1/0/-1 values to clarify the presentation. It is also important to realize that it is not the PWM pulses that are displayed, but only the polarity status of the coils during the sampling periods.

To summarize: A simple and effective method is proposed to detumble the satellite from any initial angular momentum and prepare the satellite for boom deployment. No accurate attitude knowledge is obligatory, only approximate estimates of the body angular rates are needed, for example, using only the magnetometer measurements. An indication of the instant when the boom mechanism is pointing towards zenith to trigger the boom release, will also be required. This signal can be given by a simple earth detection sensor. Simulations showed that the cross-product law can effectively achieve the right conditions for boom deployment within half an orbit, with small residual librations after gravity gradient capture. This new method outperforms all previous simple control laws, e.g. Equations (2.7-8) and (2.10-12), requiring several orbits to detumble, capture an earth pointing attitude and to damp the librations [Hodgart, 1994].

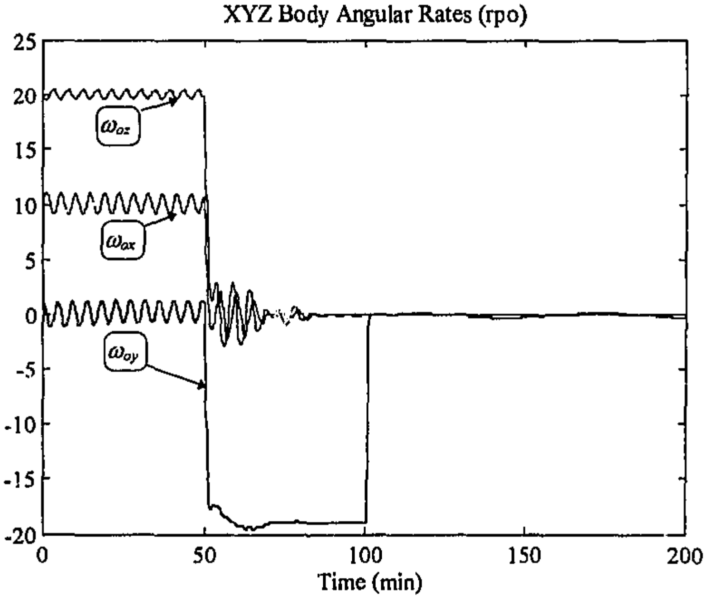

Figure 2.2 Angular rates during the boom deployment sequence

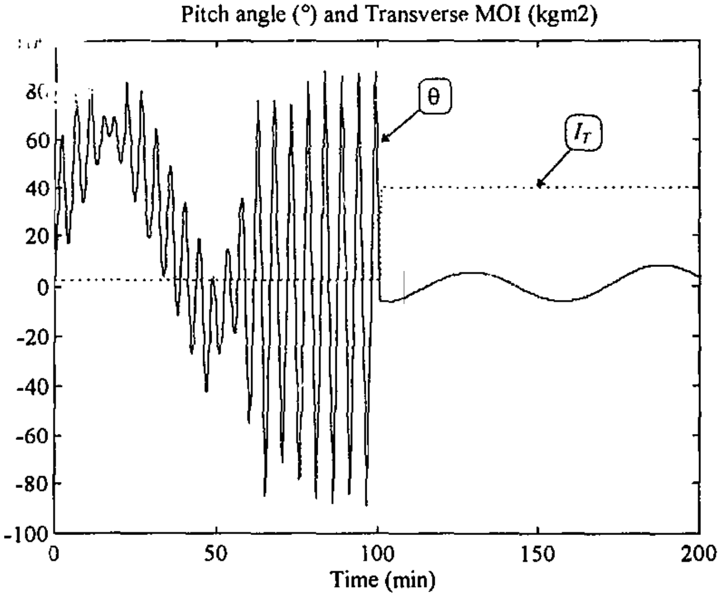

Figure 2.3 Pitch angle and transverse MOI during the boom deployment sequence

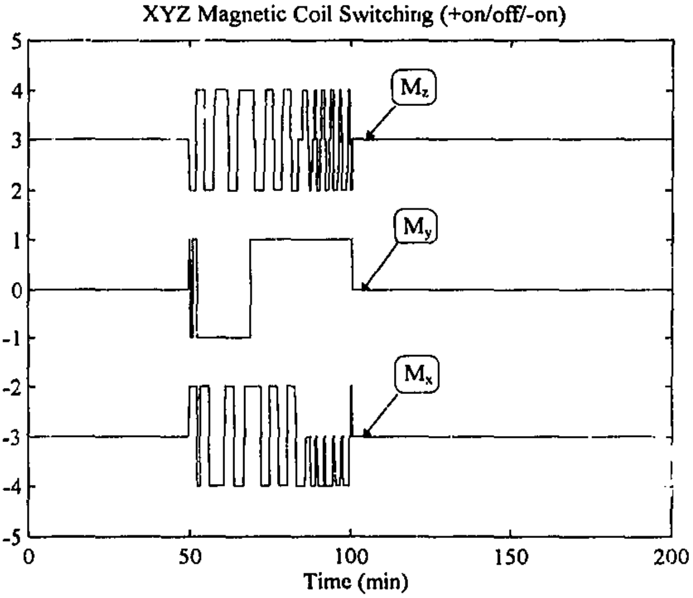

Figure 2.4 Magnetic coil switching polarities pre-boom deployment

2.3 *Libration Damping*

2.3.1 PRELIMINARIES

After boom deployment the sole stabilization purpose of the magnetic torquers will be to damp the external disturbance induced pitch and roll librations and to maintain a reference yaw rate or Z-axis spin. Small librations will ensure an approximate nadir pointing attitude (necessary for imaging and a good antenna radiation pattern). A constant Z-spin will improve the thermal distribution on the satellite's structure and cause a constant roll offset angle, see Equation (1.21). The continuous variation of the geomagnetic field vector will directly influence the control possibilities - according to Equation (2.2) the magnetic torque vector will always be orthogonal to the field vector.

These control restrictions can best be explained by looking at an example: For a polar low earth orbit (LEO), the B_x and B_y components dominates over the equatorial region due to a strong B_{ox} field value. The X- and Y-axis magnetorquers can then be used to do yaw rate control or the Z-axis magnetorquer to damp the pitch librations. Over the polar regions B_z dominates due to the strong B_{oz} component. Pitch and roll libration damping will then be possible using the X- and Y-axis magnetorquers. Limited roll damping over the polar region will also be possible using the Z-axis torquer due to the small B_{oy} field component.

Due to a large I_T/I_z MOI ratio with a deployed boom (typical 20:1 for SUNSAT), the yaw rate control loop will be much more sensitive to applied magnetic torques than damping of the pitch and roll librations. Whenever the X- and Y-axis torquers are used to do libration damping, extreme care must be taken not to disturb the yaw rate. This situation can be avoided by making use mostly of the Z-axis torquer, however, this will restrict the damping of roll librations. The main objective of any efficient magnetic control law will be to make optimal use of the 3-axis magnetic coils, such that while libration damping is taking place the yaw rate is simultaneously maintained at the required set point. All this has to be done in minimum time with the least amount of control energy.

A simple method as already discussed, will be to employ the cross-product law to do angular rate control. The angular rate error vector will be computed similar to Equation (2.15),

$$\mathbf{e} = \mathbf{K}_w\left[\boldsymbol{\omega}_B^O - \boldsymbol{\omega}_{ref}\right] = \mathbf{K}_w\begin{bmatrix} \omega_{ox} \\ \omega_{oy} \\ \omega_{oz} - \omega_{oz-ref} \end{bmatrix} \tag{2.20}$$

This time the orbit reference X- and Y-axis body rates are controlled towards zero (libration damping), while the Z-axis rate is controlled to a certain reference value.

2.3.2 LIBRATION MODEL

Similar to the derivations of Paragraph 1.5.3, a model can be derived, describing the satellite's dynamics during libration damping and Z-spin rate control. In this model we shall assume a yaw spinning satellite in a circular or near circular orbit, with nominally small pitch and roll angular deviations from the nadir direction. Furthermore, the reaction wheels will be deactivated (zero angular momentum) and the external disturbance torques will be small compared to the applied torque from the magnetorquers. Equations (1.9a) to (1.9c) can then be approximated as, ($I_T >> I_{zz}$)

$$\begin{aligned}
\dot{\omega}_x &\approx n_{mx} - 3\omega_o^2(\phi\cos\psi + \theta\sin\psi) + \omega_y\omega_z \\
\dot{\omega}_y &\approx n_{my} - 3\omega_o^2(\theta\cos\psi - \phi\sin\psi) - \omega_x\omega_z \qquad (2.21)\\
\dot{\omega}_z &= n_{mz}
\end{aligned}$$

with,

$$n_{mx} = N_{mx}/I_T, \quad n_{my} = N_{my}/I_T, \quad n_{mz} = N_{mz}/I_{zz}$$

If we transform these equations to the non-rotating body coordinates (the X',Y', and Z axes) the first two expressions can be combined and rewritten as,

$$\begin{aligned}
\dot{\omega}_{x'} &\approx n_{mx'} - 3\omega_o^2\phi + \omega_{y'}(\omega_z - \dot{\psi}) \\
\dot{\omega}_{y'} &\approx n_{my'} - 3\omega_o^2\theta - \omega_{x'}(\omega_z - \dot{\psi})
\end{aligned} \qquad (2.22)$$

where,

$$\omega_{x'} = \omega_x\cos\psi - \omega_y\sin\psi, \qquad \omega_{y'} = \omega_y\cos\psi + \omega_x\sin\psi$$

$$n_{mx'} = n_{mx}\cos\psi - n_{my}\sin\psi, \quad n_{my'} = n_{my}\cos\psi + n_{mx}\sin\psi$$

In terms of the Euler angles, Equation (2.22) and the Z-axis dynamics can be rewritten as,

X'-axis: (near roll)

$$\ddot{\phi} + 4\omega_o^2\phi = n_{mx'} + \omega_o\phi\dot{\theta} \qquad (2.23a)$$

Y'-axis: (pitch)

$$\ddot{\theta} + 3\omega_o^2\theta = n_{my'} + \omega_o\phi\dot{\phi} \qquad (2.23b)$$

Z-axis: (yaw)

$$\ddot{\psi} = n_{mz} - \omega_o\dot{\phi} \qquad (2.23c)$$

where,

$$\omega_{x'} \approx \dot{\phi}, \qquad \omega_{y'} \approx -\omega_o + \dot{\theta}, \qquad \omega_z \approx \omega_o\phi + \dot{\psi}$$

Equations (2.23a) and (2.23b) are typical harmonic oscillators at the small angle libration frequencies of $\omega_\phi = 2\omega_o$ and $\omega_\theta = \sqrt{3}\omega_o$ respectively. A roll angle ϕ, will cause cross-coupling between the two oscillators resulting in a modulation of these two fundamental frequencies. Equation (2.23c) indicates a typical double integrator inertial model for the yaw axis with a small coupling term from the roll rate. The equivalent dynamic models for the rotating X- and Y-axis are much more complicated to derive but one would expect a further modulation with the Z-spin rate ω_z. The dynamic behaviour around the body X- and Y-axis will then be modulated harmonic motion from three fundamental frequencies: $\omega_\phi, \omega_\theta, \omega_z$.

The orbit referenced body angular rates can also be written in terms of the Euler defined angular rates (see figure 1.4):

$$\begin{aligned}
\omega_{ox} &= \dot{\phi}\cos\theta\cos\psi + \dot{\theta}\sin\psi \\
\omega_{oy} &= \dot{\theta}\cos\psi - \dot{\phi}\cos\theta\sin\psi \qquad (2.24) \\
\omega_{oz} &= \dot{\psi} + \dot{\phi}\sin\psi
\end{aligned}$$

From the expressions above it becomes clear why the control law in Equation (2.20) will do libration damping. By zeroing ω_{ox} and ω_{oy} , the libration roll and pitch rates, $\dot{\phi}$ and $\dot{\theta}$, will automatically be zeroed. Furthermore, controlling ω_{oz} to a certain reference rate will automatically ensure the correct yaw rate $\dot{\psi}$.

2.3.3 FUZZY CONTROLLER STRUCTURE

In an attempt to improve the performance of the simple cross-product controller for libration damping and yaw rate control, an alternative controller based on fuzzy logic has been designed. To allow for the choice of the magnetorquer coil (or combination thereof) that will achieve the best results, given the local geomagnetic field vector, the following well structured algorithm based on fuzzy control rules was designed. Fuzzy logic is defined by Zadeh [1984] as: "A kind of logic using graded or qualified statements rather than ones that are strictly true or false. The results of fuzzy reasoning are not as definite as those derived by strict logic, but they cover a larger field of discourse".

The intention of the fuzzy controller design was to define a set of control rules and to implement them in such a way as to make the boundaries between them less strict, resulting in a more flexible system. A variation of the multi-input and single output (MISO) fuzzy controller of Sugeno and Nishida [1985] was implemented. A block diagram of the proposed fuzzy controlled system is shown in Figure 2.5. The controller actually consists of three MISO fuzzy control laws, one for each orthogonal magnetic torquer. Each control law embodies a fuzzy rule base to decide on the control desirability and output level when using the corresponding torquer. A choice is then made to use the most favourable torquer during the next control interval.

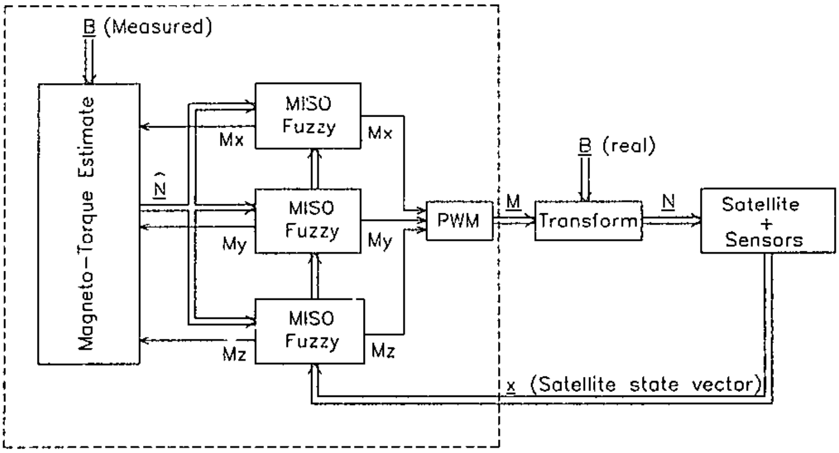

Figure 2.5 Block diagram of the full fuzzy controller

The input variables for the fuzzy controllers are the measured angular rate values from the attitude state vector of the satellite and the estimated magnetic control torques. This choice of input variables will make it possible to regulate the state variables while considering the control torque constraints (e.g. availability and cross disturbances). The torques can be estimated using Equations (2.2) and (2.18) and the magnetometer measurements. A total of six fuzzy input variables were used:

x_1 = orbit referenced X-axis angular rate: ω_{ox}

x_2 = orbit referenced Y-axis angular rate: ω_{oy}

x_3 = orbit referenced Z-axis angular rate error: $\omega_{oz} - \omega_{oz-ref}$

x_4 = estimated X-axis geomagnetic torque with a specific magnetorquer: $\hat{N}_{mx}$

x_5 = estimated Y-axis geomagnetic torque with a specific magnetorquer: $\hat{N}_{my}$

x_6 = estimated Z-axis geomagnetic torque with a specific magnetorquer: $\hat{N}_{mz}$

These variables are then mapped into fuzzy sets (e.g. ***P*** for positive, ***N*** for negative and ***Z*** for zero). The fuzzy set values are obtained from membership functions, e.g.:

$$x_1 \rightarrow m_P(x_1) \quad \text{and} \quad y_4 \rightarrow m_N(x_4)$$

The membership functions used for each input variable are shown in Figure 2.6. The reasons for choosing the functions in this specific format were multiple, most importantly to limit the number of fuzzy sets, but still obtain a linear mapping in the normal operating region of the system. The amount of overlap between the different fuzzy sets was optimized through simulation and its influence analysed (see Figures 2.7 and 2.8). The saturation point (scaling) of each input variable was done using an engineering knowledge of the system and further optimization was done through simulation trials. All inputs are grouped into two fuzzy sets: positive ***P*** and negative ***N***. The yaw control torque (Z-axis) is supplemented with an extra set: zero ***Z***. This set is used in the rules to minimize the disturbance to the sensitive yaw control loop.

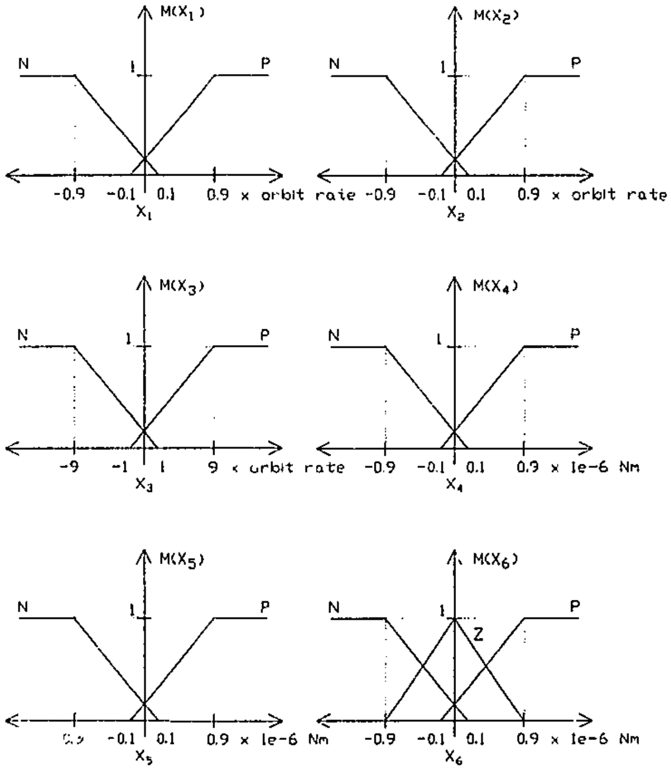

Figure 2.6 Fuzzy variable membership functions

A fuzzy rule base was constructed from a few simple linguistic rules. Each rule maps to a crisp binary output variable (+1 or -1). The output variable indicates the desired magnetorquer polarity. The M_x fuzzy controller is used to control the Y- and Z-axis angular rates, the M_y controller to control the X- and Z-axis angular rates and the M_z controller the X- and Y-axis angular rates. A set of 8 rules each, listed in Tables 2.1 to

2.3 were defined for these controllers. As an example of these rules, take rule R^2 from Table 2.1:

R^2: IF $x_2 = P$ AND $x_5 = N$ AND $x_6 = Z$ THEN $u = +1$

If the Y-axis angular rate is positive and the estimated Y-axis torque, obtainable from the M_x torquer coil (computed as if switched on positively), is negative, and the Z-axis torque (disturbance to the Z-axis) is close to zero, then the torquer polarity must stay positive.

Table 2.1 Rule table for M_x fuzzy controller

	x_2	x_3	x_5	x_6	u
R^1	***P***	-	***P***	***Z***	-1
R^2	***P***	-	***N***	***Z***	+1
R^3	***N***	-	***P***	***Z***	+1
R^4	***N***	-	***N***	***Z***	-1
R^5	-	***P***	-	***P***	-1
R^6	-	***P***	-	***N***	+1
R^7	-	***N***	-	***P***	+1
R^8	-	***N***	-	***N***	-1

Table 2.2 Rule table for M_y fuzzy controller

	x_1	x_3	x_4	x_6	u
R^1	***P***	-	***P***	***Z***	-1
R^2	***P***	-	***N***	***Z***	+1
R^3	***N***	-	***P***	***Z***	+1
R^4	***N***	-	***N***	***Z***	-1
R^5	-	***P***	-	***P***	-1
R^6	-	***P***	-	***N***	+1
R^7	-	***N***	-	***P***	+1
R^8	-	***N***	-	***N***	-1

Table 2.3 Rule table for M_z fuzzy controller

	x_1	x_2	x_4	x_5	u
R^1	***P***	-	***P***	-	-1
R^2	***P***	-	***N***	-	+1
R^3	***N***	-	***P***	-	+1
R^4	***N***	-	***N***	-	-1
R^5	-	***P***	-	***P***	-1
R^6	-	***P***	-	***N***	+1
R^7	-	***N***	-	***P***	+1
R^8	-	***N***	-	***N***	-1

The rule consequent (truth value) is then inferred using correlation-product encoding - the conjunctive (AND) combination of the antecedent fuzzy sets. For rule 2:

$$\text{R}^2\text{:} \qquad \mu\left(\text{R}^2\right) = m_P\left(x_2\right)\cdot m_N\left(x_5\right)\cdot m_Z\left(x_6\right)$$

The correlation-product norm is used rather than the more common correlation-minimum norm [Kosko, 1992], to enable all rule conditions to contribute to the rule consequent. These truth values are then used to scale the binary output to obtain the rule output:

$$\text{R}^2\text{:} \qquad y\left(\text{R}^2\right) = \mu\left(\text{R}^2\right)\cdot u\left(\text{R}^2\right)$$

All the rule outputs are then combined disjunctively (OR) to obtain the crisp rule base output:

$$y_{out} = \bigcup y\left(\text{R}^i\right) = \text{sgn}\left\{\sum_{i=1}^{8} y\left(\text{R}^i\right)\right\}\cdot \min\left\{1, \left|\sum_{i=1}^{8} y\left(\text{R}^i\right)\right|\right\} \qquad (2.25)$$

The disjunction method of Equation (2.25) can be described as a kind of signed Lukasiewicz OR logic. It is chosen to maximally negatively correlate the rule outputs. For example, opposing rule outputs (different in sign) can cancel one another to deliver a small rule base output, thus minimizing the level of cross disturbance.

The fuzzy controller with the largest absolute rule base output is the preferred one to use in controlling the angular rate error. This will be true for the case where only a single magnetorquer coil can be utilized at a time due to power limitations. When this practical constraint is not applicable, all three outputs can be applied simultaneously to the full orthogonal magnetorquer set. The control amplitude (the duration of the magnetorquing pulse) and sign (switching polarity) are also obtainable from the output values. These output values are directly related to the angular rate errors and the control torque availability through the specific choice of the membership functions. The disturbance to the sensitive yaw control dynamics ($I_T << I_{zz}$) is limited by the fuzzy set $\boldsymbol{Z}$ when the M_x and M_y rules are evaluated. Further conflicts are implicitly resolved through the summation in Equation (2.25).

To summarize, the practical computing steps to be performed every sampling period are:

1. Obtain the magnetometer measurement of the geomagnetic field vector $\mathbf{B}_{meas}$.
2. Compute the angular body rate errors of the satellite.
3. Estimate the magnetic torque components as the X-axis torquer is switched on positively ($\hat{\underline{\text{N}}}$ symbol of Figure 2.5) - use Equation (2.18) to obtain the estimated field vector in the center of the sampling period.

4. Map all the fuzzy variables into their fuzzy sets (see Figure 2.6).
5. Evaluate the rules for M_x, using Table 2.1, and combine all the rule outputs using Equation (2.25).
6. Repeat steps 3 to 5 for M_y, using Table 2.2.
7. Repeat steps 3 to 5 for M_z, using Table 2.3.
8. (Optional step if only one magnetorquer can be utilized at a time) Compare the absolute output values of the three fuzzy controllers (for M_x, M_y and M_z) and choose the magnetorquer with the largest output.
9. Activate the magnetorquer(s) around the center of the sampling period. The polarity (sign) and on-time (from Equation (2.16)) of each the magnetorquing pulse are automatically available from the fuzzy controller output.

The optimality and performance (e.g. the control response time) of the fuzzy controller are therefore solely dependent on the choice of the membership functions. Sound engineering judgement and simulation trials were used to achieve these goals. An example of a typical engineering choice is: The M_x and M_y torquers are more restricted in their rule evaluation compared to the M_z torquer, because they must limit the disturbance to the sensitive Z-axis during libration damping. The result will lead to an inherent preference in applying the M_z magnetorquer. This can, however, be to the detriment of overall control efficiency. It was found that improved performance could be obtained by scaling down the output from the M_z torquer, such that a more even utilization of the three magnetic torquers is obtained.

2.3.4 FUZZY CONTROLLER ANALYSIS

The stability and non-linearity of the fuzzy controllers can be analysed by transforming the linguistic rule base and membership functions to algebraic equations. From Equations (2.23a), (2.23b) and (2.24) it follows that the open loop linearised model for the roll and pitch libration will be oscillatory and the body X- and Y-axis angular rates will reflect these oscillations via a transformation by the yaw angle. The pitch and roll librations will be excited due to gravity gradient disturbances from the non-elliptical orbit and aerodynamic torque variations as shown in Chapter 1. The purpose of a stable control law will be to damp these oscillations using the magnetorquers. If the resulting X- and Y-axis magnetic torques can be proportional, but opposite in sign to the corresponding body angular rates, the necessary damping will take place. For example, when:

$$N_{mx} = -k_x x_1 = -k_x \omega_{ox}, \qquad N_{my} = -k_y x_2 = -k_y \omega_{oy}$$

where,

$$k_x, k_y > 0$$

then transforming to the non-rotating X'- and Y'-axis (near roll and pitch) through the yaw angle ψ, the magnetic torque around these axis can be approximated as,

$$N_{mx'} \approx -(k_x + k_y)\dot{\phi} - (k_x + k_y)\dot{\theta}\sin\psi\cos\psi$$
$$N_{my'} \approx -(k_x + k_y)\dot{\theta} - (k_x + k_y)\dot{\phi}\sin\psi\cos\psi$$

If these expressions are now included in Equations (2.23a) and (2.23b), they become:

X'-axis: (near roll)

$$\ddot{\phi} + \frac{k_x + k_y}{I_T}\dot{\phi} + 4\omega_o^2\phi = \dot{\theta}\left(\omega_o\phi - \frac{k_x - k_y}{I_T}\sin\psi\cos\psi\right) \qquad (2.26a)$$

Y'-axis: (pitch)

$$\ddot{\theta} + \frac{k_x + k_y}{I_T}\dot{\theta} + 4\omega_o^2\theta = \dot{\phi}\left(\omega_o\phi - \frac{k_x - k_y}{I_T}\sin\psi\cos\psi\right) \qquad (2.26b)$$

Equations (2.26a) and (2.26b) are typical damped oscillators with limited cross-coupling between them. Depending on the magnitude of the positive gain values k_x and k_y, the linearized closed loop pitch and roll libration model will either have real or complex, but always stable, closed loop eigenvalues. To ensure positive damping, the control law must deliver a control torque opposing the angular rate error. This will be true whenever the product of a scaled version of the angular rate error and the estimated magnetic torque around a specific body axis, gives a negative result.

For the fuzzy controller, positive damping will be accomplished by using the membership functions (scaling), rule base (pairing of the error rates and the torques) and the correlation-product conjunction of the rules. The torques are estimated as if a specific magnetorquer is switched on positively, through the computed evaluation of Equation (2.18). If the rate/torque product is positive, the magnetorquer polarity needs to be reversed, else it was chosen correctly. It can easily be shown that without any overlap between the *P* and *N* fuzzy sets and within the linear mapping range $(x_i \rightarrow m_{P/N}(x_i) < 1)$, the fuzzy controllers simplify to the following algebraic equations:

$$M_x = -k_x(\chi_2\chi_5 \cdot \xi + \chi_3\chi_6)$$
$$M_y = -k_y(\chi_1\chi_4 \cdot \xi + \chi_3\chi_6) \qquad (2.27)$$
$$M_z = -k_z(\chi_2\chi_5 + \chi_1\chi_4)$$

where,

k_x,k_y,k_z = scaling factors for X-, Y- and Z-magnetorquers
χ_i = normalized versions of the input variables
$\chi_1\chi_4,\ \chi_2\chi_5,\ \chi_3\chi_6$ = product pairs for the X-, Y- and Z-axis control

$$\begin{aligned} \xi &= 1-|\chi_6| && \text{when } |\chi_6| < 1.0 \\ &= 0 && \text{when } |\chi_6| > 1.0 \end{aligned}$$

= weighting factor to ensure small disturbances on the Z-axis

The dominant product pair (best axis to control) will be the major contributing term in Equation (2.27). It will therefore determine the correct sign and output level for each magnetic torquer. Furthermore, the best (highest output level) magnetic torquer can be selected from the three possibilities if a single torquer at a time is utilized. All this will ensure a dominant stable controlling effort with minimization of any possible cross-disturbances.

Libration damping is not required around the spacecraft's Z-axis but rather a stable feedback control law to regulate a reference yaw angular rate. If the Z-axis magnetic torque is proportional but opposite in sign to the yaw error angular rate,

$$N_{oz} = -k_z\, x_3 = -k_z\left(\omega_{oz} - \omega_{oz\ ref}\right)$$

with,

$$k_z > 0$$

Then from Equations (2.23c) and (2.24) the Z-axis dynamics can be approximated as,

$$\ddot{\psi} + \frac{k_z}{I_{zz}}\dot{\psi} \approx \frac{k_z}{I_{zz}}\omega_{oz-ref} \qquad (2.28)$$

This expression indicates a typical first order exponential tracking of the reference rate. The sign of the Z-axis (yaw) product pair will define the magnetorquer polarity in a similar way as in the previous X- and Y-axis cases. The closed loop eigenvalues of the Z-axis dynamics will then be stable and the constant k_z / I_{zz} will determine the time-constant of the closed loop response.

When the ***P*** and ***N*** fuzzy sets overlap, the scaled versions of the input variables can be rewritten as: (e.g. for a 10% overlap)

P: $m_P(\chi_i) = \chi_i + 0.1$ $\{-0.1 < \chi_i < 0.9\}$
N: $m_N(\chi_i) = -\chi_i + 0.1$ $\{-0.9 < \chi_i < 0.1\}$
and,
$m_{P/N}(\chi_i) = 1$ $\{|\chi_i| \geq 0.9\}$

Within the linear mapping range ($m_{P/N}(\chi_i) < 1$) the fuzzy controllers simplify to similar algebraic equations as in Equation (2.27). However, the product pairs $\chi_1\chi_4$, $\chi_2\chi_5$ and $\chi_3\chi_6$ now become different expressions in the scaled variables, depending on whether the input variables map inside or outside of the overlapping region. The expression to substitute the X-axis product pair is:

$$\chi_{14} = \begin{cases} \chi_1\chi_4 + 0.1\cdot\chi_1\cdot\mathrm{sgn}(\chi_1) + 0.1\cdot\chi_4\cdot\mathrm{sgn}(\chi_4) - 0.01 & \forall\ |\chi_1| \ge 0.1,\ \ |\chi_4| \ge 0.1 \\ 2\chi_1\chi_4 + 0.2\cdot\chi_4\cdot\mathrm{sgn}(\chi_1) & \forall\ |\chi_1| \ge 0.1,\ \ |\chi_4| < 0.1 \\ 2\chi_1\chi_4 + 0.2\cdot\chi_1\cdot\mathrm{sgn}(\chi_4) & \forall\ |\chi_1| < 0.1,\ \ |\chi_4| \ge 0.1 \\ 4\chi_1\chi_4 & \forall\ |\chi_1| < 0.1,\ \ |\chi_4| < 0.1 \end{cases}$$

This expression can easily be derived from the four rules pertaining the X-axis input variables, for example, rules 1 to 4 in Table 2.3. Rules 1 to 4 in Table 2.2 give a similar result with the added multiplication of ξ, the Z-axis cross disturbance constraint. Similar expressions can also be obtained to substitute the Y- and Z-axis product pairs.

The difference in control effort between the usage of non-overlapping and overlapping fuzzy sets, is illustrated clearly in the 3-D plots of figures 2.7 and 2.8. The control surface contribution of the X-axis product pair in the positive-positive quadrant (both input variables are positive) is shown The negative-negative quadrant control surface looks similar, while the positive-negative and negative-positive control surfaces have a negative output result

What is important to notice is that overlapping not only preserves the correct product pair sign (important for stability), but also increases the output (magnetic torque level) for small input values This tends to improve the sensitivity and response of the control system to small error signals, without unduly complicating the controller implementation.

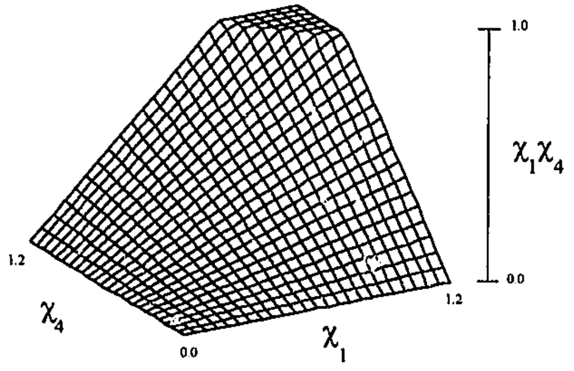

Figure 2.7 Positive-positive control surface for the non-overlapping controller

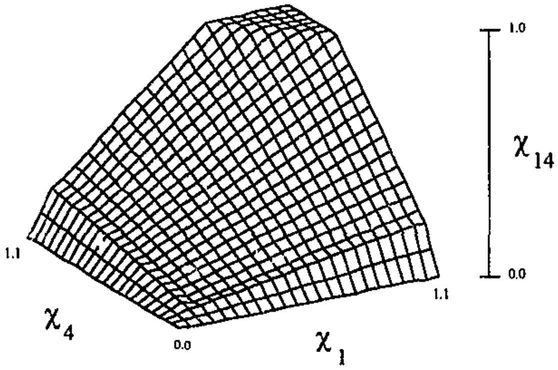

Figure 2.8 Positive-positive control surface for the overlapped controller

2.3.5 SIMULATION RESULTS

To evaluate and compare the libration damping and Z-spin regulation performance of the two controller types proposed (cross-product and fuzzy), various simulation trails were done. The SUNSAT satellite and orbit configuration were assumed, see Paragraph 2.2.3. The sampling period used by the control laws during libration damping and Z-spin regulation, was T_s = 60 seconds, i.e. 100 samples per orbit. The maximum on-time was taken at 80% of the full sampling period. Perfect knowledge of the body angular rate values were assumed. Very accurate angular rate values can be obtained in orbit from Kalman filtered sensor measurements, see Chapter 5.

The diagonal gain matrix used in the cross-product control law of Equation (2.20) is,

$$\mathbf{K}_w = diag\begin{bmatrix}\frac{0.1}{\omega_o} & \frac{0.1}{\omega_o} & \frac{0.05}{\omega_o}\end{bmatrix}$$

This gain matrix was optimized for the best performance (minimum libration damping time and power consumption) and chosen after many simulation trails.

The outputs of the three MISO fuzzy controllers are also scaled to obtain the desired magnetorquing level (similar to Equation 2.27):

$$\begin{bmatrix}M_x & M_y & M_z\end{bmatrix}_{\text{fractional on-time}} = \begin{bmatrix}0.3y_{out-x} & 0.3y_{out-y} & 0.03y_{out-z}\end{bmatrix}$$

These values were optimized for the best performance and chosen after many simulation tests.

Three variations of the fuzzy controller were tested and compared to the cross-product controller. The first variation, called the overlap fuzzy controller, makes use of 10% overlapping of the *P* and *N* fuzzy sets, similar to Figure 2.6 and applies only the magnetorquer with the highest output value during each sampling period. The second variation, called the MO (multi-output) fuzzy controller, makes use of 0% overlapping between *P* and *N*, and applies all three magnetorquers per sampling period with on-times corresponding to their respective output values. The third variation, called the SO (single-output) fuzzy controller uses 0% overlapping and a single magnetorquer per sampling period.

2.3.5.1 Disturbance damping performance

During normal nadir pointing flight conditions most of the libration energy will be removed and the satellite will have a slow and constant Z-spin. External disturbances, mainly the cyclic gravity gradient disturbance due to the slightly elliptical orbit (Paragraph 1.6.1) and aerodynamic disturbance torque due to air density variations (Paragraph 1.6.2), will induce a modulated pitch libration. If the satellite also has a Z-spin, the pitch libration will be coupled to the roll dynamics as well.
A typical example of these disturbance induced librations is presented in Figures 2.9 and 2.10. Average solar conditions and an orbit eccentricity of 0.03 were used. Furthermore, the satellite is spinning at a constant Z-axis angular rate of 5 rpo (revolutions per orbit). The roll bias angle from Equation (1.21) will be 3.78° on average. During the initial 5 orbits magnetic control does not occur and the net effect of the disturbances is a pitch and roll libration with maximum peak-to-peak amplitude of 10° and 4° respectively. During the next 5 orbits active magnetic control manages to damp these librations to 4° for pitch and 2° for roll.

Table 2.4 compares the different controllers with respect to the total on-time over the 5 orbits, this gives an indication of the power effectiveness of the various algorithms. The fuzzy controller, using overlapping of the *P* and *N* fuzzy sets, is the most power inefficient. The non-overlapped fuzzy controller using a single magnetorquer per sampling period performs the best. The cross-product controller consumes almost twice as much energy as the best fuzzy controller.

Figure 2.9 shows the simulation results obtained from the cross-product controller and Figure 2.10 from the SO fuzzy controller. As can be seen from these presentations, the SO fuzzy controller has less disturbance on the Z-spin rate and a smaller residual roll libration. The M_x, M_y and M_z magnetorquer activity are also presented on the graphs.

Table 2.4 Controller comparison for disturbance damping
(On-time in seconds over 5 orbits)

	Cross-product	*Overlap Fuzzy*	*MO Fuzzy*	*SO Fuzzy*
X-torquer	27.6	33.7	17.8	17.1
Y-torquer	26.5	37.2	20.4	19.5
Z-torquer	30.4	22.6	11.0	8.3
Total	84.5	93.5	49.2	44.9

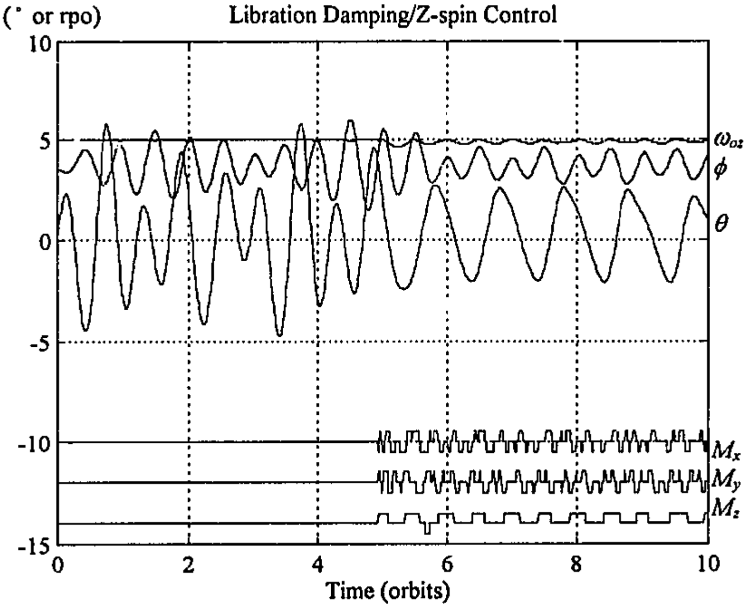

Figure 2.9 Disturbance damping using the cross-product controller

2.3.5.2 Libration damping/Spin rate control performance

In the next few simulations the performance of the various controllers were evaluated when doing libration damping from an initial pitch and roll nadir pointing error of 10° and -5° respectively. The initial Z-spin rate was 5 rpo and commanded at the end of the third orbit to 10 rpo. The magnetic control commenced at the end of the first orbit.

Table 2.5 compares the total on-time of the various controllers over the 5 controlled orbits. The SO fuzzy controller consume: the least amount of power (lowest on-time) and the overlap fuzzy the most, with the cross-product method somewhere in between.

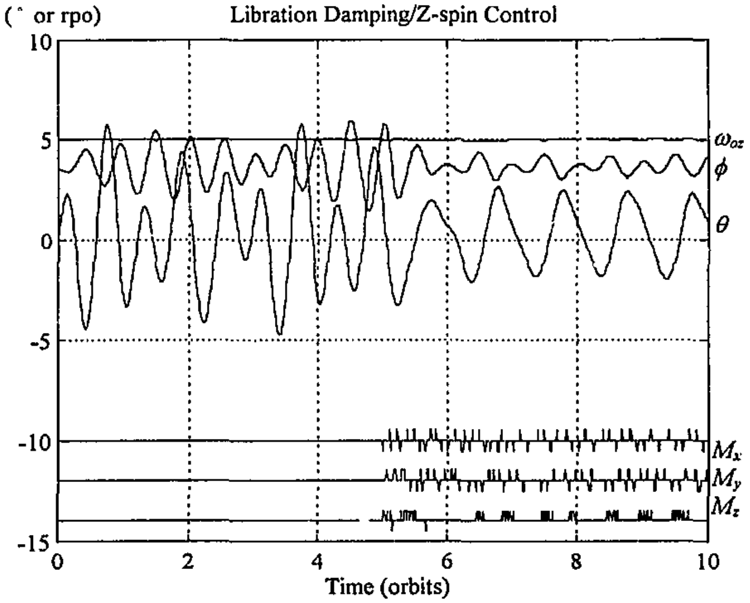

Figure 2.10 Disturbance damping using the SO fuzzy controller

Table 2.5 Controller comparison for libration damping/spin rate control (Elliptical orbit: $e = 0.03$, On-time in seconds)

	Cross-product	*Overlap Fuzzy*	*MO Fuzzy*	*SO Fuzzy*
X-torquer	61.8	95.4	60.0	63.6
Y-torquer	64.7	69.4	53.2	48.9
Z-torquer	31.9	35.8	27.0	17.1
Total	158.4	200.6	139.2	129.6

Figures 2.11 and 2.12 show the pitch, roll and Z-spin characteristics of the cross-product and the SO fuzzy controllers respectively. An elliptical orbit (e = 0.03) was used and aerodynamic disturbances (as expected during normal solar activity) implemented. The cross-product controller causes more disturbance to the Z-spin when libration damping is done. The change in Z-spin is done slightly faster by the cross-product controller. The residual pitch and roll librations during the end of the simulation is not different to those seen in Figures 2.9 and 2.10. The only difference is the expected increased roll bias to 7.56°, due to the doubling in Z-spin to 10 rpo.

Table 2.6 presents the total on-time of a simulation similar to the previous one, but this time the influence of the external disturbances was reduced by assuming a circular orbit. The SO fuzzy controller again consumes the least amount of power and the overlap fuzzy controller became much more efficient. The cross-product controller performed the worst.

Table 2.6 Controller comparison for libration damping/spin rate control (Circular orbit, On-time in seconds)

	Cross-product	*Overlap Fuzzy*	*MO Fuzzy*	*SO Fuzzy*
X-torquer	56.5	53.4	48.0	50.9
Y-torquer	45.0	40.8	48.8	32.2
Z-torquer	24.6	21.0	21.5	10.9
Total	126.1	115.2	118.3	94.0

Figures 2.13 and 2.14 show the comparative behaviour in a circular orbit of the worst (cross-product) and best (overlap fuzzy) controllers. It is clear that both the libration damping and Z-spin tracking of the overlap fuzzy controller outperforms that of the cross-product method. The performance of the SO fuzzy controller (most power efficient) is somewhere in between those of Figures 2.13 and 2.14.

2.3.5.3 Interpretation of results

After numerous simulation runs, it was concluded that the SO fuzzy controller gives the best overall performance during libration damping and Z-spin control, whenever the level of external disturbances is high. The fuzzy controller with overlapping of its ***P*** and ***N*** membership functions is more sensitive to small angular rate errors as can be seen from Figure 2.8. More energy will therefore be wasted while trying to counter the periodic disturbance torques. However, for circular orbits it was found [Steyn, 1994], as also demonstrated here (Figure 2.14), that the overlap fuzzy controller achieves the best angular rate damping and tracking performance.

The fuzzy controller which uses multiple magnetorquers (MO fuzzy) during each sampling period did not damp the librations better than the single magnetorquer case (SO fuzzy). Furthermore the power consumption indicates that the single magnetorquer case must be prefered Experimentation with the sampling period indicates that a sampling period faster the 60 seconds does not give any performance benefits. Longer periods than 60 seconds do however start to impair the performance of the various controllers.

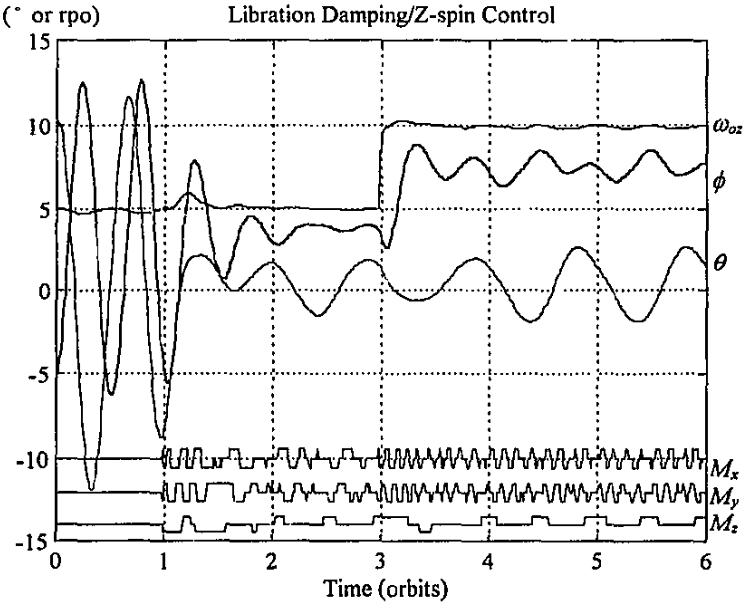

Figure 2.11 Cross-product - libration damping/Z-spin control (e = 0.03)

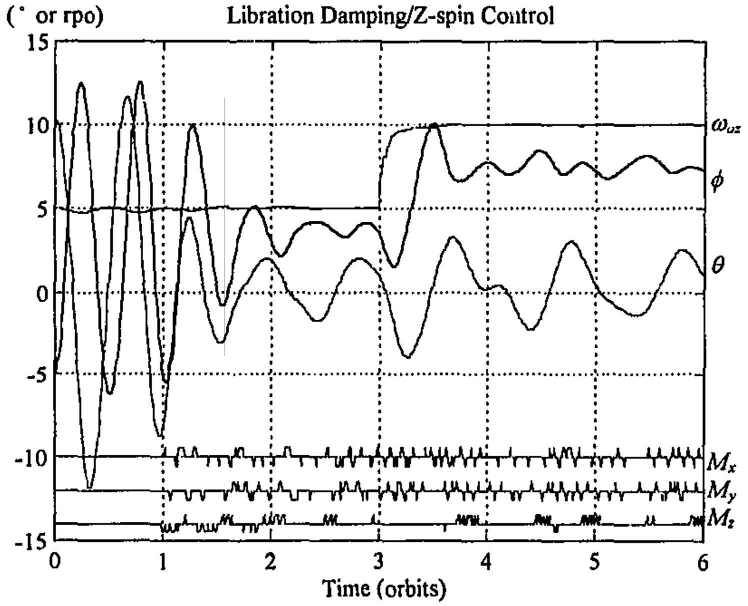

Figure 2.12 SO fuzzy - libration damping/Z-spin control (e = 0.03)

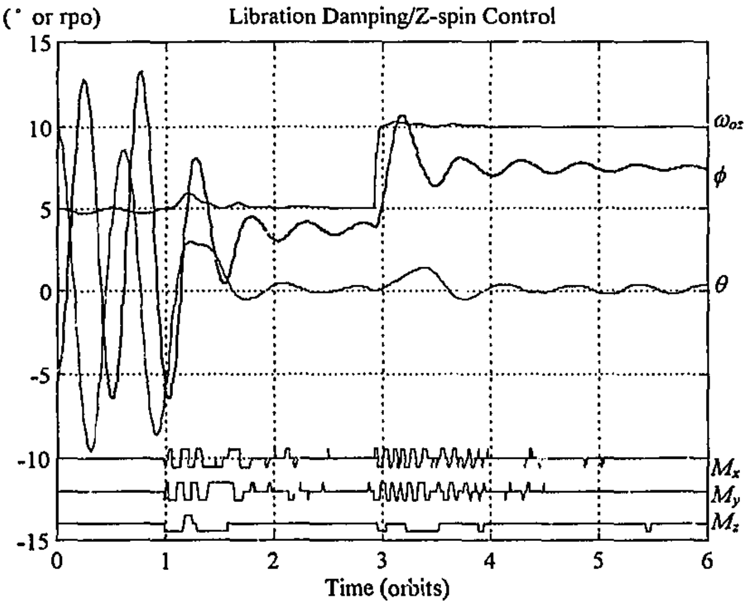

Figure 2.13 Cross-product - libration damping/Z-spin control (e = 0)

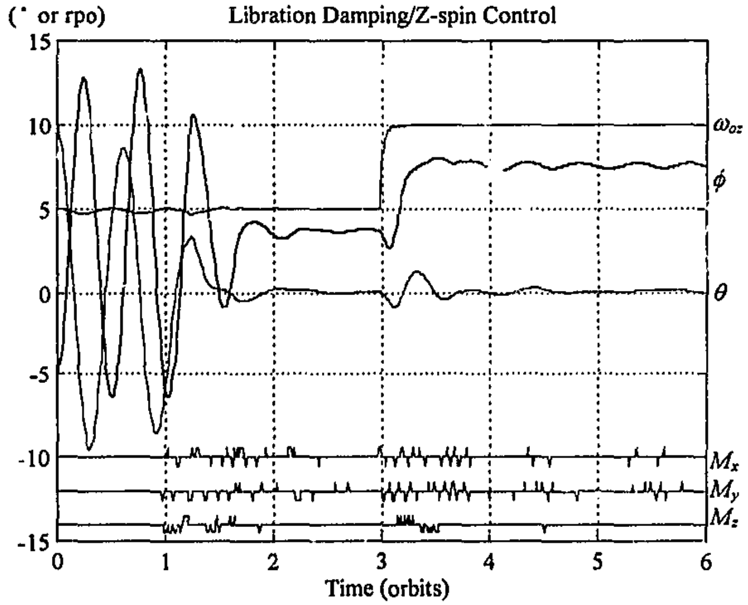

Figure 2.14 Overlap fuzzy - libration damping/Z-spin control (e = 0)

2.4 Conclusions

Magnetic torquer coils are reliable and energy efficient actuators to counter small external disturbance torques during normal flight, and to initially detumble and despin a micro satellite. Magnetic torquing, however, imposes some control constraints making the implementation of traditional linear control theory not feasible. The main constraints to account for in any control law design are,

- Control torque unavailability during certain orbital regions
- Cross disturbance torques between the spacecraft axes

In this chapter a new algorithm is proposed, based on the cross-product law [Stickler, 1974], to simultaneously detumble and prepare a gravity gradient satellite for boom deployment. The algorithm only needs approximate values (obtained from magnetometer measurements alone) of the angular body rates of the spacecraft. Simulation results showed detumbling and boom deployment within a single orbit from any reasonable initial angular momentum. The residual pitch and roll libration directly after boom deployment were always less than 20° in peak-to-peak amplitude. The results obtained during simulations of the newly proposed algorithm strongly suggests superior performance to be realised in a practical system compared to both the methods proposed by Martel *et.al.* [1988] and Hodgart *et.al.* [1994].

Two types of controllers were presented for libration damping and Z-spin rate regulation: A variant of the Martel *et.al.* [1988] cross-product controller and a newly proposed controller based on fuzzy design principles. The fuzzy controller achieved the best control energy and Z-spin tracking results with comparable libration damping performance to the cross-product method. An nearly 100% improvement in control energy consumption was obtained during simulation when the satellite was only disturbed by the cyclic type of external torques.

Proving the stability of some fuzzy controllers can sometimes be difficult if not impossible. But for this relatively simple implementation it was possible to reduce the fuzzy controller to a few algebraic equations and prove the stability through inspection of the damping term. An implementation difference between the fuzzy and the cross-product controller worth pointing out is: The fuzzy controller can select only one magnetorquer to be pulsed during each sampling interval, the cross-product controller normally uses all 3 magnetorquers and the pulses have to be interleaved if it is impossible to be applied simultaneously, due to power limitations.

The fuzzy controller with non-overlapping of its membership functions seems to perform the best, when cyclic disturbances (e.g. due to a slight elliptical orbit) will render it impossible to damp the librations completely. This fact is mainly due to the unavailability of magnetic control torque during certain regions of an orbit. The fuzzy controller with overlapping and more sensiti·ity for small libration errors, performs the best when the in..uence of external cyclic disturbances is small (e.g. in a circular orbit).

For the typical SUNSAT implementation, simulations showed that both the controller types can achieve libration damping with maximum peak-to-peak errors of 4° and 2° in pitch and roll respectively. The Z-spin rate can also easily track a reference rate and can be commanded to follow a new setpoint within a few minutes. No controller instability has ever occured during simulations and libration damping was successfully demonstrated from any initial attitude error.

2.5 References

Baron W.R., Larson M.H. and Thomsen P.L. [1994]
"The Danish Ørsted Satellite", *Proceedings of the 6th AIAA/USU Conference on Small Satellites*, Utah State University, Sept.1992.

Chang D.H. [1992]
"Magnetic and Momentum Bias Attitude Control Design for the HETE Small Satellite", *Proceedings of the 6th AIAA/USU Conference on Small Satellites*, Utah State University, Sept.1992.

Ergin E.I. and Wheeler P.C. [1965]
"Magnetic Attitude Control of a Spinning Satellite", *Journal of Spacecraft*, Vol.2, No.6, 1965, pp.846-851.

Franklin G.F., Powell J.D. and Workman M.L. [1990]
Digital Control of Dynamic Systems, Second Edition, Addison-Wesley Publishing Company, 1990.

Goel P.S. and Rajaram S. [1979]
"Apple Magnetic Attitude Control (AMAC) - A New Attitude Control Concept", *Proceedings of the 8th IFAC Symposium on Automatic Control in Space*, 2-6 July 1979, pp201-208.

Hodgart M.S. and Wright P.S. [1987]
"Attitude Determination, Control and Stabilization of UoSAT-2", *Journal IERE*, Vol.57, No.5, Sept/Oct.1987, Supplement pp.151-162.

Hodgart M.S. [1989]
"Gravity Gradient and Magnetorquing Attitude Control for Low-Cost Low Earth Orbit Satellites - The UoSAT Experience", *Ph.D. Submission at University of Surrey*, June 1989.

Hodgart M.S. and Ong W.T. [1994]
"Attitude Determination and Control for Low Earth Orbit Microsatellites", *Proceedings IEEE/SAIEE 1994 Symposium on Small Satellites and Control*, University of Stellenbosch, 4 Oct. 1994.

Kosko B. [1992]
Neural Networks and Fuzzy Systems, Prentice-Hall, Englewood Cliffs, NJ, 1992, pp.299-337.

Martel F., Pal P.K. and Psiaki M. [1988]
"Active Magnetic Control System for Gravity Gradient Stabilized Spacecraft", *Proceedings of the 2th AIAA/USU Conference on Small Satellites*, Utah State University, Sept. 1988.

Pal P.K. and Fox S.M. [1992]
"The Effects of Momentum Bias on a Gravity Gradient Stabilized Spacecraft with Active Magnetic Control", *Proceedings of the 6th AIAA/USU Conference on Small Satellites*, Utah State University, Sept. 1992.

Parkinson B.W. and Kasdin N.J. [1990]
"A Magnetic Attitude Control System for Precision Pointing of the Rolling GP-B Spacecraft", *Acta Astronautica*, Vol.21, No.6/7, 1990, pp.477-486.

Rodden J.J. [1984]
"Closed-Loop Magnetic Control of a Spin-stabililized Satellite", *Automatica*, Vol.20, No.6, 1984, pp.729-735.

Sebestyen G. [1992]
"Integrated Space Experiment Satellite (ISES) Low Cost Stabilization Subsystem Performance", *Proceedings of the 6th AIAA/USU Conference on Small Satellites*, Utah State University, Sept. 1992.

Steyn W.H. [1994]
"Comparison of Low-Earth Orbiting Satellite Attitude Controllers Submitted to Controllability Constraints", *Journal of Guidance, Control, and Dynamics*, Vol.17, No.4, 1994, pp.795-804.

Stickler A.C. and Alfriend K.T. [1974]

"An Elementary Magnetic Attitude Control System", *AIAA Mechanics and Control of Flight Conference*, Anaheim California, 5-9 Aug.1974.

Sugeno M. and Nishida M. [1985]

"Fuzzy Control of a Model Car", *Fuzzy Sets and Systems*, Vol.16, North-Holland, Amsterdam, The Netherlands, 1985, pp.103-113.

Sweeting M.N. [1994]

"UoSAT Microsatellites & Minisatellites for Affordable Access to Space", *Proceedings IEEE/SAIEE 1994 Symposium on Small Satellites and Control*, University of Stellenbosch, 4 Oct.1994.

Wertz J.R. [1986]

Spacecraft Attitude Determination and Control, D. Reidel Publishing Company Boston U.S.A., Reprint 1986.

Zadeh L.A. [1984]

"Making Computers Think Like People", *IEEE Spectrum*, Aug.1984, pp.26-32.

3. REACTION WHEEL CONTROL

3.1 Introduction

A very simple way to alter the attitude of a satellite, is to include within the satellite a disk or wheel which can be rotated (by a servo motor) with respect to the satellite. Then by conservation of angular momentum, any angular momentum imparted to this wheel is balanced by an equal momentum of opposite sense for the remainder of the satellite's body. Hence, by a trivial integration, if the wheel, starting from rest, is turned through an angle θ and then stopped, the remainder of the satellite (if the external torques are balanced) will turn through an angle $k\theta$, where k is the ratio of the wheel to body moments of inertia. If 3-axis stabilization is required, three reaction wheels with mutually perpendicular axes would be used. For redundancy reasons a fourth wheel is normally added to maintain full 3-axis controllability when one reaction wheel fails. The NASA standard configuration adds a fourth skewed wheel, such that its axis is equally inclined from the three orthogonal reference axes [Vadali, 1984; Goel, 1985]. On SUNSAT the first three reaction wheels are aligned with the body X, Y and Z axes and the fourth backup wheel is also aligned to the Z-axis. The availability of the Z-axis wheel is critical during imaging, when the body Z-spin has to be stopped and the yaw angle accurately controlled.

The total angular momentum $\mathbf{L}$ of the satellite can be written as,

$$\mathbf{L} = \mathbf{I}\boldsymbol{\omega}_B^I + \mathbf{h} \tag{3.1}$$

and,

$$\mathbf{h} = \mathbf{I}_w\boldsymbol{\omega}_w = \text{Wheel angular momentum vector}$$

with,

$\mathbf{I}_w$ = diagonal wheel MOI matrix

$\boldsymbol{\omega}_w$ = wheel angular rate vector referenced to the satellite's body

The time derivative of $\mathbf{L}$, referenced to the inertially fixed coordinate system (subscribt I) satisfies [Wertz, 1986],

$$\left.\frac{d\mathbf{L}}{dt}\right|_I = \left.\frac{d\mathbf{L}}{dt}\right|_B + \boldsymbol{\omega}_B^I \times \mathbf{L} \tag{3.2}$$

If the external torques are neglected the total angular momentum is conserved and the left-hand side of Equation (3.2) is zero. Using Equation (3.1), the dynamic equation of the satellite and wheels becomes:

$$\mathbf{I}\dot{\boldsymbol{\omega}}^{I}_{B} + \dot{\mathbf{h}} + \boldsymbol{\omega}^{I}_{B} \times \left(\mathbf{I}\boldsymbol{\omega}^{I}_{B} + \mathbf{h}\right) = \mathbf{0} \qquad (3.3)$$

The quantity $\dot{\mathbf{h}} = \mathbf{I}_w\dot{\boldsymbol{\omega}}_w = \mathbf{N}_{wheel}$ is the net torque applied to the wheels by the spacecraft body. By Newton's third law of motion, $-\dot{\mathbf{h}}$ is the torque applied to the spacecraft body by the wheels. In the remainder of the chapter the reaction wheels will be assumed to be ideal: The wheel friction and mechanical time constant will be ignored and the net torque $\mathbf{N}_{wheel}$ will be used throughout. These assumptions are reasonable as the closed loop reaction wheel control bandwidth will be lower than the open loop wheel response. Furthermore, when wheel angle or angular rate tracking is implemented, the wheel friction will automatically be compensated for by an internal servo control loop.

Equation (3.3) is equivalent to Euler's equation of motion (1.7), but without the added influence of external torques. Normally during reaction wheel control on a gravity gradient stabilized satellite, the gravity gradient torque vector $\mathbf{N}_{GG}$ and external disturbance torque vector $\mathbf{N}_{D}$ must be added to the right hand side of Equation (3.3). It is then obvious that these external torques may give rise to a momentum build-up on the reaction wheels during wheel stabilization periods. This can cause not only speed saturation of the wheels, but also an increase in gyroscopic cross-coupling during control. To prevent this situation momentum dumping must be exercised routinely. On most micro satellites where mass propulsion thrusters are impractical, momentum dumping can be done by utilizing magnetic torquers. Chapter 4 will present some new optimal algorithms on this subject.

There are not any examples of micro satellites employing reaction wheels in the open literature. A few articles on the use of momentum wheels on the following micro satellites were found:

- TUBSAT-1 (Technical University Berlin), Ginati [1989]
- HETE (Aero-Astro Inc.), Chang [1992]
- BREM-SAT (ZARM/University of Bremen), Königsmann [1994]

3.2 Pointing/Tracking Control

3.2.1 CURRENT METHODS

Extensive research has already been done on globally stable attitude control algorithms for spacecraft, some of the results will be discussed in this paragraph. The spacecraft orientation is at present commonly described in terms of quaternions (e.g. on HEAO, Space Shuttle and Galileo). Quaternions have no inherent geometrical singularity as do Euler angles; there are no singularities in the kinematic differential equations as do

Cayley-Rodiques parameters; and successive rotations follow the quaternion multiplication rules. Moreover, quaternions are well suited for onboard real-time computation, since only products and no trigonometric relations exist in the quaternion equations.

Wie and Barba [1985] used quaternions as attitude errors in three quaternion feedback control laws. They also proved the closed loop stability of these controllers for 3-axis maneuvers using the Liapunov stability theorem. The error quaternion is defined as the quaternion difference between the current quaternion and the commanded quaternion, (see Equation (1.15)). Whenever the current quaternion and the commanded quaternion coincide ($\mathbf{q} = \mathbf{q}_c$), the error quaternion $\mathbf{q}_e = [0\ 0\ 0\ 1]^T$. The quaternion feedback laws they considered for an inertial stabilized spacecraft, were:

$$\begin{aligned}
&\text{Law 1:} \quad \mathbf{N}_{control} = -\mathbf{K}\mathbf{q}_{vec} - \mathbf{D}\boldsymbol{\omega}^I_B \\
&\text{Law 2:} \quad \mathbf{N}_{control} = -\mathbf{K}\frac{\mathbf{q}_{vec}}{q_{4e}^3} - \mathbf{D}\boldsymbol{\omega}^I_B \\
&\text{Law 3:} \quad \mathbf{N}_{control} = -\operatorname{sgn}(q_{4e})\mathbf{K}\mathbf{q}_{vec} - \mathbf{D}\boldsymbol{\omega}^I_B
\end{aligned} \tag{3.4}$$

where,

$\mathbf{q}_{vec} = [q_{1e} \quad q_{2e} \quad q_{3e}]^T$ = vector part of error quaternion

$\mathbf{K} = diag[k \quad k \quad k]$ = identical positive angular control gains for each axis

$\mathbf{D} = diag[d_1 \quad d_2 \quad d_3]$ = different positive rate control gains for each axis

The three control laws they presented, are analogous to a conventional feedback control law in that the control torque is a function of position and rate. For $q_{4e} > 0$, control laws 1 and 3 are identical, but when the attitude error angle are greater than 180° ($q_{4e} < 0$), control law 3 will take the shortest rotational path. Control law 2 is similar to the asymtotical stable control law for unbounded control signals, first proposed by Mortenson [1968]. Control law 2 was also used by Van den Bosch *et.al.* [1986] for their adaptive model reference controller.

Near the equilibrium point ($q_{4e} \approx 1$) all three control laws perform similarly. The performance and some physical insight in these quaternion feedback control laws can be obtained by considering a single axis control rotation. For example, if we take a single inertially referenced rotation around the body X-axis while using control law 1, the closed loop dynamics can be written as:

$$I_{xx}\ddot{\Phi} + d_1\dot{\Phi} + k\sin\left(\frac{\Phi}{2}\right) = 0 \tag{3.5}$$

where,

$\dot{\Phi} = \omega_x$

Equation (3.5) is similar to the nonlinear equation of a simple pendulum with viscous damping. When this equation is linearized near the equilibrium point and compared to a standard damped second order system:

$$\omega_n = \sqrt{\frac{k}{2I_{xx}}}, \qquad \zeta = \frac{d_1}{\sqrt{2kI_{xx}}}, \qquad \zeta\omega_n = \frac{d_1}{2I_{xx}} \tag{3.6}$$

The settiing time can therefore be chosen by the correct choice of the rate feedback constants d_i, and the angular feedback constant k can then be adjusted for the correct amount of damping. For a critical damped ($\zeta = 1$) system, [Vadali, 1984]

$$k = \frac{d_1^2}{2I_{xx}} = \frac{d_2^2}{2I_{yy}} = \frac{d_3^2}{2I_{zz}} \tag{3.7}$$

From Equations (3.6) and (3.7) it can be seen that equal damping and settling times, for all three axes with different MOI values, will not be possible for the control laws in (3.4) near the equilibrium point. Wie and Barba [1985] (for control laws 1-3) and Vadali and Junkins [1984] (for control law 1) managed to prove asymtotic stability in the large, by using Liapunov functions.

Vadali and Junkins [1984] also presented a control law for tracking of a target frame. The target kinematics must be modelled by a quaternion representation similar to Equations (1.12) and (1.13):

$$\dot{\mathbf{q}}_c = \frac{1}{2}\mathbf{\Omega}_c\mathbf{q}_c$$

with,

$$\mathbf{\Omega}_c = \begin{bmatrix} 0 & \omega_{oz-ref} & -\omega_{oy-ref} & \omega_{ox-ref} \\ -\omega_{oz-ref} & 0 & \omega_{ox-ref} & \omega_{oy-ref} \\ \omega_{oy-ref} & -\omega_{ox-ref} & 0 & \omega_{oz-ref} \\ -\omega_{ox-ref} & -\omega_{oy-ref} & -\omega_{oz-ref} & 0 \end{bmatrix}$$

Liapunov stability could only be proven if identical feedback gains were used for both the angular and rate elements. The tracking control law for reaction wheels is then given by,

$$\mathbf{N}_{control} = \mathbf{I}\dot{\boldsymbol{\omega}}_{ref} + \boldsymbol{\omega}_B^I \times \left(\mathbf{I}\boldsymbol{\omega}_B^I + \mathbf{h}\right) - d\left[\boldsymbol{\omega}_B^I - \boldsymbol{\omega}_{ref}\right] - k\mathbf{q}_{vec} \tag{3.8}$$

Wie *et.al.* [1989] proved that eigenaxis rotations were possible for inertially referenced spacecraft when the following control law is used,

$$\mathbf{N}_{control} = \boldsymbol{\omega}_B^I \times \left(\mathbf{I}\boldsymbol{\omega}_B^I + \mathbf{h}\right) - \mathbf{D}\boldsymbol{\omega}_B^I - \mathbf{K}\mathbf{q}_{vec} \tag{3.9}$$

where,

$\mathbf{D} = d\mathbf{I}$ and $\mathbf{K} = k\mathbf{I}$

They found that global stability could be proven using a Liapunov function if $\mathbf{K}^{-1}\mathbf{D} > 0$ and $\mathbf{K}^{-1} = k_1\mathrm{I} + k_2\mathbf{1}$ (**1** is the 3×3 identity matrix). This specific selection for **K** is only necessary when precise cancellation of the gyroscopic coupling torque (first term on the right hand side of (3.9)) is not achieved. Global stability with robustness to inertia uncertainty was also proven whenever $\mathbf{K}^{-1} = k_2\mathbf{1}$ ($\mathbf{K} = diag[k \;\; k \;\; k]$ as above). It was found, however, during simulations that stable behaviour was possible even for $\mathbf{K} = k\mathbf{I}$, when perfect cancellation of the gyroscopic coupling torque is not achieved and inertia uncertainty occurs.

For the feedback gain choices of Equation (3.9) it can easily be shown that the closed loop dynamics along the eigenaxis satisfies [Wie, 1989],

$$\ddot{\Phi} + d\dot{\Phi} + k\sin\left(\frac{\Phi}{2}\right) = 0 \tag{3.10}$$

The approximate second order linear behaviour for small eigenaxis rotations present simple expressions for the gain constants in terms of the damping ratio ζ and natural frequency ω_n,

$$d = 2\zeta\omega_n \qquad \text{and} \qquad k = 2\omega_n^2 \tag{3.11}$$

Salehi and Ryan [1985] presented an asymtotically stabilizing, non-linear feedback law for attitude regulation. The control law takes the form of a linear combination of non-linear functions of attitude (expressed in quaternion format) and of angular momentum. The control law can be expressed as,

$$\mathbf{N}_{control} = -\sum_{j=1}^{m}\mathbf{K}_j\left\|\mathbf{q}_{vec}\right\|^{2(p_j-1)}\mathbf{q}_{vec} - \sum_{j=1}^{n}\mathbf{D}_j\left\|\boldsymbol{\omega}_B^I\right\|^{2(r_j-1)}\boldsymbol{\omega}_B^I \tag{3.12}$$

where,

$$\mathbf{K}_j = diag\begin{bmatrix}k_j & k_j & k_j\end{bmatrix}$$
$$\mathbf{D}_j = diag\begin{bmatrix}d_{1j} & d_{2j} & d_{3j}\end{bmatrix}$$

with,

$$k_j, d_{ij} > 0$$
$$p_j, r_j > 0.5$$

The asymtotic stability of this control law was proven similar to the others, by using a Liapunov function. The control law in Equation (3.12) becomes equal to control law 1 in Equation (3.4) when $m = n = 1$ and $p_1 = r_1 = 1$. When $p_i > 1$ and $r_i > 1$ a 'higher order" feedback term is introduced in the control. On the other hand, if $p_i \in (0.5, 1)$ or $r_i \in (0.5, 1)$ for some j, then a 'lower order" feedback term is introduced. The latter case exhibits efficient regulation near the equilibrium state (higher gains near zero state).

3.2.2 SUNSAT IMPLEMENTATION

The SUNSAT reaction wheel pointing and tracking controllers will be implemented somewhat differently to all the controllers of the previous paragraph, because all pointing and tracking maneuvers must be referenced to the orbit coordinate system and not to the inertial coordinate system. The angular rate values used as feedback variables will therefore be those referenced to the orbital axes. Likewise, the error quaternion is also the quaternion difference of the current and commanded quaternion referenced to the orbital axes. The pointing reaction wheel control law 1 (Equation (3.4)) can then be written as,

$$\mathbf{N}_{wheel} = \mathbf{K}\mathbf{q}_{vec} + \mathbf{D}\boldsymbol{\omega}_{B}^{O} \tag{3.13}$$

with,

Controller 1:
$$\mathbf{K} = k\mathbf{1} = diag\begin{bmatrix}0.05 & 0.05 & 0.05\end{bmatrix}$$
$$\mathbf{D} = d\,\mathbf{1} = diag\begin{bmatrix}2 & 2 & 2\end{bmatrix}$$

The feedback gain values k and d have been chosen from Equations (3.6) and (3.7) to give a critically damped system with a 2% settling time of 160 seconds around the X- and Y-axis. The Z-axis, however, will perform similarly due to the gyroscopic coupling between the various axes.

The eigenaxis rotation control law of Equation (3.9) can be approximated (SUNSAT's attitude is not inertially referenced) as,

$$\mathbf{N}_{wheel} = \mathbf{K}\mathbf{q}_{vec} + \mathbf{D}\boldsymbol{\omega}_{B}^{O} - \boldsymbol{\omega}_{B}^{I} \times \left(\mathbf{I}\boldsymbol{\omega}_{B}^{I} + \mathbf{h}\right) + \mathbf{N}_{GG} \tag{3.14}$$

with,

Controller 2:
$$\mathbf{K} = k\mathbf{I} = diag\begin{bmatrix}0.05 & 0.05 & 0.0025\end{bmatrix}$$
$$\mathbf{D} = d\mathbf{I} = diag\begin{bmatrix}2 & 2 & 0.1\end{bmatrix}$$

The main additional torques to be compensated for in an open-loop fashion are the gyroscopic torque and the gravity gradient torque $\mathbf{N}_{GG}$. The latter torque is computed as indicated in Equation (1.8).

These two pointing controllers were then implemented in the simulation program for SUNSAT. A slightly elliptical orbit (e = 0.03) with average aerodynamic disturbance torque was utilized. A sampling period T_s of 1 second was used for the reaction wheel controllers. An initial angular momentum on the wheels of $\mathbf{h}$ = [30 20 10] kgm^2.rpo was assumed. The pointing control accuracy of less than 3 mrad (0.17°) in pitch and roll, and 6 mrad (0.34°) in yaw must be satisfied as specified in Paragraph 1.2.

Figures 3.1 and 3.2 show typical step responses and pointing accuracies for controller 1 and controller 2 respectively. Both controllers satisfy the required pointing accuracy, but controller 2 gives the smallest pointing errors, mainly due to the cancellation of the gyroscopic coupling torques. The magnitude of the pointing error when utilizing controller 1 is, however, dependent on the initial wheel angular momentum and can under certain conditions exceed the accuracy limits. Figure 3.3 presents the reaction wheel angular momentum when either controllers 1 or 2 is used.

A tracking control law when SUNSAT is not 3-axis stabilized, but rather tracking a reference angular rate was also implemented and simulated. This controller is an adapted version of Equations (3.9) and (3.14) combined:

$$\mathbf{N}_{wheel} = \mathbf{K}\mathbf{q}_{vec} + \mathbf{D}\left(\boldsymbol{\omega}_B^O - \boldsymbol{\omega}_{ref}\right) - \mathbf{I}\dot{\boldsymbol{\omega}}_{ref} - \boldsymbol{\omega}_B^I \times \mathbf{h} + \mathbf{N}_{GO} \tag{3.15}$$

Note: The gyroscopic coupling compensation in the equation above only provides for the reaction wheel coupling components. The body coupling components is not cancelled in the dynamic equation as they are still needed to modulate the body inertially referenced angular rates, for example, when the satellite is both Z-spinning and nadir pointing (rotates inertially once per orbit). Additionally, the commanded quaternion must be modelled by the target's kinematic equation as was shown in Paragraph 3.2.1.

Figures 3.4 and 3.5 are typical results obtained from the tracking controller. The pitch and roll pointing reference angles are 2° and -1° during the tracking maneuver and the yaw angle is tracking a target body Z-spin rate ($\omega_{oz\text{-}ref}$) of 2 rpo (revolutions per orbit). The initial reaction wheel angular momentum was taken as $\mathbf{h}$ = [30 20 10] kgm^2.rpo. The pitch and roll pointing errors and yaw tracking error were more than an order of magnitude smaller than the required specification. However, it must be stressed that the simulation assumed perfect attitude, angular rate and MOI knowledge for the satellite during simulation.

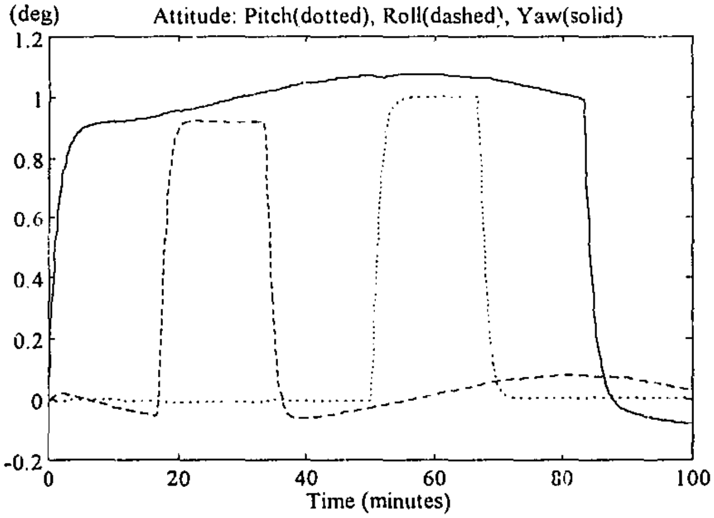

Figure 3.1 Attitude control of quaternion feedback controller 1

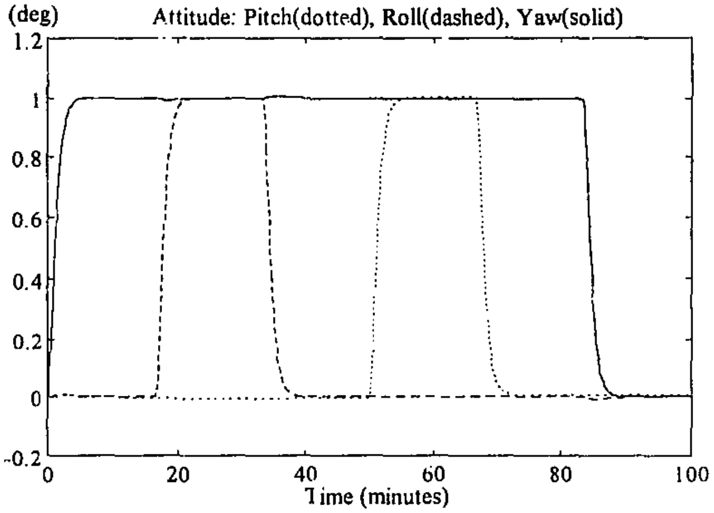

Figure 3.2 Attitude control of quaternion feedback controller 2

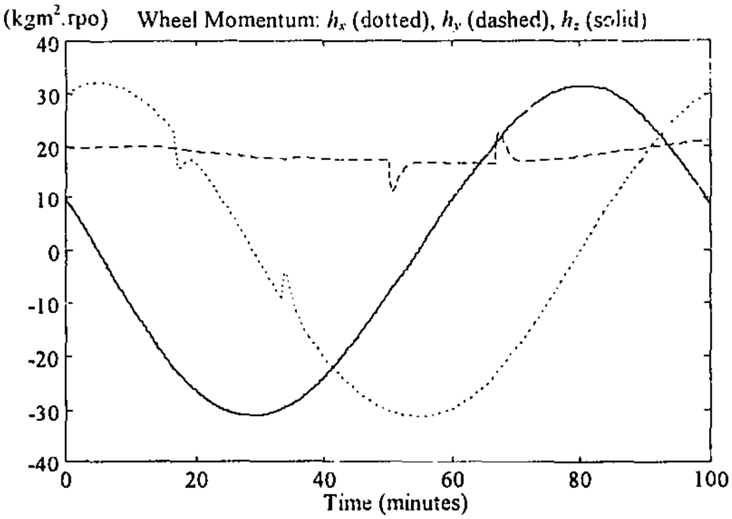

Figure 3.3 Reaction wheel angular momentum of controller 1 and 2

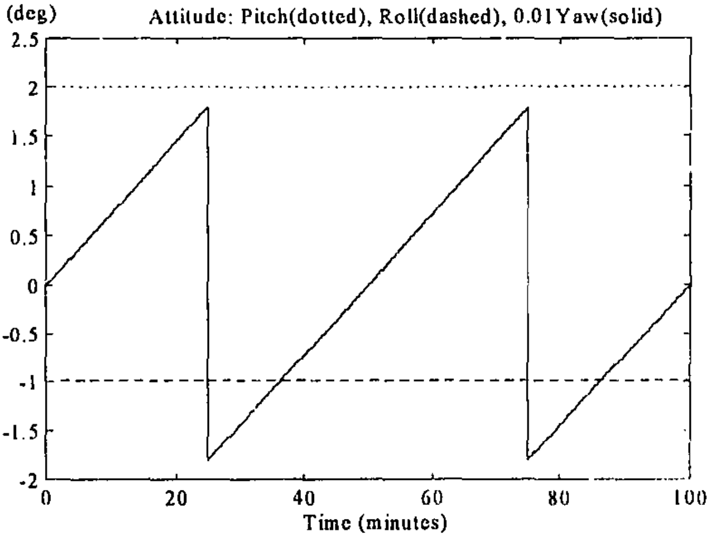

Figure 3.4 Attitude performance of SUNSAT tracking controller

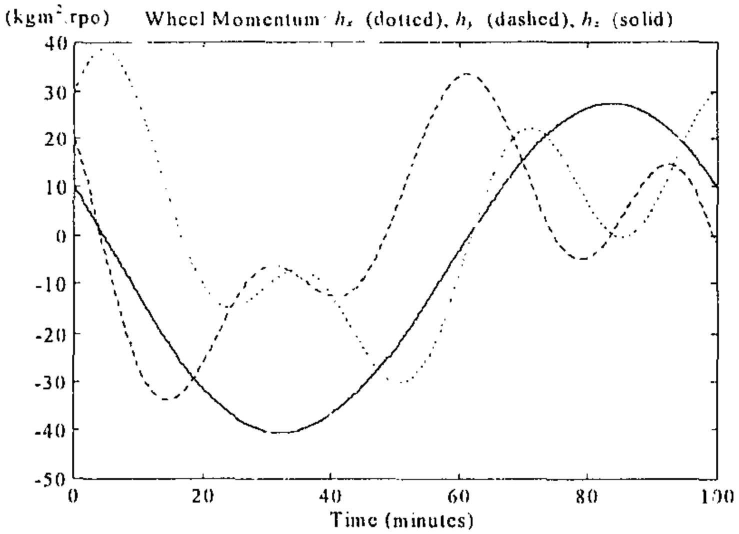

Figure 3.5 Reaction wheel angular momentum for the tracking controller

3.3 Large Angular Slew Maneuvers

3.3.1 INTRODUCTION

Remote sensing satellites often have to embark on minimum time large slew maneuvers to point their payload (e.g. body fixed cameras) at different targets within a short span of time. This problem has already been addressed by various researchers in the past: Wie *et.al.* [1989] proposed a linear quaternion feedback regulator with open loop decoupling control torque for gyroscopic forces to ensure inertially referenced eigenaxis rotations. Reaction wheel speed constraints were however not considered; it was assumed that the rotation will be slow enough resulting in no violation of the wheel saturation limits. Van den Bosch *et.al.* [1986] presented an adaptive control method to enable the IRAS satellite to track a linear reference model around an eigenaxis. Although reaction wheel constraints were considered in their proposed algorithm, model updating was needed to ensure only a near eigenaxis rotation and the rotation time was not minimized. Both these papers developed and applied their algorithms only to inertially stabilized spacecraft.

D'Amario *et.al.* [1979] designed a single-rotation-axis (eigenaxis) autopilot for rapid attitude maneuvers on spacecraft using jet thrusters. Pre-maneuver calculations are done to obtain a staircase time history of the commanded eigenaxis angular acceleration/deceleration. The calculations ensure that at least one thruster will be on near full time (near maximum acceleration and deceleration) except during coasting when a maximum rotation speed limit is reached. Only gyroscopic coupling non-linearities were considered in their autopilot model. Although the autopilot commands were computed open-loop, rate feedback tracking was provided during the maneuver.

Redding *et.al.* [1987] described fuel-optimal jet thruster maneuvers for the Space Shuttle Linearized dynamics is assumed and a two-point boundary-value problem solved in real-time. The solution provides open-loop jet firing commands and set-points for standard autopilot feedback loops. Angular rate limits are satisfied by adopting a fixed-end-time formulation and by correct choice of the final time.

Vadali [1986] used the principles of variable structure control theory to implement robust large-angle maneuvers on a spacecraft Initially maximum torque is used to reach a sliding manifold, then the state trajectory is controlled to slide on the manifold towards the target attitude. The sliding motion was chosen to be optimal in the sense of a quadratic performance index in the Euler parameters and angular velocities. Although the maneuvers were not time-optimal, they were robust against modelling errors and disturbance torques.

Li and Bainum [1990] presented an iterative numerical approach to find the minimum slew time and the corresponding optimal controls of a general rigid spacecraft. Bilimoria and Wie [1993] described a time-optimal 3-axis reorientation method for a rigid spacecraft They showed that in general the eigenaxis rotation maneuver is not time-optimal. Reductions in the slew time of a symmetric body compared to eigenaxis rotations of less than 10 % were obtained during simulations. However, this was achieved at the expense of a substantial increase in control energy (each actuator per axis is commanded simultaneously in a full bang-bang manner). Their algorithm was exclusively derived for an inertially symmetric (e.g. spherical or cubical) rigid body.

Byers and Vadali [1993] extended the time-optimal reorientation problem by presenting approximate solutions to the switching times for non-symmetric rigid bodies. A feedforward/feedback control law to approximate the time-optimal solution was also proposed. This control law took care of modelling errors caused by gyroscopic coupling and differences in moment of inertia. The algorithms in Byers *et.al.* [1993] are, however, computationally very demanding. Both Bilimoria and Byers *et.al.* [1993] demonstrated a marginal decrease in rotation time compared to an eigenaxis rotation at the expense of a substantial increase in control energy

The algorithm presented in this paragraph attempts to balance the opposing goals of minimum time and minimum control effort when using reaction wheels. A practical solution to these opposing goals might be to minimize the maneuver time around the eigenaxis. An eigenaxis rotation results in the shortest angular path and therefore also in near minimum control effort. The unique constant ratio properties of the Euler vector components and body angular rates during an eigenaxis rotation are used to design a computational undemanding algorithm for real-time implementation. Reaction wheel torque and speed limitations are also explicitly satisfied by the algorithm. Compensation feedback is used during the eigenaxis rotations to minimize any perturbations that might be caused by modelling errors (especially inertia uncertainty) and external disturbances.

Only three orthogonally mounted reaction wheels are assumed in this paragraph, however, the NASA standard configuration with an extra skew reaction wheel can also be used. The required three element control torque vector must then be transformed by any suitable transformation matrix to a four element wheel vector. Likewise, the reaction wheel speed and torque saturation constraints must be inversely transformed to the three element vector values as used in the newly proposed algorithm. Momentum dumping will be considered in chapter 4. It is assumed in this paragraph that a suitable dumping mechanism is implemented to ensure small values of wheel angular momentum before each slew maneuver commence.

3.3.2 EIGENAXIS MANEUVER

During an eigenaxis slew, the rotation axis ($\mathbf{e} = [e_x\ e_y\ e_z]^T$) stays constant with respect to the orbital reference frame for a nadir pointing satellite. Then for the vector components of the error quaternion as defined in Equation (1.15):

$$\frac{q_{ie}(t)}{q_{je}(t)} = const. \qquad \text{i, j} = 1,2,3 \text{ and } i \neq j \qquad (3.16)$$

Likewise, during the eigenaxis maneuver the angular body rate vector (orbit referenced) will point in the same direction as the Euler axis:

$$\omega_B^O = \left[e_x\dot{\Phi} \quad e_y\dot{\Phi} \quad e_z\dot{\Phi}\right]^T = \left[\omega_{ox} \quad \omega_{oy} \quad \omega_{oz}\right]^T \qquad (3.17)$$

thus,

$$\frac{\omega_{oi}(t)}{\omega_{oj}(t)} = const \qquad \text{i, j} = x,y,z \text{ and } i \neq j \qquad (3.18)$$

If we assume relatively small gyroscopic, gravity gradient and disturbance torques compared to the maximum reaction wheel torque during a slew maneuver, a near minimum time eigenaxis rotation would be possible when (see Figure 3.6):

$$\ddot{\Phi}(t) = \begin{cases} +a & \{t \in t_0, t_h\} \\ -a & \{t \in t_h, t_f\} \end{cases} \tag{3.19}$$

where,

a = near maximum acceleration/deceleration possible with reaction wheels
t_h = time to reach the halfway mark during the slew maneuver
$t_f = 2t_h$ = time to complete the slew maneuver

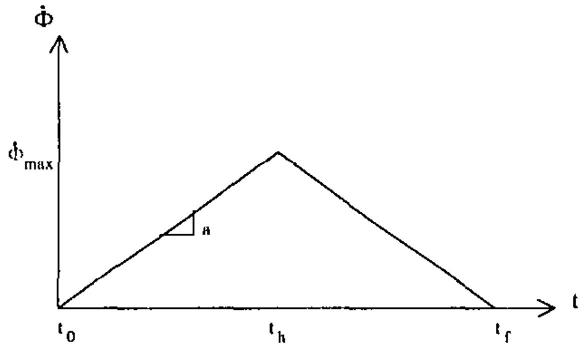

Figure 3.6 Angular rate during eigenaxis rotation without wheel speed limiting

If the wheel speed limit is reached during the acceleration phase, further acceleration has to be stopped and the satellite must be allowed to coast until the appropriate time to enter the deceleration phase. The rule to follow is to ensure equal periods of time, t_c before and after the halfway mark during the coasting phase (see Figure 3.7).

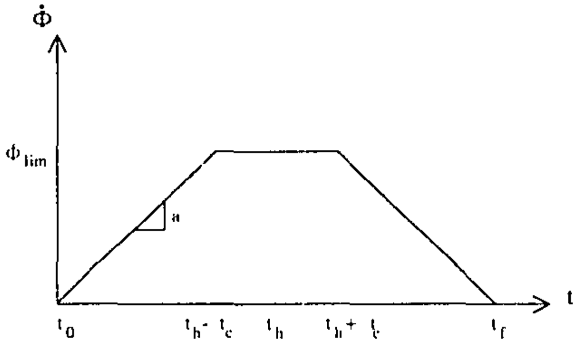

Figure 3.7 Angular rate during eigenaxis rotation with wheel speed limiting

From Equation (3.17) it is clear that the angular body rates ω_{ox}, ω_{oy} and ω_{oz} will have similar trajectories to Figure 3.6 and 3.7, but with slopes ± $e_x a$, ± $e_y a$ and ± $e_z a$ respectively. A linear increase/decrease in the angular body rates can be obtained by a large constant acceleration/deceleration slew torque from the reaction wheels plus a small additional wheel torque to cancel gyroscopic and gravity gradient torques. The assumption of small gyroscopic torques will be valid if the spacecraft is 3-axis stabilized with low reaction wheel moments before the eigenaxis rotation commences.. Therefore, the satellite has a small angular momentum initially. During the maneuver, the only external torques to marginally change the angular momentum, will be small gravity gradient and external disturbance torques. A further small wheel torque will be needed to ensure the necessary momentum transfer between the body axes, as to maintain the -ω_o inertial body rate[1] around the Y_o-axis (to stay nominally nadir pointing during attitude changes). These additional torques are:

$$\mathbf{N}_{add} = \mathbf{N}_{gyro} + \mathbf{N}_{GG} + \mathbf{N}_{\omega o} \qquad (3.20)$$

where,

$$\begin{aligned} \mathbf{N}_{gyro} &= -\boldsymbol{\omega}_B^I \times \left(\mathbf{I}\boldsymbol{\omega}_B^I + \mathbf{h}\right) \\ \mathbf{N}_{\omega o} &= -\mathbf{I}\dot{\boldsymbol{\omega}}_o^B \end{aligned} \qquad (3.20a)$$

with,

$$\begin{aligned} \dot{\boldsymbol{\omega}}_o^B &= \text{Change in orbit angular rate vector along the body reference axes} \\ &= \dot{\mathbf{A}}\begin{bmatrix}0 & -\omega_o & 0\end{bmatrix}^T = -\omega_o \dot{\mathbf{A}}_2 \end{aligned}$$

where,

$\dot{\mathbf{A}}_2$ is the second column vector of matrix $\dot{\mathbf{A}}$

From the quaternion format transformation matrix $\mathbf{A}$ in Equation (1.4) and the kinematic update in Equation (1.12), $\mathbf{N}_{\omega o}$ can be rewritten as:

$$\mathbf{N}_{\omega o} = \omega_o \mathbf{I}\begin{bmatrix} 2\left(q_2\dot{q}_1 + q_1\dot{q}_2 - q_4\dot{q}_3 + q_3\dot{q}_4\right) \\ 2\left(-q_1\dot{q}_1 + q_2\dot{q}_2 - q_3\dot{q}_3 + q_4\dot{q}_4\right) \\ 2\left(-q_4\dot{q}_1 + q_3\dot{q}_2 + q_2\dot{q}_3 - q_1\dot{q}_4\right) \end{bmatrix} = \omega_o \mathbf{I}\begin{bmatrix} q_2 & q_1 & q_4 & q_3 \\ -q_1 & q_2 & -q_3 & q_4 \\ -q_4 & q_3 & q_2 & -q_1 \end{bmatrix}\boldsymbol{\Omega}\mathbf{q} \qquad (3.20b)$$

The total reaction wheel torque can then be taken as (ignoring wheel friction):

$$\mathbf{N}_{wheel} = \mathbf{I}_w\dot{\boldsymbol{\omega}}_w = \dot{\mathbf{h}} = \mathbf{N}_{add} - \mathbf{N}_{slew} \qquad (3.21)$$

[1] We shall assume a circular orbit with constant body angular rate around the Y_o-axis to keep the satellite nadir pointing for all derivations in the rest of this chapter. The small errors induced due to this assumption in an orbit with low eccentricity, will be accounted for by a feedback term in the control law.

If Equation (3.21) is now included in Euler's dynamic equation (1.7):

$$\mathbf{I}\dot{\boldsymbol{\omega}}_B^I = -\mathbf{N}_{\omega\omega} + \mathbf{N}_{slew} + \mathbf{N}_D \tag{3.22}$$

Differentiating Equation (1.15)[1] and using Equation (3.22):

$$\mathbf{I}\dot{\boldsymbol{\omega}}_B^O = \mathbf{I}\dot{\boldsymbol{\omega}}_B^I - \mathbf{I}\boldsymbol{\omega}_o\dot{\mathbf{A}}_2 = \mathbf{N}_{slew} + \mathbf{N}_D \tag{3.23}$$

If we ignore the disturbance torque as small being compared to the reaction wheel slew torques and assume $\mathbf{I} = diag(I_{xx}, I_{yy}, I_{zz})$, we have during the acceleration and deceleration phase of the slew maneuver:

$$\frac{I_{ii}\dot{\omega}_{oi}}{I_{jj}\dot{\omega}_{oj}} = \left.\frac{N_{wheel-i}}{N_{wheel-j}}\right|_{slew} = const. \qquad \mathrm{i,j} = \mathrm{x,y,z} \text{ and } \mathrm{i} \neq \mathrm{j} \tag{3.24}$$

We already have from Equations (3.16-3.18), for example, when i = x and j = y, during an eigenaxis slew:

$$\frac{\omega_{ox}}{\omega_{oy}} = \frac{e_x}{e_y} = \frac{q_{1e}}{q_{2e}}$$

thus,

$$\left.\frac{N_{wheel-x}}{N_{wheel-y}}\right|_{slew} = \frac{I_{xx}\,q_{1e}}{I_{yy}\,q_{2e}} \tag{3.25}$$

Relationship (3.25) - in all combinations of x, y, z and q_{1e}, q_{2e}, q_{3e} - can now be used at the start of an eigenaxis slew, to determine the respective reaction wheel slew torques during acceleration and deceleration. The slew torque can then be written as:

$$\mathbf{N}_{slew} = \begin{cases} +k\,\min\limits_i \left|\dfrac{N_{sat-i}}{I_{ii}q_{ie}}\right| diag(\mathbf{I})\mathbf{q}_{vec}, & t \in (0, t_h - t_c) \\ \mathbf{0}, & t \in (t_h - t_c, t_h + t_c) \\ -k\,\min\limits_i \left|\dfrac{N_{sat-i}}{I_{ii}q_{ie}}\right| diag(\mathbf{I})\mathbf{q}_{vec}, & t \in (t_h + t_c, 2t_h) \end{cases} \tag{3.26}$$

with,

$\mathbf{q}_{vec}$ = $\begin{bmatrix} q_{1e} & q_{2e} & q_{3e} \end{bmatrix}^T$

N_{sat-i} = Saturated wheel torque in body axis *i*

$k \in (0,1)$ is the amount of total saturated torque which is used for the maneuver

To satisfy the torque constraint on the reaction wheels, the wheel with the saturated slew torque requirement (*i* from Equation (3.26)) will be fixed to slightly less than the maximum limit, i.e. $k < 1$. This is done to enable a near minimum time slew while providing for the small additional torques as explained above.

While the actual reaction wheel control torque ([illegible]) is computed during the maneuver in an open loop manner, the halfway mark is determined using feedback from the error quaternion. The largest error quaternion vector component (q_{1e}(t), q_{2e}(t) or q_{3e}(t)) is compared to its pre-computed value at the halfway mark (q_{half}):

$$\max_t |q_{ie}(t)| - q_{half} \quad = \quad \begin{cases} > 0 & \forall \quad t < t_h \\ < 0 & \forall \quad t > t_h \end{cases} \tag{3.27}$$

where,

$$q_{half} = \frac{\max_i |q_{ie}(t_0)|}{\left|\sin\left(\frac{\Phi}{2}\right)\right|} \left|\sin\left(\frac{\Phi}{4}\right)\right| \tag{3.28}$$

The full algorithm for a near time optimal eigenaxis rotation can be summarized as:

1. Determine the initial error quaternion $\mathbf{q}_e(t_0)$ from Equation (1.15).
2. Compute the respective x-, y- and z-axis reaction wheel slew torques from Equation (3.26).
3. Use Equation (3 28) to obtain q_{half} for the dominant torque axis error quaternion vector component.
4. Use the acceleration phase slew torques plus the additional torques, Equations (3.20-3.21) to compute the total amount of wheel torque to be applied to each wheel.
5. Apply these torques and repeat step 4 until $t > t_h$ using Equation (3.27), then go to step 8, or until one of the wheel speeds approach its limit (say 95% of maximum to provide for the additional torques during the coasting phase), then go to step 6.
6. If a wheel speed limit is encountered, zero all the wheel slew torques and apply only the additional torques during a coasting phase.
7. Repeat step 6 until an equal coasting period t_c is completed before and after the half-way mark, then go to step 8.
8. Use the deceleration phase slew torques plus the additional torques, Equations (3.20-3.21) to compute the total amount of wheel torque to be applied to each wheel.
9. Apply these torques and repeat step 8, until $t = 2t_h = t_f$

To compensate for any small attitude and rate errors at the completion of the eigenaxis rotation due to unmodelled disturbances or small inertia mismatches, a standard linear quaternion feedback regulator (Paragraph 3.2.1) will be used.

In case of larger inertia mismatches, greater deviations can occur from the eigenaxis during the rotation. A feedback compensation torque using the measured body angular rate components (referenced to the orbit following coordinates) can solve this problem. The constant ratio property of the angular body rate components (3.18) during an eigenaxis rotation will be used to formulate a compensation feedback law. The compensation torque will be proportional to the difference between a reference angular rate vector and the measured angular rate vector. The compensation feedback torque to be added to the total wheel torque (3.21) is:

$$\mathbf{N}_{comp} = \mathbf{C}\left[\boldsymbol{\omega}_B^O - \boldsymbol{\omega}_{ref}\right] \qquad (3.29)$$

where,

$$\mathbf{I}\dot{\boldsymbol{\omega}}_{ref} = \mathbf{N}_{slew} \qquad (3.30)$$

In the next section near eigenaxis convergence and stability of the compensation control law (3.29), in spite of inertia mismatches, will be proven. As shown in the next paragraph, convergence stability will be achieved, whenever **C** is larger than a certain minimum diagonal positive matrix.

3.3.3 EIGENAXIS CONVERGENCE

Let $\mathbf{I}_n$ denote the nominal value of the inertia matrix and $\Delta\mathbf{I}$ the uncertainty, therefore $\mathbf{I} = \mathbf{I}_n + \Delta\mathbf{I}$. From Euler's equation (1.7) using **I**, and Equations (3.20-3.22) using $\mathbf{I}_n$, follows:

$$\left(\mathbf{I}_n + \Delta\mathbf{I}\right)\dot{\boldsymbol{\omega}}_B^I = 3\omega_o^2\left[\mathbf{z}_o^B \times \Delta\mathbf{I}\,\mathbf{z}_o^B\right] - \boldsymbol{\omega}_B^I \times \Delta\mathbf{I}\,\boldsymbol{\omega}_B^I - \omega_o\mathbf{I}_n\dot{\mathbf{A}}_2 + \mathbf{N}_{slew} \qquad (3.31)$$

or in terms of the orbit referenced body angular rates using equation (1.14):

$$\left(\mathbf{I}_n + \Delta\mathbf{I}\right)\dot{\boldsymbol{\omega}}_B^O = 3\omega_o^2\left[\mathbf{z}_o^B \times \Delta\mathbf{I}\,\mathbf{z}_o^B\right] - \boldsymbol{\omega}_B^I \times \Delta\mathbf{I}\,\boldsymbol{\omega}_B^I + \omega_o\Delta\mathbf{I}\dot{\mathbf{A}}_2 + \mathbf{N}_{slew} \qquad (3.32)$$

If we assume that the orbit angular rate ω_o is much smaller than the body angular rate components during the slew maneuver; 1) $\boldsymbol{\omega}_B^I \approx \boldsymbol{\omega}_B^O$ and 2) the first and third terms on the right hand side of Equation (3.32) can be ignored. For an axially symmetric spacecraft (diagonal inertia matrix), Equation (3.32) can then be approximated as:

$$\begin{aligned}
\dot{\omega}_{ox} &\approx \frac{\Delta I_Y - \Delta I_Z}{I_X + \Delta I_X}\,\omega_{oy}\omega_{oz} - \frac{\Delta I_X}{I_X + \Delta I_X}\,\eta_{slew\,x} + \eta_{slew\,x} \\
\dot{\omega}_{oy} &\approx \frac{\Delta I_Z - \Delta I_X}{I_Y + \Delta I_Y}\,\omega_{ox}\omega_{oz} - \frac{\Delta I_Y}{I_Y + \Delta I_Y}\,\eta_{slew\,y} + \eta_{slew\,y} \\
\dot{\omega}_{oz} &\approx \frac{\Delta I_X - \Delta I_Y}{I_Z + \Delta I_Z}\,\omega_{ox}\omega_{oy} - \frac{\Delta I_Z}{I_Z + \Delta I_Z}\,\eta_{slew\,z} + \eta_{slew\,z}
\end{aligned} \qquad (3.33)$$

with,

$$\eta_{slew} = \begin{cases} +k \min\limits_{i} \left| \dfrac{N_{sat\_i}}{I_{ii} q_{ie}} \right| \mathbf{q}_{vec} & \text{for the acceleration phase} \\ 0 & \text{for the coasting phase} \\ -k \min\limits_{i} \left| \dfrac{N_{sat\_i}}{I_{ii} q_{ie}} \right| \mathbf{q}_{vec} & \text{for the deceleration phase} \end{cases}$$

The first two terms on the right hand side of Equation (3.33) will be the main cause of perturbations from the eigenaxis due to inertia mismatches. The last term will be the normalised slew torque needed for a true eigenaxis rotation. If we take the body angular rate components as the sum of an eigenaxis reference component and a perturbational component,

$$\boldsymbol{\omega}_B^O = \boldsymbol{\omega}_{ref} + \Delta\boldsymbol{\omega}$$
$$\dot{\boldsymbol{\omega}}_{ref} = \boldsymbol{\eta}_{slew}$$

Then using Equation (3.33) and subtracting (adding to wheel torque) the normalized compensation torque of Equation (3.29) the perturbational dynamics becomes:

$$\begin{bmatrix} I_X + \Delta I_X & 0 & 0 \\ 0 & I_Y + \Delta I_Y & 0 \\ 0 & 0 & I_Z + \Delta I_Z \end{bmatrix} \Delta\dot{\omega} = \begin{bmatrix} -C_X & k_1\omega_{oz\_ref} & k_1\omega_{oy\_ref} \\ k_2\omega_{oz\_ref} & -C_Y & k_2\omega_{ox\_ref} \\ k_3\omega_{oy-ref} & k_3\omega_{ox\_ref} & -C_Z \end{bmatrix} \Delta\omega$$
$$+ \begin{bmatrix} k_1\left(\Delta\omega_y \Delta\omega_z + \omega_{oy-ref}\omega_{oz\_ref}\right) \\ k_2\left(\Delta\omega_x \Delta\omega_z + \omega_{ox\_ref}\omega_{oz-ref}\right) \\ k_3\left(\Delta\omega_x \Delta\omega_y + \omega_{ox\_rex}\omega_{oy\_ref}\right) \end{bmatrix} - \begin{bmatrix} \Delta I_X & 0 & 0 \\ 0 & \Delta I_Y & 0 \\ 0 & 0 & \Delta I_Z \end{bmatrix} \eta_{slew} \tag{3.34}$$

with,

$$k_1 = \Delta I_Y - \Delta I_Z$$
$$k_2 = \Delta I_Z - \Delta I_X$$
$$k_3 = \Delta I_X - \Delta I_Y$$
$$\mathbf{C} = diag[C_X \quad C_Y \quad C_Z]$$

In Equation (3.34) k_1 to k_3 are small constants depending only on the inertia uncertainty, with the property that they can never all have the same sign The eigenaxis reference angular rate components increase linearly with time during the acceleration phase, stay constant during a coasting phase or decrease linearly with time during the deceleration phase. The dynamics of Equation (3.34) will be stable if the time varying system matrix of $\Delta\omega$ have stable eigenvalues. It can easily be shown by applying the Routh's stability criterion to the characteristic equation of the system that a sufficient condition for stable eigenvalues will be when,

$$
\begin{aligned}
C_X &> \max\left\{\left|k_1\omega_{oy\ ref}\right|,\left|k_1\omega_{oz\ ref}\right|\right\}\\
C_Y &> \max\left\{\left|k_2\omega_{ox\ ref}\right|,\left|k_2\omega_{oz\ ref}\right|\right\}\\
C_Z &> \max\left\{\left|k_3\omega_{ox\ ref}\right|,\left|k_3\omega_{oy\ ref}\right|\right\}
\end{aligned}
\tag{3.35}
$$

There is a limit to the size of the time varying forcing terms in Equation (3.34) due to the wheel speed and torque constraints. Their influence on $\Delta\omega$ can be reduced further by increasing the positive constants C_X, C_Y and C_Z - this will speedup convergence and reduce tracking errors to the eigenaxis. The simple compensation control law of (3.29) will therefore result in stable feedback with arbitrary small perturbations from the eigenaxis due to inertia modelling errors.

3.3.4 SIMULATION RESULTS

SUNSAT, a small cubical micro satellite of 60 kg, with a gravity gradient boom of 2.3 meter and tip mass of 6 kg deployed along the nominal zenith (-Z_o-axis) direction, will be used to test the new algorithm. This structure gives a MOI tensor $\mathbf{I}$ = *diag*[40, 40, 2] kgm^2. The orbit configuration will be slightly elliptical with a perigee altitude of 400 km and an apogee altitude of 840 km. Three identical reaction wheels with a wheel MOI I_w = 5e-4 kgm^2, a maximum wheel torque N_{sat} = 4e-3 Nm and a maximum wheel speed = 4800 rpm, will be used.

To compare the various simulations all the rotations will be start from a stable nadir pointing attitude and rotate to a certain 1-2-3 Euler attitude of roll, pitch and yaw. The satellite is then kept at this specific pointing attitude for a while and finally commanded back towards nadir. The new algorithm is then compared to an eigenaxis quaternion feedforward/feedback regulator [Wie *et.al.*, 1989]:

$$\mathbf{N}_{wheel} = k\mathbf{I}\mathbf{q}_{vec} + d\mathbf{I}\,\boldsymbol{\omega}_B^O + \mathbf{N}_{add} \tag{3.36}$$

Equation (3.36) will ensure eigenaxis rotations when not subjected to the wheel speed and wheel torque constraints as proven by Wie *et.al.* [1989]. In the simulations of this paragraph, however, saturation of wheel speed and torque can occur during certain large angular slew maneuvers. Gain values for k = 0.00175 and d = 0.05 were used. These values for k and d were arrived at during simulation optimization, the criteria being to minimize the slew time of large angle slew maneuvers.

Table 3.1 lists some typical simulation results. The average slew time and time integrated torque values between the new algorithm and the quaternion regulator are compared. A positive presentage value lists the improvement due to the new algorithm, and a negative value the opposite. The new algorithm outperforms the quaternion regulator with respect to the slew time in all cases, although not by much when speed and torque saturation of the quaternion regulator occur. Torque saturations normally leads to a decrease in energy effectiveness when compared to the new algorithm. As expected, an unsaturated quaternion regulator (small angular or yaw rotations) delivers eigenaxis rotations These rotations are slower but more energy efficient compared to the new algorithm

Table 3.1 Comparison between the new algorithm and a quaternion regulator.

Roll / Pitch / Yaw	*Slew Time*	*Integrated Wheel Torque*
50°/ -70°/ 90°	+1.0%	+46.6%
20°/ -10°/ -60°	+29.9%	-19.4%
-10°/ 10°/ 0°	+68.0%	-85.5%
80°/ 0°/ 0°	+1.0%	+1.0%
0°/ 0°/ 175°	+83.9%	-52.5%

Figure 3.8 shows a large angular rotation with the corresponding roll, pitch and yaw angles during the maneuvers. Initially the new near-minimum-time eigenaxis algorithm is used, and at time = 1500 seconds the quaternion feedforward/feedback regulator is employed. A more symmetric behaviour and pitch, roll and yaw angles reaching their target values simultaneously, are observed for the new algorithm. Although the slew times were almost identical (approximately 300 seconds), the new eigenaxis rotation results in almost 50% less control effort. The reason being the controller saturation and non-eigenaxis rotation of the quaternion regulator (see Figures 3.10 and 3.11).

Figure 3.9 shows he corresponding reaction wheel spin moments. (Note: the angular rate is expressed in revolutions per orbit - rpo units.) The effect of the wheel constraints can be noticed: The maximum slope of the spin moments is limited by the wheel torque constraint. The magnitude of the spin angular momentum is also limited to within the maximum wheel momentum of ± 240 kgm^2.rpo. The wheel speed is limited to this maximum value by zeroing of the corresponding wheel torque of the quaternion regulator. The coasting phase is entered at ± 228 kgm^2.rpo (5% lower than maximum) for the newly proposed algorithm, to provide for the additional torque influence during this phase.

Figure 3.10 shows the corresponding reaction wheel torques. It can clearly be seen from the integrated area under these curves that the eigenaxis rotation results in less control effort. The maximum wheel torque constraint of ± 4e-3 Nm is also adhered to by saturation torque limiting of the linear feedback regulator. The new eigenaxis algorithm satisfies the torque constraint by using 3.6e-3 Nm (10% lower than maximum) as the upper limit on the wheel slew torque, therefore providing 0.4e-3 Nm for the additional small torques.

Figure 3.11 shows the corresponding behaviour of the error quaternion vector components. The straight lines in Figure 3.11 indicate the pure eigenaxis rotation (constant ratios in the error quaternion vector components) of the new method. The large excursions from the straight lines points to the non-eigenaxis rotation of the quaternion regulator. Figure 3.12 shows a smaller angular rotation and the quaternion regulator also delivers an eigenaxis maneuver. The slew time improvement of the new algorithm is now clearly observable.

Figure 3.13 presents the same angular maneuver as in Figure 3.8, but this time with ± 10% modelling errors in the MOI values of the satellite: $\mathbf{I}_n$ = *diag*[44 44 1.8] kgm². The first part (until 1500 seconds) shows the rotations using the new method, but without the compensation torque of Equation (3.29). Under-compensation in roll and over-compensation in yaw is clearly visible. The attitude and rate errors are then corrected at the end of the rotation by the quaternion feedback regulator. The latter part of the simulation demonstrates the effect when the feedback rate tracker is added. The roll, pitch and yaw angles are almost identical to the idealy modelled case of Figure 3.8. **C** was chosen as *diag*[2.0 2.0 2.0], this gives an effective perturbation dynamics time constant of 20 seconds - see Equation (3.34).

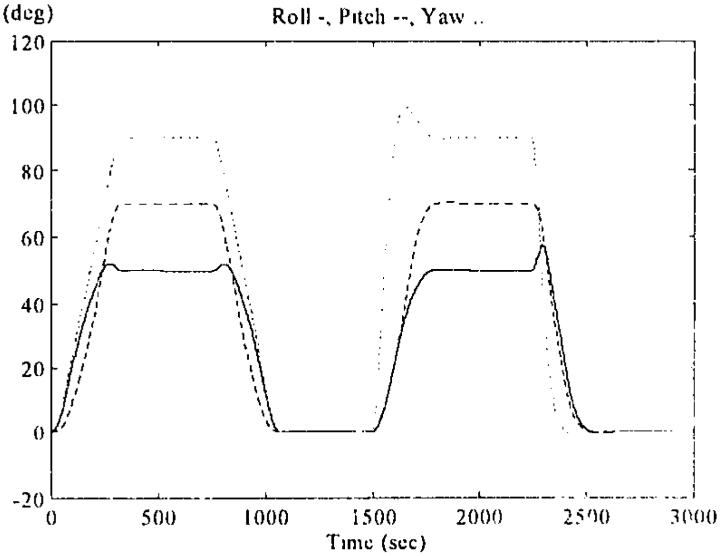

Figure 3.8 Euler angles for a large angular slew (New and Q-regulator)

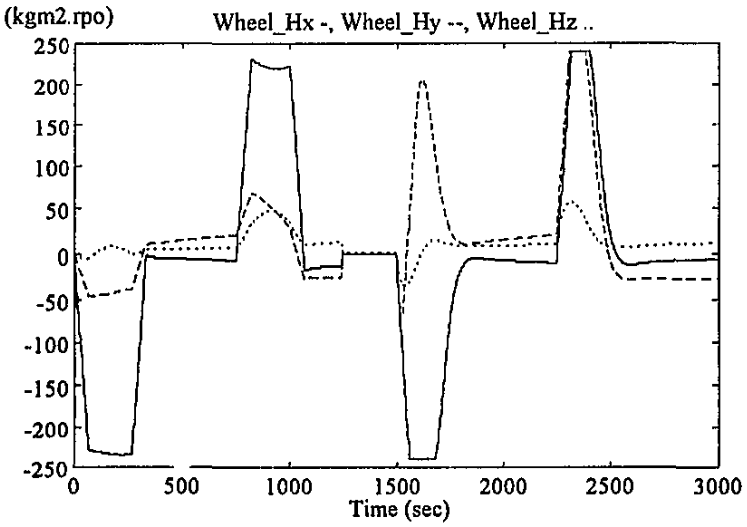

Figure 3.9 Large slew reaction wheel angular momentum (New and Q-regulator)

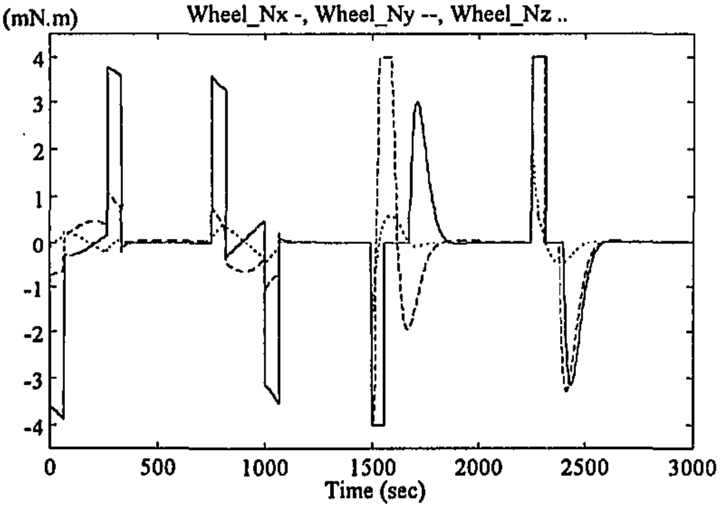

Figure 3.10 Large slew reaction wheel torque (New and Q-regulator)

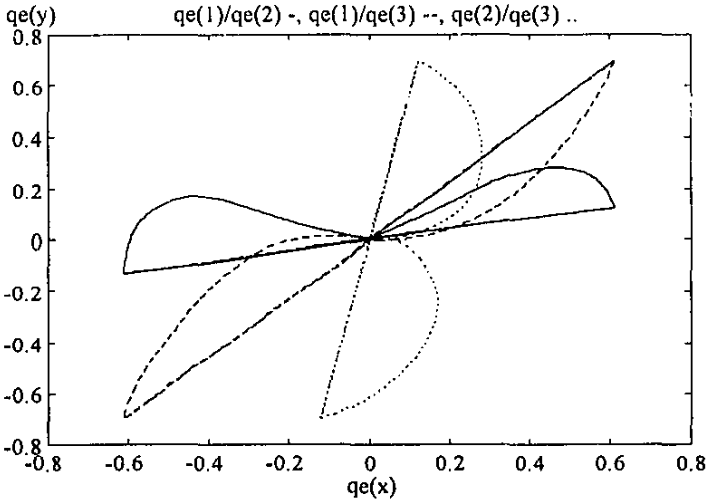

Figure 3.11 Large slew quaternion error plots (New and Q-regulator)

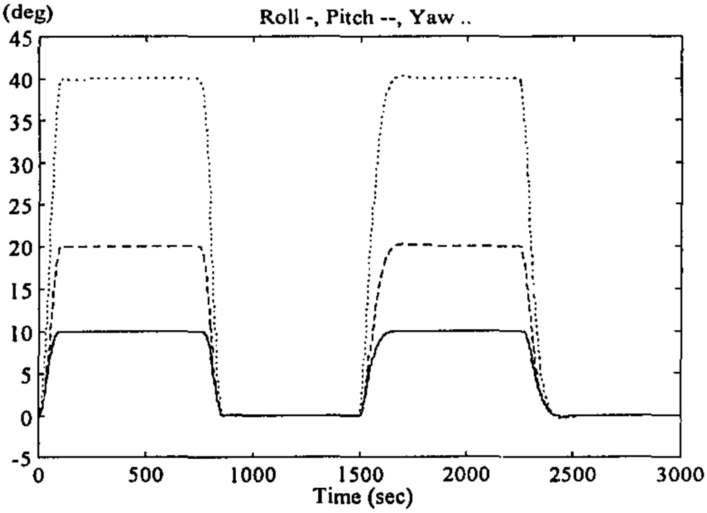

Figure 3.12 Euler angles during a small angular slew (New and Q-regulator)

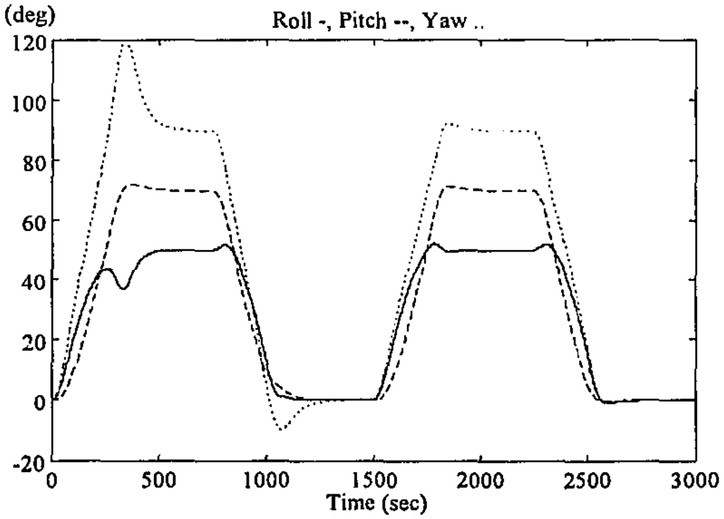

Figure 3.13 Euler angles during a large angular slew with 10% MOI mismatch

3.4 Conclusions

In this chapter slight variations of published globally stable quaternion feedback control laws for reaction wheels are presented, to enable accurate pointing and tracking control on a nominally nadir pointing satellite. Simple expressions are also given to design the feedback gains for specific closed loop settling time and damping specifications. Pointing and tracking accuracies of better than 0.02° (Figures 3.2 and 3.4) in pitch, roll and yaw commanded angles were achieved during simulation (assuming perfect attitude measurements). These accuracies are almost an order of magnitude better than the specification for the SUNSAT mission. It can therefore be concluded that the final pointing accuracy will depend only on the sensor measurements and performance of the attitude estimators, see Chapter 5 for more detail.

A practical, near minimum-time, eigenaxis rotational method is proposed for a 3-axis reaction wheel and gravity gradient stabilized nadir pointing satellite. This method considers all the important constraints of reaction wheels. Compared to a simulation optimized linear feedback control law, the new method improved the slew time versus control effort performance for all the target rotations investigated. Due to the open-loop computation of the reaction wheel torque during a large angular rotation, unmodelled disturbances and MOI errors can degrade the overall performance to some extent. Disturbance torques will normally be a few orders of magnitude smaller than the near maximum reaction wheel torque employed during the rotation. These disturbances will therefore not cause significant deviations from the eigenaxis in the limited time of a slew maneuver.

The main cause for degradation can be MOI modelling mismatches. Simulations indicated that modelling errors of less than 5% still result in near eigenaxis rotations. For larger MOI mismatches a feedback compensation law can be utilized to minimize any large deviations from the eigenaxis. The stability of the closed loop system with this addition is also proven. The algorithm presented in this chapter can easily be adapted and used on inertially stabilized spacecraft as well.

3.5 References

Bilimoria K.D. and Wie B. [1993]
"Time optimal Three-Axis Reorientation of a Rigid Spacecraft", *Journal of Guidance, Control and Dynamics*, Vol.16 No.3, May-June1993, pp. 446-452.

Byers R.M. and Vadali S.R.
"Quasi-Closed-Form Solution to the Time-Optimal Rigid Spacecraft Reorientation Problem", *Journal of Guidance, Control and Dynamics*, Vol.16 No.3, May-June 1993, pp. 453-461.

D'Amario L.A. and Stubbs G.S. [1979]
"A New Single-Rotation-Axis Autopilot for Rapid Spacecraft Attitude Maneuvers", *Journal of Guidance, Control and Dynamics*, Vol.2 No.4, July-Aug. 1979, pp. 339-346.

Goel P.S., Krishnan A. and Upadhyaya J.S. [1985]
"Auto Reconfiguration of Reaction Wheels in IRS", *IEEE Transactions on Aerospace and Electronic Systems*, Vol.21, No.1, Jan. 1985, pp.160-163.

Ginati A. [1989]
"The TUBSAT-1 Attitude Control and Stabilization System", *Proceedings of the 3rd Annual AIAA/USU Conference on Small Satellites*, Utah State University, Sept. 1989.

Königsmann H.J., Oelze H.W. and Rath H.J. [1994]
"BREM-SAT - First Flight Results", *Proceedings of the 8th Annual AIAA/USU Conference on Small Satellites*, Utah State University, Sept. 1994.

Li F. and Bainum P.M. [1990]
"Numerical Approach for Solving Rigid Spacecraft Minimum Time Attitude Maneuvers", *Journal of Guidance, Control and Dynamics*,Vol.13 No.1, Jan.-Feb. 1990, pp. 38-43.

Mortensen R.E. [1968]
"A Globally Stable Linear Attitude Regulator", *International Journal of Control*, Vol.8, No.3, 1968, pp.297-302.

Redding D.C. and Adams N.J. [1987]
"Optimized Rotation-Axis Attitude Maneuver Controller for the Space Shuttle Orbiter", *Journal of Guidance, Control and Dynamics*, Vol.10 No.1, Jan.-Feb. 1987, pp. 4-13.

Salehi S.V. and Ryan E.P. [1985]
"A Non-Linear Feedback Attitude Regulator", *International Journal of Control*, Vol.41, No.1, 1985, pp.281-287.

Vadali S.R. [1986]
"Variable-Structure Control of Spacecraft Large-Angle Maneuvers", *Journal of Guidance, Control and Dynamics*, Vol.9 No.2, March-April 1986, pp. 235-239.

Van den Bosch P.P.J., Jongkind W. and Van Swieten A.C.M. [1986]
"Adaptive Attitude Control for Large-angle Slew Manoeuvres", *Automatica*, Vol.22 No.2, 1986, pp. 209-215.

Wie B. and Barba P.M. [1985]
"Quaternion Feedback for Spacecraft Large Angle Maneuvers", *Journal of Guidance, Control and Dynamics*, Vol.8 No.3, May-June 1985, pp. 360-365.

Wie B., Weiss H. and Arapostathis A. [1989]
"Quaternion Feedback Regulator for Spacecraft Eigenaxis Rotations", *Journal of Guidance, Control and Dynamics*, Vol.12 No.3, May-June 1989, pp. 375-380.

4. MOMENTUM DUMPING

4.1 Introduction

Any reaction wheel 3-axis stabilized satellite must employ a momentum management .ilgorithm to restrict the wheel momentum within allowable limits. Momentum build-up naturally occurs due to the influence of external disturbance torques, for example, from gravity gradient torques, aerodynamic and solar forces. A cheap and effective means of active unloading of this momentum, is by making use of magnetic torquers [Johnson, 1971; Glaese, 1976; Burns, 1992; Chang, 1992].

Johnson and Skelton [1971] proposed an optimal desaturation control scheme, where the natural external environmental torques (gravity gradient, aerodynamic etc.) are used. A minimum energy optimization problem is solved assuming a priori knowledge of the external disturbance torques, or using an on-line, real-time disturbance estimator. The solution of the optimization problem is a time-varying feedback gain matrix for the system state, to obtain an angular rate reference command vector for the spacecraft. The desaturation maneuver is done within a predetermined fixed time interval during which no attitude pointing will be possible. The prefered final time attitude can howevei be specified.

Glaese *et.al.* [1976] described a minimum energy desaturation law (MEDL) to dump angular momentum on an inertially stabilized space telescope. They also considered the simpler but more inefficient "cross-product law". The MEDL method minimizes the power dissipated by the magnetic torquers over a predetermined fixed desaturation period. Burns and Flashner [1992] presented an adaptive control technique making use of simultaneous magnetic, gravity gradient and aerodynamic torques to do momentum unloading. They assumed a 3-axis stabilized, nadir pointing spacecraft. Feedback gain parameters are adapted in real-time, to ensure tracking of the reaction wheel momentum vector by using an exponentially decayi.,g momentum model. Chang [1992] used magnetic torquers to contain the momentum bias on a single wheel, 3-axis inertially stabilized satellite, at a nominal value. Simple "penalty factor" laws are proposed to reduce the undesirable disturbance torques during magnetic torquing.

In this chapter, two new optimal desaturation algorithms for a nadir-pointing, 3-axis reaction wheel stabilized satellite, using 3-axis magnetic torquing, will be described and simulated. A circular orbit will be assumed throughout the chapter to obtain solvable mathematical models. Simulation results indicate marginal degradation in performance when near-circular (low eccentricity) orbits are actually utilized. The advantages and disadvantages of the two new algorithms will be discussed in the conclusions section.

The results will also be compared to the performance of a non-optimal "cross-product law".

4.2 Preliminaries

The dynamical model of a satellite, using reaction wheels as internal torque actuators and magnetic torquers as external torque actuators, is: {see also Equations (1.7) and (3.3)}

$$\mathbf{I}\dot{\boldsymbol{\omega}}_B^I + \dot{\mathbf{h}} = -\boldsymbol{\omega}_B^I \times \left(\mathbf{I}\boldsymbol{\omega}_B^I + \mathbf{h}\right) + \mathbf{N}_M + \mathbf{N}_{EXT} \tag{4.1}$$

$\mathbf{I} = diag(I_{xx}, I_{yy}, I_{zz})$ is the moment of inertia tensor of an axially symmetric spacecraft.
$\boldsymbol{\omega}_B^I = \begin{bmatrix}\omega_x & \omega_y & \omega_z\end{bmatrix}^T$ is the inertially referenced body angular rate vector.
$\mathbf{h} = \begin{bmatrix}h_x & h_y & h_z\end{bmatrix}^T$ is the reaction wheel angular momentum vector.
$\mathbf{N}_M$ is the torque vector due to magnetic torquing.
$\mathbf{N}_{EXT}$ is the total external torque disturbance vector.

If the satellite is 3-axis stabilized at a fixed nadir-pointing attitude in a circular orbit, we have:

$$\begin{aligned}\boldsymbol{\omega}_B^I &= \begin{bmatrix}0 & -\omega_o & 0\end{bmatrix}^T = \text{ constant vector}\\ \mathbf{I}\dot{\boldsymbol{\omega}}_B^I &= \boldsymbol{\omega}_B^I \times \mathbf{I}\boldsymbol{\omega}_B^I = \mathbf{0}\end{aligned} \tag{4.2}$$

where,

ω_o = orbital angular rate

The magnetic torque vector can be written as the cross-product of the magnetic dipole moment **M** of the magnetic coils with the local geomagnetic field strength **B**:

$$\mathbf{N}_M = \mathbf{M} \times \mathbf{B} = \boldsymbol{\Psi}(t)\mathbf{M} \tag{4.3}$$

with,

$$\boldsymbol{\Psi}(t) = \begin{bmatrix} 0 & B_z(t) & -B_y(t) \\ -B_z(t) & 0 & B_x(t) \\ B_y(t) & -B_x(t) & 0 \end{bmatrix}$$

The desaturation model of a fixed nadir pointing satellite can then be written from Equations (4.1) - (4.3), as:

$$\dot{\mathbf{h}} = \boldsymbol{\Omega}\mathbf{h} + \boldsymbol{\Psi}(t)\mathbf{M} + \mathbf{N}_{EXT} \tag{4.4}$$

with,

$$\Omega = \begin{bmatrix} 0 & 0 & \omega_o \\ 0 & 0 & 0 \\ -\omega_o & 0 & 0 \end{bmatrix}$$

4.3 Optimal Controllers

4.3.1 LQR CONTROLLER

We can find a stable and optimal feedback control law to regulate the wheel moment vector **h** towards the zero vector, by minimizing the following cost functional per orbit:

$$J = \int_{t_0}^{t_f} \left\{ \mathbf{h}^T \mathbf{Q} \mathbf{h} + \mathbf{M}^T \mathbf{R} \mathbf{M} \right\} dt \qquad (4.5)$$

where,

t_0 - t_f = full orbital period
Q = weighting matrix for the wheel angular momentum
R = weighting matrix for the magnetic coil moments

To minimize the cost functional in Equation (4.5) subject to the desaturation model constraint of Equation (4.4), we have to solve a matrix Riccati equation:

$$\dot{\mathbf{K}}(t) = -\mathbf{K}(t)\boldsymbol{\Omega} - \boldsymbol{\Omega}^T\mathbf{K}(t) - \mathbf{Q} + \mathbf{K}(t)\boldsymbol{\Psi}(t)\mathbf{R}^{-1}\boldsymbol{\Psi}^T(t)\mathbf{K}(t) \qquad (4.6)$$

The control law then becomes:

$$\mathbf{M}(t) = -\mathbf{R}^{-1}\boldsymbol{\Psi}(t)\mathbf{K}(t)\mathbf{h}(t) = \mathbf{F}_{LQR}(t)\mathbf{h}(t) \qquad (4.7)$$

The Riccati equation in Equation (4.6) has to be solved by backwards integration from $\mathbf{K}(t_f)$ to $\mathbf{K}(t_0)$. If we assume an orbital periodic nature of the local geomagnetic field vector:

$$\boldsymbol{\Psi}(t_0) \approx \boldsymbol{\Psi}(t_f) \quad \text{and therefore} \quad \mathbf{K}(t_0) \approx \mathbf{K}(t_f) \qquad (4.8)$$

A sufficiently accurate orbital solution to the matrix function $\mathbf{K}(t)$ can then be found by solving the Riccati equation twice: Firstly, the unknown $\mathbf{K}(t_f)$ is chosen as the zero matrix, Equation (4.6) is integrated backwards until $\mathbf{K}(t_0)$ is found. Then $\mathbf{K}(t_f)$ is taken as the $\mathbf{K}(t_0)$ of the first iteration and the integration process is repeated to find the orbital solution for **K**(t). Values of **K**(t) at sampled intervals can then be stored in an onboard look-up table to be used at the corresponding orbital locations for desaturation control. Due to the irregular behaviour of the geomagnetic field closer to the earth's surface, $\Psi(t)$ may vary too much for some low earth orbits for the orbital periodic assumption to hold. It may then be necessary to repeat the solution process for several orbits to ensure full coverage of the sub-satellite ground track on the earth's surface. The corresponding sampled values of the matrix function K(t) are then stored in a 2-dimensional look-up table indexed by the sub-satellite position coordinate.

Due to the slowly varying nature of the geomagnetic field a fairly accurate quasi-static LQR feedback control law can be computed by an on-line solution of the infinite time LQR control problem (at time t):

$$J = \int_{t_0}^{\infty} \left\{ \mathbf{h}^T \mathbf{Q} \mathbf{h} + \mathbf{M}^T \mathbf{R} \mathbf{M} \right\} dt \qquad (4.9)$$

To minimize the cost functional we have to solve the following Riccati equation by eigenvector decomposition of an associated Hamiltonian matrix:

$$\mathbf{0} = -\mathbf{K}_\infty \boldsymbol{\Omega} - \boldsymbol{\Omega}^T \mathbf{K}_\infty - \mathbf{Q} + \mathbf{K}_\infty \boldsymbol{\Psi}(t) \mathbf{R}^{-1} \boldsymbol{\Psi}^T(t) \mathbf{K}_\infty \qquad (4.10)$$

The control law at time t then becomes similar to Equation (4.7) with **K**(t) substituted by **K**$_\infty$, the solution of Equation (4.10). This is a computational demanding effort because the Hamiltonian is a 6 th-order matrix, but no large look-up tables are needed as in the previous method.

4.3.2 MINIMUM-ENERGY CONTROLLER

A fixed terminal time, minimum-energy, optimal controller can be found by minimizing the cost functional subject to the desaturation model constraint of Equation (4.4):

$$J = \frac{1}{2} \int_{t_0}^{t_f} \left\{ \mathbf{M}^T \mathbf{M} \right\} dt \qquad (4.11)$$

The boundary conditions to be satisfied, are:

$$\begin{aligned} \mathbf{h}(t_0) &= \begin{bmatrix} h_{x0} & h_{y0} & h_{z0} \end{bmatrix}^T \\ \mathbf{h}(t_f) &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \end{aligned} \qquad (4.12)$$

The Hamiltonian of the system is defined as:

$$\mathbf{H} = \frac{1}{2}\mathbf{M}^T\mathbf{M} + \mathbf{p}^T\left[\boldsymbol{\Omega}\mathbf{h} + \boldsymbol{\Psi}(t)\mathbf{M}\right] \tag{4.13}$$

The costate equations can be obtained in the standard fashion, as:

$$\dot{\mathbf{p}} = -\frac{\partial \mathbf{H}}{\partial \mathbf{h}} = -\boldsymbol{\Omega}^T\mathbf{p} = \boldsymbol{\Omega}\,\mathbf{p} \tag{4.14}$$

These equations are linear and can easily be solved analytically as:

$$\mathbf{p}(t) = \boldsymbol{\Xi}(t)\,\mathbf{p}(t_0) \tag{4.15}$$

with,

$$\boldsymbol{\Xi}(t) = \begin{bmatrix} \cos\omega_0 t & 0 & \sin\omega_0 t \\ 0 & 1 & 0 \\ -\sin\omega_0 t & 0 & \cos\omega_0 t \end{bmatrix}$$

The optimal controls can be obtained from the optimality condition:

$$\frac{\partial \mathbf{H}}{\partial \mathbf{M}} = \mathbf{0} \tag{4.16}$$

Differentiating Equation (4.13), we obtain the optimal controls:

$$\mathbf{M}(t) = \boldsymbol{\Psi}(t)\mathbf{p}(t) \tag{4.17}$$

The modified state equations can then be written as:

$$\dot{\mathbf{h}}(t) = \boldsymbol{\Omega}\,\mathbf{h}(t) + \boldsymbol{\Psi}(t)\boldsymbol{\Psi}(t)\boldsymbol{\Xi}(t)\mathbf{p}(t_0) \tag{4.18}$$

The variation of extremals method [Kirk, 1970] can now be used to solve $\mathbf{p}(t_0)$:

1. Assume an initial $\mathbf{p}(t_0)$.
2. Solve h(t) by numerically integrating Equation (4.18) from t_0 to t_f, using $\mathbf{h}(t_0)$ and $\mathbf{p}(t_0)$ as initial conditions. Also, solve the state and costate influence function matices $\mathbf{P}_h$ (t) and $\mathbf{P}_p$ (t) by numerical integration.

where,

$$\mathbf{P}_h(t) \equiv \frac{d\mathbf{h}(t)}{d\mathbf{p}(t_0)} \qquad \text{and} \qquad \mathbf{P}_h(t_0) = [0] \quad \text{(the 3x3 zero matrix)}$$

$$\mathbf{P}_p(t) \equiv \frac{d\mathbf{p}(t)}{d\mathbf{p}(t_0)} \qquad\qquad\quad \mathbf{P}_p(t_0) = [1] \quad \text{(the 3x3 identity matrix)}$$

thus,

$$\begin{aligned}\dot{\mathbf{P}}_h(t) &= \frac{\partial^2 \mathbf{H}}{\partial \mathbf{p} \partial \mathbf{h}} \mathbf{P}_h(t) + \frac{\partial^2 \mathbf{H}}{\partial \mathbf{p}^2} \mathbf{P}_p(t) = \boldsymbol{\Omega} \mathbf{P}_h(t) + \boldsymbol{\Psi}(t) \boldsymbol{\Psi}(t) \mathbf{P}_p(t) \\ \dot{\mathbf{P}}_p(t) &= -\frac{\partial^2 \mathbf{H}}{\partial \mathbf{h}^2} \mathbf{P}_h(t) - \frac{\partial^2 \mathbf{H}}{\partial \mathbf{h} \partial \mathbf{p}} \mathbf{P}_p(t) = \boldsymbol{\Omega} \mathbf{P}_p(t)\end{aligned} \tag{4.19}$$

Note: $\mathbf{P}_p\,(t)$ can actually be solved analytically as: $\mathbf{P}_p(t) = \Xi(t)$

3. Adjust $\mathbf{p}(t_0)$ based on Newton's method:

$$\mathbf{p}^{i+1}(t_0) = \mathbf{p}^i(t_0) - \left[\mathbf{P}_h^i(t_f)\right]^{-1} \mathbf{h}^i(t_f) \tag{4.20}$$

4. Repeat step 2 and 3, until $\left\|\mathbf{h}^i(t_f)\right\| < \varepsilon$, where ε is an arbitrarily small constant.

A closer look at Equation (4.18) reveals that a solution of this linear differential expression will be:

$$\mathbf{h}(t) = \boldsymbol{\Theta}(t) \mathbf{p}(t_0)$$

with,

$$\boldsymbol{\Theta}(t) = \mathcal{L}^{-1}\left\{\left[s\mathbf{1} - \boldsymbol{\Omega}\right]^{-1} \mathcal{L}\left\{\boldsymbol{\Psi}(t)\boldsymbol{\Psi}(t)\Xi(t)\right\}\right\} \quad \{\text{using the Laplace transform}\}$$

therefore,

$$\mathbf{P}_h(t_f) = \frac{d\mathbf{h}(t_f)}{d\mathbf{p}(t_0)} = \boldsymbol{\Theta}(t_f) \qquad \text{, a constant matrix}$$

The variational of extremals algorithm will thus converge to the correct value for $\mathbf{p}(t_0)$ within a single iteration. The optimal magnetic moment control vector can then be computed from Equations (4.15), (4.17) and (4.19), assuming an initial $\mathbf{p}(t_0) = \mathbf{0}$:

$$\mathbf{M}(t) = -\boldsymbol{\Psi}(t)\Xi(t)\left[\mathbf{P}_h^0(t_f)\right]^{-1} \mathbf{h}^0(t_f) \tag{4.21}$$

with,

$$\mathbf{h}^0(t_f) = \Xi(t_f)\mathbf{h}(t_0)$$

The constant part of Equation (4.21) can be pre-computed for a specific part and duration of an orbit - the major numerical computations being the numerical integration of $\mathbf{P}_h(t)$ from t_0 to t_f and the inversion of the 3×3 matrix $\mathbf{P}_h(t_f)$. The on-line computations needed to obtain the optimal magnetic moment vector are only the multiplication of a few time-varying and constant terms with the initial wheel moment vector.

4.4 Simulation Results

A small cubical micro satellite of 60 kg (SUNSAT), with three identical orthogonally mounted reaction wheels ($h_{wheel\text{-}max}$ = 0.25 Nms or 240 kgm^2.rpo) and magnetic torquer coils ($M_{coil\text{-}max}$ = 20 Am^2), was used to test the desaturation algorithms. The satellite was 3-axis stabilized, using a quaternion feedback regulator (see Chapter 3) for the reaction wheel torques, with a nadir pointing Z-axis. The newly proposed optimal algorithms were compared to a standard "cross-product law" algorithm:

$$\mathbf{M} = K_m (\mathbf{h} \times \mathbf{B}) / \|\mathbf{B}\| \qquad (4.22)$$

The gain constant K_m = -80.0 (**h** in Nms units) was optimized for the best simulated results (minimum desaturation energy versus time performance). A polar and near-circular orbit at an altitude of 800 km was utilized. A simple, dipole geomagnetic model was used during the simulations:

$$\mathbf{B} = \begin{bmatrix} 20\cos\omega_o t \\ 5 \\ 40\sin\omega_o t \end{bmatrix} \mu\mathrm{T} \qquad (4.23)$$

All the simulations were started from an initial wheel momentum of $\mathbf{h}(t_0) = [100, 100, -100]^T$ kgm^2.rpo (rpo = revolutions per orbit) and conducted over half an orbit (3000 seconds). All the controllers were implemented using a sampling period T_s = 10 seconds. Figures 4.1 to 4.5 show the respective simulated spin moment desaturation trajectories of the LQR (t_f = full orbit), infinite-time LQR, Minimum Energy (t_f = 0.25 orbit), Minimum Energy (t_f = 0.5 orbit) and the ‘cross-product law" controllers. Table 4.1 summarizes these results with respect to the effective on-time of the magnetic coils (an indication of the energy consumed) and the residual wheel moment after half an orbit.

Table 4.1 Comparison of the various desaturation controllers

	LQR	*∞LQR*	*0.25 MEDL*	*0.5 MEDL*	*XPROD*
X-time (sec)	164.4	175.3	174.0	155.3	170.6
Y-time (sec)	333.9	338.6	347.8	271.7	342.8
Z-time (sec)	138.3	125.4	37.2	39.7	141.3
Total (s)	636.6	639.3	559.0	466.7	654.7
h_x (0.5orbit)	-0.53	-0.40	0.48	-0.77	-2.52
h_y (0.5orbit)	0.02	-0.11	0.80	1.75	-1.18
h_z (0.5orbit)	0.41	-0.23	0.68	0.20	-5.40

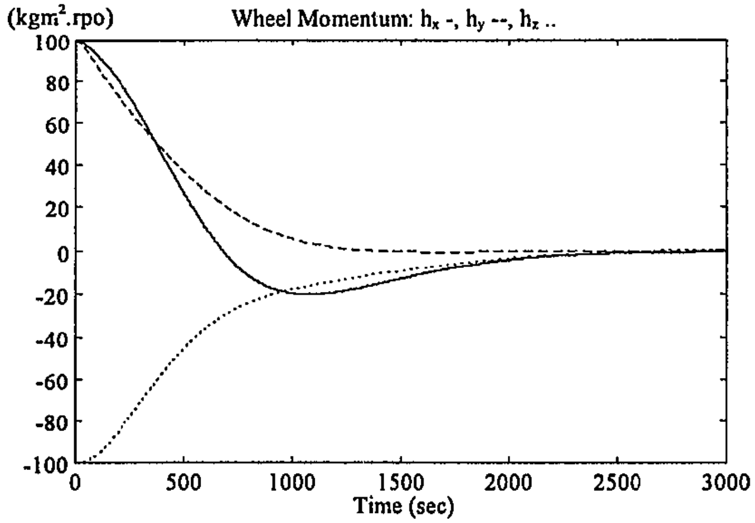

Figure 4.1 Desaturation using an optimal LQR controller

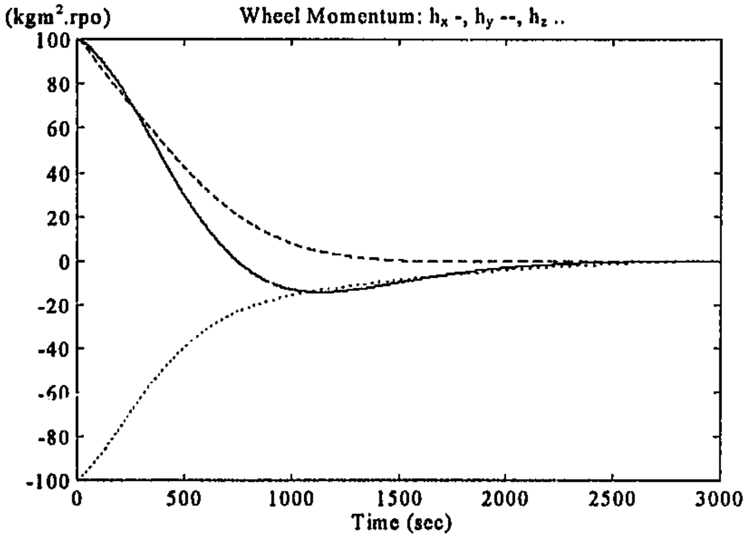

Figure 4.2 Desaturation using an infinite time LQR controller

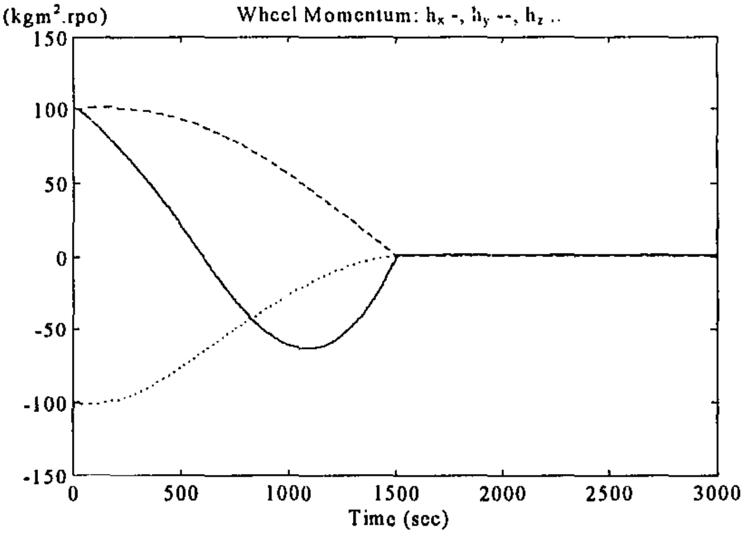

Figure 4.3 Desaturation using a 0.25 orbit MEDL controller

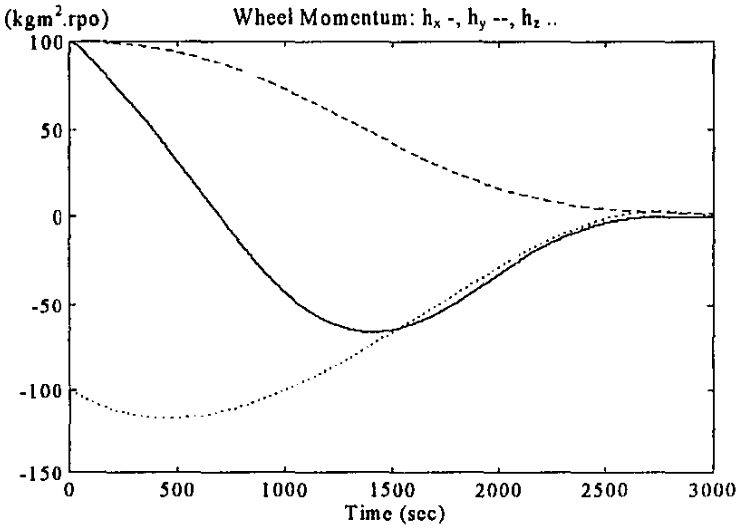

Figure 4.4 Desaturation using a 0.5 orbit MEDL controller

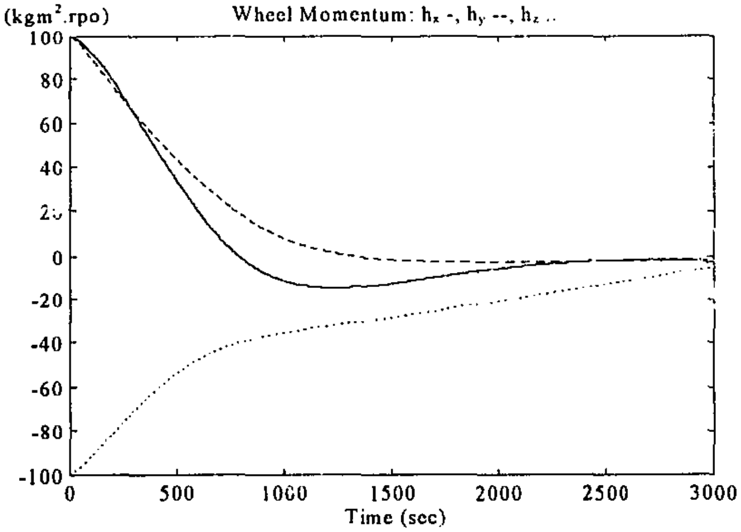

Figure 4.5 Desaturation using the "Cross-product law" controller

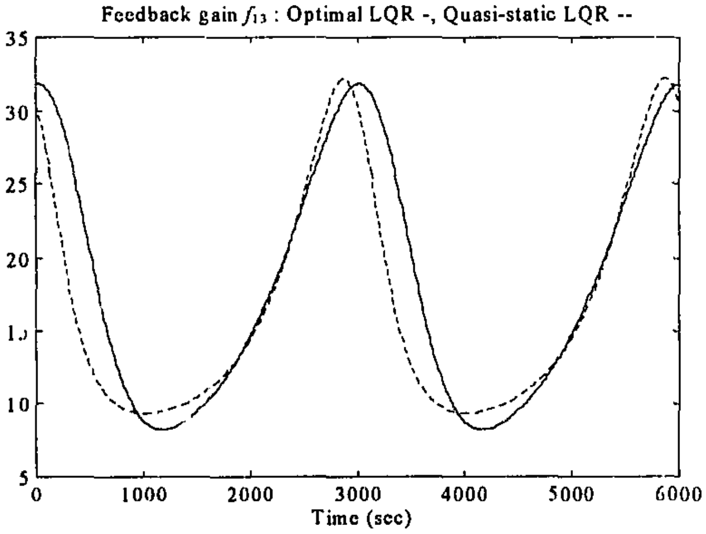

Figure 4.6 Comparison of a quasi-static and an optimal LQR feedback gain term

4.5 Conclusions

All the newly proposed optimal controllers outperform the "cross-product" XPROD controller with respect to the desaturation efficiency (e.g. less magnetic coil energy consumed at a certain desaturation rate). The truely optimal time-variant LQR controller performed almost identical to the infinite time, static LQR controller, although the desaturation model is time-variant. A comparison of the respective LQR feedback gains showed similarly shaped time trajectories with only small variations in magnitude and phase - e.g. see Figure 4.6 for the f_{13} term of $\mathbf{F}_{LQR}$. The Minimum Energy MEDL controllers consumed the least amount of energy as expected, but due to their open-loop nature and non-ideal simulation conditions, terminated with small residual wheel momentum components. Due to modelling errors (e.g. geomagnetic field and non-circular orbit) and external torque disturbances on the stabilized satellite, the residual moments are expected to increase for the actual implementation.

The feedback nature of the LQR controllers would be preferred where an absolute minimum energy demand is not so important. These controllers will therefore ensure some robustness against modelling errors and external disturbances. If the geomagnetic field does not change much between successive orbits, a LQR orbital gain lookup table can be calculated off-line and then used onboard. Else, an on-line infinite time LQR algorithm can be used to recompute the quasi-static feedback gain matrix at every sampling period.

The MEDL algorithm must be implemented for a specific part and duration of an orbit and most of the computations can be done off-line to obtain a constant optimal matrix. The on-line computation then consists of only a few multiplications, as already explained.

To summarize: Two optimal and easily implementable desaturation controllers were presented. These controllers can be used during 3-axis reaction wheel momentum desaturation of nadir pointing spacecraft in near circular orbits.

4.6 References

Burns T.F., Flashner H. [1992]
"Adaptive Control Applied to Momentum Unloading Using the Low Earth Orbital Environment", *Journal of Guidance, Control, and Dynamics*, Vol.15, No.2, March-April 1992, pp.325-333.

Chang D.H. [1992]
"Magnetic and Momentum Bias Attitude Control Design for the HETE Small Satellite", *Proceedings of the 6th AIAA/USU Conference on Small Satellites,* Utah State University, Sept.1992.

Glaese J.R., Kennel H.F., Nurre G.S., Seltzer S.M. and Sheldon H.L. [1976]
"Low-Cost Space Telescope Pointing Control System", *Journal of Spacecraft and Rockets*, Vol.13, No.7, July 1976, pp.400-405.

Johnson C.D. and Skelton R.E. [1971]
"Optimal Desaturation of Momentum Exchange Control Systems", *AIAA Journal,* Vol.9, No.1, January 1971, pp.12-22.

Kirk D.E. [1970]
Optimal Control Theory, An Introduction, Prentice Hall, Englewood Cliffs, New Jersey, 1970, pp.343-357.

5. ATTITUDE DETERMINATION

5.1 Introduction

Various sensors can be employed to measure the attitude of an orbiting body to a varying degree of accuracy. On SUNSAT for example: A magnetometer can measure the direction of the local geomagnetic field vector and when compared to the expected direction obtained from a model of the geomagnetic field, attitude information can be extracted. Six solar cells, one on each facet of the satellite cube can give a rough indication of the direction of the sun relative to the satellite body and when the sun position is known relative to the spacecraft position by using orbital models (sun and satellite), attitude information can be obtained. More accurate attitude measurements can be acquired by using CCD based earth horizon and sun sensors or taking images of the stars with a strapdown star camera.

Unfortunately the most accurate sensors, due to a limited field of view, can only give useful measurements during a fraction of the orbit, whereas the least accurate sensors with an increased measurement range and longer availability of useful data during an orbit, have mostly increased levels of measurement noise. To maximize the attitude determination accuracy and to have continuous availability of attitude information, some form of onboard state estimation will be needed.

This chapter presents a new Kalman filter based state estimator (observer) to extract the approximate angular rate vector of the satellite from either magnetometer measurements or the six omnidirectional solar cells. This estimator can be used during the initial phase of the satellite's life after separation from the launcher, when the satellite is still tumbling. The detumbling and rate tracking controllers (Paragraph 2.2) will use the measured angular rate values from this estimator. Two new extended Kalman filters (EKF) are also introduced to determine the full satellite state from vector observations. The attitude, attitude rate and dominant disturbance torque values are estimated. The first EKF uses only magnetometer measurements and an IGRF geomagnetic field model to determine the satellite's attitude to an angular accuracy of better than 1°. The second EKF uses measurements from the more accurate horizon and sun sensors plus orbital models, and/or star sensor plus star cataloque to determine the attitude to below 0.1° accuracy levels. The new EKF algorithms are compared to a slightly modified version of the EKF estimator of Psiaki *et.al.* [1990].

The application of Kalman filters in the aerospace field has gone a long way since the original work by R.E. Kalman on prediction theory [Kalman, 1960]. Lefferts *et.al.* [1982] reviews the methods of Kalman filtering and their development during the

sixties and seventies when applied to the Apollo program and subsequent spacecraft attitude estimation problems. The Kalman filter is also used extensively in orbit determination problems (Bierman, 1977; Psiaki 1993]. The advantages and disadvantages of Kalman filtering to estimate the direction of the spin vector on spinning satellites with optical sensors are discussed in a paper by Fraiture [1986].
Most of the attitude determination applications use Kalman filters only to determine the attitude and gyro bias values, the angular rates are directly measured from strapdown 3-axis rate gyros [Lefferts, 1982; Dodds, 1984; Zwartbol, 1986; Bar-Itzhack, 1985]. Gai *et.al.* [1985] and Psiaki *et.al.* [1990] are the only papers that could be found utilizing Kalman filter algorithms, where angular rates are also estimated from angular sensor measurements. Gai *et.al.* presented a star sensor based, gyroless EKF estimator for a 3-axis stabilized, zero-momentum bias satellite. Continuous star sensor measurements at a 10 Hz data rate are needed for attitude performance levels of approximately 10 μrad. Psiaki *et.al.* used only magnetometer measurements and data from an IGRF model to estimate the attitude, attitude rate and constant disturbance torques on a gravity gradient 3-axis stabilized spacecraft.

Not all attitude estimation problems are solved through Kalman filtering. Other methods employed, are: An iterative least squares method developed by Grusas [1969] to do high accuracy attitude estimation from celestial sightings. In the Japanese interplanetary explorers "Sakigake" and "Suisei" a sequential least-squares method was used to determine the attitude using ground based software [Ninomiya, 1986]. Methods to determine 3-axis attitude from two or more vector observations, for example, the deterministic TRIAD algorithm [Shuster, 1981; Fisher 1993] as implemented on the Small Astronomy Satallite (SAS), Seasat, Atmospheric Explorer Missions (AEM), Magsat and the Dynamics Explorer (DE) missions. Optimal variations of TRIAD, where a certain weighted loss function is minimized, include the QUEST algorithm of Shuster and Oh [1981], the recursive direction cosine matrix (DCM) estimation method of Bar-Itzhack and Reiner [1984] and the new quaternion attitude estimation method of Markley [1994].

Since the first application of quaternions to the spacecraft attitude problem by Ickes [1970], it has become the most popular means through which to represent the kinematics of orbiting bodies. Friedland [1978] and Shibata [1986] did analysis of strapdown inertial navigation, where the attitude of the sensor box with respect to inertial space is represented using quaternions. Shuster and Oh [1981], Lefferts *et.al.* [1982], Bar-Itzhack and Oshman [1985], Gai *et.al.* [1985], Psiaki *et.al.* [1990] and Markley [1994] to name just a few, all use quaternions in attitude estimation problems.

5.2 Robust Angular Rate Estimation

5.2.1 MOTIVATION

A robust method is needed to measure the body angular rate components of SUNSAT after separation from the launcher and before the gravity gradient boom is deployed. During this initial stage of the satellite's life it can have a certain unspecified angular momentum, with spin vector direction oriented anywhere in inertial space. An angular rate estimator will be needed to implement the detumbling/rate tracking control law of Equation (2.15).

Direct measurements of the body angular rate components are normally done by rate gyroscopes. However, due to power and cost constraints these sensors are seldom considered for a micro satellite application. Body rate values therefore must be computed from attitude sensors. As the orientation of the satellite in inertial space is undefined during this initial phase, an attitude sensor with an omnidirectional application mode will be the preferred one to use. On SUNSAT the magnetometer or the six cubically mounted solar cells can be used. The former sensor gives an accurate measurement of the geomagnetic field vector with respect to the satellite's body and the latter a rough measurement of the sun vector direction with respect to the satellite's body. The rate of change of these vectors relative to the body axes can then be used to extract angular body rate measurements.

The sun vector can for all practical purposes be taken as an inertial reference as the satellite's orbit diameter is about 4 orders of magnitude smaller than it's distance to the sun and the orbit/earth rotates once per year around the sun. Angular rate measurements when using the six solar cell sensor will therefore be inertially referenced. The geomagnetic field vector though, rotates inertially twice per polar orbit and once if referenced to orbital coordinates - see Equation (2.4) for a dipole model. The magnetometer can then be expected to give more accurate orbit referenced angular rate measurements. As the rotation is approximately around the orbit normal direction (Y_o-axis) for a polar orbit, an expected average error in the measured, orbit referenced body angular rate components will be ω_o (the orbital rate) in the orbit normal direction. Simulations of an IGRF modelled geomagnetic field indicate (see Paragraph 5.2.3) that the magnetometer alone can give Kalman filtered body rate estimates at a maximum error of about ω_o.

In the SUNSAT case, the magnetometer will preferably be used to supply measurements to the rate estimator. The reasons being: 1) The control law (2.15) needs orbit referenced angular rates, not inertially referenced rates. 2) The magnetometer gives a much more accurate vector measurement compared to the six solar cells (sun vector).

5.2.2 KALMAN FILTER RATE ESTIMA'.OR

The state vector $\mathbf{x}(k)$, to be estimated, will be the body referenced angular rate vector $\boldsymbol{\omega}_B^O(k)$.

System model:

With an undeployed boom, the satellite can be modelled as being symmetrical with almost equal moments of inert' around all axes, i.e. $I_{xx} = I_{yy} = I_{zz} = I_{xyz}$. The dynamic equation of motion can then l written as,

$$\dot{\boldsymbol{\omega}}_B^I = I_{xyz}^{-1}(\mathbf{N}_M + \mathbf{N}_D) \tag{5.1}$$

Using Equation (1.14) and assuming a near circular orbit, Equation (5.1) can also be written in terms of the orbit referenced angular rates:

$$\dot{\boldsymbol{\omega}}_B^O = I_{xyz}^{-1}(\mathbf{N}_M + \mathbf{N}_D) + \dot{\mathbf{A}}\omega_o \tag{5.2}$$

The last t in Equation (5.2) is normally in the same order of magnitude as the disturbance torque term. It can therefore be modelled with the disturbance torque as system noise. Note: The attitude matrix will in any case be unknown during the initial stages of the satellite s life.

The continuous time system model can then be written as,

$$\dot{\mathbf{x}}(t) = \dot{\boldsymbol{\omega}}_B^O(t) = \mathbf{G}\mathbf{u}(t) + \mathbf{s}(t) \tag{5.3}$$

with,

$$\mathbf{G} = I_{xyz}^{-1}\mathbf{1}, \qquad \mathbf{u}(t) = \mathbf{N}_M(t), \qquad \mathbf{s}(t) = I_{xyz}^{-1}\mathbf{N}_D(t) + \dot{\mathbf{A}}(t)\omega_o$$

and,

$$\mathbf{s}(t) = N\{0, \mathbf{Q}(t)\}$$

(Continuous zero mean system noise with covariance matrix **Q**)

A discrete system model (sampling period T_s) will be used in the Kalman filter:

$$\mathbf{x}(k+1) = \boldsymbol{\omega}_B^O(k+1) = \boldsymbol{\Phi}\mathbf{x}(k) + \boldsymbol{\Gamma}\mathbf{u}(k) + \mathbf{s}(k) \tag{5.4}$$

with,

$$\boldsymbol{\Phi} = 1, \qquad \boldsymbol{\Gamma} = I_{xyz}^{-1}T_s\mathbf{1}$$

and,

$$\mathbf{s}(k) = N\{0, \mathbf{Q}(k)\}$$

(Discrete zero mean system noise with covariance matrix **Q**)

Measurement model:

The measurements for the Kalman filter are obtained by making use of the small-angle approximation of the direction cosine matrix. From Equation (1.2) for a small rotation:

$$\mathbf{A} \approx \begin{bmatrix} 1 & \psi & -\theta \\ -\psi & 1 & \phi \\ \theta & -\phi & 1 \end{bmatrix} \tag{5.5}$$

Arbitrary small rotations between two successive sampling instances can be obtained on a spinning spacecraft by reducing the sampling period T_s. If we further assume almost constant angular rates during sampling period k, the small roll, pitch and yaw rotation angles can be approximated as,

$$\phi(k) \approx \omega_{ox}(k)T_s\ , \qquad \theta(k) \approx \omega_{oy}(k)T_s\ , \qquad \psi(k) \approx \omega_{oz}(k)T_s \tag{5.6}$$

The rotation matrix during sampling period k can then be approximated as,

$$\mathbf{A}\left\{\boldsymbol{\omega}_B^O(k)\right\} \approx 1 + \Lambda\left\{\boldsymbol{\omega}_B^O(k)\right\} \tag{5.7}$$

with,

$$\Lambda\left\{\boldsymbol{\omega}_B^O(k)\right\} = \begin{bmatrix} 0 & \omega_{oz}(k)T_s & -\omega_{oy}(k)T_s \\ -\omega_{oz}(k)T_s & 0 & \omega_{ox}(k)T_s \\ \omega_{oy}(k)T_s & -\omega_{ox}(k)T_s & 0 \end{bmatrix}$$

An orbit referenced vector in body coordinates can then be updated between sampling periods as,

$$\mathbf{v}(k) = \mathbf{A}\left\{\boldsymbol{\omega}_B^O(k)\right\}\mathbf{v}(k-1) \tag{5.8}$$

or,

$$\delta\mathbf{v}(k) = \mathbf{v}(k) - \mathbf{v}(k-1) = \Lambda\left\{\boldsymbol{\omega}_B^O(k)\right\}\mathbf{v}(k-1) \tag{5.9}$$

The time-variant measurement model for a discrete Kalman filter follows directly from Equation (5.9):

$$\mathbf{y}(k) = \delta\mathbf{v}(k) = \mathbf{H}(k)\mathbf{x}(k) + \mathbf{m}(k) \tag{5.10}$$

with,

$$\mathbf{H}(k) = \begin{bmatrix} 0 & -v_z(k-1)T_s & v_y(k-1)T_s \\ v_z(k-1)T_s & 0 & -v_x(k-1)T_s \\ -v_y(k-1)T_s & v_x(k-1)T_s & 0 \end{bmatrix} \tag{5.11}$$

and,

$$\mathbf{m}(k) = N\{\mathbf{0},\mathbf{R}\}$$

(Discrete zero mean measurement noise with covariance matrix **R**)

Kalman filter algorithm:

To reduce the notation, the sampling instant will be indicated in the following text by using a subscribt rather than the bracketing method e.g. $\mathbf{x}_k = \mathbf{x}(k)$. Firstly, define the state covariance matrix as, $\mathbf{P}_k = E\{\mathbf{x}_k.\mathbf{x}_k^T\}$.

Between measurements

1. Propagation of state vector using Equation (5.4):

$$\hat{\mathbf{x}}_{k+1/k} = \hat{\mathbf{x}}_{k/k} + \Gamma\mathbf{u}_k \tag{5.12a}$$

2. Propagation of the perturbation covariance matrix:

$$\mathbf{P}_{k+1/k} = \mathbf{P}_{k/k} + \mathbf{Q} \tag{5.12b}$$

Across measurements

1. Gain update, compute $\mathbf{H}_{k+1}$ from Equation (5.11):

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1/k}\mathbf{H}_{k+1}^T\left[\mathbf{H}_{k+1}\mathbf{P}_{k+1/k}\mathbf{H}_{k+1}^T + \mathbf{R}\right]^{-1} \tag{5.12c}$$

2. State update:

$$\hat{\mathbf{x}}_{k+1/k+1} = \hat{\mathbf{x}}_{k+1/k} + \mathbf{K}_{k+1}\left(\mathbf{y}_{k+1} - \mathbf{H}_{k+1}\hat{\mathbf{x}}_{k+1/k}\right) \tag{5.12d}$$

5. Covariance update:

$$\mathbf{P}_{k+1/k+1} = \left[\mathbf{1} - \mathbf{K}_{k+1}\mathbf{H}_{k+1}\right]\mathbf{P}_{k+1/k} \tag{5.12e}$$

5.2.3 SIMULATION RESULTS

The proposed Kalman filter was evaluated by doing several simulation tests. The SUNSAT satellite and orbit configuration were used as already specified in previous chapters. An eighth order IGRF model was used to obtain the geomagnetic field values. The satellite was initialized with an arbitrary (only limited by the sampling period) angular rate vector. The MOI matrix used, was: **I** = *diag*(2.0, 2.0, 2.0) kgm^2.

A sampling period of T_s = 10 seconds was utilized for the discrete Kalman filter algorithm. The measurement and system noise covariance matrices (**R** and **Q**) were chosen as diagonal and adjusted through simulation optimization to give a Kalman filter with the best filtering properties - smallest tracking error.

Uniformly distributed measurement noise within the range -1 to 1 μT was added to each channel of the magnetometer output. The geomagnetic field vector obtained, was then normalized to obtain the unit vector v(k) to be used in the algorithm of Paragraph 5.2.2. The normalization step is necessary because the body angular rate information is only reflected in the rate of change of the vector direction and not in its magnitude variation.

Figure 5.1 presents a typical performance of the Kalman rate estimator. The initial unknown orbit referenced body angular rate vector was: $\boldsymbol{\omega}_B^O = \begin{bmatrix} 20 & 0 & 40 \end{bmatrix}^T$ rpo. Within about 50 minutes (half an orbit) estimator tracking has been achieved. The real body rate values show small oscillations due to a modulation with the orbital rate ω_o (see Equation 5.2). The Kalman filter's system model ignored these modulation terms due to a lack of any attitude knowledge. A maximum tracking error of $\pm\omega_o$ is therefore expected in the estimated rate components. This also explains the rate errors as shown in Figure 5.2. After about one orbit (100 minutes) the preboom deployment controller (2.15), using the estimated angular rates, is activated and a similar result to Figure 2.2 is obtained. The orbit referenced X and Z-axis angular rates are controlled to near zero and the Y-axis rate to the reference rate of -19 rpo. The satellite is now ready for the deployment of the boom.

The performance of the controller of Equation (2.15) has not been impaired much by using the Kalman filter, although it possesses maximum estimation errors of $\pm\omega_o$. The newly proposed filter also proved to be robust during simulation against measurement noise and always converged within half an orbit from any initial unknown body rate vector value.

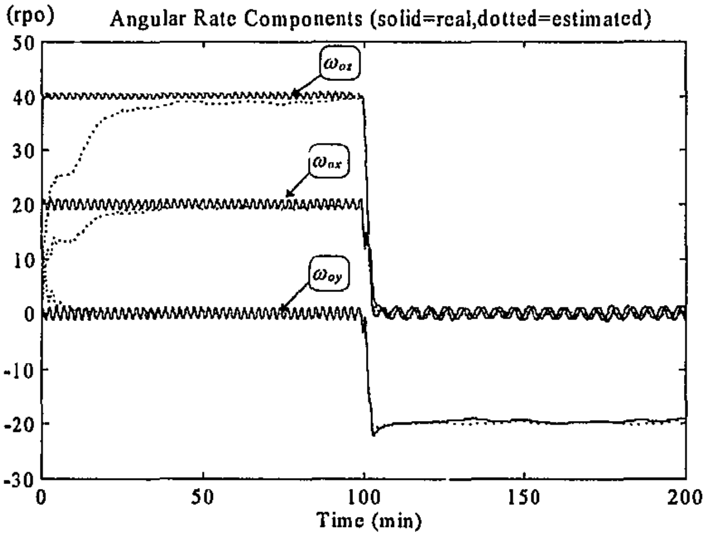

Figure 5.1 A typical performance of the new angular rate Kalman filter

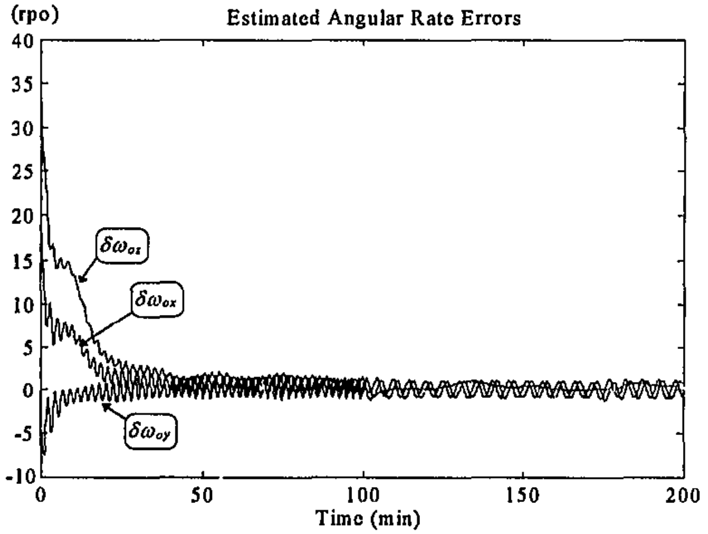

Figure 5.2 The estimated angular rate errors of the new angular rate Kalman filter

5.3 Full Satellite State Determination

5.3.1 BACKGROUND

This paragraph will present two new extended Kalman (EKF) filters to determine the full satellite state from pairs of vector information. The first vector will be obtained from any angular sensor measurement, e.g. utilizing a magnetometer, earth horizon, sun and star sensors. The second vector will be obtained from a reference model of the sensor data in a fixed coordinate frame, e.g. using geomagnetic field, satellite orbit, earth oblateness, sun orbit and star catalogue models.

The use of pairs of vector information to determine the attitude quaternion of a satellite has been treated in the literature to some extent. Shuster and Oh [1981] presented batch least square algorithms (TRIAD and QUEST) to determine the direction cosine matrix from a set of vector measurements. Bar-Itzhack and Reiner [1984] extended the idea to recursive algorithms. Lefferts *et.al.* [1982] and Bar-Itzhack and Oshman [1985] also showed how Kalman filtering can be used to determine the attitude. Both these algorithms assumed the usage of rate gyroscopes to obtain the angular velocities. Psiaki *et.al.* [1990] presented a Kalman filter estimating attitude, rates and constant disturbance torques similar to the new algorithm dealt with in this paper. Their filter, however, uses an error quaternion (the difference between the true and estimated quaternion is also a quaternion - thus implying quaternion multiplication), while the filter presented here uses a perturbation vector difference between the true and estimated quaternion. The innovation in Psiaki's paper [1990] uses the cross-product to obtain the rotational error between the pair of vectors, while the innovation in the new algorithm uses the standard vector difference method. The final significant difference is that Psiaki *et.al.* [1990] assumed a 3-axis stabilized nominal earth pointing satellite, while the algorithms presented here can be used for spinning (e.g. yaw spin) satellites as well.

A slightly modified version of the EKF estimator of Psiaki *et.al.* [1990] will also be presented to compare its simulation performance to the new EKF algorithm. The main reason for the modification will be to extend the algorithm in Psiaki's paper to spinning satellites.

Two implementations of the basic EKF algorithms will be used for SUNSAT. The first EKF will make use of magnetometer measurements and an IGRF geomagnetic field model to supply the vector pairs needed for determination of the satellite's attitude to an angular accuracy of below 1°. The second EKF will use measurements from the higher accuracy sensors (CCD type horizon, sun and star sensors) and orbital models plus star catalogues to determine SUNSAT's attitude to below 0.1° angular accuracy.

The derivations of the EKF algorithms presented in the rest of this chapter will make the following assumptions:

1. The satellite will be axially symmetric with a deployed boom along the body Z-axis. The MOI matrix will be,

$$\mathbf{I} = \begin{bmatrix} I_T & 0 & 0 \\ 0 & I_T & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{5.13}$$

with,

$I_T = I_{xx} = I_{yy}$ = transverse moment of inertia

2. The satellite orbit will be near circular with almost constant angular rate ω_o. The gravity gradient torque vector of equation (1.8) can then be approximated as,

$$\mathbf{N}_{GG} = 3\omega_o^2 (I_T - I_{zz}) A_{33} \begin{bmatrix} -A_{23} \\ A_{13} \\ 0 \end{bmatrix} \tag{5.14}$$

3. The satellite will nominally be earth pointing with either a certain Z-spin rate or be 3-axis stabilized.

4. The disturbance torque vector in body coordinates $\mathbf{N}_D$ is obtained from a slowly varying (constant) disturbance torque n_{doy} around the orbital Y_o-axis,

$$\mathbf{N}_D = \mathbf{A}\begin{bmatrix} 0 & n_{doy} & 0 \end{bmatrix}^T \tag{5.15}$$

where,

$$\dot{n}_{doy} = 0 \tag{5.16}$$

The disturbance torque n_{doy} is modelled as a random walk (slowly varying) process acting only around the orbital Y_o-axis. This torque is generated mainly by aerodynamic pressure on the earth pointed satellite body (see Paragraph 1.6.2). However, due to the axial symmetry of the satellite, little disturbance torque will occur around the body Z-axis. To ensure that no disturbance torque will occur around the body Z-axis, the last row of the DCM **A** in Equation (5.15) will normally be zeroed.

For an elliptical orbit the assumption of a slow torque variation will no longer be valid due to the large variation in atmospheric density between perigee and apogee. An improved disturbance model will then be: (see Figure 1.10)

$$\mathbf{N}_D = \mathbf{A}\begin{bmatrix}0 & n_{doy}\{1.0+\cos(\omega_o t)\} & 0\end{bmatrix}^T \qquad (5.17)$$

where,

$\dot{n}_{doy} = 0$, and t = time since perigee

5.3.2 NEW EXTENDED KALMAN FILTER

The continuous full (8 element) state vector to be estimated can be defined as:

$$\mathbf{x}(t) = \begin{bmatrix}\boldsymbol{\omega}_B^{I\,T}(t) & \mathbf{q}^T(t) & n_{doy}(t)\end{bmatrix}^T \qquad (5.18)$$

System model:

The full non-linear model of the satellite can then be obtained using Equations (1.7), (1.12) and (5.16),

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t),t) + \mathbf{s}(t) \qquad (5.19)$$

where,

$$\mathbf{s}(t) = N\{\mathbf{0},\mathbf{Q}(t)\} \qquad (5.20)$$

(Zero mean white system noise with covariance matrix **Q**)

If we define the state perturbation $\delta\mathbf{x}$ as the difference between the actual state and the estimated state,

$$\delta\mathbf{x}(t) = \mathbf{x}(t) - \hat{\mathbf{x}}(t) \qquad (5.21)$$

a first order Taylor expansion of the non-linear state function can be done:

$$\mathbf{f}(\mathbf{x}(t),t) \approx \mathbf{f}(\hat{\mathbf{x}}(t),t) + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}} \delta\mathbf{x}(t) \qquad (5.22)$$

Define,

$$\mathbf{F}(\hat{\mathbf{x}}(t),t) = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}}$$

The linearized perturbation state model to be used by the EKF is then,

$$\delta\dot{\mathbf{x}}(t) = \mathbf{F}(\hat{\mathbf{x}}(t),t)\,\delta\mathbf{x}(t) + \mathbf{s}(t) \qquad (5.23)$$

From Equations (1.7), (1.12) and (5.16) follows for the satellite model,

$$\delta\dot{\boldsymbol{\omega}} = \mathbf{I}^{-1}\left[\delta\mathbf{N}_{GG} + \delta\mathbf{N}_{D} - \delta\boldsymbol{\omega} \times \left(\mathbf{I}\,\boldsymbol{\omega}_{B}^{I} + \mathbf{h}\right) - \boldsymbol{\omega}_{B}^{I} \times \mathbf{I}\,\delta\boldsymbol{\omega}\right] \tag{5.24}$$

and,

$$\delta\dot{\mathbf{q}} = \frac{1}{2}\hat{\boldsymbol{\Omega}}\delta\mathbf{q} + \hat{\boldsymbol{\beta}}\delta\boldsymbol{\omega} \tag{5.25}$$

and,

$$\delta\dot{n}_{doy} = 0 \tag{5.26}$$

where,

$$\delta\mathbf{N}_{GG} = 6\omega_{o}^{2}\left(I_{T} - I_{zz}\right) \times$$

$$\begin{bmatrix} -\hat{A}_{33}\hat{q}_{4} + \hat{A}_{23}\hat{q}_{1} & -\hat{A}_{33}\hat{q}_{3} + \hat{A}_{23}\hat{q}_{2} & -\hat{A}_{33}\hat{q}_{2} - \hat{A}_{23}\hat{q}_{3} & -\hat{A}_{33}\hat{q}_{1} - \hat{A}_{23}\hat{q}_{4} \\ \hat{A}_{33}\hat{q}_{3} - \hat{A}_{13}\hat{q}_{1} & -\hat{A}_{33}\hat{q}_{4} - \hat{A}_{13}\hat{q}_{2} & \hat{A}_{33}\hat{q}_{1} + \hat{A}_{13}\hat{q}_{3} & -\hat{A}_{33}\hat{q}_{2} + \hat{A}_{13}\hat{q}_{4} \\ 0 & 0 & 0 & 0 \end{bmatrix}\delta\mathbf{q} \tag{5.27}$$

$$\delta\mathbf{N}_{D} = \begin{bmatrix} \hat{A}_{12} \\ \hat{A}_{22} \\ 0 \end{bmatrix}\delta n_{dyo} + 2\begin{bmatrix} \hat{q}_{2}\hat{n}_{dyo} & \hat{q}_{1}\hat{n}_{dyo} & \hat{q}_{4}\hat{n}_{dyo} & \hat{q}_{3}\hat{n}_{dyo} \\ -\hat{q}_{1}\hat{n}_{dyo} & \hat{q}_{2}\hat{n}_{dyo} & -\hat{q}_{3}\hat{n}_{dyo} & \hat{q}_{4}\hat{n}_{dyo} \\ 0 & 0 & 0 & 0 \end{bmatrix}\delta\mathbf{q} \tag{5.28}$$

$\hat{\boldsymbol{\Omega}}$ is similarly defined as Equation (1.13), but using the estimated rate values

$$\hat{\boldsymbol{\beta}} = \frac{1}{2}\begin{bmatrix} \hat{q}_{4} & -\hat{q}_{3} & \hat{q}_{2} \\ \hat{q}_{3} & \hat{q}_{4} & -\hat{q}_{1} \\ -\hat{q}_{2} & \hat{q}_{1} & \hat{q}_{4} \\ -\hat{q}_{1} & -\hat{q}_{2} & -\hat{q}_{3} \end{bmatrix} \tag{5.29}$$

A discrete version of the EKF will be used, therefore a discrete perturbation model will be:

$$\delta\mathbf{x}_{k+1} = e^{\mathbf{F}(\hat{\mathbf{x}}(t_k),t_k)\delta\mathrm{T}}\delta\mathbf{x}_{k} = \boldsymbol{\Phi}_{k}\delta\mathbf{x}_{k} \tag{5.30}$$

The discrete system matrix can be approximated for a short sampling period δT as,

$$\boldsymbol{\Phi}_{k} \approx \left[\mathbf{I} + \mathbf{F}\left(\hat{\mathbf{x}}(t_{k}),t_{k}\right)\delta\mathrm{T}\right] \tag{5.31}$$

Measurement model:

For an EKF a discrete non-linear measurement model is normally assumed [Gelb, 1989]·

$$\mathbf{y}_{k} = \mathbf{h}_{k}\left(\mathbf{x}(t_{k}),t_{k}\right) + \mathbf{m}_{k} \tag{5.32}$$

where,

$$\mathbf{m}_k = N\{\mathbf{0},\mathbf{R}_k\} \tag{5.33}$$

(Discrete zero mean white measurement noise with covariance matrix **R**)

We can now use a first order Taylor expansion of the non-linear output function:

$$\mathbf{h}_k(\mathbf{x}(t_k),t_k) \approx \mathbf{h}(\hat{\mathbf{x}}(t_k),t_k) + \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}} \delta\mathbf{x}_k \tag{5.34}$$

Define,

$$\mathbf{H}_k(\hat{\mathbf{x}}_k) = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}}$$

Then the linearized innovation error model to be used by the EKF:

$$\mathbf{e}_k = \mathbf{H}_k(\hat{\mathbf{x}}_k)\delta\mathbf{x} + \mathbf{m}_k \tag{5.35}$$

Innovation computation:

The derivation presented here is similar to the one showed by Bar-Itzhack and Oshman [1985]. In this new EKF algorithm the innovation is computed as the vector difference between a measured normalized vector in body coordinates $\mathbf{v}_{meas}$ and a modelled normalized vector $\mathbf{v}_{body}$ in body coordinates. The modelling is actually done in orbital coordinates and the vector $\mathbf{v}_{orb}$ is then transformed by the DCM **A** to $\mathbf{v}_{body}$. The vectors are normalized to minimize the effect of any magnitude errors due to inaccurate modelling while still preserving all the directional information.

For an error free measurement and modelling process:

$$\mathbf{v}^{true}_{meas,k} = \mathbf{A}(\mathbf{q}_k)\mathbf{v}^{true}_{orb,k} = \mathbf{v}^{true}_{body,k} \tag{5.36}$$

With measurement ($\mathbf{m}_{meas}$) plus modelling ($\mathbf{m}_{orb}$) noise (errors) and using the estimated quaternion, Equation (5.34) now becomes:

$$\mathbf{v}_{meas,k} - \mathbf{m}_{meas,k} = \mathbf{A}(\hat{\mathbf{q}}_k + \delta\mathbf{q})(\mathbf{v}_{orb,k} - \mathbf{m}_{orb,k}) \tag{5.37}$$

A first order Taylor expansion can be used on the DCM:

$$\mathbf{A}(\hat{\mathbf{q}}_k + \delta\mathbf{q}_k) \approx \mathbf{A}(\hat{\mathbf{q}}_k) + \sum_{i=1}^{4}\frac{\delta\mathbf{A}(\hat{\mathbf{q}}_k)}{\delta q_{i,k}}\delta q_{i,k} \tag{5.38}$$

Then the innovation will be,

$$\begin{aligned}
\mathbf{e}_k &= \mathbf{v}_{meas,k} - \mathbf{A}(\hat{\mathbf{q}}_k)\mathbf{v}_{orb,k} \\
&= \left[\sum_{i=1}^{4} \frac{\delta \mathbf{A}(\hat{\mathbf{q}}_k)}{\delta q_{i,k}} \delta q_{i,k}\right] \mathbf{v}_{orb,k} + \mathbf{m}_k \\
&= \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3 & \mathbf{h}_4 \end{bmatrix} \delta \mathbf{q}_k + \mathbf{m}_k \qquad (5.39) \\
&= \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3 & \mathbf{h}_4 & \mathbf{0}_{3\times1} \end{bmatrix} \delta \mathbf{x}_k + \mathbf{m}_k \\
&= \mathbf{H}_k(\hat{\mathbf{q}}_k)\delta \mathbf{x}_k + \mathbf{m}_k
\end{aligned}$$

with,

$$\mathbf{h}_i = \frac{\delta \mathbf{A}(\hat{\mathbf{q}}_k)}{\delta q_{i,k}} \mathbf{v}_{orb,k} \quad , \quad i = 1,2,3,4 \qquad (5.40)$$

e.g.

$$\mathbf{h}_1 = 2\begin{bmatrix} \hat{q}_{1,k} & \hat{q}_{2,k} & \hat{q}_{3,k} \\ \hat{q}_{2,k} & -\hat{q}_{1,k} & \hat{q}_{4,k} \\ \hat{q}_{3,k} & -\hat{q}_{4,k} & -\hat{q}_{1,k} \end{bmatrix} \mathbf{v}_{orb,k} \qquad (5.41)$$

and, (ignoring the small higher order noise terms)

$$\mathbf{m}_k \approx \mathbf{m}_{meas,k} - \mathbf{A}(\hat{\mathbf{q}}_k)\mathbf{m}_{orb,k} \qquad (5.42)$$

EKF algorithm:

Firstly, define the perturbation covariance matrix as,

$$\mathbf{P}_k = E\left\{\delta \mathbf{x}_k . \delta \mathbf{x}_k{}^T\right\}$$

Between measurements

1. Propagation of the full state Equations (1.7) and (1.12), by using a numerical integration method:

$$\hat{\mathbf{x}}_{k+1/k} = \hat{\mathbf{x}}_{k/k} + \int_k^{k+1} \mathbf{f}_k\left(\hat{\mathbf{x}}_{k/k}, k\right) dt \qquad (5.43a)$$

2. Propagation of the perturbation covariance matrix, compute $\mathbf{\Phi}$ from Equation (5.31):

$$\mathbf{P}_{k+1/k} = \mathbf{\Phi}_{k+1/k}\mathbf{P}_{k/k}\mathbf{\Phi}_{k+1/k}^T + \mathbf{Q}_{k+1} \qquad (5.43b)$$

Across measurements

1. Gain update, compute $\mathbf{H}_{k+1/k}$ with the propagated state from Equations (5.39) and (5.40):

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1/k}\mathbf{H}^T_{k+1/k}\left[\mathbf{H}_{k+1/k}\mathbf{P}_{k+1/k}\mathbf{H}^T_{k+1/k} + \mathbf{R}_{k+1}\right]^{-1} \quad (5.43c)$$

2. Obtain the innovation:

$$\mathbf{e}_{k+1} = \mathbf{v}_{meas,k+1} - \hat{\mathbf{v}}_{body,k+1/k} \quad (5.43d)$$

3. Perturbation update:

$$\delta\mathbf{x}_{k+1} = \mathbf{K}_{k+1}\mathbf{e}_{k+1} \quad (5.43e)$$

4. State update:

$$\hat{\mathbf{x}}_{k+1/k+1} = \hat{\mathbf{x}}_{k+1/k} + \delta\mathbf{x}_{k+1} \quad (5.43f)$$

5. Covariance update, recompute $\mathbf{H}_{k+1/k+1}$ with updated state:

$$\mathbf{P}_{k+1/k+1} = \left[1-\mathbf{K}_{k+1}\mathbf{H}_{k+1/k+1}\right]\mathbf{P}_{k+1/k}\left[1-\mathbf{K}_{k+1}\mathbf{H}_{k+1/k+1}\right]^T + \mathbf{K}_{k+1}\mathbf{R}_{k+1}\mathbf{K}^T_{k+1} \quad (5.43g)$$

In step 4, the quaternion elements of the updated state vector are normalized each time by dividing each quaternion element by the updated quaternion norm:

$$\hat{\mathbf{x}}_{norm,k+1} = \frac{\hat{\mathbf{x}}_{k+1/k+1}}{\left\|\hat{\mathbf{x}}_{k+1/k+1}\right\|} \quad (5.44)$$

This step is done to preserve the normalization constraint of a quaternion. It was also shown by Bar-Itzhack and Oshman [1985] that the convergence performance improves significantly when this is done. The normalization step does not result in a full reset of the quaternion perturbation and the unresetted part needs to be propagated between measurements. However, simulations indicates no improvement in the EKF performance when this was done. The added complexity to the algorithm was therefore not considered to be worthwhile.

5.3.3 MODIFIED PSIAKI EXTENDED KALMAN FILTER

The original EKF algorithm presented in this paragraph was first proposed by Psiaki *et.al.* [1990] for a 3-axis stabilized nadir pointing satellite, using only magnetometer data. In order to compare their filter to the new EKF of this thesis, it is now modified and extended to handle Z-spinning satellites as well. Duplication of the equations presented in the previous paragraph will be avoided and all the implementation differences will be highlighted.

The state vector to be estimated is similar to Equation (5.18), but the EKF will only be used to compute a 7 element state perturbation vector, i.e. one element less than the New EKF. The reason for this reduction in state dimension is: The attitude quaternion has been linearized in a special way. Instead of expressing the actual quaternion in terms of an estimated value plus a perturbation (see Equation (5.21)), it is expressed in terms of a perturbation quaternion times the estimated quaternion using quaternion multiplication:

$$\mathbf{q}(t) = \hat{\mathbf{q}}(t) \otimes \begin{bmatrix} \delta\mathbf{q}(t) \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & \delta q_3 & -\delta q_2 & \delta q_1 \\ -\delta q_3 & 1 & \delta q_1 & \delta q_2 \\ \delta q_2 & -\delta q_1 & 1 & \delta q_3 \\ -\delta q_1 & -\delta q_2 & -\delta q_3 & 1 \end{bmatrix} \begin{bmatrix} \hat{q}_1(t) \\ \hat{q}_2(t) \\ \hat{q}_3(t) \\ \hat{q}_4(t) \end{bmatrix} \qquad (5.45)$$

The perturbation quaternion has just three unknowns, the fourth is not needed because the perturbation Euler angle is small and no attitude singularity can occur.

System model:

The full non-linear model of the satellite can again be obtained from Equations (1.7), (1.12) and (5.16). The linearized perturbation state model (Equation (5.23)) to be used, will now be given by:

$$\delta\dot{\boldsymbol{\omega}} = \mathbf{I}^{-1}\left[\delta\mathbf{N}_{GG} + \delta\mathbf{N}_D - \delta\boldsymbol{\omega} \times \left(\mathbf{I}\,\boldsymbol{\omega}_B^I + \mathbf{h}\right) - \boldsymbol{\omega}_B^I \times \mathbf{I}\,\delta\boldsymbol{\omega}\right] \qquad (5.46)$$

and,

$$\delta\dot{\mathbf{q}} = \frac{1}{2}\delta\boldsymbol{\omega} \qquad (5.47)$$

and,

$$\delta\dot{n}_{doy} = 0 \qquad (5.48)$$

where,

$$\delta\mathbf{N}_{GG} = 6\omega_o^2\left(I_T - I_{zz}\right) \begin{bmatrix} \hat{A}_{23}^2 - \hat{A}_{33}^2 & -\hat{A}_{13}\hat{A}_{23} & \hat{A}_{13}\hat{A}_{33} \\ -\hat{A}_{13}\hat{A}_{23} & \hat{A}_{13}^2 - \hat{A}_{33}^2 & \hat{A}_{23}\hat{A}_{33} \\ 0 & 0 & 0 \end{bmatrix} \delta\mathbf{q} \qquad (5.49)$$

$$\delta\mathbf{N}_D = \begin{bmatrix} \hat{A}_{12} \\ \hat{A}_{22} \\ 0 \end{bmatrix} \delta n_{dyo} + 2\begin{bmatrix} 0 & -\hat{A}_{32}n_{dyo} & \hat{A}_{22}n_{dyo} \\ \hat{A}_{32}n_{dyo} & 0 & -\hat{A}_{12}n_{dyo} \\ 0 & 0 & 0 \end{bmatrix}\delta\mathbf{q} \tag{5.50}$$

The discrete perturbation system matrix will be computed similarly to Equations (5.30) and (5.31).

Measurement model:

The discrete non-linear measurement model will be linearized similarly to Equation (5.34) to obtain the innovation error model of Equation (5.35).

Innovation computation:

The innovation is computed as the cross-product between the normalized measurement $\mathbf{v}_{meas}$ and modelled $\mathbf{v}_{body}$ vector pair in body coordinates. The modelling is actually done in orbital coordinates and the vector $\mathbf{v}_{orb}$ has to be transformed by the DCM **A** to the vector $\mathbf{v}_{body}$ in body coordinates. The innovation can then be written as,

$$\mathbf{e}_k = \mathbf{v}_{meas,k} \times \hat{\mathbf{v}}_{body,k} = \mathbf{v}_{meas,k} \times \mathbf{A}(\hat{\mathbf{q}}_k)\mathbf{v}_{orb,k} \tag{5.51}$$

For an error free measurement and modelling process Equation (5.36) still applies. With measurement plus modelling noise (errors) and using the estimated values the error free Equation (5.36) becomes:

$$\mathbf{v}_{meas,k} - \mathbf{m}_{meas,k} = \mathbf{A}\left(\begin{bmatrix} \delta\mathbf{q}_k \\ 1 \end{bmatrix}\right)\mathbf{A}(\hat{\mathbf{q}}_k)\left(\mathbf{v}_{orb,k} - \mathbf{m}_{orb,k}\right) \tag{5.52}$$

therefore,

$$\mathbf{v}_{meas,k} = \mathbf{A}\left(\begin{bmatrix} \delta\mathbf{q}_k \\ 1 \end{bmatrix}\right)\hat{\mathbf{v}}_{body,k} + \tilde{\mathbf{m}}_k \tag{5.53}$$

with,

$$\mathbf{A}\left(\begin{bmatrix} \delta\mathbf{q}_k \\ 1 \end{bmatrix}\right) \approx \begin{bmatrix} 1 & 2\delta q_3 & -2\delta q_2 \\ -2\delta q_3 & 1 & 2\delta q_1 \\ 2\delta q_2 & -2\delta q_1 & 1 \end{bmatrix}$$

$$\tilde{\mathbf{m}}_k = \mathbf{m}_{meas,k} - \mathbf{A}(\mathbf{q}_k)\mathbf{m}_{orb,k} \tag{5.54}$$

The innovation vector of Equation (5.51) can then be written as,

$$\begin{aligned}
\mathbf{e}_k &= \mathbf{A}\left(\begin{bmatrix}\delta\mathbf{q}_k \\ 1\end{bmatrix}\right)\hat{\mathbf{v}}_{body,k} \times \hat{\mathbf{v}}_{body,k} + \tilde{\mathbf{m}}_k \times \hat{\mathbf{v}}_{body,k} \\
&= 2\begin{bmatrix} \hat{v}^2_{by,k} + \hat{v}^2_{bz,k} & -\hat{v}_{bx}\hat{v}_{by} & -\hat{v}_{bx}\hat{v}_{bz} \\ -\hat{v}_{bx}\hat{v}_{by} & \hat{v}^2_{bx,k} + \hat{v}^2_{bz,k} & -\hat{v}_{by}\hat{v}_{bz} \\ -\hat{v}_{bx}\hat{v}_{bz} & -\hat{v}_{by}\hat{v}_{bz} & \hat{v}^2_{bx,k} + \hat{v}^2_{by,k} \end{bmatrix}\delta\mathbf{q}_k + \mathbf{m}_k \\
&= \tilde{\mathbf{H}}_k\left(\hat{\mathbf{v}}_{body,k}\right)\delta\mathbf{q}_k + \mathbf{m}_k \\
&= \left[\mathbf{0}_{3\times3} \quad \tilde{\mathbf{H}}_k\left(\hat{\mathbf{v}}_{body,k}\right) \quad \mathbf{0}_{3\times1}\right]\delta\mathbf{x}_k + \mathbf{m}_k \qquad (5.55)\\
&= \mathbf{H}_k\left(\hat{\mathbf{v}}_{body,k}\right)\delta\mathbf{x}_k + \mathbf{m}_k
\end{aligned}$$

$$\mathbf{m}_k = \tilde{\mathbf{m}}_k \times \hat{\mathbf{v}}_{body,k}$$

EKF algorithm:

The algorithm is similar to the previous one presented in Equations (5.43a) to (5.43g). In step 4, however, the quaternion part of the state vector is updated not by vector addition but by quaternion multiplication:

$$\hat{\mathbf{q}}_{k+1/k+1} = \hat{\mathbf{q}}_{k+1/k} \otimes \begin{bmatrix}\delta\mathbf{q}_k \\ \sqrt{1 - \|\delta\mathbf{q}_k\|}\end{bmatrix} \qquad (5.56)$$

The scalar part of the full perturbation quaternion is not chosen as 1, as in Equation (5.45), but chosen to preserve the normalization constraint of the quaternion.

5.3.4 VECTOR COMPUTATIONS FOR EKF

Innovation data for both EKF types presented, can be obtained from any attitude sensor able to supply vector directional measurements. For example: A magnetometer measuring the geomagnetic field vector and giving a full 3-degree of freedom direction. An optical earth horizon sensor measuring the angle within a satellite referenced plane towards the horizon where the illuminated earth's disc, as seen from the satellite, meets the dark space background. Such a sensor only produces an unit vector with 1-degree of freedom. A sun sensor measures the angular direction towards the sun, within a satellite referenced plane. This sensor also produces an unit vector with only 1-degree of freedom. The combination of two horizon sensors and a sun sensor, all measuring in different satellite referenced planes, will then be able to supply full 3-dimensional attitude information.

All the sensor measured vectors must be paired with modelled vectors, as explained in the previous paragraphs. These modelled vectors give the expected directional information of the sensors, whenever the satellite's body coordinate frame is aligned to a certain reference coordinate frame. In SUNSAT's case the reference coordinate frame will be orbit-defined.

Two EKF full state estimators will be employed on SUNSAT. The first EKF will use a magnetometer to supply full 3-dimensional directional measurements on a continuous basis. The modelling vector to be paired to the measurement vector will be obtained from an IGRF geomagnetic field model. Due to uncertainties in the IGRF model, giving rise to 1-σ errors of 0.4° per axis [Psiaki, 1990], this EKF will not be able supply highly accurate attitude information.

The second EKF will use two orthogonal looking earth horizon sensors and a sun sensor in a third plane, orthogonal to the horizon sensor planes. Due to FOV limitations on all these sensors they will not supply continuous or even simultaneous innovations to the filter. The Kalman filter, therefore, will run in open-loop (it executes only the prediction part of the algorithm) when no innovations are available. The horizon and sun sensors plus models are able to supply attitude data to high accuracy (better than 0.1° per axis). The second EKF are hence expected to provide highly accurate attitude information, especially during the periods when state updates are taking place. 3-Dimensional measurements from a star mapper can also be added for an even further improvement to the filter's performance.

5.3.4.1 Magnetometer EKF

Every 10 seconds, 3-axis measurements will be taken from a fluxgate magnetometer. For an improved measurement accuracy the magnetometer will be continuously calibrated while in orbit (Chapter 6). The geomagnetic field vector measured in spacecraft body coordinates will then be normalized to obtain the measured innovation vector:

$$\mathbf{v}_{meas,k} = \frac{\mathbf{B}_{meas,k}}{\left\|\mathbf{B}_{meas,k}\right\|} \qquad (5.57)$$

Every 10 seconds, an eighth-order IGRF model will recalculate the reference geomagnetic field vector in orbit coordinates (Appendix A). The model needs as inputs, the sub-satellite latitude plus longitude and the satellite's geocentric distance.

To obtain these parameters in real-time, an onboard orbital model of the satellite must be utilized. The modelled innovation vector in orbital coordinates will then be,

$$\mathbf{v}_{orb,k} = \frac{\mathbf{B}_{o,k}}{\left\|\mathbf{B}_{o,k}\right\|} \tag{5.58}$$

5.3.4.2 Horizon/Sun sensor EKF

The sensor placement is done such that the two horizon sensors and one sun sensor each measure attitude angles in a different body plane. The arrangement on SUNSAT is as shown in figure 5.3. The sun sensor measure the sun's azimuth angle within the XY body plane, limited to ± 60° from the nominal -Y direction. The X-horizon sensor measures the elevation angle within the XZ body plane, limited to ± 15° around the nominal horizon at 24.3° from the X-axis towards the Z-axis (nadir). The latter angle is the average angle at an altitude of 620 km (for SUNSAT's 400 km to 840 km orbit). The Y-horizon sensor measures the elevation angle within the YZ body plane, limited to ± 15° around the nominal horizon at 24.3° from the -Y-axis towards the Z-axis.

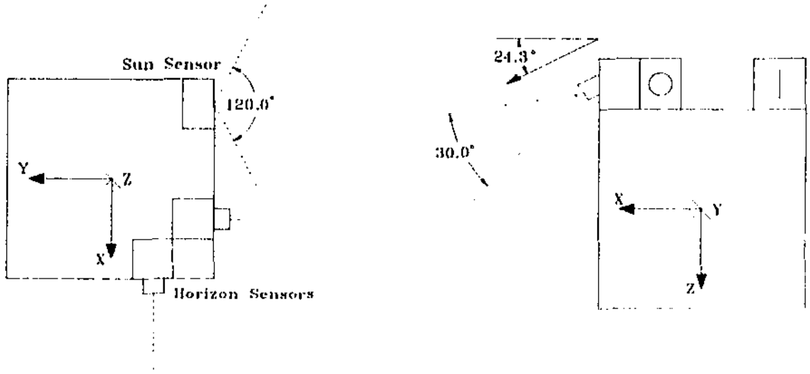

Figure 5.3 Placement of the horizon and sun sensors on SUNSAT

To determine whether the horizon sensors are seeing a valid horizon and not looking at the sun's terminator on the earth, orbit models of both the sun and satellite have to be implemented (a geocentered system is assumed). From the respective orbital parameters the terminator can be determined and each horizon sensor output validated. The spheroidal shape of the earth must also be modelled to determine the horizon angles to the required accuracy.

The innovation vector pairs for the respective sensors can then be computed by executing the following steps every second, when valid measurements are available:

X-Horizon sensor

1. Obtain the sensor horizon angle measurement within the XZ plane and compute the measured unit vector,

$$\mathbf{v}^{hx}_{meas,k} = \begin{bmatrix} \cos\theta_{hx} & 0 & \sin\theta_{hx} \end{bmatrix}^T \tag{5.59}$$

with,

θ_{hx} = sensor horizon angle measurement from the X to Z-axis

2. Compute the estimated azimuth angle in the X_oY_o local horizon plane from the measurement vector and the estimated DCM,

$$\hat{Azim}_k = \arctan\left\{\frac{\hat{v}_{oy,k}}{\hat{v}_{ox,k}}\right\} \tag{5.60}$$

with,

$$\begin{bmatrix} \hat{v}_{ox,k} & \hat{v}_{oy,k} & \hat{v}_{oz,k} \end{bmatrix}^T = \mathbf{A}^T(\hat{\mathbf{q}}_k)\mathbf{v}^{hx}_{meas,k} \tag{5.61}$$

3. Compute the modelled unit vector, in orbit coordinates, from the earth oblateness model and the estimated azimuth angle,

$$\mathbf{v}^{hx}_{orb,k} = \begin{bmatrix} \cos Elev \cdot \cos\hat{Azim} & \cos Elev \cdot \sin\hat{Azim} & \sin Elev \end{bmatrix}^T \tag{5.62}$$

where, [Wertz, 1986]

$$Elev = \frac{\pi}{2} - \operatorname{arccot}\left\{\begin{array}{l} \sqrt{\dfrac{\left(R_s^2 - R_e^2\right)}{a^2}\left[1 + \dfrac{(2-f)fR_e^2\cos^2\lambda}{(1-f)^2a^2}\sin^2\Psi\right]} \\ + \dfrac{(2-f)fR_e^2\sin 2\lambda}{2(1-f)^2a^2}\sin\Psi \end{array}\right\} \tag{5.63}$$

with,

$$R_e = \frac{a(1-f)}{\sqrt{1-(2-f)f\cos^2\lambda}} = \text{radius of oblate earth at latitude } \lambda \tag{5.64}$$

R_s = distance from geocenter to satellite

a = 6378.14 km (earth's radius at the equator)

f = 0.00335281 (earth's flattening factor)

Ψ = azimuth angle of the horizon vector, within the X_oY_o plane, measured from the east direction.

The azimuth and elevation angles are defined as shown in Figure 5.4.

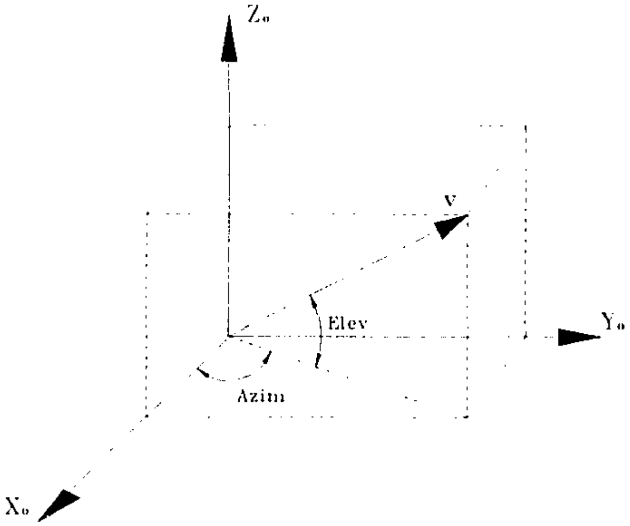

Figure 5.4 Definition of the azimuth and elevation angles within the orbit coordinates

Y-Horizon sensor

1. Obtain the sensor horizon angle measurement within the YZ plane and compute the measured unit vector,

$$\mathbf{v}_{meas,k}^{hy} = \begin{bmatrix} 0 & -\cos\theta_{hy} & \sin\theta_{hy} \end{bmatrix}^T \tag{5.65}$$

with,

θ_{hy} = sensor horizon angle measurement from the -Y to Z-axis

2. Compute the estimated azimuth angle in the X_oY_o local horizon plane from the measurement vector and the estimated DCM similarly to Equations (5.60) and (5.61).

3. Compute the modelled unit vector, in orbit coordinates, from the earth oblateness model and the estimated azimuth angle similarly to Equations (5.62) to (5.64)

Sun sensor

1. Compute the modelled sun unit vector, in orbit coordinates, by making use of the satellite and sun orbit models:

$$\mathbf{v}_{orb,k}^{sun} = \begin{bmatrix} v_{ox,k}^{sun} & v_{oy,k}^{sun} & v_{oz,k}^{sun} \end{bmatrix}^T \tag{5 66}$$

2. Obtain the sun azimuth angle measurement in the XY body plane and the estimated Z-axis component of the sun unit vector calculated from the modelled vector and the estimated DCM,

$$\mathbf{v}^{sun}_{meas,k} = \left[\sqrt{1-\hat{v}^{sun}_{z,k}}\cos Azim \quad \sqrt{1-\hat{v}^{sun}_{z,k}}\sin Azim \quad \hat{v}^{sun}_{z,k}\right] \tag{5.67}$$

where,

$$Azim = \theta_{sun} - \frac{\pi}{2} \tag{5.68}$$

$$\hat{v}^{sun}_{z,k} = \hat{A}_{31}v^{sun}_{ox} + \hat{A}_{32}v^{sun}_{oy} + \hat{A}_{33}v^{sun}_{oz} \tag{5.69}$$

with,

θ_{sun} = sensor sun angle, measured around the -Y-axis direction

EKF update

The update part of the EKF algorithm (Equations (5.43c) to (5.43g)) is then repeatedly executed each sampling period for each innovation vector pair available. If no sensor can supply valid measurements, the update part will be skipped and the filter will be run in open-loop. Each innovation pair per sensor has additional attitude information due to the non-alignment of the various measurements. The repeated update method, therefore, has to improve the estimation error each time. The iterative EKF method is a well known higher-order nonlinear filtering technique used to improve the estimation error, also with only one set of innovation data [Gelb, 1989].

5.3.5 EKF PROPAGATION OF THE SATELLITE STATE

According to Equation (5.43a) the satellite's state has to be propagated between measurements. The nonlinear dynamic equation (1.7) and kinematic equation (1.12) must be numerically integrated to obtain a discrete solution of the state at each EKF sampling interval. Numerical integration solutions will never be exact but suffer from truncation errors, roundoff errors and even error unstability. Unstable behaviour occurs where errors introduced at some stage in the calculation propagate without bound as the integration proceeds.

According to Wertz [1986] the most common numerical integration method used for spacecraft state propagation, is the classical R-stage Runge-Kutta method. For example, the 4-stage Runge-Kutta method, which introduces errors at each stage of the order h^5 (h is the integration step size):

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{h}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4) \tag{5.70a}$$

with,

$$\begin{aligned} \mathbf{k}_1 &= \mathbf{f}(\mathbf{x}_k, k) \\ \mathbf{k}_2 &= \mathbf{f}\left(\mathbf{x}_k + \frac{1}{2}h\mathbf{k}_1, k + \frac{1}{2}h\right) \\ \mathbf{k}_3 &= \mathbf{f}\left(\mathbf{x}_k + \frac{1}{2}h\mathbf{k}_2, k + \frac{1}{2}h\right) \\ \mathbf{k}_4 &= \mathbf{f}(\mathbf{x}_k + h\mathbf{k}_3, k + h) \end{aligned} \tag{5.70b}$$

The main drawback of Runge-Kutta methods is the many function evaluations required per integration step. On SUNSAT, the reaction wheel controllers use a sampling period of one second All high accuracy attitude sensors will also be sampled at one second intervals. It is therefore imperative to implement the Horizon/Sun sensor EKF at a T_s or h of one second. Simulation studies showed that even an ordinary trapezium integration method performed satisfactory at this small step size.

During stages when the reaction wheels are not utilized for attitude control, when only magnetic torquing is active, longer sampling periods can be used. Due to the slowly changing reference geomagnetic field vector and low bandwidth magnetic torquer controllers, the Magnetometer EKF will be implemented at a sampling period of 10 seconds. An accurate and fast single-step method to do the numerical integration at large time steps for axially symmetric satellites, was introduced by Hodgart [1991]. This method is especially suited for conditions during which the satellite is spinning around its Z-axis (nominal nadir pointing) - when $\omega_z > \omega_x$ and ω_y. In SUNSAT's case this will be true during normal orbiting conditions when the magnetic torquers are utilized.

A brief overview will now be given of the Hodgart's single-step method. For an axially symmetric satellite the simplified dynamic equations around the X and Y body axes can be written as,

$$\begin{aligned} N_{x\ total} &= I_T\dot{\omega}_x - \omega_y\omega_z(I_T - I_{zz}) \\ N_{y\ total} &= I_T\dot{\omega}_y + \omega_x\omega_z(I_T - I_{zz}) \end{aligned} \tag{5.71}$$

If we define two complex variables,

$$\begin{aligned} \omega &\equiv \omega_x + j\omega_y \\ n &\equiv (N_{x\ total} + jN_{y\ total})/I_T \end{aligned} \tag{5.72}$$

then,

$$\dot{\omega} + j\omega_A\omega = n \tag{5.73}$$

with,

$$\omega_A = \omega_z\left(1 - \frac{I_{zz}}{I_T}\right) \quad = \quad \text{body nutation rate} \tag{5.74}$$

A solution to Equation (5.73) can easily be found by using the Laplace transform, given an initial value $\omega(0)$ and assuming a constant normalized torque n and nutation rate ω_A:

$$\begin{aligned} \omega(t) &= -j\frac{n}{\omega_A}\cdot\left(1-\exp(-j\omega_A t)\right) \quad + \quad \omega(0)\exp(-j\omega_A t) \\ &= \frac{2n}{\omega_A}\exp(-j\omega_A t/2)\sin\omega_A t/2 \quad + \quad \omega(0)\exp(-j\omega_A t) \end{aligned} \tag{5.75}$$

If the solution above is applied to a discrete system over a small time interval h, from time $t = kh$ to $(k+1)h$, Equation (5.75) can be approximated by,

$$\omega_{k+1} \approx nh\exp(-j\omega_A h/2) \quad + \quad \omega_k\exp(-j\omega_A h) \tag{5.76}$$

The normalized torque and nutation rate will not change much during a small time interval and the constant value assumption is therefore considered to be sound. Equation (5.76) shows that updates to the X and Y body angular rates must involve a rotation of the current complex angular rate variable through an angle $-\omega_A h$ due to gyroscopic coupling of the nutation rate, plus a further rotation through an angle $-\omega_A h/2$ for the complex angular rate increment nh.

While Equation (5.76) is used as a basis to the discrete propagation of the body angular rates, the closed form solution to the kinematic equation from Chapter 17 in Wertz [1986] can be used to propagate the quaternion in discrete time steps: (Assuming constant angular rates over the sampling period)

$$\begin{aligned} \mathbf{q}_{k+1} &= \exp\left[\frac{1}{2}\mathbf{\Omega}_k h\right]\mathbf{q}_k \\ &= \left[\cos\left(\frac{\overline{\omega}_k h}{2}\right)\mathbf{1} \quad + \quad \frac{1}{\overline{\omega}_k}\sin\left(\frac{\overline{\omega}_k h}{2}\right)\mathbf{\Omega}_k\right]\mathbf{q}_k \end{aligned} \tag{5.77}$$

with,

$$\overline{\omega}_k \quad \equiv \quad \sqrt{\omega^2_{ox,k} + \omega^2_{oy,k} + \omega^2_{oz,k}} \tag{5.78}$$

$\mathbf{\Omega}_k$ is the matrix defined in equation (1.13) and evaluated at time $t = kh$

The accurate single-step propagation method for the EKF can now be implemented by following the next set of steps during each sampling interval: (sampling step $h = T_s$)

1. Calculate the current body nutation rate from Equation (5.74),

$$\omega_A = \hat{\omega}_{z,k/k}\left(1-\frac{I_{zz}}{I_T}\right) \qquad (5.79a)$$

2. Calculate the current angular rate increments from Equations (1.17), (5.14) and (5 15), (Assuming small constant reaction wheel angular momentum components)

$$\begin{aligned}
d\omega_x &= \frac{T_s}{I_T}\left[N_{mx} + \hat{n}_{dry}\hat{A}_{12} - \frac{3GM_\oplus}{R_{s,k}^3}(I_T - I_{zz})\hat{A}_{23}\hat{A}_{13} - \hat{\omega}_{y,k/k}h_{z,k} + \hat{\omega}_{z,k/k}h_{y,k}\right] \\
d\omega_y &= \frac{T_s}{I_T}\left[N_{my} + \hat{n}_{dry}\hat{A}_{22} + \frac{3GM_\oplus}{R_{s,k}^3}(I_T - I_{zz})\hat{A}_{13}\hat{A}_{33} + \hat{\omega}_{x,k/k}h_{z,k} - \hat{\omega}_{z,k/k}h_{x,k}\right] \qquad (5.79b) \\
d\omega_z &= \frac{T_s}{I_T}\left[N_{mz} - \hat{\omega}_{x,k/k}h_{y,k} + \hat{\omega}_{y,k/k}h_{x,k}\right]
\end{aligned}$$

3. Correct the X- and Y-axis rate increments by a rotation of $-\omega_A T_s/2$,

$$\begin{aligned}
&[d\omega_x, d\omega_y]_{RECT} \rightarrow [d\omega, \phi]_{POL} \\
&[d\omega, \phi - \omega_A T_s/2]_{POL} \rightarrow [d\omega'_x, d\omega'_y]_{RECT}
\end{aligned} \qquad (5.79c)$$

4. Correct the X- and Y-axis current angular rates by a rotation of $-\omega_A T_s$,

$$\begin{aligned}
&[\hat{\omega}_{x,k/k}, \hat{\omega}_{y,k/k}]_{RECT} \rightarrow [\omega, \phi]_{POL} \\
&[\omega, \phi - \omega_A T_s]_{POL} \rightarrow [\hat{\omega}'_{x,k/k}, \hat{\omega}'_{y,k/k}]_{RECT}
\end{aligned} \qquad (5.79d)$$

5. Propagate the angular rate values,

$$\begin{aligned}
\hat{\omega}_{x,k+1/k} &= \hat{\omega}'_{x,k/k} + d\omega'_x \\
\hat{\omega}_{y,k+1/k} &= \hat{\omega}'_{y,k/k} + d\omega'_y \qquad (5.79e) \\
\hat{\omega}_{z,k+1/k} &= \hat{\omega}_{z,k/k} + d\omega_z
\end{aligned}$$

6. Calculate the transformation increments from the inertially referenced to orbit referenced angular rates, {see Equation (1.14)}

$$d\omega_{ox} = \hat{A}_{12}\tilde{\omega}_{o,k}, \qquad d\omega_{oy} = \hat{A}_{22}\tilde{\omega}_{o,k}, \qquad d\omega_{oz} = \hat{A}_{32}\tilde{\omega}_{o,k} \qquad (5.79f)$$

7. Obtain the latest estimate of the Z-axis orbit referenced angular rate,

$$\hat{\omega}_{oz,k+1} = \hat{\omega}_{z,k+1/k} + d\omega_{oz} \qquad (5.79g)$$

8. Improve the -X and Y-axis transformation increments by a rotation of $-0.8\hat{\omega}_{oz,k+1}T_s$ (This step was arrived at heuristically and the constant 0.8 obtained by trial and error during simulation to give the best propagation accuracy),

$$\begin{aligned} &\left[d\omega_{ox}, d\omega_{oy}\right]_{RECT} \rightarrow \left[d\omega, \phi\right]_{POL} \\ &\left[d\omega, \phi - 0.8\hat{\omega}_{oz,k+1}T_s\right]_{POL} \rightarrow \left[d\omega'_{ox}, d\omega'_{oy}\right]_{RECT} \end{aligned} \tag{5.79h}$$

9. Obtain the improved latest estimates of the X- and Y-axis orbit referenced angular rates,

$$\begin{aligned} \hat{\omega}_{ox,k+1} &= \hat{\omega}_{x,k+1/k} + d\omega'_{ox} \\ \hat{\omega}_{oy,k+1} &= \hat{\omega}_{y,k+1/k} + d\omega'_{oy} \end{aligned} \tag{5.79i}$$

10. Use the angular rates of Equations (5.79g) and (5.79i) and propagate the estimated quaternion using Equations (5.77) and (5.78).

Figures 5.5 anu 5.6 show typical simulation results over 5 orbits when the satellite's dynamics and kinematics are propagated using the single-step method and a 4-stage Runge-Kutta method, implemented at a time step of 1, 10 and 30 seconds respectively.

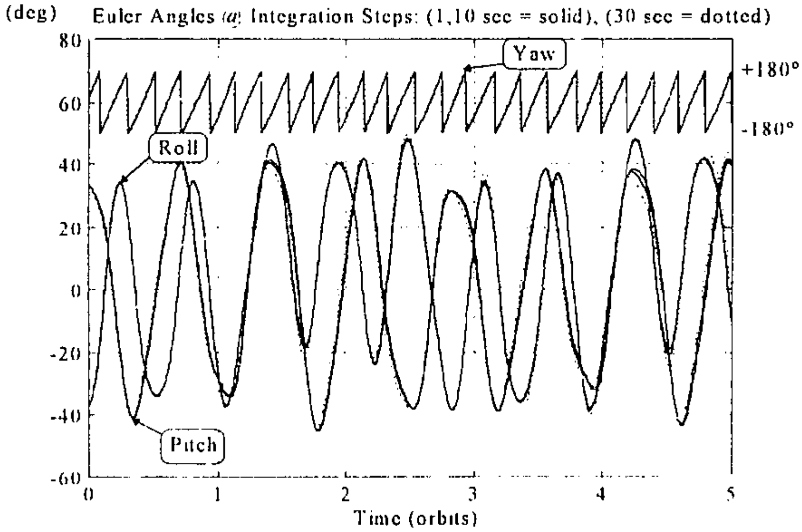

Figure 5.5 Various integration step size results using the new single-step method

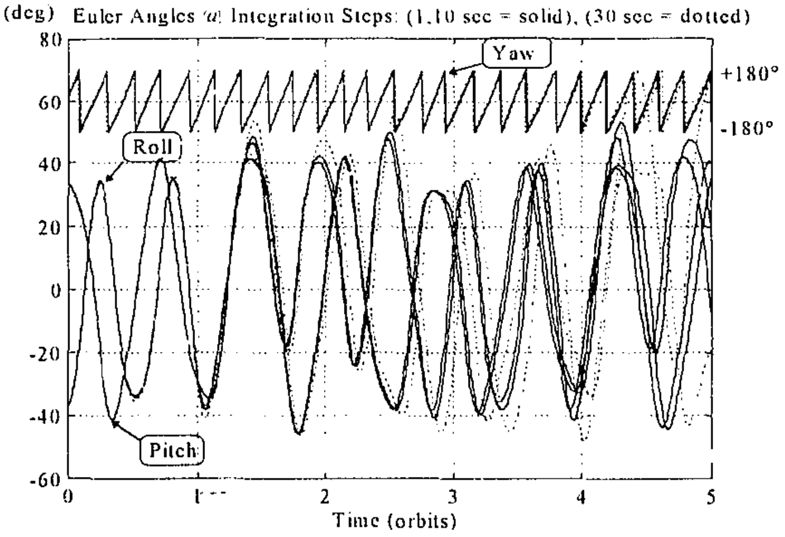

Figure 5.6 Various integration step size results using 4-stage Runge-Kutta method

It is clear from these graphs that the new single-step method outperforms the Runge-Kutta method at larger time steps. The 4-stage Runge-Kutta method is also computationally more expensive than the new method. For these reasons, it was decided to utilized the new method for all EKF state propagation calculations on SUNSAT.

During reaction wheel control periods, the Magnetometer EKF with a 10 second sampling interval increases its state propagation frequency to 1 Hz, hence a 10 times iteration of Equation (5.43a). The reason being the higher bandwidth dynamics and a one second sampling period of the reaction wheel controllers. The reaction wheel angular momentum vector is measured every second and the wheel torque vector computed every second. These values are then used during the calculation of the angular rate increments in Equation (5.79b). Note: The wheel torque $\mathbf{N}_{wheel}$ components must be included in the expressions of Equation (5.79b) during reaction wheel control. Simulation results indicate excellent state propagation accuracy even during reaction wheel control.

5.3.6 EKF SIMULATION RESULTS

The proposed full state extended Kalman filters were evaluated through simulation testing. The SUNSAT micro satellite with $I_T = 40$ kgm^2 and $I_{zz} = 2$ kgm^2 in an 400 km to 840 km elliptical polar orbit, was assumed. The geomagnetic field was modelled by using a second order IGRF model to speed up the simulation time.

In all the simulations the following default conditions were adhered to:

The satellite was not actively controlled but left in a free librating mode except where explicitly stated otherwise. The IGRF modelled vector $\mathbf{B}_o$ was obtained by adding uniformly distributed noise components, within the range -0.3 to 0.3 µT, to each calculated vector element. IGRF modelling errors will be the dominant contributing factor to the EKF innovation [Psiaki, 1990]. Note: The expected maximum and RMS error in the field magnitude when using the IGRF model are 0.54 µT and 0.18 µT respectively at an altitude of 445 km, see Table 5-1 in Wertz [1986]. The horizon angle measurement (θ_{hx} and θ_{hy}) was obtained by adding uniform distributed noise, within the range -0.03° to 0.03° (0.5 mrad), to the true horizon sensor model. Note: The horizon sensor is accurate to within ± 2 CCD pixels or ± 0.03°. The sun angle measurement (θ_{sun}) was obtained by adding uniform distributed noise, within the range -0.1° to 0.1° (1.75 mrad), to the true sun sensor model. Note: The sun sensor is accurate to within ± 2 CCD pixels or ± 0.1° An offset-sinusoidal aerodynamic disturbance torque (Equation (5.17)) with an amplitude of $n_{doy} = 3$ µNm (for average atmospheric density conditions) was employed during the simulations.

The initial satellite angular body rate components were zero for the X and Y body axes and either 4 or 5 rpo (revolutions per orbit) for the Z-axis (yaw spin). The estimator state vector was initialised with zero values, except for the estimated Z-axis angular rate $\hat{\omega}_z$, which was initialised to 90% of the correct value to speedup the conversion rate. It is reasonable to expect that angular rate values can be calculated to within 10% accuracy from the sensor measurements, for example, by employing the robust angular estimator of Paragraph 5.2. These rough estimations can then be used to prime the full state Kalman filters presented in this chapter.

The Magnetometer EKF was implemented at a sampling period $T_s = 10$ seconds and the Horizon/Sun sensor EKF at $T_s = 1$ second. The initial filter covariance matrix $\mathbf{P}_0$ was initialized each time as a diagonal matrix with elements approximately 3 orders of magnitude larger than the system covariance matrix $\mathbf{Q}$. The system and measurement noise covariance matrices $\mathbf{Q}$ and $\mathbf{R}$ were not chosen from stochastic considerations, but rather in a pragmatic manner to optimize the filter's simulation performance This approach was perceived to be reasonable, considering the non-linear satellite and measurement models

The first few simulations were done to compare the performance of the New magnetometer sensor EKF (Paragraph 5.3.2) with the modified Psiaki EKF (Paragraph 5.3.3). Figures 5.7 and 5.8 show the comparative pointing angle estimation over 2 orbits. The initial pitch and roll angles of the satellite were chosen as 10° and -5° respectively, and the initial Z-spin rate ω_z = 4 rpo. Both filters converge to the true (real) pointing angles within an orbit, with the New EKF having the smallest tracking errors during the initial orbit. Figures 5.9 and 5.10 compare the estimated Z-axis angular rates. Figures 5.11 and 5.12 compare the estimated Y_o-axis disturbance torque amplitudes. Similar observations regarding the improved convergence speed of the New EKF can be made. *It is also evident that the estimated disturbance torque amplitude of the Psiaki EKF is more sensitive to measurement noise. Figures 5.13 and 5.14 show the RMS pointing (combined pitch and roll) error, using magnetometer measurements only, of the New and Psiaki EKF over a few orbits after conversion has been achieved. The maximum error for the New EKF is 0.55° and the average about 0.3°, compared to a maximum of 1.05° and average of about 0.5° for the Psiaki EKF.*

Another performance characteristic worth investigating, was the ability of the EKF to converge from any initial unknown attitude state. Figure 5.15 and 5.16 present the ability of the two different EKF implementations (New and Psiaki) to start tracking the pitch and roll pointing angles to within a combined RMS value of 0.5°. The initial true pitch and roll Euler angles were chosen uniformly within the range -60° to +60° and perfect measurements were assumed. Figure 5.15 indicates a 100% convergence success within a single orbit for the New EKF, with an average time of only 0.5 orbits. Figure 5.16 indicates convergence success for the Psiaki filter within less than 0.5 orbits only if the estimator initial pointing attitude is close to 0° for pitch and roll. Large initial pointing errors (> 40°) seldom converge within the maximum allowed time of 2 orbits. Note: The Z-axis (orbital fraction) in Figure 5.16 is limited to 2 orbits. Convergence problems was also reported by Psiaki [1990] for their filter in a 3-axis stabilization application, whenever initial angular errors larger than 60° were utilized.

Figures 5.17 and 5.18 display typical magnetic torquing delibration and Z-spin rate control sequences, when the cross-product control law (Equation (2.20)) is used in combination with the New magnetometer sensor EKF. The initial pitch and roll attitude angles of the satellite are 40° and -30° respectively. During the first orbit the controller is deactivated to enable the filter to converge to the true satellite state - from Figure 5.18 it can be seen that convergence took about half an orbit. At the start of the second orbit the controller is activated and the satellite is delibrated over a span of 2 orbits without any noticable increase in estimation pointing error. Note: Perfect knowledge of the effective magnetic torque was assumed during simulation.

Figures 5.19 to 5.21 present typical performance of the New horizon/sun sensor EKF An initial pitch attitude of 5° and a Z-spin rate of 5 rpo was assumed, with zero attitude knowledge and 90% angular rate knowledge for the filter. Included in Figure 5.19 is the status of the innovation windows indicating the sensor measurement availability (high = available, low = unavailable). Convergence is achieved within one orbit and accurate attitude tracking (< 0.1°) is established during periods when data from all three sensors becomes available simultaneously. Larger tracking errors can occur during periods when the filter runs open-loop, as Figure 5.20 clearly indicates.

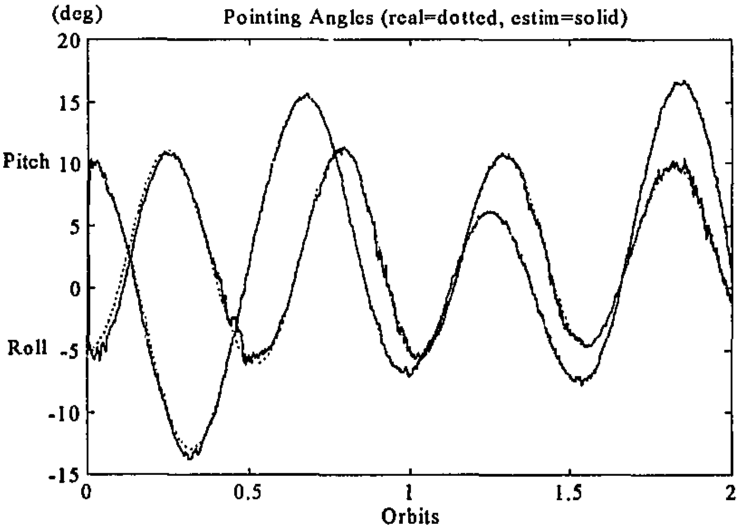

Figure 5.7 Pointing angle estimation of New Magnetometer EKF

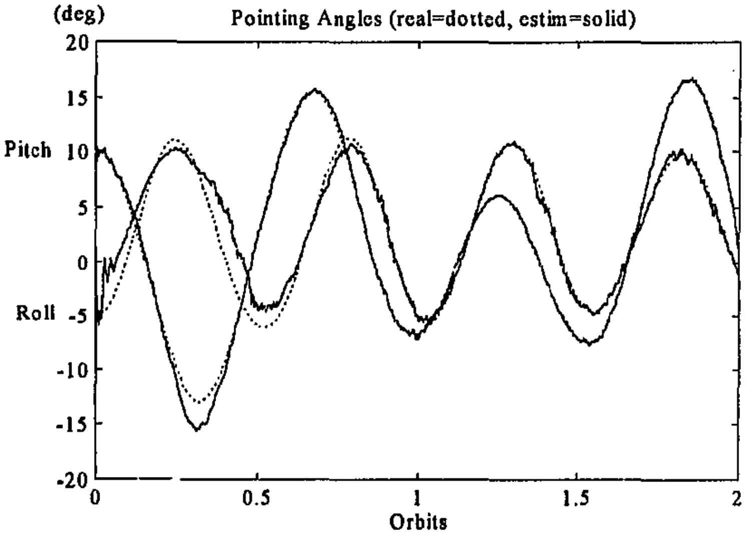

Figure 5.8 Pointing angle estimation of modified Psiaki EKF

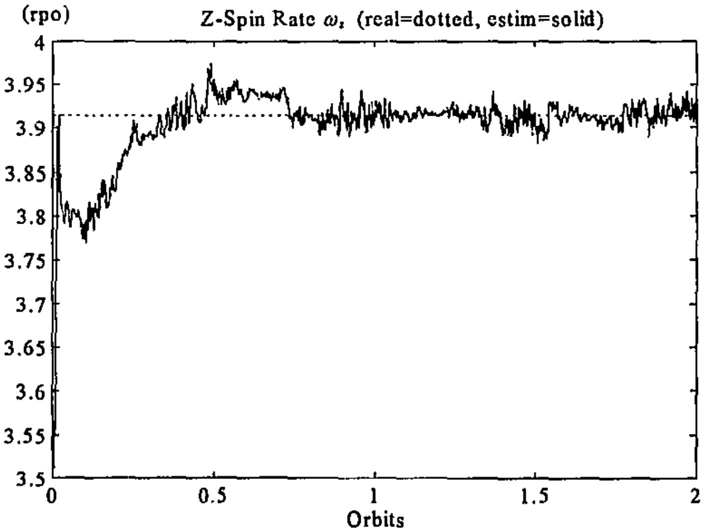

Figure 5.9 Z-spin estimation of New Magnetometer EKF

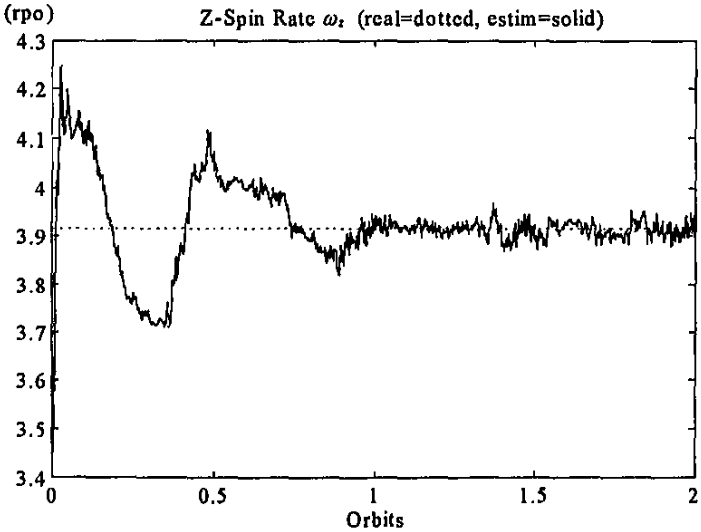

Figure 5.10 Z-spin estimation of modified Psiaki EKF

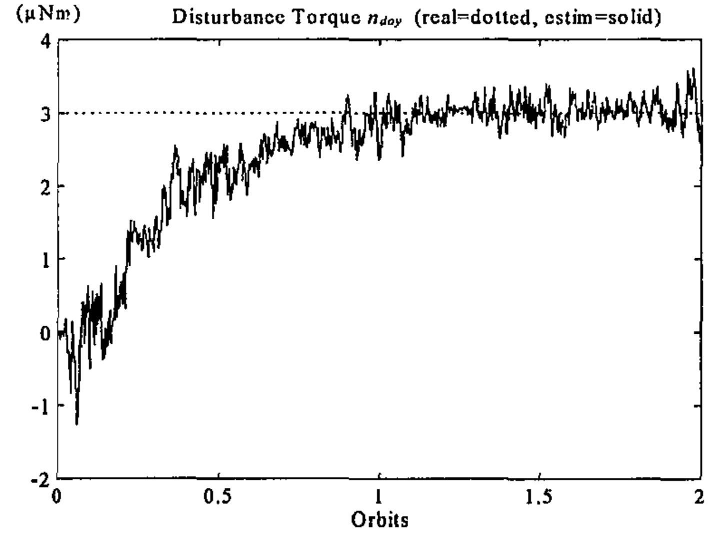

Figure 5.11 Y$_o$-axis disturbance torque estimation of New Magnetometer EKF

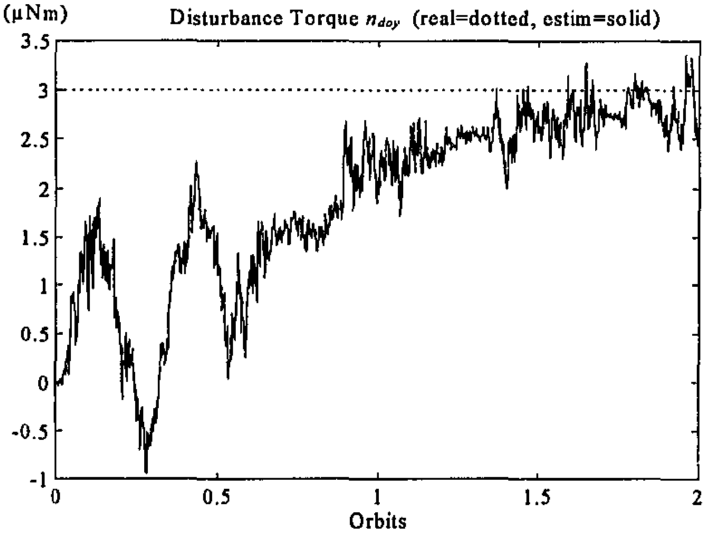

Figure 5.12 Y$_o$-axis disturbance torque estimation of modified Psiaki EKF

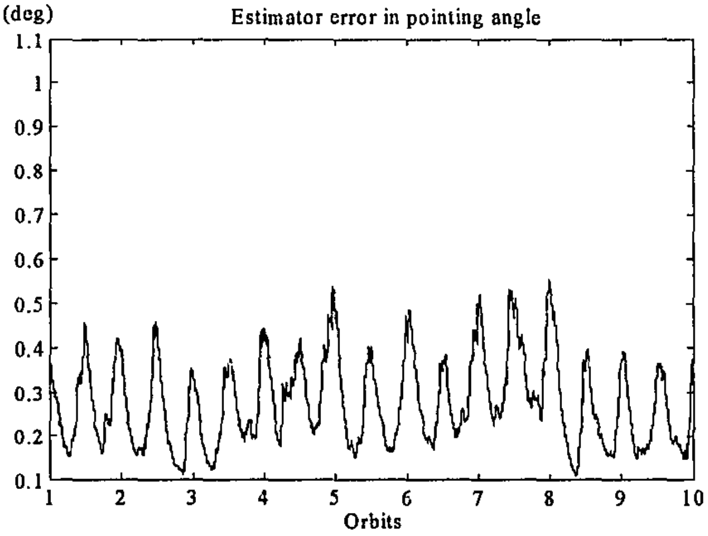

Figure 5.13 Estimation RMS pointing error of New Magnetometer EKF

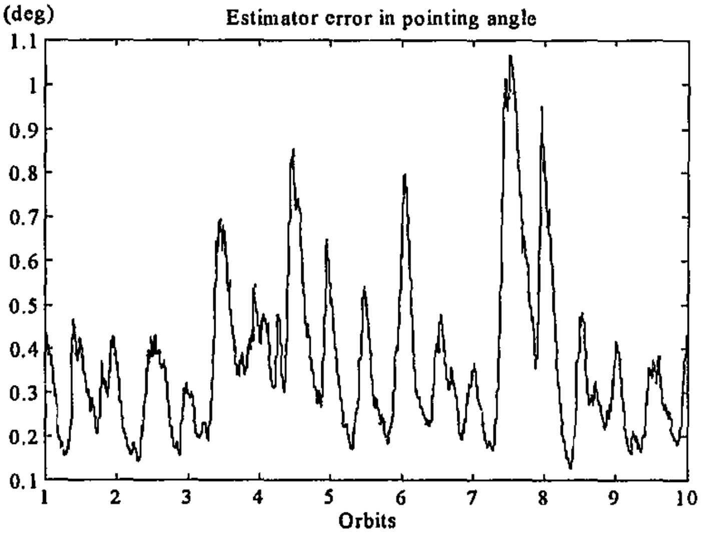

Figure 5.14 Estimation RMS pointing accuracy of modified Psiaki EKF

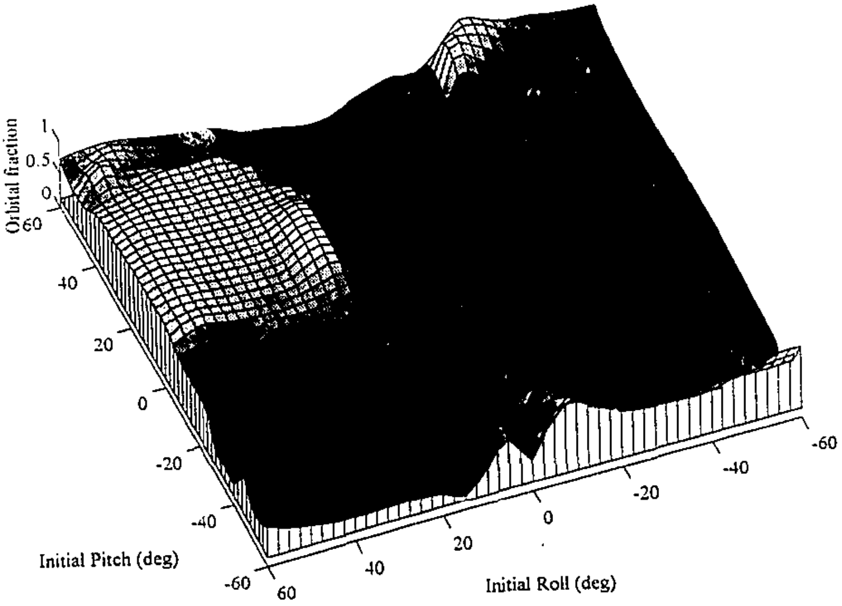

Figure 5.15 Convergence performance of New Magnetometer EKF

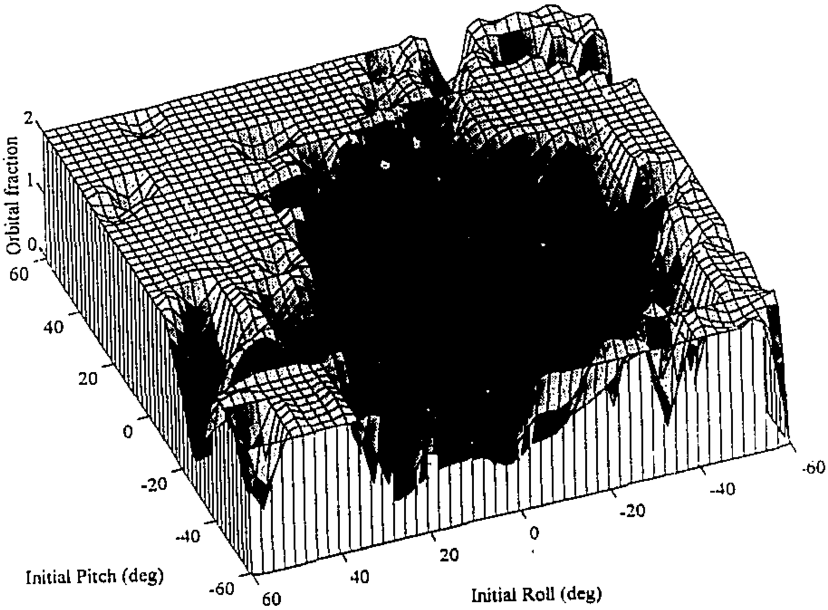

Figure 5.16 Convergence performance of modified Psiaki EKF

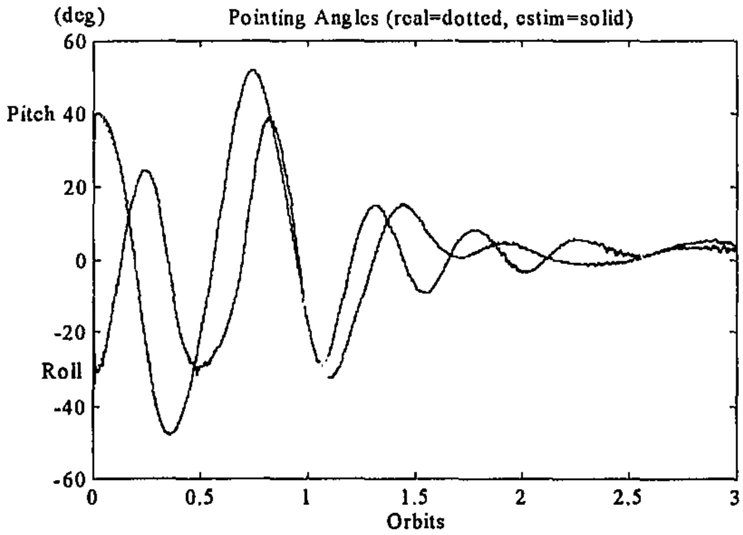

Figure 5.17 New Magnetometer EKF tracking performance during magnetic torquing

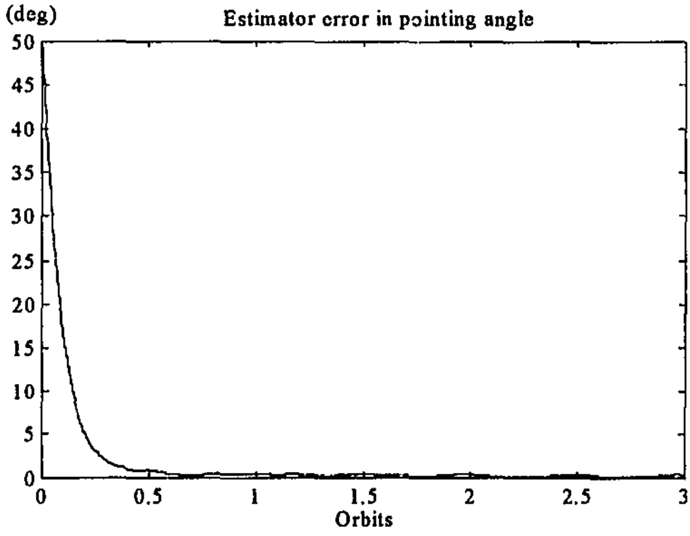

Figure 5.18 New Magnetometer EKF RMS pointing error during n.ᵢᵹ ιetic torquing

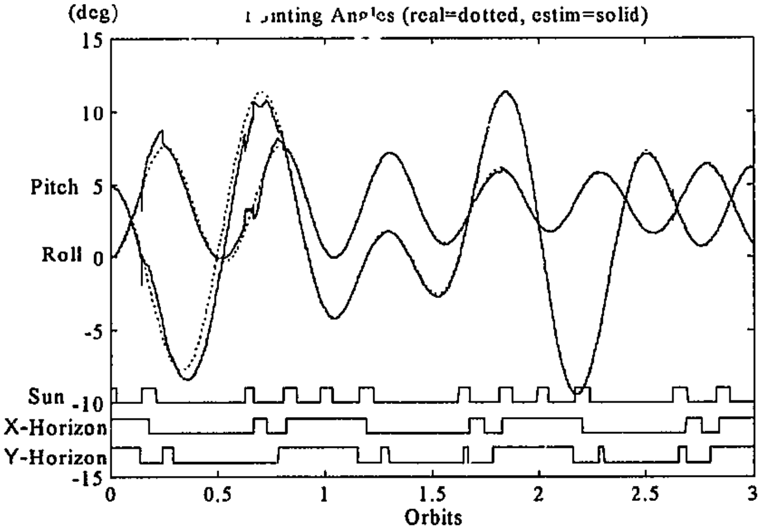

Figure 5.19 Typical pointing angle estimation of the New Horizon/Sun EKF

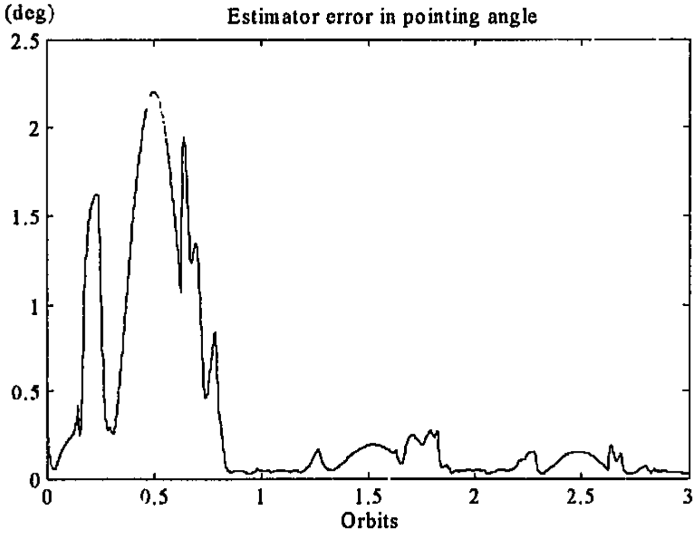

Figure 5.20 Estimation RMS pointing accuracy of the New Horizon/Sun EKF

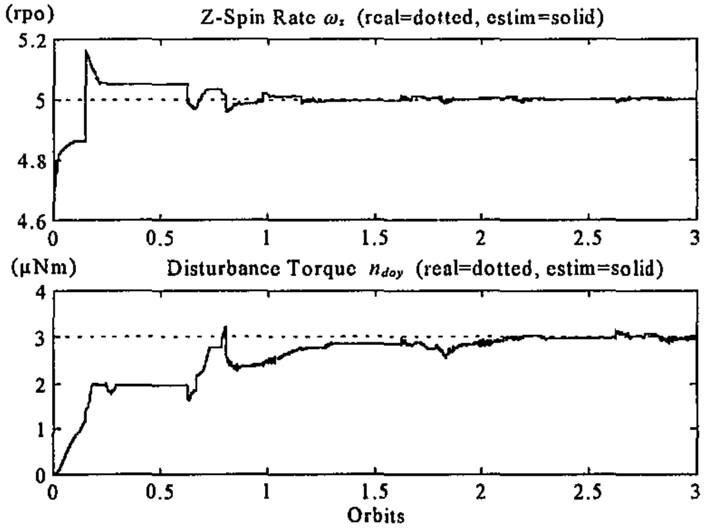

Figure 5.21 Z-spin and Disturbance torque estimation of the New Horizon/Sun EKF

5.4 Conclusions

A robust angular rate Kalman filter was presenteu ... Paragraph 5.2 to estimate the orbit referenced angular rates using successive magnetometer measurements only. The practical application of the filter will be reserved for the initial stages before boom deployment has taken place. Convergence within half an orbit from any initial body angular rate vector was demonstrated during simulation trials. Tracking accuracy is obtainable at maximum errors of $\pm\omega_o$ and the proposed filter proved to be robust in the presence of high levels of measurement noise.

The extended Kalman filters described in Paragraph 5.3 were able to extract full attitude, body rate and disturbance torque information from noisy vector observations and applied to a LEO gravity gradient stabilized satellite. The superiour performance of a New EKF algorithm compared to the Psiaki method [Psiaki, 1990] demonstrated convergence from various initial conditions within one orbital period. Full state tracking after initial attitude angular errors of up to ± 90° was successfully tested. The modified Psiaki EKF experienced convergence problems for initial angular errors greater than ± 40°.

The filter model as presented is able to extract full attitude information from any initial libration and Z-spin rate values, the only limitation being the sampling period of the discrete model. It was, however, found that the convergence speed improved to more acceptable levels when the initial filter state is primed with rough estimates of the true angular rates.

The Magnetometer EKF uses only geomagnetic vector measurements and modelling data to obtain the filter innovations. Attitude angular errors of less than 0.5° are expected during normal operating conditions. These errors are caused mainly by modelling errors of the geomagnetic field. Geomagnetic measurements have two main advantages compared to other methods: Magnetometers are relatively inexpensive and measurements can be done throughout an orbit. A disadvantage, however, is the inaccurate knowledge of the local modelled geomagnetic field vector due to disturbances during sun activity, for example.

The Horizon/Sun EKF utilized two orthogonal looking horizon sensors and a sun sensor measuring in a third orthogonal body plane. While measurements are not available continuously due to FOV limitations, this EKF is capable of improved continuous attitude angular errors compared to the Magnetometer EKF. An overall attitude accuracy of below 0.2°, assuming a reasonable disturbance torque model, and below 0.1°, during periods when all sensor measurements become available concurrently, is expected. Further improvement will be possible when star sensor vector measurements are included as well.

To conclude, low-cost and light-weight methods of robust angular estimation and accurate full satellite state determination were proposed to be used by gravity gradient stabilized satellites.

5.5 References

Bar-Itzhack I.Y. and Reiner J. [1984]

"Recursive Attitude Determination from Vector Observations: Direction Cosine Matrix Identification", *Journal of Guidance, Control, and Dynamics*, Vol.7, No.1, Jan.-Feb.1984, pp.51-56.

Bar-Itzhack I.Y. and Oshman Y. [1985]

"Attitude Determination from Vector Observations: Quaternion Estimation", *IEEE Transactions on Aerospace and Electronic Systems*, Vol.21, No.1, Jan.1985, pp.128-135.

Bierman G.J. and Thornton C.L. [1977]

"Numerical Comparison of Kalman Filter Algorithms: Orbit Determination Case Study", *Automatica*, Vol.13, No.1, 1977, pp.23-35.

Dodds S.J. [1984]

"A gyro drift and dynamics state estimator for satellite attitude control systems utilizing a star mapper", *International Journal of Control*, Vol.39, No.6, 1984, pp.1275-1293.

Fisher H.L., Musser K.L. and Shuster M.D. [1993]

"Coarse Attitude Detennination from Earth Albedo Measurements", *IEEE Transactions on Aerospace and Electronic Systems*, Vol.29, No.1, Jan.1993, pp.22-25.

Fraiture L. [1986]

"Attitude reconstruction filters: Yes or No ?", *Proceedings of the 2nd International Symposium of Spacecraft Flight Dynamics*, Darmstadt Germany, 20-23 Oct.1986, pp.403-414.

Friedland B. [1978]

"Analysis Strapdown Navigation Using Quaternions", *IEEE Transactions on Aerospace and Electronic Systems*, Vol.14, No.5, Sept..1978, pp.764-768.

Gai E., Daly K., Harrison J. and Lemos L. [1985]

"Star-Sensor Based Satellite Attitude/Attitude Rate Estimator", *Journal of Guidance, Control, and Dynamics*, Vol.8, No.5, Sept.-Oct.1985, pp.560-565.

Gelb A. [1989]

Applied Optimal Estimation, The Analytical Sciences Corporation, M.I.T. Press, Eleventh Printing 1989.

Grusas P.A. [1969]

"Satellite Attitude Determination from Celestial Sightings", *Journal of Spacecraft and Rockets*, Vol.6, No.9, Sept.1969, pp.1007-1012.

Hodgart M.S. [1992]

"Accurate and Fast Numerical Integration for Simulation of Dynamics of Axially Symmetric Satellites", *Internal Document: University of Surrey*, Nov.1992.

Ickes B.P. [1970]
"A New Method for Performing Digital Control System Attitude Computations Using Quaternions", *AIAA Journal*, Vol.8, No.1, Jan.1970, pp.13-17.

Lefferts E.J., Markley F.L. and Shuster M.D. [1982]
"Kalman Filtering for Spacecraft Attitude Estimation", *Journal of Guidance, Control, and Dynamics*, Vol.5, No.5, Sept.-Oct.1982, pp.417-429.

Markley F.L. [1994]
"New Quaternion Attitude Estimation Method", *Journal of Guidance, Control, and Dynamics*, Vol.17, No.2, March-April 1994, pp.407-409.

Ninomiya K., Muranaka M. and Uo M. [1986]
"Attitude determination for Japan's first interplanetary flights, 'Sakigake' and 'Suisei'", *Proceedings of the 2nd International Symposium of Spacecraft Flight Dynamics*, Darmstadt Germany, 20-23 Oct.1986, pp.415-420.

Psiaki M.L., Martel F. and Pal P.K. [1990]
"Three-Axis Attitude Determination via Kalman Filtering of Magnetometer Data", *Journal of Guidance, Control, and Dynamics*, Vol.13, No.3, May-June 1990, pp.506-514.

Psiaki M.L., Huang L. and Fox S.M. [1993]
"Ground Tests of Magnetometer-Based Autonomous Navigation (MAGNAV) for Low-Earth-Orbiting Spacecraft", *Journal of Guidance, Control, and Dynamics*, Vol.16, No.1, Jan.-Feb.1993, pp.206-214.

Shibata M. [1986]
"Error Analysis Strapdown Inertial Navigation Using Quaternions", *Journal of Guidance, Control, and Dynamics*, Vol.9, No.3, May-June 1986, pp.379-381.

Shuster M.D. and Oh S.D. [1981]
"Three-Axis Attitude Determination from Vector Observations", *Journal of Guidance, Control and Dynamics*, Vol.4, No.1, Jan.-Feb.1981, pp.70-77.

Wertz J.R. [1986]
Spacecraft Attitude Determination and Control, Astrophysics and Space Science Library, Vol.73, D.Reidel Publishing Company, 1978 (reprint 1986).

Zwartbol T., Van den Dam R.F., Terpstra A.P. and Van Woerkom P.Th.L.M. [1985]
"Attitude Estimation and Control of Manoeuvring Spacecraft", *Automatica*, Vol.21, No.5, 1985, pp.513-526.

6. MAGNETOMETER CALIBRATION

6.1 Introduction

Magnetometers are low cost, low power, rugged and reliable sensors used for attitude determination and control of small satellites. A three-axis magnetometer can measure the geomagnetic field strength and direction to high accuracy if it is well calibrated and free from magnetic disturbances. If the magnetometer is used in magnetorquing control applications only, a crude sensor will suffice, for example, when doing libration damping or momentum dumping of reaction wheels. In many of these applications [Alfriend, 1975; Hodgart, 1990] only the sign of the geomagnetic field vector is used. In contrast, when doing attitude determination to accuracies better than 0.5° per axis an accurate sensor will be required.

It is important to realize that the accuracy of an attitude solution can not exceed that of the reference geomagnetic field model used. An IGRF (International Geomagnetic Reference Field) model can have a maximum directional error of 0.5° at an altitude of 800 km [Lerner, 1981]. Therefore, to achieve attitude determination accuracies limited by the reference model only, a calibrated magnetometer is needed at all times. Sources of magnetometer errors are pre-launch gain and offset miscalibration of the electronic circuitry, internal misalignment of the magnetometer pickup coils, external misalignment of the magnetometer mounting and analog to digital (A/D) conversion errors. Zero mean noise sources and the resolution of the A/D convertor also need to be considered.

Lerner and Shuster [1981] presented a batch linear regression algorithm to determine simultaneously magnetometer biases, scale factor and misalignment errors. Their algorithm has been successfully used to calibrate magnetometers on the Seasat-1 and AEM-1/HCMM satellites. The availability of a source of attitude knowledge independent of the magnetometer data was assumed. This source may be from any combination of star sensors, sun and earth sensors. Their algorithm uses the following calibration model:

$$\mathbf{A}(t)\mathbf{B}_o(t) = (\mathbf{1}_3 + \mathbf{S})\mathbf{B}_m(t) + \mathbf{b} \tag{6.1}$$

$\mathbf{A}(t)$ is the 3 × 3 attitude matrix at time t that transforms vectors from orbit reference to body coordinates. $\mathbf{B}_o(t)$ is the modelled geomagnetic field vector in reference (orbital) coordinates and $\mathbf{B}_m(t)$ is the pre-flight calibrated magnetometer measurement. $\mathbf{S}$ is the scale factor/misalignment matrix and $\mathbf{b}$ the bias vector to be determined. A nonzero diagonal element of the S matrix is indicative of an error in the corresponding axis's scale factor. Nonzero off-diagonal elements present a misaligned magnetometer axis. If a nondiagonal 1 + S matrix is orthogonal, the three magnetometer axes are orthonormal and coherently misaligned to the spacecraft reference axes. The resulting solution obtained when doing linear regression is then:

$$\mathbf{S} = \langle \mathbf{H} | \mathbf{B}_m \rangle \langle \mathbf{B}_m | \mathbf{B}_m \rangle^{-1} \qquad \mathbf{b} = \langle \mathbf{H} \rangle - \mathbf{S} \langle \mathbf{B}_m \rangle \qquad (6.2)$$

wher?,

$$\mathbf{H}_i = \mathbf{A}(t_i)\mathbf{B}_o(t_i) - \mathbf{B}_m(t_i)$$

with,

$\langle \mathbf{F} | \mathbf{G} \rangle$ = sampied covariance matrix of two vectors

$\langle \mathbf{F} \rangle$ = sampled mean of a vector

Thompson *et.al.* [1984] described a batch estimator utilizing a long segment of magnetometer data to determine the unknown magnetometer bias vector. The iterative solution was obtained independent of any attitude knowledge and only uses information from a geomagnetic field model.

Lee *et.al.* [1994] proposed a variable step size LMS (Least Mean Square) algorithm to determine the scaling factor and bias coefficients of a magnetometer. The algorithm is simple and compact and can be applied effectively to do in-flight sensor calibration. No attitude knowledge is assumed and only the modelled geomagnetic field's magnitude is used during the calibration process. The algorithm has been used with success in calibrating magnetometers on the KITSAT-1 and KITSAT-2 micro satellites. Misalignment errors can unfortunately not be corrected by this algorithm, however, improvements to the pre-launch calibration of at least an order of magnitude in measurement errors have been shown on the KITSAT satellites.

Just after separation from the launching vehicle, the spacecraft normally tumbles in an unknown manner. In this situation, accurate sun, earth or star sensors are not practical to use for attitude determination. A magnetometer will then be ideally suited to initially determine the attitude of the spacecraft. The accuracy of this method will then depend solely on how well the magnetometer has been calibrated and on the error of the geomagnetic field model used. The self-calibrating LMS algorithm can be employed to correct the former source of error to some extent. The details of this method will be described in the next paragraph and some results will be presented in the simulation section.

A new RLS (Recursive Least Squares) algorithm for in-flight magnetometer calibration (scale factor, misalignment and bias) of magnetometers will also be presented. The accuracy and convergence speed of this method shows great improvement when compared to the LMS algorithm. This method, however, uses attitude knowledge (the transformation matrix from the reference orbital coordinates to body coordinates) and the vector output of an IGRF model. Measurements from other attitude sensors are therefore essential for the accuracy of this improved self-calibrating algorithm. The RLS algorithm and simulation results will be given in the rest of this chapter.

6.2 LMS Self-Calibration Algorithm

The major part of this paragraph was first presented by Lee *et.al.* [1994] and included here for completeness. The first order calibration model for the magnetometer is,

$$\mathbf{B}_c(k) = \mathbf{G}\mathbf{B}_m(k) + \mathbf{b} \tag{6.3}$$

where,

$\mathbf{G} = diag\left[g_x, g_y, g_z\right] =$ Scale factor (gain) matrix

$\mathbf{b} = \left[b_x \quad b_y \quad b_z\right]^T =$ Bias vector

The error to be minimized is defined as the difference between the norm of the orbit reference magnetic field $\mathbf{B}_o$(k) and the calibrated measurement vector $\mathbf{B}_c$(k).

$$e(k) = \left\|\mathbf{B}_o(k)\right\| - \left\|\mathbf{B}_c(k)\right\| \tag{6.4}$$

The LMS algorithm is a gradient approximation algorithm that adjusts the six calibration parameters g_i and b_i to minimize the instantaneous squared error. Each parameter is adjusted simply as,

$$p(k+1) = p(k) - \mu(k)\frac{\partial e^2(k)}{\partial p(k)} \tag{6.5}$$

where,

$\mu(k)$ = variable step size

From Equations (6.3) to (6.5) the update expressions for the scale factor and bias parameters are written as,

$$\begin{aligned} g_i(k+1) &= g_i(k) + 2\mu_g(k)e(k)\left\|\mathbf{B}_o(k)\right\|^2 B_{ci}(k)B_{mi}(k) \\ b_i(k+1) &= b_i(k) + 2\mu_b(k)e(k)\left\|\mathbf{B}_o(k)\right\|^2 B_{ci}(k) \end{aligned} \tag{6.6}$$

with,

$i = x, y, z$

In order for the LMS algorithm to converge, the step size must be selected smaller than a certain upper bound. A conservative estimate of the step size upper bound is given [Lee, 1994] as,

$$\mu_g \leq \frac{1}{3\langle \alpha^2(k) \rangle} \quad \text{and} \quad \mu_b \leq \frac{1}{3\langle \beta^2(k) \rangle} \tag{6.7}$$

where,

$$\alpha(k) = \left\|\mathbf{B}_o(k)\right\|^2 B_{ci}(k) B_{mi}(k)$$

$$\beta(k) = \left\|\mathbf{B}_o(k)\right\|^2 B_{ci}(k)$$

To improve the convergence speed a variable step size can be implemented:

$$\mu(k) = \mu_{\max}\left\{1 - \exp\left(-e^n(k)\right)\right\} \tag{6.8}$$

Lee *et.al.* [1994] proposed a choice of n = 4, however, implementation of the LMS algorithm (see simulation section) showed improved convergence accuracy for n = 2.

6.3 RLS Self-Calibration Algorithm

A new magnetometer calibration algorithm based on real-time parameter estimation is proposed for improved convergence and accuracy. The algorithm is a recursive implementation of the least squares minimization technique. The vector error to be minimized can be written as:

$$\mathbf{e}(k) = \mathbf{y}_{model}(k) - \mathbf{y}_{calib}(k) = \mathbf{A}(k)\mathbf{B}_o(k) - \left[\mathbf{G}(k)\mathbf{B}_m(k) + \mathbf{b}(k)\right] \tag{6.9}$$

with,

$$\mathbf{G}(k) = \begin{bmatrix} g_{11}(k) & g_{12}(k) & g_{13}(k) \\ g_{21}(k) & g_{22}(k) & g_{23}(k) \\ g_{31}(k) & g_{32}(k) & g_{33}(k) \end{bmatrix}$$

$$\mathbf{b}(k) = \left[b_x(k) \quad b_y(k) \quad b_z(k)\right]^T$$

A(k) is the 3 × 3 attitude matrix at time-step k that transforms vectors from orbit reference to body coordinates. **B**$_o$(k) is the modelled geomagnetic field vector in reference (orbital) coordinates and **B**$_m$(k) is the uncalibrated magnetometer measurement. The vector error problem can then be divided into three standard scalar least square parameter estimation problems:

$$e_i(k) = y_i(k) - \varphi^T(k)\theta_i(k) \qquad i = 1, 2, 3 \tag{6.10}$$

with,

$$\begin{aligned} y_1(k) &= a_{11}(k)B_{ox}(k)+a_{12}(k)B_{oy}(k)+a_{13}(k)B_{oz}(k) \\ y_2(k) &= a_{21}(k)B_{ox}(k)+a_{22}(k)B_{oy}(k)+a_{23}(k)B_{oz}(k) \\ y_3(k) &= a_{31}(k)B_{ox}(k)+a_{32}(k)B_{oy}(k)+a_{33}(k)B_{oz}(k) \end{aligned} \tag{6.10a}$$

and,

$$\varphi^T(k) = \begin{bmatrix} B_{mx}(k) & B_{my}(k) & B_{mz}(k) & 1 \end{bmatrix} \tag{6.10b}$$

$$\begin{aligned} \theta_1^T(k) &= \begin{bmatrix} g_{11}(k) & g_{12}(k) & g_{13}(k) & b_x(k) \end{bmatrix} \\ \theta_2^T(k) &= \begin{bmatrix} g_{21}(k) & g_{22}(k) & g_{23}(k) & b_y(k) \end{bmatrix} \\ \theta_3^T(k) &= \begin{bmatrix} g_{31}(k) & g_{32}(k) & g_{33}(k) & b_z(k) \end{bmatrix} \end{aligned} \tag{6.10c}$$

If the least square cost function to be minimized is taken as,

$$J = \frac{1}{2}\sum_{k=1}^{t} \lambda^{t-k} e^2(k) \tag{6.11}$$

The forgetting factor λ is a constant such that $0 < \lambda \leq 1$ to introduce a time-varying weighting of the data. The most recent data is given more weight than data in the past. This will ensure that calibration changes will be tracked. The full RLS algorithm for any of the three parameter estimation problems will be given as, [Åström, 1989]

1. Compute the regression vector $\varphi(k)$ and the residual $e(k)$ from Equation (6.10).

2. Compute the update gain vector:

$$\mathbf{K}(k) = \mathbf{P}(k-1)\varphi(k)\left[\lambda + \varphi^T(k)\mathbf{P}(k-1)\varphi(k)\right]^{-1} \tag{6.12}$$

   where **P** is defined as the covariance matrix of the regression vector $\varphi(k)$.

3. Update the parameter vector:

$$\theta(k) = \theta(k-1) + \mathbf{K}(k)e(k) \tag{6.13}$$

4. Update the covariance matrix:

$$\mathbf{P}(k) = \left[\mathbf{1}_4 - \mathbf{K}(k)\varphi^T(k)\right]\mathbf{P}(k-1)/\lambda \tag{6.14}$$

5. Repeat steps 1-4 every sampling period.

To initialize the RLS algorithm, the unknown scale factor/misalignment matrix **G**(0) and bias vector **b**(0) can be primed with the pre-launch calibration parameters or the results from a LMS algorithm (only the diagonal elements of **G** will be known, the other elements will be zeroed). If no calibration knowledge is available, **G**(0) = $\mathbf{1}_3$ (3 ×

3 identity matrix) and **b**(0) = **0** (zero vector). The covariance matrix can be initialized as a diagonal matrix, for example, **P**(0) = *diag* [10, 10, 10, 10] was found to give the best conversion results during simulation. Smaller initial **P** elements reduce the conversion rate and larger elements increase initial fluctuations in the parameters.

To improve the robustness of the RLS algorithm especially against large initial residuals or against measurement outliers (noise spikes), the residual can be modified by a non-linear function. A typical example is, [Åström, 1989]

$$f\{e(k)\} = \frac{e(k)}{1+a|e(k)|} \qquad (6.15)$$

The constant a is designed such that the function is still linear for normal values of $e(k)$, but to decrease the consequences of larger abnormal errors.

6.4 Simulation Results

The LMS and RLS algorithms were both implemented in a full simulation program of the satellite's dynamics, sensors and environmental models. An eighth order 1990 IGRF model was used to model the geomagnetic environment (Appendix A). A slightly elliptical orbit (eccentricity = 0.03), at an inclination 96° and orbital period of 100 minutes was employed to generate latitude, longitude and altitude inputs to the IGRF model. The field vector component output of the model was then transformed to orbit referenced coordinates.

The magnetometer was modelled with its axes parallel to the satellite's body axes, but with scale factor and bias errors. Measurement noise, uniformly distributed in the interval -0.5 μT to +0.5 μT, was added to each vector component. Table 6.1 lists the calibration error values used for the magnetometer. Figure 6.1 shows the deviation in geomagnetic magnitude when using the uncalibrated magnetometer, compared to the true magnitude. It is therefore obvious that the uncalibrated magnetometer will not deliver any useful results.

Table 6.1 Calibration Errors

	Scale Factor	*Bias Error*
X-axis	0 8	-5.0 μT
Y-axis	1.2	+5.0 μT
Z-axis	1.1	-8.0 μT

Figures 6.2 to 6.4 illustrate the performance of the LMS algorithm. The conservative upper bounds on the step sizes (Equation 6.7) were further optimized during simulation to improve the convergence rate. The final values used were:

$$\mu_{g-\max} = 10^{-9} \qquad \text{and} \qquad \mu_{b-\max} = 10^{-6}$$

The calibration parameters took about three full orbits (300 minutes) to converge to the values of Table 6.1. A 10 second sampling period was used (600 samples/orbit) for the LMS algorithm. It was, however, found that an increase in the sampling period up to 60 seconds (100 samples/orbit) gives almost the same results. The only provision is that the upper bounds on the step sizes also have to be increased by the same factor.

Figure 6.5 represents a case where the magnetometer axes are non-orthogonal and where the LMS algorithm is used. Due to the diagonal restriction of the scaling factor matrix **G** in the LMS calibration model, it will be impossible to reduce the calibration errors further below a certain limit. In this example, a magnetometer is modelled with its Z-axis tilted by 10° towards the X-axis. The same scaling factor and bias calibration error values of Table 6.1 are also included in the simulation to enable comparison of this result to the one presented in Figure 6.2. It is clear from Figure 6.5 that conversion errors will be reduced up to a certain level but small errors will persist due to the LMS modelling problem.

Figures 6.6 to 6.8 illustrate the performance of the RLS algorithm. An exponential weighting factor λ of 0.99 was used. This value of near unity ensures that the algorithm gradually discards previous measurements and will therefore be able to track slow calibration changes. These changes will mainly be caused by ageing of the magnetometer electronics and unmodelled temporal plus secular changes of the geo-magnetic field.

Convergence of the RLS calibration parameters was achieved within half an orbit (50 minutes) and the parameter variation after convergence was also much smaller compared to the LMS algorithm. A 10 second sampling period was used for the recursion of the RLS algorithm. It was found that an increase in sampling period also leads to an increase in the convergence time. For example, a 30 second sampling period increased the convergence time to appoximately 80 minutes.

Figure 6.7 only shows the diagonal elements of the scaling factor matrix **G**. The off-diagonal elements stay near zero due to the nature of this calibration example - i.e. the magnetometer model assumed orthogonal magnetometer axes and no internal or external misalignment between the magnetometer axes and satellites's body axes.

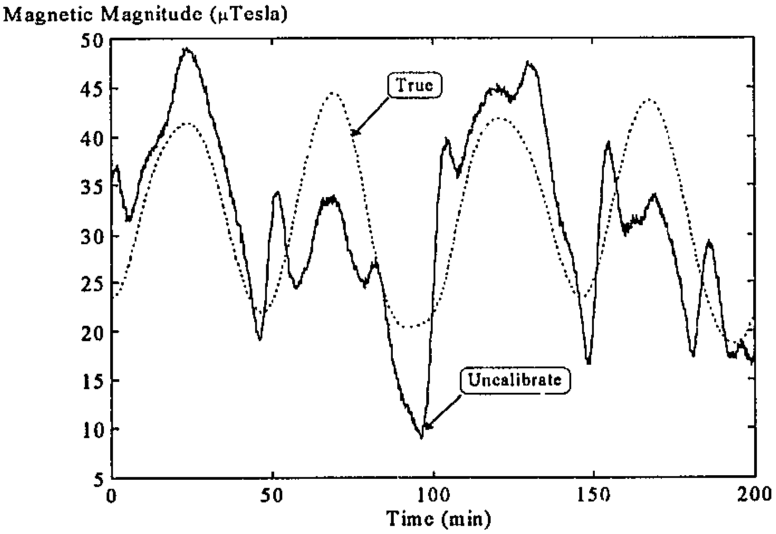

Figure 6.1 Geomagnetic field magnitude of the uncalibrated magnetometer

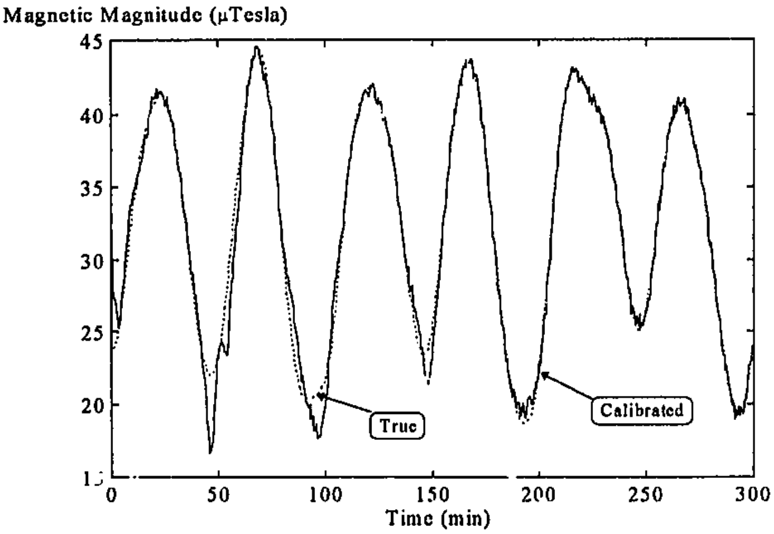

Figure 6.2 Geomagnetic field magnitude for the LMS calibrated magnetometer

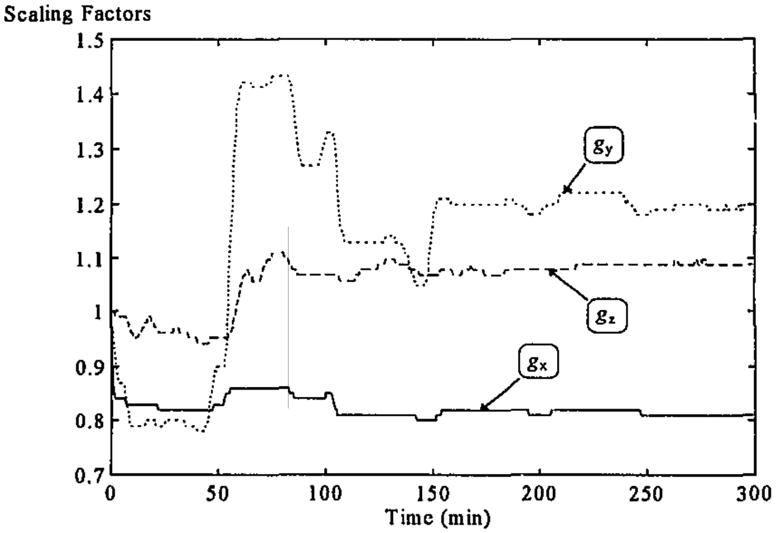

Figure 6.3 Scaling factor convergence for the LMS calibrated magnetometer

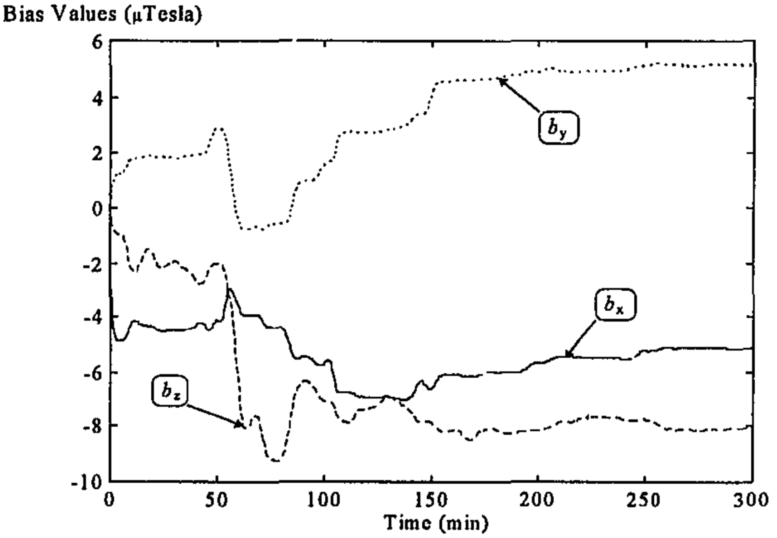

Figure 6.4 Bias value convergence for the LMS calibrated magnetometer

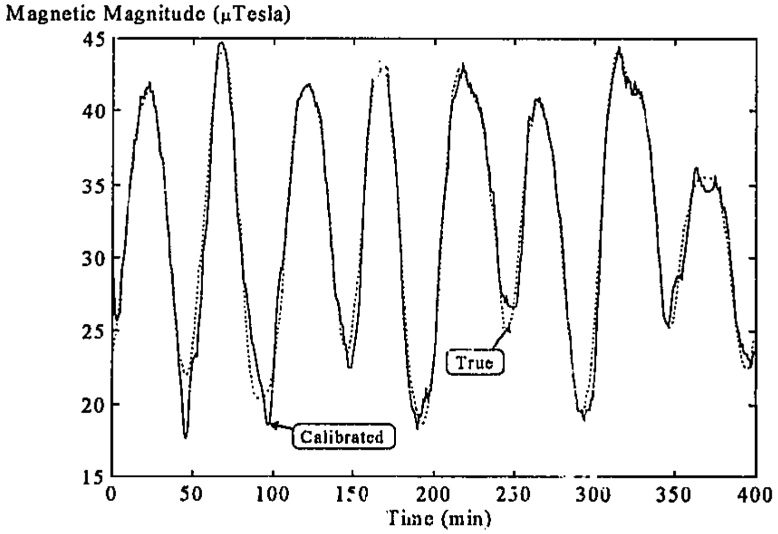

Figure 6.5 Convergence errors for non-orthogonal axes during LMS calibration

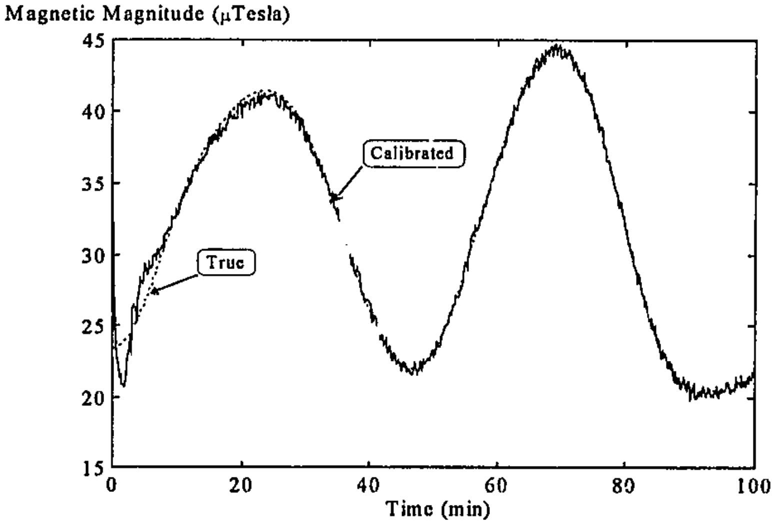

Figure 6.6 Geomagnetic field magnitude for the RLS calibrated magnetometer

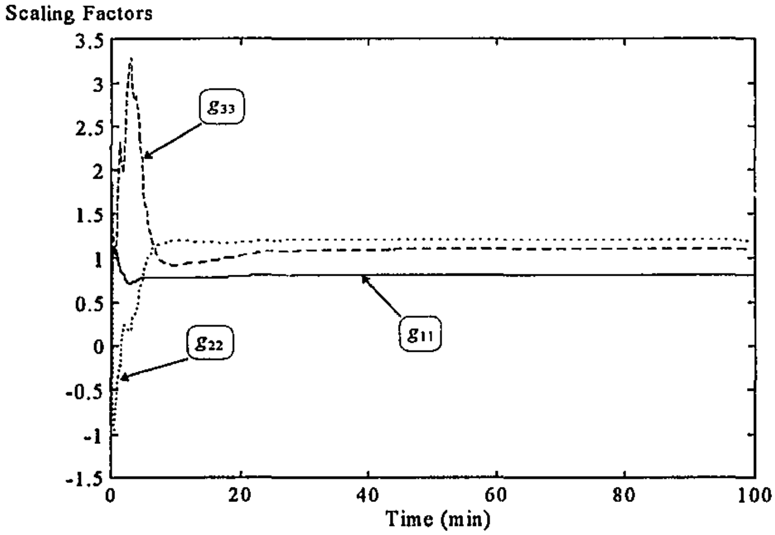

Figure 6.7 Scaling factor convergence for the RLS calibrated magnetometer

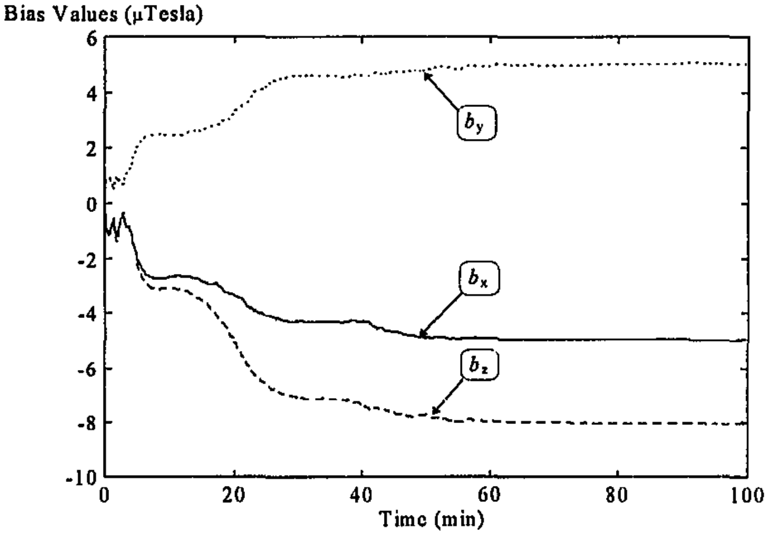

Figure ċ.8 Bias value convergence for the RLS calibrated magnetometer

6.5 Conclusions

Two magnetometer calibration algorithms were compared in this chapter. The LMS method as first presented by Lee *et.al.* [1994] is a simple, computationally undemanding algorithm and no attitude knowledge of the satellite is required. It is, therefore, the ideal method to use during the initial phase of a satellite's mission, before attitude stability has been obtained and when attitude determination is still difficult to do accurately. This method does, however, deliver slow convergence and the calibration accuracy is also affected by measurement noise. Misalignment and non-orthogonality of the magnetometer axes are not compensated for during the calibration process and this can lead to unacceptable errors.

A new, slightly more involved RLS algorithm is proposed in this chapter to solve or improve most of the problems of the LMS algorithm. This method, however, requires attitude knowledge of the spacecraft. It is proposed that this method be used during normal mission conditions, when the satellite is stabilized and attitude measurement sensors are operational. The performance of the RLS algorithm with respect to the convergence speed and immunity to measurement noise is far superior if compared to the LMS method. It is also robust against variations in the initial conditions of the parameters and covariance matrix **P**, whereas the LMS algorithm is much more sensitive to changes in the step sizes.

6.6 References

Alfriend K.T. [1975]
"Magnetic Attitude Control System for Dual Spin Satellites", *AIAA Journal*, Vol.13, June 1975, pp.817-822.

Åström K.J. and Wittenmark B. [1989]
Adaptive Control, Addison-Wesley, Massachusetts.

Hodgart M.S. [1989]
"Gravity Gradient and Magnetorquing Attitude Control for Low-Cost Low Earth Orbit Satellites - The UoSAT Experience", *Ph.D submission at the University of Surrey*, June 1989.

Lee H., Park D. and Choi S.D. [1994]
"In-Flight Calibration of Magnetometer by LMS Algorithm for the Korean Experimental Micro Satellite KITSAT-1 & 2", *IEEE/SAIEE Symposium on Small Satellites and Control*, University of Stellenbosch, South Africa, October 1994.

Lerner G.M. and Shuster M.D. [1981]
"In-Flight Magnetometer Calibration and Attitude Determination for Near-Earth Spacecraft", *Journal of Guidance, Control, and Dynamics*, Vol.4, No.5, Sept.-Oct. 1981, pp.518-522.

Thompson R.H., Neal G.F. and Shuster M.D. [1984]
"Magnetometer Bias Determination ans Spin-Axis Attitude Estimation for the AMPTE Mission", *Journal of Guidance, Control, and Dynamics*, Vol.7, No.4, July-August 1984, pp.505-507.

7. CONCLUSION

7.1 Summary of Contributions

Several new attitude control and estimation techniques for earth orbiting satellites were presented in this thesis document. The research was directed mainly to the development of advanced attitude determination and control algorithms for LEO micro satellites with stringent performance, cost and power requirements, such as SUNSAT. Most of the results are general enough and can be applied to a wide range of space missions. The new results were extensively tested by computer simulation and compared to current state of the art methods where applicable.

7.1.1 MAGNETIC TORQUER CONTROL

Magnetic torquer actuation is a reliable and cheap way of applying an external torque to a satellite. Initial detumbling and boom deployment of a gravity gradient stabilized satellite were successfully demonstrated using a new method based on the cross-product control law. Simultaneous detumbling from any initial angular momentum (assuming a high enough sampling rate) and preparation for boom deployment are achievable within a single orbit. No accurate attitude knowledge is needed, only approximate estimates of the angular body rates are used. These estimates can easily be obtained from magnetometer measurements alone. Any residual libration after boom deployment depends mainly on the time instant the boom is released. A simple earth detection sensor (albedo detector) and a sampling period of 10 seconds showed pitch librations of less than 20° peak-to-peak after boom extension during a SUNSAT simulation.

A novel magnetic torquer controller based on fuzzy design principles was designed to do libration damping and spin rate regulation for gravity gradient satellites. The fuzzy controller showed damping performance comparable to an optimized cross-product controller, but required less power. An reduction of almost 100% in power consumption was obtained during simulation when a cylic external disturbance torque was applied to the satellite. The spin rate regulation performance of the fuzzy controller was also more accurate (less disturbed) compared to the cross-product method. Stability of the fuzzy algorithm was furthermore proven analytically. Residual maximum peak-to-peak libration errors of 4° and 2° in pitch and roll respectively, were achieved for SUNSAT in a slightly elliptical orbit (e = 0.03) during normal solar activity. This can be compared to a minimum energy, uncontrolled, peak-to-peak pitch libration of 11° under similar simulation conditions.

7.1.2 REACTION WHEEL CONTROL

Accurate attitude control and fast large angle maneuvers can easily be done using at least three reaction wheels with mutually perpendicular axes. A variation of a globally stable quaternion feedback control law was implemented to do accurate pointing and tracking control on a nominally nadir pointing satellite. Expressions to design the feedback gains for specific closed loop settling time and damping factor specifications were also given. With a perfect attitude knowledge assumption, pointing and tracking errors of less 0.02° in attitude were achieved during simulation of these controllers when applied to SUNSAT.

A practical, near minimum-time, eigenaxis rotation method was developed for a 3-axis reaction wheel and gravity gradient stabilized, nadir pointing satellite. This new method considers the maximum torque and speed constraints of reaction wheels and is robust against modelling errors, e.g. unmodelled external disturbance torques and spacecraft MOI uncertainty. Simulations showed an overall improved slew time versus control effort performance compared to a simulation optimized, eigenaxis, quaternion feedback controller. Stability, in spite of MOI mismatches for the new controller, was proven analytically.

7.1.3 MOMENTUM DUMPING

Two new optimal desaturation algorithms for a nadir-pointing, 3-axis reaction wheel stabilized satellite, using magnetic torquing only, were designed. The first method uses LQR feedback gains to obtain the magnetic dipole moment of the magnetic coils from the wheel angular momentum vector. Due to the time-varying nature of the local geomagnetic field, the feedback gain matrix will also vary with orbital position. One solution shown for a practical implementation, is to compute discrete feedback gain matrices off-line and to store it onboard in a 2-dimensional look-up table indexed by the sub-satellite position coordinate. Another practical solution demonstarted, is to solve a quasi-static LQR Riccati equation on-line to obtain the feedback gain matrix. This is a computationally demanding effort, but no large look-up tables are needed.

The second new method uses the optimal solution of a fixed terminal time, minimum-energy problem. The solution is however specific to the part and duration of the orbit where the desaturation control will be implemented. The computationally intensive part of the solution can be computed off-line and the resultant constant gain matrix uploaded, preceding the desaturation period. The time-varying magnetic dipole moment is then calculated on-line using a few multiplications to enable minimum-energy desaturation of any initial wheel momentum vector.

The new optimal controllers outperformed a cross-product desaturation controller, with respect to the desaturation efficiency, during simulation. The minimum-energy (MEDL) controllers consumed the least amount of energy as expected, but was less robust against modelling errors due to its open-loop structure. The feedback nature of the LQR desaturation controllers make them more robust and practical in an application where momentum dumping is not done specific to the satellite's orbital location.

7.1.4 ATTITUDE DETERMINATION

A new Kalman filter based estimator was designed to extract angular rate information from magnetometer vector measurements. This estimator can be used during the initial stage of an earth orbiting satellite's life, when it still tumbling in an uncontrolled manner. The estimated angular rate vector can then be used by the magnetic detumbling controller and to prepare the satellite for boom deployment. Simulations showed tracking of the true orbit referenced angular rate vector within half an orbit, from any initial unknown angular rate vector. Due to the rotation of the geomagnetic field vector once per orbit within the orbit referenced coordinates, a maximum estimation error of $\pm\ \omega_o$ occurred. Performance of the detumbling controller was, however, not impaired much by this relative small error.

Two new extended Kalman filters were presented to determine the full satellite's state (attitude, angular rate and main external disturbance torque) from pairs of vector measurements. The first vector is obtained from any angular sensor measurement and the second from a model of the sensor within a fixed reference frame. The first EKF is an extension of a known magnetometer based estimator, but adapted to be applicable to spinning and nadir pointing satellites as well. The second EKF is new and based on a known quaternion estimation method. Simulations indicated superior performance of the second EKF and convergence within a single orbit from a wider range of initial conditions compared to the first EKF.

Two implementations of the basic EKF algorithms were applied to SUNSAT. The first obtained its vector measurement pair from magnetometer sensor data plus an onboard IGRF geomagnetic field model. This implementation should result in practical attitude determination errors of less than ± 1°. These estimation errors are expected, based on published data, indicating IGRF modelling errors to be the main contributing factor. The second implementation used a combination of the higher accuracy attitude sensors, e.g. the sun and horizon sensors and their respective models. Simulations with expected sensor noise and external disturbances showed maximum attitude estimation errors as small as ± 0.1° during active sensor measurement periods.

7.1.5 MAGNETOMETER CALIBRATION

It will be crucial to have a well calibrated magnetometer during vector measurements for attitude determination to accuracies only limited by the IGRF model. A new RLS algorithm for in-flight magnetometer calibration was presented and compared to a published self-calibrating LMS algorithm. The accuracy, robustness against noise, step size and convergence speed of the new RLS method were superior to the LMS method. Both methods calibrate the scaling and bias of the respective magnetometer channels, but the RLS method can also compensate for misalignment and non-orthogonality of the magnetometer axes. The RLS method, however, needs attitude information and vector data from an IGRF model, whereas only magnitude data from an IGRF model is required by the LMS method.

The LMS calibration routine can therefore be used during the initial part of the satellite's mission, when attitude estimation is not yet feasible, and the RLS calibration routine used later when the EKF estimators become operational.

7.2 Onboard Software Structure

A few recommendations for the future implementation of the ADCS algorithms in onboard software will be given in this final paragraph and in Appendix G. Although the software structure presented, is tailored specifically for the SUNSAT hardware, the overall functionality is general and sets up a framework to include the various algorithms presented in the thesis.

The software structure of the attitude control processor (ACP), which will be a T800 transputer in SUNSAT's case, is given in Figure 7.1. As stated in Paragraph 1.3.4, the software will be implemented as tasks scheduled by a hard real-time kernel. Some tasks will be triggered by asynchronous events, such as the interface control processor (ICP) and onboard computer (OBC) communication input tasks. Others will be timer driven, such as the direct magnetorquer, reaction wheel and sensor interface tasks. The rest will be data (message) driven, e.g. the sensor data calibration, environmental modelling, control law algorithms, attitude test and validation, attitude estimators, ADCS manager, ICP and OBC communication output tasks.

Under normal conditions the ICP will be selected to directly interface to the actuator and sensor hardware. The ACP will then receive, using the ICP communication input task, filtered sensor data via an UART interface (see Figure 1.2) every second. The control data for the actuators will be send through the same UART interface from the ACP, using the ICP communication output task. The data transfer will take place every second for the reaction wheels and every 10 seconds for the magnetorquers.

The ICP must then determine and update the reference wheel speed commands and the magnetorquer switching instances at 100 ms intervals. During abnormal conditions, e.g. when the ICP is not functioning properly, the ACP has to be connected directly to the actuators and sensors. This is achieved by switching the multiplexor of Figure 1.2. The functionality of the ICP will then be implemented in the direct magnetorquer, reaction wheel and sensor interface tasks.

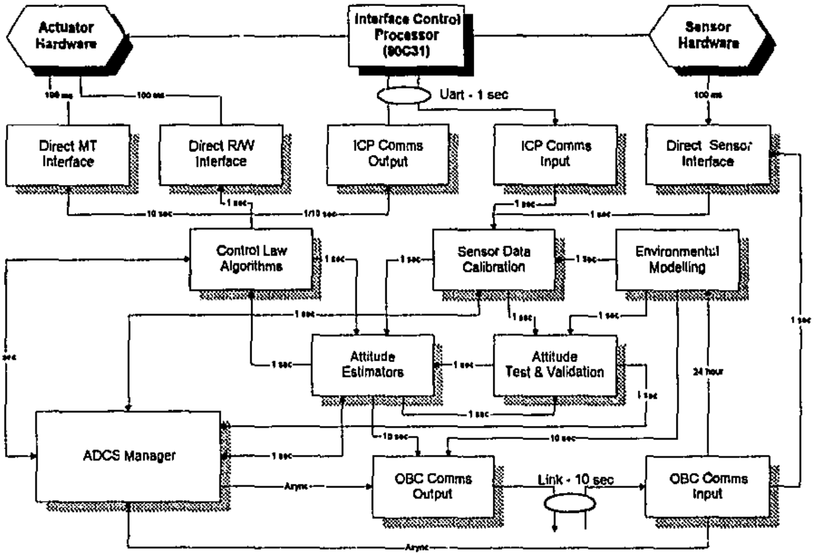

Figure 7.1 Software structure of the Attitude Control Processor

The proposed ACP tasks are listed in Appendix G to convey ideas regarding task interdependency, functionality and scheduling.

Appendix A. IGRF Magnetic Field Modelling

Note: The following theory were extracted from Appendix H in Wertz [1986].

A spherical harmonic IGRF (International Geomagnetic Reference Field) model is used to obtain the reference geomagnetic field vector on SUNSAT. The magnetic field, **B** can be represented as the gradient of a scalar potential function, V:

$$\mathbf{B} = -\nabla V \tag{A.1}$$

V can be conveniently represented by a series of spherical harmonics,

$$V(r,\theta,\phi) = a\sum_{n=1}^{k}\left(\frac{a}{r}\right)^{n+1}\sum_{m=0}^{n}\left(g_n^m \cos m\phi + h_n^m \sin m\phi\right)P_n^m(\theta) \tag{A.2}$$

where,

a = equatorial radius of the earth (6371.2 km)

g and h = Gaussian coefficients of the IGRF model

P = Legendre functions (Schmidt normalized)
r = geocentric distance
θ = coelevation (south positive)
ϕ = east longitude from Greenwich

The Gaussian coefficients are determined empirically by a least-squares fit to measurements of the field and are updated every 5 years. Table A.1 gives the coefficients for the period 1990 to 1995. First-order time derivatives (secular terms) are also given to determine values within the 5 year period. Computation time required for the field model can be reduced significantly by solving the Legendre functions recursively. The first step is to convert the Legendre functions from Schmidt to Gauss normalization. The Gauss functions are related to the Schmidt functions by,

$$P_n^m = S_{n,m}P^{n,m} \tag{A.3}$$

where,

$$S_{n,m} \equiv \left[\frac{\left(2-\delta_m^0\right)(n-m)}{(n+m)!}\right]^{0.5}\frac{(2n-1)!!}{(n-m)!} \tag{A.4}$$

with,

$\delta_i^j = 1$, if $i = j$, and 0 otherwise

The $S_{n,m}$ factors are best combined with the Gaussian coefficients because they are independent of r, θ and ϕ and must be calculated only once. Thus, we define,

$$\begin{aligned} g^{n,m} &\equiv S_{n,m} g_n^m \\ h^{n,m} &\equiv S_{n,m} h_n^m \end{aligned} \tag{A.5}$$

Using mathematical induction, it is possible to derive the following recursion relations for $S_{n,m}$:

$$\begin{aligned} S_{0,0} &= 1 \\ S_{n,0} &= S_{n-1,0}\left[\frac{2n-1}{n}\right] && n \geq 1 \\ S_{n,m} &= S_{n,m-1}\sqrt{\frac{(n-m+1)(\delta_m^1+1)}{n+m}} && m \geq 1 \end{aligned} \tag{A.6}$$

The $P^{n,m}$ can be similarly obtained from the following recursive relations:

$$\begin{aligned} P^{0,0} &= 1 \\ P^{n,n} &= \sin\theta P^{n-1,n-1} \\ P^{n,m} &= \cos\theta P^{n-1,m} - K^{n,m} P^{n-2,m} \end{aligned} \tag{A.7}$$

where,

$$\begin{aligned} K^{n,m} &\equiv \frac{(n-1)^2 - m^2}{(2n-1)(2n-3)} && n > 1 \\ K^{n,m} &\equiv 0 && n = 1 \end{aligned} \tag{A.8}$$

Because the gradient in Eq. (A.1) will lead to partial derivatives of the $P^{n,m}$, we need:

$$\begin{aligned} \frac{\partial P^{0,0}}{\partial\theta} &= 0 \\ \frac{\partial P^{n,m}}{\partial\theta} &= (\sin\theta)\frac{\partial P^{n-1,n-1}}{\partial\theta} + \cos(\theta) P^{n-1,n-1} \\ \frac{\partial P^{n,m}}{\partial\theta} &= (\cos\theta)\frac{\partial P^{n-1,m}}{\partial\theta} - \sin(\theta) P^{n-1,m} - K^{n,m}\frac{\partial P^{n-2,m}}{\partial\theta} \end{aligned} \tag{A.9}$$

Also note that,

$$\begin{aligned} \cos m\phi &= \cos((m-1)\phi)\cos\phi - \sin((m-1)\phi)\sin\phi \\ \sin m\phi &= \sin((m-1)\phi)\cos\phi + \cos((m-1)\phi)\sin\phi \end{aligned} \tag{A.10}$$

Given the IGRF coefficients $g^{n,m}$ and $h^{n,m}$ and the recursive relations in Eqs. (A.6) through (A.10), the **B** field vector is calculated from Eqs. (A.1) and (A.2) as:

$$
\begin{aligned}
B_r &= \frac{-\partial V}{\partial r} = \sum_{n=1}^{k}\left(\frac{a}{r}\right)^{n+2}(n+1)\sum_{m=0}^{n}\left(g^{n,m}\cos m\phi + h^{n,m}\sin m\phi\right)P^{n,m}(\theta) \\
B_\theta &= \frac{-1}{r}\frac{\partial V}{\partial \theta} = -\sum_{n=1}^{k}\left(\frac{a}{r}\right)^{n+2}\sum_{m=0}^{n}\left(g^{n,m}\cos m\phi + h^{n,m}\sin m\phi\right)\frac{\partial P^{n,m}(\theta)}{\partial \theta} \qquad \text{(A.11)} \\
B_\phi &= \frac{-1}{r\sin\theta}\frac{\partial V}{\partial \phi} = \frac{-1}{\sin\theta}\sum_{n=1}^{k}\left(\frac{a}{r}\right)^{n+2}\sum_{m=0}^{n} m\left(-g^{n,m}\sin m\phi + h^{n,m}\cos m\phi\right)P^{n,m}(\theta)
\end{aligned}
$$

where,

B_r = radial component (Outward positive)
B_θ = coelevation component (South positive)
B_ϕ = azimuthal component (East positive)

Table A.1 Eighth order IGRF Gaussian Coefficients for EPOCH 1990-1995

n	m	g (nT)	h (nT)	dg/dt (nT/yr)	dh/dt (nT/yr)	n	m	g (nT)	h (nT)	dg/dt (nT/yr)	dh/dt (nT/yr)
1	0	-29775	--	18.0	--	6	2	60	83	1.8	-1.3
1	1	-1851	5411	10.6	-16.1	6	3	-178	68	1.3	0.0
2	0	-2136	--	-12.9	--	6	4	2	-52	-0.2	-0.9
2	1	3058	-2278	2.4	-15.8	6	5	17	2	0.1	0.5
2	2	1693	-380	0.0	-13.8	6	6	-96	27	1.2	1.2
3	0	1315	--	3.3	--	7	0	77	--	0.6	--
3	1	-2240	-287	-6.7	4.4	7	1	-64	-81	-0.5	0.6
3	2	1246	293	0.1	1.6	7	2	4	-27	-0.3	0.2
3	3	807	-348	-5.9	-10.6	7	3	28	1	0.6	0.8
4	0	939	--	0.5	--	7	4	1	20	1.6	-0.5
4	1	782	248	0.6	2.6	7	5	6	16	0.2	-0.2
4	2	324	-240	-7.0	1.8	7	6	10	-23	0.2	0.0
4	3	-423	87	0.5	3.1	7	7	0	-5	0.3	0.0
4	4	142	-299	-5.5	-1.4	8	0	22	--	0.2	--
5	0	-211	--	0.6	--	8	1	5	10	-0.7	0.5
5	1	353	47	-0.1	-0.1	8	2	-1	-20	-0.2	-0.2
5	2	244	153	-1.6	0.5	8	3	-11	7	0.1	0.3
5	3	-111	-154	-3.1	0.4	8	4	-12	-22	-1.1	0.3
5	4	-166	-69	-0.1	1.7	8	5	4	12	0.0	0.4
5	5	-37	98	2.3	0.4	8	6	4	11	-0.1	-0.5
6	0	61	--	1.3	--	8	7	3	-16	-0.5	-0.3
6	1	64	-16	-0.2	0.2	8	8	-6	-11	-0.6	0.6

Software Code

Model:

IGRF 8th order model

Inputs:

COEF	: array[1..80] of double;	*{IGRF Gaussian Coefficients $g^{m,n}$ and $h^{m,n}$ (nT) at current EPOCH from Eqs. A.4-A.5}*
LON	: double;	*{Sub-satellite Longitude (rad)}*
LAT	: double;	*{Sub-satellite Latitude (rad)}*
PLON	: double;	*{Previous (-10 sec) sub-sat. Longitude (rad)}*
PLAT	: double;	*{Previous (-10 sec) sub-sat. Latitude (rad)}*
RADIUS	: double;	*{Satellite distance from geocentre (km)}*
AA	: array[1..3,1..3] of double;	*{DCM to transform from orbit to body axes}*

Outputs:

BET	: double;	*{Angle between velocity (X_o) vector and geometric north (rad), see Figure A.1}*
BX0,BY0,BZ0	: double;	*{Mag. field components in orbital axes (μT)}*
BX,BY,BZ	: double;	*{Mag. field components in body axes (μT)}*

Pascal Code:

```
procedure MAGFIELD;
var
 CBE,SBE,BET,NBX,NBY,NBZ     : double;
 THETA,PHI,BR,BN,BE          : double;

 procedure MAGMODEL;                {IGRF Model}
 const
  NMAX = 8;
 var
  PP,DP                  : array[0..NMAX,0..NMAX] of double;
  CPHM,SPHM              : array[0..NMAX] of double;
  AR,KK,BB1,BB2,BB3      : double;
  STH,CTH,SPH,CPH        : double;
  N,M,II                 : integer;
 begin
  STH:=sin(THETA); CTH:=cos(THETA);
  for N:=0 to NMAX do               {Obtain Legendre functions, Eqs. A.7-A.9}
  for M:=0 to NMAX do begin
   PP[N,M]:=0.0; DP[N,M]:=0.0;
  end;
  PP[0,0]:=1.0; DP[0,0]:=0.0;
  for N:=1 to NMAX do
   for M:=0 to N do begin
    if (N=1) then KK:=0.0
```

```
      else KK:=((N-1)*(N-1)-M*M)/((2*N-1)*(2*N-3));
      if (N=M) then begin
        PP[N,N]:=STH*PP[N-1,N-1];
        DP[N,N]:=STH*DP[N-1,N-1]+CTH*PP[N-1,N-1];
      end
      else begin
        PP[N,M]:=CTH*PP[N-1,M];
        DP[N,M]:=CTH*DP[N-1,M]-STH*PP[N-1,M];
        if (N-2)>=M then begin
          PP[N,M]:=PP[N,M]-KK*PP[N-2,M];
          DP[N,M]:=DP[N,M]-KK*DP[N-2,M];
        end;
      end;
    end;
  SPH:=sin(PHI); CPH:=cos(PHI);             {Obtain trigonometric functions, Eq. A.10}
  CPHM[0]:=1.0; SPHM[0]:=0.0;
  for M:=1 to NMAX do begin
    CPHM[M]:=CPHM[M-1]*CPH-SPHM[M-1]*SPH;
    SPHM[M]:=CPHM[M-1]*SPH+SPHM[M-1]*CPH;
  end;
  BR:=0.0; BN:=0.0; BE:=0.0;                {Compute magnetic field components, Eq. A.11}
  AR:=6371.2/RADIUS; KK:=AR*AR;
  II:=1;
  for N:=1 to NMAX do begin
    KK:=KK*AR;
    BB1:= COEF[II]*PP[N,0];
    BB2:= COEF[II]*DP[N,0];
    BB3:= 0.0;
    inc(II);
    for M:=1 to N do begin
      BB1:=BB1+(COEF[II]*CPHM[M]+COEF[II+1]*SPHM[M])*PP[N,M];
      BB2:=BB2+(COEF[II]*CPHM[M]+COEF[II+1]*SPHM[M])*DP[N,M];
      BB3:=BB3+(COEF[II+1]*CPHM[M]-COEF[II]*SPHM[M])*M*PP[N,M];
      II:=II+2;
    end;
    BR:=BR+KK*(N+1)*BB1;
    BN:=BN+KK*BB2;
    BE:=BE-KK*BB3;
  end;
  BR:=1e-3*BR;                              {Radial component - outwards positive}
  BN:=1e-3*BN;                              {North positive component}
  BE:=1e-3*BE/STH;                          {East positive component}
 end;{MAGMODEL}

begin                                       {Compute rotation angle BET, see Figure A.1}
 BET:=arctan((PLON-LON)/(LAT-PLAT));
 if BET<0.0 then BET:=pi+BET;
 CBE:=cos(BET); SBE:=sin(BET);
 THETA:=0.5*pi-LAT;                         {Obtain coelevation angle θ}
 PHI:=-LON;                              {Obtain azimuthal angle φ }
 MAGMODEL;                               {IGRF computation}
 BX0:= BN*CBE-BE*SBE;                    {Transform to orbital coordinates}
 BY0:= BN*SBE+BE*CBE;
 BZ0:=-BR;                               {Transform to body coordinates}
 BX := BX0*AA[1,1] + BY0*AA[1,2] + BZ0*AA[1,3];
 BY := BX0*AA[2,1] + BY0*AA[2,2] + BZ0*AA[2,3];
 BZ := BX0*AA[3,1] + BY0*AA[3,2] + BZ0*AA[3,3];
end;{MAGFIELD}
```

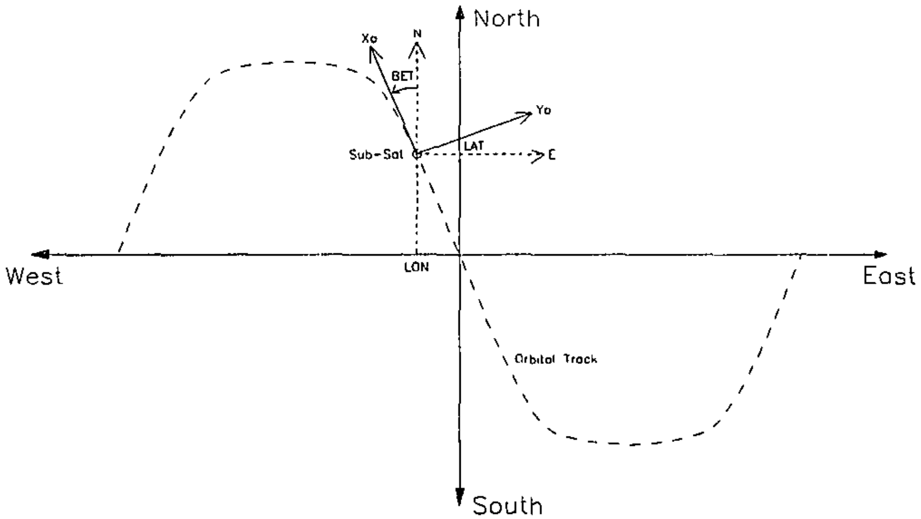

Figure A.1 North/East to X_o/Y_o coordinate transformation

Reference

Wertz J.R. [1986].
Spacecraft Attitude Determination and Control, D.Reidel Publishing Company Boston U.S.A., Reprint 1986.

Appendix B. Magnetorquer Controllers

B.1 Pre-boom Deployment Controller

This controller is presented in Paragraph 2.2. To prepare for boom deployment, the satellite's angular rate vector must first be aligned to the orbital normal. This can be achieved by controlling the orbit referenced X- and Z-axis body rate components towards zero and the Y-axis component towards a certain reference value. The Y-axis reference angular rate must be chosen to conserve the angular momentum before and after boom deployment. Conservation of angular momentum means that the ratio of increase in the MOI around the body Y-axis must be equal to the ratio of decrease in the inertially referenced body Y-axis angular rate (Eq. 2.9). The cross-product control law (Eq. 2.6) utilizing the scaled rate error vector of Equation 2.15, will then be used to achieve these goals. The complete control law is coded below as the Pascal procedure DETUMBLE.

Due to the expected high initial spin rates, the geomagnetic field vector can change direction rapidly within the body coordinates. The directional change can easily be a few degrees within the controller sampling period of 10 seconds. The PWM method of switching the torquer coils centers the control pulses in the middle of the sampling interval (see Figure 2.1). It is therefore more accurate to use an estimated magnetic field vector as expected midway through the sampling interval, than the magnetometer measurement obtained at the beginning of the sampling interval, when computing the magnetorquing vector (Eq. 2.6). The local procedure CENTRE employs an Euler axis rotation (Eqs. 2.17 to 2.19) to compute this expected geomagnetic field vector in body coordinates.

Software Code

Control Law:

Cross-product algorithm to prepare for boom deployment

Inputs:

```
BX, BY, BZ      : double;    {Magnetometer field measurement (μT)}
WX,WY,WZ        : double;    {Orbit referenced angular body rates (rad/s)}
W0              : double;    {Average orbit angular rate (rad/s)}
TS              : double;    {Sampling period = 10 seconds}
```

Outputs:

```
MTX,MTY,MTZ     : double;    {Magnetorquer on-time fraction of Ts (0-0.8}
MPX,MPY,MPZ     : integer;   {Magnetorquer switching polarity (±1)}
```

Pascal Code:

```
procedure DETUMBLE;
var
 EBX,EBY,EBZ       : double;
 ERX,ERY,ERZ       : double;
 BABS              : double;

 procedure CENTRE;              {Mag.vector estimation in centre of control pulse}
 var
  PHI,CPH,SPH,WRS        : double;
  E1,E2,E3               : double;
  AE                     : array[1..3,1..3] of double;
 begin
  WRS := sqrt(WX*WX+WY*WY+WZ*WZ);
  if WRS = 0.0 then begin
   EBX := BX;
   EBY := BY;
   EBZ := BZ;
  end
  else begin
   PHI := 0.5*TS*WRS;                                   {Eq. 2.17a}
   CPH := cos(PHI); SPH := sin(PHI);
   E1 := WX/WRS;                                        {Eq. 2.17b}
   E2 := WY/WRS;
   E3 := WZ/WRS;
   AE[1,1] := CPH+E1*E1*(1.0-CPH);                      {Eq. 2.19}
   AE[1,2] := E1*E2*(1.0-CPH)+E3*SPH;
   AE[1,3] := E1*E3*(1.0-CPH)-E2*SPH;
   AE[2,1] := E1*E2*(1.0-CPH)-E3*SPH;
   AE[2,2] := CPH+E2*E2*(1.0-CPH);
   AE[2,3] := E2*E3*(1.0-CPH)+E1*SPH;
   AE[3,1] := E1*E3*(1.0-CPH)+E2*SPH;
   AE[3,2] := E2*E3*(1.0-CPH)-E1*SPH;
   AE[3,3] := CPH+E3*E3*(1.0-CPH);
   EBX := AE[1,1]*BX + AE[1,2]*BY + AE[1,3]*BZ;     {Eq. 2.18}
   EBY := AE[2,1]*BX + AE[2,2]*BY + AE[2,3]*BZ;
   EBZ := AE[3,1]*BX + AE[3,2]*BY + AE[3,3]*BZ;
  end;
 end; {CENTRE}
```

```
begin {DETUMBLE}
 CENTRE;
 ERX := 0.2*WX/W0;                                  {Eq. 2.15}
 ERY := 0.2*(WY/W0+19.0);
 ERZ := 0.1*WZ/W0;
 BABS := sqrt(BX*BX+BY*BY+BZ*BZ);                   {Eq. 2.6}
 MTX := (ERY*EBZ-ERZ*EBY)/BABS;
 MTY := (ERZ*EBX-ERX*EBZ)/BABS;
 MTZ := (ERX*EBY-ERY*EBX)/BABS;
 if MTX>0.01 then MPX := 1              {Obtain magnetorquer on-time fraction}
 else if MTX<-0.01 then begin           {and switching polarity for X-axis}
  MPX := -1;
  MTX := abs(MTX);
 end
 else MTX:=0.0;
 if MTX>0.8 then MTX := 0.8;
 if MTY>0.01 then MPY := 1              {... for Y-axis}
 else if MTY<-0.01 then begin
  MPY := -1;
  MTY := abs(MTY);
 end
 else MTY:=0.0;
 if MTY>0.8 then MTY := 0.8;
 if MTZ>0.01 then MPZ := 1              {... for Z-axis}
 else if MTZ<-0.01 then begin
  MPZ := -1;
  MTZ := abs(MTZ);
 end
 else MTZ:=0.0;
 if MTZ>0.8 then MTZ := 0.8;
end; {DETUMBLE}
```

B.2 Libration Damping Fuzzy Controller

This controller is presented in Paragraph 2.3.3. It is used only as an angular rate regulator: For example, to regulate the orbit referenced X- and Y-axis angular body rates towards zero (do libration damping) and to maintain a certain reference Z-axis angular rate. The full fuzzy controller consists of three MISO fuzzy control blocks (see Figure 2.5), one for each magnetorquer axis. Six fuzzy input variables are used: Three of these are obtained from the satellite's angular rate measurement and the other three from an estimation of the magnetorquer vector.

The input variables are then mapped using the membership functions of Figure 2.6. A set of eight rules for each magnetorquer (see Table 2.1 to 2.3) is then evaluated to determine the correct switching polarity for each rule. The truth value for each rule is then inferred and used to scale the binary output (using the correllation-product norm) to obtain, not only the correct switching polarity, but also the required magnetorquing magnitude. All the rule outputs are then combined to obtain the overall magnetorquer polarity and magnitude, while minimizing any cross disturbances between opposing rules (Eq. 2.25).

The software code in procedure FUZZY below, implements a fuzzy algorithm with no overlapping between the membership functions and selects only one magnetorquer (most favourable - one with the highest magnitude) during each sampling period. This controller is the most power efficient and gives the best rate regulation performance during simulation of slight elliptical orbits. The sampling period used for libration damping and Z-spin regulation is 60 seconds. The maximum on-time of a magnetorquer can not exceed 80% (48 seconds) of the sampling period to enable the sensitive magnetometer to function during the torque-free (no magnetic disturbance) window.

Software Code

Control Law:

Fuzzy controller for libration damping and Z-spin regulation

Inputs:

BX, BY, BZ	: double;	*{Magnetometer field measurement (μT)}*
WX,WY,WZ	: double;	*{Orbit referenced angular body rates (rad/s)}*
W0	: double;	*{Average orbit angular rate (rad/s)}*
WZ_REF	: double;	*{Reference Z-spin rate (rpo)}*
TS	: double;	*{Sampling period = 60 seconds}*

Outputs:

MTX,MTY,MTZ	: double;	*{Magnetorquer on-time fraction (0-0.8*Ts}*
MPX,MPY,MPZ	: integer;	*{Magnetorquer switching polarity (±1)}*

Constant:

MTQ = 2.0e-5;	{Mag. torquer dipole moment (20 $Am^2 \times 10^{-6}$)}

Pascal Code:

```
procedure FUZZY;
type
 RULEMAP = array[1..8,1..5] of byte;
const
 MAP : RULEMAP =        ((1,0,1,0,1),             {Rule table 2.3, P=1, N=2, u=0/1}
                         (1,0,2,0,0),
                         (2,0,1,0,0),
                         (2,0,2,0,1),
                         (0,1,0,1,1),
                         (0,1,0,2,0),
                         (0,2,0,1,0),
                         (0,2,0,2,1));
var
 NX,NY,NZ,OUTX,OUTY,OUTZ           : double;
 POSW1,NEGW1,POSW2,NEGW2           : double;
 POSN1,NEGN1,POSN2,NEGN2,ZRON2     : double;
 MAGX,MAGY,MAGZ,WZERR              : double;
 MX,MY,MZ,EBX,EBY,EBZ              : double;
 WZFLAG                            : boolean;

 procedure CENTRE;              {Mag vector estimation in centre of control pulse}
 var
  PHI,CPH,SPH,WRS        : double;
  E1,E2,E3               : double;
  AE                     : array[1..3,1..3] of double;
 begin
  WRS := sqrt(WX*WX+WY*WY+WZ*WZ);
  if WRS = 0.0 then begin
   EBX := BX;
   EBY := BY;
   EBZ := BZ;
  end
  else begin
   PHI := 0.5*TS*WRS;                            {Eq. 2.17a}
   CPH := cos(PHI); SPH := sin(PHI);
   E1 := WX/WRS;                                 {Eq. 2.17b}
   E2 := WY/WRS;
   E3 := WZ/WRS;
   AE[1,1] := CPH+E1*E1*(1.0-CPH);               {Eq. 2.19}
   AE[1,2] := E1*E2*(1.0-CPH)+E3*SPH;
   AE[1,3] := E1*E3*(1.0-CPH)-E2*SPH;
```

```
    AE[2,1] := E1*E2*(1.0-CPH)-E3*SPH;
    AE[2,2] := CPH+E2*E2*(1.0-CPH);
    AE[2,3] := E2*E3*(1.0-CPH)+E1*SPH;
    AE[3,1] := E1*E3*(1.0-CPH)+E2*SPH;
    AE[3,2] := E2*E3*(1.0-CPH)-E1*SPH;
    AE[3,3] := CPH+E3*E3*(1.0-CPH);
    EBX := AE[1,1]*BX + AE[1,2]*BY + AE[1,3]*BZ;     {Eq. 2.18}
    EBY := AE[2,1]*BX + AE[2,2]*BY + AE[2,3]*BZ;
    EBZ := AE[3,1]*BX + AE[3,2]*BY + AE[3,3]*BZ;
  end;
end; {CENTRE}

procedure CALC;                                   {Magnetic torque calculation}
begin
  NX := MY*EBZ - MZ*EBY;                                {Eq. 2.2}
  NY := MZ*EBX - MX*EBZ;
  NZ := MX*EBY - MY*EBX;
end; {CALC}

procedure MEMBER(W1,W2,N1,N2:double);             {Evaluate membership functions}
const                                                   {Figure 2.6}
   X1 = 0.0; {0.1}                {Overlapping}
   X2 = 0.0; {1e-5}
begin
  POSW1:=0.0; NEGW1:=0.0;
  if W1 > -X1 then begin          {W1 membership}
    POSW1:=(W1+X1);
    if POSW1 > 1.0 then POSW1:=1.0;
  end;
  if W1 < X1 then begin
    NEGW1:=-(W1-X1);
    if NEGW1 > 1.0 then NEGW1:=1.0;
  end;
  POSW2:=0.0; NEGW2:=0.0;
  if W2 > -X1 then begin          {W2 membership}
    POSW2:=(W2+X1);
    if POSW2 > 1.0 then POSW2:=1.0,
  end;
  if W2 < X1 then begin
    NEGW2:=-(W2-X1);
    if NEGW2 > 1.0 then NEGW2:=1.0;
  end;
  POSN1:=0.0; NEGN1:=0.0;
  if N1 > -X2 then begin              {N1 membership}
    POSN1:=(N1+X2)/1e-4;
    if POSN1 > 1.0 then POSN1:=1.0;
  end;
  if N1 < X2 then begin
    NEGN1:=-(N1-X2)/1e-4;
    if NEGN1 > 1.0 then NEGN1:=1.0;
  end;
  POSN2:=0.0; NEGN2:=0.0;
  if N2 > -X2 then begin              {N2 membership}
    POSN2:=(N2+X2)/1e-4;
    if POSN2 > 1.0 then POSN2:=1.0;
  end;
  if N2 < X2 then begin
    NEGN2:=-(N2-X2)/1e-4;
    if NEGN2 > 1.0 then NEGN2:=1.0;
  end;
  if abs(N2) < 1e-4 then ZRON2:=1.0-abs(N2)/1e-4
  else ZRON2:=0.0;
end; {MEMBER}

function RULE : double;                           {Weighing of rules - Eq. 2.25}
var
  WW              : array[1..8] of double;
  SOUT,MAPIJ      : double;
  I,J             : integer;
begin
  SOUT:=0.0;
  for I := 1 to 8 do begin
    WW[I]:=1.0;
    for J := 1 to 4 do begin
      MAPIJ := MAP[I,J];
      if MAPIJ <> 0 then case J of
        1 : if (MAPIJ = 1) then WW[I]:=WW[I]*POSW1
            else WW[I]:=WW[I]*NEGW1;
        2 : if (MAPIJ = 1) then WW[I]:=WW[I]*POSW2
            else WW[I]:=WW[I]*NEGW2;
        3 : if (MAPIJ = 1) then WW[I]:=WW[I]*POSN1
            else WW[I]:=WW[I]*NEGN1;
        4 : if (MAPIJ = 1) then WW[I]:=WW[I]*POSN2
            else WW[I]:=WW[I]*NEGN2;
      end;
    end;
    if (I < 5) and WZFLAG then WW[I]:=WW[I]*ZRON2;
    if MAP[I,5] = 0 then SOUT:=SOUT+WW[I]
    else SOUT:=SOUT-WW[I];
  end;
  SOUT:=0.3*SOUT;                          {Scaling for magnetorquers - Eq. 2.27}
  if abs(SOUT)<0.001 then SOUT:=0.0;
  RULE:=SOUT;
 end; {RULE}

begin {FUZZY}
 CENTRE;                                   {Estimate geomagnetic vector}
 WZERR:=0.1*(WZ/W0-FRAC_WZ);               {Scale for membership evaluation}
 MX:=MTQ; MY:=0.0; MZ:=0.0;                {Evaluate MX torquer}
 CALC;
 WZFLAG:=true;
 MEMBER(WY/W0,WZERR,NY,NZ);
 OUTX:=RULE;
 MX:=0.0; MY:=MTQ; MZ:=0.0;                {Evaluate MY torquer}
 CALC;
 WZFLAG:=true;
 MEMBER(WX/W0,WZERR,NX,NZ);
 OUTY.=RULE,
 MX:=0.0; MY:=0.0; MZ:=MTQ;                {Evaluate MZ torquer}
 CALC;
 WZFLAG:=false;
 MEMBER(0.1*WX/W0,0.1*WY/W0,NX,NY);
 OUTZ:=RULE;
 MTX := 0.0; MTY := 0.0; MTZ := 0.0;       {Choose most favourable magnetorquer}
 if (abs(OUTX)>abs(OUTY)) and (abs(OUTX)>abs(OUTZ)) then
 begin
  MTX := OUTX;
  if MTX>0.0 then MPX := 1                 {Obtain magnetorquer on-time fraction}
  else begin                               {and switching polarity for X-axis}
   MPX := -1;
   MTX := abs(MTX);
  end
  if MTX > 0.8 then MTX := 0.8;
 end;
 if (abs(OUTY)>abs(OUTX)) and (abs(OUTY)>abs(OUTZ)) then
 begin
  MTY := OUTY;
  if MTY>0.0 then MPY := 1                 {Obtain magnetorquer on-time fraction}
  else begin                               {and switching polarity for Y-axis}
```

```
      MPY := -1;
      MTY := abs(MTY);
    end
    if MTY > 0.8 then MTY := 0.8;
  end;
  if (abs(OUTZ)>abs(OUTX)) and (abs(OUTZ)>abs(OUTY)) then
  begin
    MTZ := OUTZ;
    if MTZ>0.0 then MPZ := 1                {Obtain magnetorquer on-time fraction}
    else begin                              {and switching polarity for Z-axis}
      MPZ := -1;
      MTZ := abs(MTZ);
    end
    if MTZ > 0.8 then MTZ := 0.8;
  end;
end; {FUZZY}
```

B.3 Libration Damping Cross-Product Controller

This controller is presented in Paragraph 2.1.3. It can be used as an alternative to the fuzzy controller. Although it is less power efficient than the fuzzy controller, it has comparable libration damping performance. The main advantage of this controller, however, is the simplicity thereof and its similarity to the detumbling controller. The same software code can then be used to do both detumbling and libration damping. The only difference will be the sampling period (10 seconds for detumbling, 60 seconds for libration damping and Z-spin control) and the angular rate error vector (Eq. 2.15 for detumbling, Eq. 2.20 for damping/Z-spin control).

Software Code

Similar to the procedure DETUMBLE. Only differences: TS = 60 seconds and,

```
  ERX := 0.1*WX/W0;                       {Eq. 2.15}
  ERY := 0.1*WY/W0;
  ERZ := 0.05*(WZ/W0 - WZ_REF);
```

Appendix C. Reaction Wheel Control

3-Axis Stabilization

The main purpose of the reaction wheels on SUNSAT will be to 3-axis stabilize the satellite during imaging. Due to the high pointing accuracy (< 3 mrad in pitch and roll, < 6 mrad in yaw) required, the reaction wheel quaternion feedback controller of Eq. 3.14 will be used during fine attitude pointing. Various pointing and tracking controllers were presented in Paragraph 3.2. The near minimum-time eigenaxis rotation (presented in Paragraph 3.3.2) will be used during large angular maneuvers, to enable fast slewing of the satellite to a new reference pointing attitude.

The Pascal code to implement the pointing regulation and fast slew rotations will be presented next. The procedure COMMAND will be called with the required pointing attitude, specified as pitch, roll and yaw Euler angles, to initialize the respective controllers. If the resultant eigenaxis rotation is less than 10°, the quaternion feedback pointing controller will be selected (RWMODE = 0), else the minimum-time eigenaxis rotation controller (RWMODE = 1) will be selected first. The reaction wheel controllers implemented in procedure RWHEEL must then be called at an one second sampling interval, to compute the X, Y and Z-axis wheel control torques.

Software Code

Reaction Wheel Controller:

Pointing and Large Angular Slew

Inputs:

QQ	. array[1..4] of double;	*{Satellite Quaternion}*
WX,WY,WZ	: double;	*{Orbit ref. angular rate vector (rad/s)}*
WXI,WYI,WZI	: double;	*{Inertial ref. angular rate vector (rad/s)}*
HX,HY,HZ	: double;	*{RWheel momentum vector (kgm^2.rad/s)}*
RADIUS	: double;	*{Satellite distance from geocenter (km)}*

Outputs:

NWX,NWY,NWZ	: double;	*{Reaction wheel torque vector (Nm)}*

Global Variables:

QC	: array[1..4] of double;	*{Commanded Quaternion}*
QE	: array[1. 4] of double;	*{Error Quaternion}*
AA	: array[1..3,1..3] of double;	*{DCM to transform from orbit to body}*

```
WRX,WRY,WRZ       : double;        {Reference slew angular rate vector}
W0                : double;        {Orbit angular rate (rad/s)}
W0X,W0Y,W0Z       : double;        {Orbit angular rate vector in body axes}
RWMODE            : integer;       {RWheel controller mode}
KX,KY,KZ          : double;        {Slew torque fraction per axis}
QTEST             : integer;       {Number of max. QE-vector component}
QHALF             : double;        {Halfway max. QE-vector component}
STIME             : integer;       {Slew maneuver timer (seconds)}
SATURATE          : boolean;       {RWheel momentum saturation flag}
SATTIM            : integer,       {Rwheel saturation timer}
```

Constants:

```
IT = 40.0;        {Transverse MOI (kgm^2)}
IZ = 2.0;         {Z-axis MOI (kgm^2)}
```

Pascal Code:

```
procedure TRANSFORM;
begin                                          {DCM Transformation, Eq. 1.4}
 AA[1,1] := QQ[1]*QQ[1]-QQ[2]*QQ[2]-QQ[3]*QQ[3]+QQ[4]*QQ[4];
 AA[1,2] := 2.0*(QQ[1]*QQ[2]+QQ[3]*QQ[4]);
 AA[1,3] := 2.0*(QQ[1]*QQ[3]-QQ[2]*QQ[4]);
 AA[2,1] := 2.0*(QQ[1]*QQ[2]-QQ[3]*QQ[4]);
 AA[2,2] :=-QQ[1]*QQ[1]+QQ[2]*QQ[2]-QQ[3]*QQ[3]+QQ[4]*QQ[4];
 AA[2,3] := 2.0*(QQ[2]*QQ[3]+QQ[1]*QQ[4]);
 AA[3,1] := 2.0*(QQ[1]*QQ[3]+QQ[2]*QQ[4]);
 AA[3,2] := 2.0*(QQ[2]*QQ[3]-QQ[1]*QQ[4]);
 AA[3,3] :=-QQ[1]*QQ[1]-QQ[2]*QQ[2]+QQ[3]*QQ[3]+QQ[4]*QQ[4];
                                               {Error Quaternion, Eq. 1.15}
 QE[1] := QC[4]*QQ[1]+QC[3]*QQ[2]-QC[2]*QQ[3]-QC[1]*QQ[4];
 QE[2] :=-QC[3]*QQ[1]+QC[4]*QQ[2]+QC[1]*QQ[3]-QC[2]*QQ[4];
 QE[3] := QC[2]*QQ[1]-QC[1]*QQ[2]+QC[4]*QQ[3]-QC[3]*QQ[4];
 QE[4] := QC[1]*QQ[1]+QC[2]*QQ[2]+QC[3]*QQ[3]+QC[4]*QQ[4];
end; {TRANSFORM}

procedure COMMAND(PITCH,ROLL,YAW : double);
const
 DEG = pi/180.0;
var
 CP,SP,CR,SR,CY,SY          : double,
 E1,E2,E3,A1,A2,A3,PHI      : double;
begin
 CP:=cos(PITCH); SP:=sin(PITCH);
 CR:=cos(ROLL); SR:=sin(ROLL);
 CY:=cos(YAW); SY:=sin(YAW);                 {Get commanded quaternion, Eq. 1.6}
 QC[4] .= 0.5*sqrt(1.0+CY*CP-SY*SP*SR+CY*CR+CP*CR);
 QC[1] := (0.25/QC[4])*(SY*SP*CR+CY*SR+CP*SR);
 QC[2] := (0.25/QC[4])*(SP+CY*SP*CR-SY*SR);
 QC[3] := (0.25/QC[4])*(CY*SP*SR+SY*CR+SY*CP);
 TRANSFORM;                                  {Compute DCM and Error Quaternion}
 W0X := W0*AA[1,2]; W0Y := W0*AA[2,2]; W0Z := W0*AA[3,2];
 PHI := arctan(sqrt(1.0-QE[4]*QE[4])/QE[4]); {Get the Euler axis/angle, Eq. 1.3}
 if abs(PHI*DEG) < 10.0 then
  RWMODE := 0;                               {Do pointing control only}
  exit,                                      {if Euler angle < 10
 end;                                        {else do a large angular slew maneuver}
 SP := sin(PHI);
 E1 := IT*QE[1]/SP; E2 := IT*QE[2]/SP; E3 := IZ*QE[3]/SP;
 A1 := abs(E1); A2 := abs(E2); A3 := abs(E3); {Obtain the maximum torque axis}
 if A1 > A2 then
  if A1 > A3 then begin
   KX := E1/A1; KY := E2/A1; KZ := E3/A1;
   QTEST := 1; QHALF := abs(E1*sin(0.5*PHI))/IT;
  end
  else begin
   KZ := E3/A3; KX := E1/A3; KY := E2/A3;
   QTEST := 3; QHALF := abs(E3*sin(0.5*PHI))/IZ;
  end
 else if A2 > A3 then begin
  KY := E2/A2; KX := E1/A2; KZ := E3/A2;
  QTEST := 2; QHALF := abs(E2*sin(0.5*PHI))/IT;
 end
 else begin
  KZ := E3/A3; KX:= E1/A3; KY := E2/A3;
  QTEST := 3; QHALF := abs(E3*sin(0.5*PHI))/IZ;
 end;
 STIME := 0; RWMODE := 1;                    {Start slew maneuver}
 WRX := 0.0; WRY := 0.0; WRZ := 0.0;        {Zero reference angular rate vector}
end; {COMMAND}

function SLEW : double;
const
 HLIM = 228.0;       {Wheel angular momentum limit (kgm^2.rpo) - 5% below max.}
 NLIM = 3.6e-3,      {Wheel torque limit (Nm) - 10% below max.}
var
 NS      : double,
 ACCEL : boolean,
begin
  ACCEL := false;
  case QTEST of                               {Determine halfway mark, Eq 3.27}
1: if abs(QE[1]) > QHALF then ACCEL:=true;
2: if abs(QE[2]) > QHALF then ACCEL:=true;
3: if abs(QE[3]) > QHALF then ACCEL:=true;
  end;
  if ACCEL then begin                         {Acceleration phase}
    inc(STIME);
    if not(SATURATE) then                     {Test for wheel momentum saturation}
      if (abs(HX/W0) > HLIM) or (abs(HY/W0) > HLIM)
        or (abs(HZ/W0) > HLIM) then begin
      SATURATE := true;
      SATTIM := 0;
    end;
    if SATURATE then begin
      inc(SATTIM);
      NS := 0.0;
    end
    else NS := NLIM;
  end
  else begin                                  {Deceleration phase}
    dec(STIME);
    if STIME < 0 then begin                   {Test for end of slew maneuver}
      RWMODE := 0;                            {Return to pointing mode}
      NS := 0.0;
    end;
    if SATURATE then begin
      dec(SATTIM);
      if SATTIM = 0 then SATURATE := false;
      NS := 0.0;
    end
    else NS := -NLIM;
  end;
  return NS;
end; {SLEW}

procedure RWHEEL;
const
  GM = 1.195801e6;          {3 × GM$_\oplus$ - geogravitational constant (km$^3$/s$^2$)}
  NSAT = 4.0e-3;            {Wheel saturation torque (Nm)}
var
  WW,NS               : double;
```

```
 GG,NGX,NGY       : double;
 NAX,NAY,NAZ      : double;
 WDX,WDY,WDZ      : double;
begin
 TRANSFORM;                                  {Update DCM and Error Quaternion}
 WDX := W0X - W0*AA[1,2];                    {Change in orbit angular rate vector}
 WDY := W0Y - W0*AA[2,2];                    {along the body reference axis}
 WDZ := W0Z - W0*AA[3,2];
 W0X := W0*AA[1,2];  W0Y := W0*AA[2,2];  W0Z := W0*AA[3,2];
 WW := WZI*(IT-IZ);
 GG := GM*(IT-IZ)/(RADIUS*RADIUS*RADIUS);
 NGX := -GG*AA[2,3]*AA[3,3];                 {GG torque components, Eq. 1.8}
 NGY := GG*AA[1,3]*AA[3,3];
                                             {Additional non-linear torques, Eq. 3.20}
 NAX := NGX + WW*WYI - WYI*HZ + WZI*HY - WDX*IT;
 NAY := NGY - WW*WXI + WXI*HZ - WZI*HX - WDY*IT;
 NAZ := WYI*HX - WXI*HY - WDZ*IZ;
 case RWMODE of
0: begin                                     {Pointing control, Eq. 3.14}
    NWX := 2.0*WX + 0.05*QE[1] + NAX;
    NWY := 2.0*WY + 0.05*QE[2] + NAY;
    NWZ := 0.1*WZ + 0.0025*QE[3] + NAZ;
  end;
1: begin                                     {Large angular slew control}
    NS := SLEW;                              {Compute max. slew torque, Eq. 3.26}
    WRX := WRX + NS*KX/IT;                   {Update reference angular rate vector}
    WRY := WRY + NS*KY/IT;
    WRZ := WRZ + NS*KZ/IZ;
    NWX := NAX - NS*KX + 2.0*(WX-WRX);       {Compute total wheel torque, Eq. 3.21}
    NWY := NAY - NS*KY + 2.0*(WY-WRY);       {plus compensation feedback, Eq. 3.29}
    NWZ := NAZ - NS*KZ + 2.0*(WZ-WRZ);
  end
  else begin
    NWX := 0.0;  NWY := 0.0;  NWZ := 0.0;
  end;
 end;
 if abs(NWX) > NSAT then                     {Implement wheel torque saturation}
  if NWX > NSAT then NWX := NSAT
  else NWX:= -NSAT;
 if abs(NWY) > NSAT then
  if NWY > NSAT then NWY := NSAT
  else NWY:= -NSAT;
 if abs(NWZ) > NSAT then
```

```
   if NWZ > NSAT then NWZ := NSAT
   else NWZ:= -NSAT;
end; {RWHEEL}
```

Appendix D. Momentum Dumping

Optimal magnetorquer controllers to do reaction wheel momentum dumping were presented in Paragraph 4.3. Their performance was then compared to the standard "cross-product" algorithm (see Eq. 4.22). All the momentum dumping magnetorquer controllers must be used in combination with a 3-axis nadir stabilization reaction wheel controller. The basic control principle to accomplish momentum dumping, can be described as follows: Disturbances caused by the magnetorquers are compensated for by the reaction wheels in such a way as to dump the undesirable angular momentum from the wheels.

A LQR optimal controller can be obtained by using feedback from the reaction wheel momentum vector. The time-varying feedback gain matrix can be used to compute the optimal magnetorquing vector at each sampling instant. The feedback gain matrix varies due to the changing geomagnetic field vector along the satellite's orbit. A gain matrix look-up table can be computed off-line and stored onboard for easy reference. Indexing can be done by using the sub-satellite latitude and longitude values in general. Procedure DESAT_LQR (listed below) is then all that is required onboard to implement this optimal controller. The Matlab m-file to generate quasi-static LQR gains from a simple dipole geomagnetic field model, is also presented here. Note: A dipole geomagnetic field model assumes changes in the field vector due to the orbital phase (true anomaly) only. Simulations showed that for SUNSAT's polar orbit this can be a used effectively to simplify the look-up table.

A minimum energy desaturation (MEDL) controller can also be used in an open-loop fashion to dump the reaction wheel momentum. The only limitation being that the desaturation effort must be done for a specific part and duration of an orbit. Most of the computations can then be done off-line beforehand to obtain an optimal (minimum energy) constant gain matrix. The Matlab m-file (listed below) show a typical off-line computation when using a dipole geomagnetic field model. In practice an IGRF and satellite orbit model will normally be used for improved accuracy. The procedures START_MEDL and DESAT_MEDL are all that is typically required for an onboard implementation of this controller.

Software Code

Control Law:

Reaction wheel desaturation using magnetorquers:

1. Cross-product law (DESAT_XP)
2. Minimum Energy law (DESAT_MEDL)
3. LQR law (DESAT_LQR)

Inputs:

```
BX, BY, BZ          : double;       {Magnetometer field measurement (μT)}
HX,HY,HZ            : double;       {Rwheel momentum vector (kgm^2.rad/s}
HZOFF               : double;       {Z-axis wheel momentum offset}
W0                  : double;       {Average orbit angular rate (rad/s)}
LAT                 : double;       {Sub-satellite Latitude (rad)}
LON                 : double;       {Sub-satellite Longitude (rad)}
KM  : array[1..3,1..3] of double;   {MEDL constant gain matrix}
KL : array[0..IMAX,0..8] of double; {LQR gain lookup table}
```

Outputs:

```
MTX,MTY,MTZ         : double;       {Magnetorquer on-time fraction of Ts (0-0.8}
MPX,MPY,MPZ         : integer;      {Magnetorquer switching polarity (±1)}
```

Global Variables: (for MEDL only)

```
HX0,HY0,HZ0         : double;       {Initial wheel momentum vector to dump}
MTIME               : integer;      {Desaturation timer}
MFINAL              : integer;      {MEDL desaturation period}
DESAT               : boolean;      {MEDL active flag}
```

Constant:

```
MTQ = 20.0;         {Magnetorquer moment (Am^2)}
```

Pascal Code:

```
procedure MAGTORQ;
begin
 if MTX > 0.01 then MPX := 1                    {X-torquer on-time fraction}
 else if MTX < -0.01 then begin                 {and switching polarity}
  MPX := -1;
  MTX := abs(MTX);
 end;
 if MTY > 0.01 then MPY := 1                    {Y-torquer on-time fraction}
 else if MTY < -0.01 then begin                 {and switching polarity}
  MPY := -1;
  MTY := abs(MTY);
 end;
 if MTZ > 0.01 then MPZ := 1                    {Z-torquer on-time fraction}
 else if MTZ < -0.01 then begin                 {and switching polarity}
  MPZ := -1;
  MTZ .= abs(MTZ);
 end;
end;
```

```
procedure DESAT_XP;                           {Cross-Product Desaturation}
var
 BABS,HZN : double;
begin
 HZN := HZ - HZOFF*W0;                        {Determine Z-wheel offset error}
 BABS := sqrt(BX*BX + BY*BY + BZ*BZ);         {Cross-product law, Eq. 4.22}
 MTX := -80.0*(HY*BZ - HZN*BY)/(BABS*MTQ);
 MTY := -80.0*(HZN*BX - HX*BZ)/(BABS*MTQ);
 MTZ := -80.0*(HX*BY - HY*BX)/(BABS*MTQ);
 MAGTORQ;
end; {DESAT_XP}

procedure START_MEDL;                         {Initialize MEDL controller}
begin                                         {Determine initial wheel momentum}
 HX0 := HX;  HY0 := HY;  HZ0 := HZ - HZOFF*W0;
 MTIME := 0;
 DESAT := true;
end;

procedure DESAT_MEDL;                         {Minimum Energy Desaturation}
const
 TS = 10.0;              {10 second sampling period}
var
 CX,CY,CZ,DX,DY,DZ : double;
 CC,SS,PHASE : double;
begin
 PHASE := W0*MTIME;                           {MEDL law, Eq. 4.21}
 CC := cos(PHASE);  SS := sin(PHASE);
 CX := KM[1,1]*HX0 + KM[1,2]*HY0 + KM[1,3]*HZ0;
 CY := KM[2,1]*HX0 + KM[2,2]*HY0 + KM[2,3]*HZ0;
 CZ := KM[3,1]*HX0 + KM[3,2]*HY0 + KM[3,3]*HZ0;
 DX :=  CC*CX + SS*CZ;
 DY :=  CY;
 DZ := -SS*CX + CC*CZ;
 MTX := (DY*BZ - DZ*BY)/MTQ;
 MTY := (DZ*BX - DX*BZ)/MTQ;
 MTZ := (DX*BY - DY*BX)/MTQ;
 MTIME := MTIME + TS;                         {Test for terminal time}
 if MTIME > MFINAL then begin
  MTX := 0.0; MTY := 0.0; MTZ := 0.0;
  DESAT := false;
  exit;
 end;
```

```
  MAGTORQ;
end; {DESAT_MEDL}

procedure DESAT_LQR;                          {LQR Desaturation}
var
  II     : integer;
  HZN  : double;
begin
  II := LOOKUP(LAT,LON);                      {Obtain index into LQR gain table}
  HZN := HZ-HZOFF*W0;                         {LQR desaturation law, Eq. 4.7}
  MTX := (KL[II,0]*HX + KL[II,1]*HY + KL[II,2]*HZN)/MTQ;
  MTY := (KL[II,3]*HX + KL[II,4]*HY + KL[II,5]*HZN)/MTQ;
  MTZ := (KL[II,6]*HX + KL[II,7]*HY + KL[II,8]*HZN)/MTQ;
  MAGTORQ;
end; {DESAT_LQR}
```

MATLAB Code

MEDL Gain Matrix KM:

```
% Gain matrix computation for Minimum Energy
% Reaction Wheel Desaturation
% Variation of Extremals Method
% WH Steyn 26/3/94

%Initialize
tmax = input('No. of Orbits ? ');
tmax = tmax*6000;                       % Assume 100 minute orbit
dt = 1;                                 % 1 Second integration period
wo = 2*pi/6000;                         % Orbit angular rate
Omega = [0 0 wo;0 0 0;-wo 0 0];         % State matrix

%Simulation loop
t  = dt:dt:tmax;
PH = zeros(3,3);
for i = 1:tmax
  CC = cos(wo*t(i));
  SS = sin(wo*t(i));
  bx = 20.0*CC;                         % Dipole Geomagnetic field model
  by = 5.0;
  bz = 40.0*SS;                         % Time varying state input matrix
  Phi = [0 bz -by;-bz 0 bx;by -bx 0];
  Phi2 = Phi*Phi;                      % Compute state + costate influence matrices, Eq. 4.19
  PP = [CC 0.0 SS;0.0 1.0 0.0 -SS 0.0 CC];
  PH = PH + dt*(Omega*PH + Phi2*PP);
end
KM = inv(PH)*PP*1e6                    % Final MEDL gain matrix, constant part Eq. 4.21

%End
```

LQR Gain Matrix KL:

```
% Quasi-static LQR Gain computation for Reaction Wheel
% Momentum dumping by Magnetorquing
% WH Steyn 24/2/94

%Initialize
tmax = input('No. of Orbits ? ');
tmax = 6000*tmax;                      % Assume 100 minute orbit
smax = tmax/10;                        % 10 Second sampling period
wo = 2*pi/6000;                        % Orbit angular rate
Omega = [0 0 wo;0 0 0;-wo 0 0];        % State matrix
QQ = eye(3);                           % State weighting matrix
RR = 1e8*input('R weight ? ');         % Control weighting matrix
RR = RR*eye(3);
co = zeros(smax,1);                    % Controllability minimum singular value
KK = zeros(3,3);                       % LQR gain matrix
CG = eye(3);                           % Controllability gramian
k1 = zeros(smax,3);                    % 1st row LQR gain matrix trajectory
k2 = zeros(smax,3);                    % 2nd row ...
k3 = zeros(smax,3);                    % 3rd row ...

%Simulation loop
t  = 0:10:tmax-10;
for i = 1:smax
  bx = 20.0*cos(wo*t(i));              % Dipole Geomagnetic field model
  by = 5.0;
  bz = 40.0*sin(wo*t(i));              % Time varying state input matrix
  Phi = [0 bz -by;-bz 0 bx;by -bx 0];
  [AD,BD] = c2d(Omega,Phi,10);         % Get discrete state model
  SV = svd(ctrb(Omega,Phi))';          % Compute controllability singular values
  co(i) = SV(3);                       % Obtain minimum SV
  KK = dlqr(AD,BD,QQ,RR);              % Compute quasi-static discrete LQR gains
  k1(i,:) = KK(1,:)*1e6;
  k2(i,:) = KK(2,:)*1e6;
  k3(i,:) = KK(3,:)*1e6;
  CG = CG*(AD-BD*KK);              % Compute controllability grammian
  if rem(i,10) == 0
    disp(i)
  end
end

%Show Results
plot(t,co);                        % Show min'mum SV variation
title('SV Controllability');
xlabel('Time (sec)');
ylabel('Sigma');
pause                              % Show LQR gains
subplot(221), plot(t,k1(:,1),'-w');
title('k(1,1)');
subplot(222), plot(t,k1(:,2),'-w');
title('k(1,2)');
subplot(223), plot(t,k1(:,3),'-w');
title('k(1,3)');
pause
subplot(111);
subplot(221), plot(t,k2(:,1),'-w');
title('k(2,1)');
subplot(222), plot(t,k2(:,2),'-w');
title('k(2,2)');
subplot(223), plot(t,k2(:,3),'-w');
title('k(2,3)');
pause
subplot(111);
subplot(221), plot(t,k3(:,1),'-w');
title('k(3,1)');
subplot(222), plot(t,k3(:,2),'-w');
title('k(3,2)');
subplot(223), plot(t,k3(:,3),'-w');
title('k(3,3)');
pause
subplot(111);
CGpoles = eig(CG)                  % Eigenvalues of controllability gramian
KL = [k1 k2 k3];                   % Final LQR Gain Matrix trajectory

%End
```

Appendix E. Attitude Determination

E.1 Robust Angular Rate Estimator

This estimator is derived in Paragraph 5.2.2 and can be implemented as shown in procedure KALMAN. The main purpose of the estimator will be to do robust angular rate (orbit referenced) estimation from magnetometer measurements during the pre-boom deployment phase of the satellite's life. The Kalman filter type estimator is called every 10 seconds and the current and previous magnetometer measurement vector are used to obtain the innovation. The filter then updates an orbit referenced angular rate estimation vector to be used by the detumbling (pre-boom deployment) magnetorquer controller. The procedure INIT_KALMAN must be called initially to set up the filter's variables.

Software Code

Rate Estimator:

Kalman filter using rate of change of magnetometer vector

Inputs:

BX,BY,BZ	: double;	*{Current Mag.field measurement (μT)}*
OBX,OBY,OBZ	: double;	*{Previous Mag.field measurement (μT)}*
NMX,NMY,NMZ	: double;	*{Magnetic torque vector (Nm)}*
PMAT : array[1..3,1..3] of double;		*{State covariance matrix}*
QMAT: array[1..3,1..3] of double;		*{System noise covariance matrix}*

Outputs:

EWX,EWY,EWZ	: double;	*{Estim. orbit ref. angular rate vector (rad/s)}*

Constants:

IXYZ = 2.0;	*{X, Y, Z-axis MOI (kgm^2)}*
TS = 10.0;	*{10 second Sampling Period}*

Matrix Toolbox:

KFillM	*{Fill matrix with constant value}*
MAddM	*{Add two matrices A+B}*
MSubM	*{Subtract two matrices A-B}*
MxM	*{Multiply two matrices A*B}*
MxMT	*{Multiply matrix with transposed matrix $A*B^T$}*
MxMxMT	*{$A*B*A^T$}*
InvMatr	*{Matrix inversion A^{-1}}*

Pascal Code:

```
procedure INIT_KALMAN;                      {Initialize Kalman filter}
begin
  EWX := 0.0;  EWY := 0.0;  EWZ := 0.0;
  KFillM(PMAT,0.0,3,3,3);
  PMAT[1,1] := 1e-3;  PMAT[2,2] := 1e-3;  PMAT[3,3] := 1e-3;
  KFillM(QMAT,0.0,3,3,3);
  QMAT[1,1] := 1e-5;  QMAT[2,2] := 1e-5;  QMAT[3,3] := 1e-5;
end;{INIT_KALMAN}

procedure KALMAN;
var
  KMAT,HMAT,DUM1,DUM2 : array[1..3,1..3] of double;
  ABS,OBXN,OBYN,OBZN,BXN,BYN,BZN,VX,VY,VZ : double;
  II  : integer;
begin
  EWX := EWX + TS*NMX/IXYZ;                     {Propagate estimated state, Eq. 5.12a}
  EWY := EWY + TS*NMY/IXYZ;
  EWZ := EWZ + TS*NMZ/IXYZ;
  MAddM(PMAT,PMAT,QMAT,3,3,3,3,3);              {Propagate state covariance, Eq. 5.12b}
                                                {Obtain normalized previous B vector}
  ABS := sqrt(OBX*OBX+OBY*OBY+OBZ*OBZ);
  OBXN := OBX/ABS;  OBYN := OBY/ABS;  OBZN := OBZ/ABS;
                                                {Update meas. matrix H, Eq. 5.11}
  KFillM(HMAT,0.0,3,3,3);
  HMAT[1,2] := -TS*OBZN;  HMAT[1,3] := TS*OBYN;
  HMAT[2,1] :=  TS*OBZN;  HMAT[2,3] := -TS*OBXN;
  HMAT[3,1] := -TS*OBYN;  HMAT[3,2] := TS*OBXN;
                                                {Update gain matrix K, Eq. 5.12c}
  MxMxMT(DUM1,HMAT,PMAT,3,3,3,3,3);
  for II := 1 to 3 do DUM1[II,II] := 1.0 + DUM1[II,II];
  InvMatr(DUM1,3,3,ABS);
  MxMT(DUM2,PMAT,HMAT,3,3,3,3,3,3);
  MxM(KMAT,DUM2,DUM1,3,3,3,3,3,3);
                                                {Obtain normalized current B vector}
  ABS := sqrt(BX*BX+BY*BY+BZ*BZ);
  BXN := BX/ABS;  BYN := BY/ABS;  BZN := BZ/ABS;
                                                {Obtain measurement innovation}
  VX := BXN - OBXN - HMAT[1,2]*EWY - HMAT[1,3]*EWZ;
  VY := BYN - OBYN - HMAT[2,1]*EWX - HMAT[2,3]*EWZ;
  VZ := BZN - OBZN - HMAT[3,1]*EWX - HMAT[3,2]*EWY;
                                                {Update estimated state, Eq. 5.12d}
  EWX := EWX + KMAT[1,1]*VX + KMAT[1,2]*VY + KMAT[1,3]*VZ;
```

```
 EWY := EWY + KMAT[2,1]*VX + KMAT[2,2]*VY + KMAT[2,3]*VZ;
 EWZ := EWZ + KMAT[3,1]*VX + KMAT[3,2]*VY + KMAT[3,3]*VZ;
                                        {Update state covariance, Eq. 5.12e}
 KFillM(DUM1,0.0,3,3,3);
 for II := 1 to 3 do DUM1[II,II] := 1.0;
 MxM(DUM2,KMAT,HMAT,3,3,3,3,3,3);
 MSubM(DUM1,DUM1,DUM2,3,3,3,3,3);
 MxM(PMAT,DUM1,PMAT,3,3,3,3,3,3);
end; {KALMAN}
```

E.2 Full Satellite State Estimators

i. Magnetometer Extended Kalman Filter

This estimator is derived in Paragraph 5.3.2-4 and can be implemented as shown in procedure EKF_MAGNETOMETER. The estimator will be used to do full state determination from magnetometer measurements and the output of a geomagnetic field model. The extended Kalman filter type estimator is called every 10 seconds to update the estimated attitude quaternion, the orbit plus inertial referenced angular rate vectors and aerodynamic disturbance torque magnitude. The procedure INIT_MAG_EKF must be called initially to set up the filter's variables.

Software Code

Full State Estimator:

Extended Kalman filter using magnetometer vector measurements

Inputs:

```
BX,BY,BZ            : double;                {Mag.field magnetometer measurement (μT)}
BX0,BY0,BZ0         : double;                {Mag.field model vector in orbit axes (μT)}
NMX,NMY,NMZ         : double;                {Magnetic torque vector (Nm)}
NWX,NWY,NWZ         : double;                {Reaction Wheel torque vector (Nm)}
PMAT : array[1..8,1..8] of double;           {State covariance matrix}
QMAT : array[1..8,1..8] of double;           {System noise covariance matrix}
RADIUS              : double;                {Satellite distance from geocenter (km)}
TANOM               : double;                {True anomaly of the orbit}
MANOM               : double;                {Mean anomaly of the orbit}
```

Outputs:

```
EQQ   : array[1..4] of double;               {Estim. quaternion}
```

```
        EQE    : array[1..4] of double;         {Estim. error quaternion}
        EWXI,EWYI,EWZI  : double;               {Estim. inertial ref. angular rate vector (rad/s)}
        EWX,EWY,EWZ     : double;               {Estim. orbit ref. angular rate vector (rad/s)}
        EDY0            : double;               {Estim. aerodynamic disturbance torque (Nm)}
```

Constants:

```
        IT = 40.0;              {Transverse MOI (kgm²)}
        IZ = 2.0;               {Z-axis MOI (kgm²)}
        TS = 10.0;              {10 second Sampling Period}
```

Matrix Toolbox:

```
        KFillM          {Fill matrix with constant value}
        CopyM           {Copy matrix B = A}
        MAddM           {Add two matrices A+B}
        MSubM           {Subtract two matrices A-B}
        KxM             {Multiply constant with a matrix k*A}
        MxM             {Multiply two matrices A*B}
        MxMT            {Multiply matrix with transposed matrix A*B^T}
        MxMxMT          {A*B*A^T}
        InvMatr         {Matrix inversion A^-1}
```

Pascal Code:

```
procedure ETRANSFORM;                       {Obtain estimated DCM, Eq. 1.4}
begin
  EAA[1,1] :=  EQQ[1]*EQQ[1] - EQQ[2]*EQQ[2] - EQQ[3]*EQQ[3] + EQQ[4]*EQQ[4];
  EAA[1,2] := 2.0*(EQQ[1]*EQQ[2] + EQQ[3]*EQQ[4]);
  EAA[1,3] := 2.0*(EQQ[1]*EQQ[3] - EQQ[2]*EQQ[4]);
  EAA[2,1] := 2.0*(EQQ[1]*EQQ[2] - EQQ[3]*EQQ[4]);
  EAA[2,2] := -EQQ[1]*EQQ[1] + EQQ[2]*EQQ[2] - EQQ[3]*EQQ[3] + EQQ[4]*EQQ[4];
  EAA[2,3] := 2.0*(EQQ[2]*EQQ[3] + EQQ[1]*EQQ[4]);
  EAA[3,1] := 2.0*(EQQ[1]*EQQ[3] + EQQ[2]*EQQ[4]);
  EAA[3,2] := 2.0*(EQQ[2]*EQQ[3] - EQQ[1]*EQQ[4]);
  EAA[3,3] := -EQQ[1]*EQQ[1] - EQQ[2]*EQQ[2] + EQQ[3]*EQQ[3] + EQQ[4]*EQQ[4];
end;{ETRANSFORM}

procedure INIT_MAG_EKF;                     {Initialize Magnetometer EKF}
  EQQ[1] := 0.0;  EQQ[2] := 0.0;  EQQ[3] := 0.0;  EQQ[4] := 1.0;
  ETRANSFORM;
  EQE[1] := 0.0;  EQE[2] := 0.0;  EQE[3] := 0.0;  EQE[4] := 1.0;
  EWXI := EWX;  EWYI := EWY;  EWZI := EWZ;          {From Rate Estimator}
  EDY0 := 0.0;
  KFillM(PMAT,0.0,8,8,8);
  PMAT[1,1] := 1e-2;  PMAT[2,2] := 1e-2;  PMAT[3,3] := 1e-2;
  PMAT[4,4] := 1e4;  PMAT[5,5] := 1e4;  PMAT[6,6] := 1e4;  PMAT[7,7] := 1e4;
  PMAT[8,8] := 5e-6;
  KFillM(QMAT,0.0,8,8,8);
  QMAT[1,1] := 2e-5;  QMAT[2,2] := 2e-5;  QMAT[3,3] := 2e-5;
  QMAT[4,4] := 2e-1;  QMAT[5,5] := 2e-1;  QMAT[6,6] .= 2e-1;  QMAT[7,7] := 2e-1;
  QMAT[8,8] := 5e-9;
end;{INIT_MAG_EKF}

procedure MODEL;                              {Propagate full satellite state, Eq. 5.43a}
const
  GM = 1.195801e6;           {3 × GM⊕ - geogravitational constant ($km^3/s^2$}
var
  WW,RX,RY,RZ,EDX,EDY                         : double;
  SS,CC,DWX,DWY,DWZ,NGX,NGY                   : double;

  procedure EQTERNIONS;                       {Propagate the estimated quaternion, Eq. 5.77}
  var
    WW,CC,SS : double;
    TQQ      : array[1..4] of double;
  begin
    WW := sqrt(EWX*EWX + EWY*EWY + EWZ*EWZ);
    CC := cos(0.5*WW*TS);
    if WW = 0.0 then SS := 1.0
    else SS := sin(0.5*WW*TS)/WW;
    TQQ[1] := CC*EQQ[1] + EWZ*SS*EQQ[2] - EWY*SS*EQQ[3] + EWX*SS*EQQ[4];
    TQQ[2] :=-EWZ*SS*EQQ[1] + CC*EQQ[2] + EWX*SS*EQQ[3] + EWY*SS*EQQ[4];
    TQQ[3] := EWY*SS*EQQ[1] - EWX*SS*EQQ[2] + CC*EQQ[3] + EWZ*SS*EQQ[4];
    TQQ[4] :=-EWX*SS*EQQ[1] - EWY*SS*EQQ[2] - EWZ*SS*EQQ[3] + CC*EQQ[4];
    EQQ[1] := TQQ[1]; EQQ[2] := TQQ[2]; EQQ[3] := TQQ[3]; EQQ[4] := TQQ[4];
{Compute the error quaternion, Eq. 1.15}
    EQE[1] := QC[4]*EQQ[1] + QC[3]*EQQ[2] - QC[2]*EQQ[3] - QC[1]*EQQ[4];
    EQE[2] :=-QC[3]*EQQ[1] + QC[4]*EQQ[2] + QC[1]*EQQ[3] - QC[2]*EQQ[4];
    EQE[3] := QC[2]*EQQ[1] - QC[1]*EQQ[2] + QC[4]*EQQ[3] - QC[3]*EQQ[4];
    EQE[4] := QC[1]*EQQ[1] + QC[2]*EQQ[2] + QC[3]*EQQ[3] + QC[4]*EQQ[4];
    ETRANSFORM;
  end;{EQTERNIONS}

  procedure CONVERT(var XX,YY : double; AA : double);
  var MAG,ARG : double;                       {Rotate complex variable, Eq. 5.79c,d,h}
  begin
    if XX = 0.0 then begin
      if YY = 0.0 then exit;
      if YY > 0.0 then ARG := pi/2 else ARG := -pi/2;
```

```
   end
   else if XX < 0.0 then ARG := pi + arctan(YY/XX)
   else ARG := arctan(YY/XX);
   MAG := sqrt(XX*XX + YY*YY);
   XX := MAG*cos(ARG - AA);
   YY := MAG*sin(ARG - AA);
 end;{CONVERT}

{Main part of MODEL}
begin
{Propagate the aerodynamic disturbance torque, Eq. 5.17}
 EDX := EAA[1,2]*EDY0*(1.0 + cos(TANOM));
 EDY := EAA[2,2]*EDY0*(1.0 + cos(TANOM));
{Compute the gravitational torque in an elliptical orbit, Eq. 5.14}
 SS := -sin(TANOM - MANOM);  CC := cos(TANOM - MANOM);
 RX := EAA[1,1]*SS + EAA[1,3]*CC;
 RY := EAA[2,1]*SS + EAA[2,3]*CC;
 RZ := EAA[3,1]*SS + EAA[3,3]*CC;
 GG := GM*(IT - IZ)/(RADIUS*RADIUS*RADIUS);
 NGX := -GG*RY*RZ;  NGY := GG*RX*RZ;
{Propagate the angular rate vector}
 WW := EWZI*(1.0 - IZ/IT);                                          {Eq.5.79a}
 DWX := TS*(EDX + NGX - NWX + NMX - EWYI*HZ + EWZI*HY)/IT;          {Eq.5.79b}
 DWY := TS*(EDY + NGY - NWY + NMY + EWXI*HZ - EWZI*HX)/IT;
 DWZ := TS*(NMZ - NWZ - EWXI*HY + EWYI*HX)/IZ;
 CONVERT(DWX,DWY,0.5*WW*TS);                                        {Eq.5.79c}
 CONVERT(EWXI,EWYI,WW*TS);                                          {Eq.5.79d}
 EWXI := EWXI + DWX;                                                {Eq.5.79e}
 EWYI := EWYI + DWY;
 EWZI := EWZI + DWZ;
 DWX := W0*EAA[1,2];                                                {Eq.5.79f}
 DWY := W0*EAA[2,2];
 EWZ := EWZI + W0*EAA[3,2];                                         {Eq.5.79g}
 CONVERT(DWX,DWY,0.8*EWZ*TS);                                       {Eq.5.79h}
 EWX := EWXI + DWX;  EWY := EWYI + DWY;                             {Eq.5.79i}
 EQTERNIONS;
end;{MODEL}

procedure PROPAGATE;             {Propagate perturbation cov. matrix, Eq. 5.43b}
var
 KI,KA,KB,KC,KD   : double;
 II               : integer;
 DUM              : array[1..8,1..8] of double;
```

```
begin
{Compute the continuous pertubation system matrix F = FMAT, Eqs. 5.24-5.30}
 KFillM(FMAT,0.0,8,8,8);
 KI := 1.0 - IZ/IT;  KD := 2.0*EDY0/IT;
 KA := 2.0*GG*EAA[1,3]/IT;  KB := 2.0*GG*EAA[2,3]/IT;  KC := 2.0*GG*EAA[3,3]/IT;
 FMAT[1,2] :=  EWZI*KI;  FMAT[1,3] :=  EWYI*KI;
 FMAT[2,1] := -EWZI*KI;  FMAT[2,3] := -EWXI*KI;
 FMAT[1,4] := -EQQ[4]*KC + EQQ[1]*KB + EQQ[2]*KD;
 FMAT[1,5] := -EQQ[3]*KC + EQQ[2]*KB + EQQ[1]*KD;
 FMAT[1,6] := -EQQ[2]*KC - EQQ[3]*KB + EQQ[4]*KD,
 FMAT[1,7] := -EQQ[1]*KC - EQQ[4]*KB + EQQ[3]*KD;
 FMAT[2,4] :=  EQQ[3]*KC - EQQ[1]*KA - EQQ[1]*KD;
 FMAT[2,5] := -EQQ[4]*KC - EQQ[2]*KA + EQQ[2]*KD;
 FMAT[2,6] :=  EQQ[1]*KC + EQQ[3]*KA - EQQ[3]*KD;
 FMAT[2,7] := -EQQ[2]*KC + EQQ[4]*KA + EQQ[4]*KD;
 FMAT[1,8] :=  EAA[1,2]/IT;
 FMAT[2,8] :=  EAA[2,2]/IT;
 FMAT[4,1] :=  0.5*EQQ[4];  FMAT[4.2] := -0.5*EQQ[3];  FMAT[4,3] :=  0.5*EQQ[2];
 FMAT[5,1] :=  0.5*EQQ[3];  FMAT[5,2] :=  0.5*EQQ[4];  FMAT[5,3] := -0.5*EQQ[1];
 FMAT[6,1] := -0.5*EQQ[2];  FMAT[6,2] :=  0.5*EQQ[1];  FMAT[6,3] :=  0.5*EQQ[4];
 FMAT[7,1] := -0.5*EQQ[1];  FMAT[7,2] := -0.5*EQQ[2];  FMAT[7,3] := -0.5*EQQ[3];
 FMAT[4,5] :=  0.5*EWZ;  FMAT[4,6] := -0.5*EWY;  FMAT[4,7] :=  0.5*EWX;
 FMAT[5,4] := -0.5*EWZ;  FMAT[5,6] :=  0.5*EWX;  FMAT[5,7] :=  0.5*EWY;
 FMAT[6,4] :=  0.5*EWY;  FMAT[6,5] := -0.5*EWX;  FMAT[6,7] :=  0.5*EWZ;
 FMAT[7,4] := -0.5*EWX;  FMAT[7,5] := -0.5*EWY;  FMAT[7,6] := -0.5*EWZ;
{Compute the discrete perturbation system matrix Φ = FMAT, Eq. 5.31}
 KxM(FMAT,TS,FMAT,8,8,8,8);
 MxM(DUM,FMAT,FMAT,8,8,8.8,8,8);
 KxM(DUM,0.5,DUM,8,8,8,8);
 MAddM(FMAT,FMAT,DUM,8,8,8,8,8);
 for II := 1 to 8 do FMAT[II,II] := 1.0 + FMAT[II,II];
{Compute the perturbation covariance matrix P = PMAT, Eq.5.43b}
 MxMxMT(DUM,FMAT,PMAT,8,8,8,8,8);
 CopyM(PMAT,DUM,8,8,8,8);
 MAddM(PMAT,PMAT,QMAT,8,8,8,8,8);
end;{PROPAGATE}

procedure UPDATE_MAG;                    {Update the estim. state vector and cov. matrix}
var
 DUM,DUM1          : array[1..8,1..8] of double;
 DUM2              : array[1..3,1..3] of double;
 HMAT              : array[1..3,1..8] of double;
 DUM3,KMAT         : array[1..8,1..3] of double;
 II                           : integer;
 BX0N,BY0N,BZ0N,ABS           : double;
 VVX,VVY,VVZ                  : double;

 procedure OBTAINH;                          {Compute measurement matrix, Eqs. 5.40,5.41}
 begin
  KFillM(HMAT,0.0,3,8,8);
  HMAT[1,4] :=  EQQ[1]*BX0N + EQQ[2]*BY0N + EQQ[3]*BZ0N;
  HMAT[2,4] :=  EQQ[2]*BX0N - EQQ[1]*BY0N + EQQ[4]*BZ0N;
  HMAT[3,4] :=  EQQ[3]*BX0N - EQQ[4]*BY0N - EQQ[1]*BZ0N;
  HMAT[1,5] := -EQQ[2]*BX0N + EQQ[1]*BY0N - EQQ[4]*BZ0N;
  HMAT[2,5] :=  EQQ[1]*BX0N + EQQ[2]*BY0N + EQQ[3]*BZ0N;
  HMAT[3,5] :=  EQQ[4]*BX0N + EQQ[3]*BY0N - EQQ[2]*BZ0N;
  HMAT[1,6] := -EQQ[3]*BX0N + EQQ[4]*BY0N + EQQ[1]*BZ0N;
  HMAT[2,6] := -EQQ[4]*BX0N - EQQ[3]*BY0N + EQQ[2]*BZ0N;
  HMAT[3,6] :=  EQQ[1]*BX0N + EQQ[2]*BY0N + EQQ[3]*BZ0N;
  HMAT[1,7] :=  EQQ[4]*BX0N + EQQ[3]*BY0N - EQQ[2]*BZ0N;
  HMAT[2,7] := -EQQ[3]*BX0N + EQQ[4]*BY0N + EQQ[1]*BZ0N;
  HMAT[3,7] :=  EQQ[2]*BX0N - EQQ[1]*BY0N + EQQ[4]*BZ0N;
  KxM(HMAT,2.0,HMAT,3,8,8,8);
 end;{OBTAINH}

 procedure INNOVATION;                       {Obtain the innovation}
 var
  BXN,BYN,BZN,EBX,EBY,EBZ,ABS : double;
 begin
  ABS := sqrt(BX*BX + BY*BY + BZ*BZ);                                    {Eq.5.57}
  BXN := BX/ABS;  BYN := BY/ABS;  BZN := BZ/ABS;
  EBX := BX0N*EAA[1,1] + BY0N*EAA[1,2] + BZ0N*EAA[1,3];                  {Eq.5.39}
  EBY := BX0N*EAA[2,1] + BY0N*EAA[2,2] + BZ0N*EAA[2,3];
  EBZ := BX0N*EAA[3,1] + BY0N*EAA[3,2] + BZ0N*EAA[3,3];
  VVX := BXN - EBX;
  VVY := BYN - EBY;
  VVZ := BZN - EBZ;
 end;{INNOVATION}

{Main part of UPDATE_MAG}
begin
{Gain Update}
 ABS := sqrt(BX0*BX0 + BY0*BY0 + BZ0*BZ0);                               {Eq.5.58}
 BX0N := BX0/ABS;  BY0N := BY0/ABS;  BZ0N := BZ0/ABS;
 OBTAINH;                                                                {Eq.5.43c}
 MxMxMT(DUM2,HMAT,PMAT,3,8,3,8,8);
 for II := 1 to 3 do DUM2[II,II] := RMAT + DUM2[II,II];
 InvMatr(DUM2,3,3,ABS);
 MxMT(DUM3,PMAT,HMAT,8,8,3,3,8,8);
 MxM(KMAT,DUM3,DUM2,8,3,3,3,3,3);
 INNOVATION;                                                  {Eq.5.43d}
{State Update}                                                {Eq.5.43f}
 EWXI := EWXI + KMAT[1,1]*VVX + KMAT[1,2]*VVY + KMAT[1,3]*VVZ;
 EWYI := EWYI + KMAT[2,1]*VVX + KMAT[2,2]*VVY + KMAT[2,3]*VVZ;
 EWZI := EWZI + KMAT[3,1]*VVX + KMAT[3,2]*VVY + KMAT[3,3]*VVZ;
 EQQ[1] := EQQ[1] + KMAT[4,1]*VVX + KMAT[4,2]*VVY + KMAT[4,3]*VVZ;
 EQQ[2] := EQQ[2] + KMAT[5,1]*VVX + KMAT[5,2]*VVY + KMAT[5,3]*VVZ;
 EQQ[3] := EQQ[3] + KMAT[6,1]*VVX + KMAT[6,2]*VVY + KMAT[6,3]*VVZ;
 EQQ[4] := EQQ[4] + KMAT[7,1]*VVX + KMAT[7,2]*VVY + KMAT[7,3]*VVZ;
{Normalize quaternion, Eq. 5.44}
 ABS := sqrt(EQQ[1]*EQQ[1] + EQQ[2]*EQQ[2] + EQQ[3]*EQQ[3] + EQQ[4]*EQQ[4]);
 EQQ[1] := EQQ[1]/ABS;  EQQ[2] := EQQ[2]/ABS;
 EQQ[3] := EQQ[3]/ABS;  EQQ[4] := EQQ[4]/ABS;
 EDY0 := EDY0 + KMAT[8,1]*VVX + KMAT[8,2]*VVY + KMAT[8,3]*VVZ;
{Covariance matrix Update}
 OBTAINH;                                                     {Eq.5.43g}
 KFillM(DUM,0.0,8,8,8);
 for II := 1 to 8 do DUM[II,II] := 1.0 + DUM[II,II];
 MxM(DUM1,KMAT,HMAT,8,3,8,8,3,8);
 MSubM(DUM,DUM,DUM1,8,8,8,8,8);
 MxMxMT(DUM1,DUM,PMAT,8,8,8,8,8);
 MxMT(DUM,KMAT,KMAT,8,3,8,8,3,3);
 KxM(DUM,RMAT,DUM,8,8,8,8);
 MAddM(PMAT,DUM1,DUM,8,8,8,8,8);
end;{UPDATE_MAG}

procedure EKF_MAGNETOMETER;       {Extended Kalman Filter - Magnetometer}
begin
 MODEL;
 PROPAGATE;
 UPDATE_MAG;
end;{EKF_MAGNETOMETER}
```

ii) Sun/Horizon Sensor EKF

This estimator is derived in Paragraphs 5.3.2-4 and can be implemented as shown in procedure EKF_SUNHORIZON. The estimator will be used to do full state determination from a sun sensor, two orthogonal looking horizon sensors and the outputs of both sun and satellite orbit models. The extended Kalman filter type estimator is called every 1 second to update the estimated attitude quaternion, the orbit plus inertial referenced angular rate vectors and aerodynamic disturbance torque magnitude. The procedure INIT_SH_EKF must be called initially to set up the filter's variables. Two of the procedures called from the main procedure EKF_SUNHORIZON are identical to those of the magnetometer EKF (i.e. MODEL and PROPAGATE), the changed procedure UPDATE_SH is the only one listed below.

Software Code

Full State Estimator:

Extended Kalman filter using sun and horizon sensor measurements

Inputs:

Same as for magnetometer EKF plus:

LAT	: double;	*{Sub-satellite latitude (rad)}*
X0EAST	: double;	*{Angle between X_o-axis and East, see Fig. A.1}*
YAW	: double;	*{Satellite yaw angle ψ, see Eq. 1.5 (rad)}*
BETA	: double;	*{Sun angle from local zenith (rad)}*
OMEGA	: double;	*{Sun azimuth angle within the X_oY_o-plane (rad)}*
SUNOK	: boolean;	*{Flag to indicate a valid sun measurement}*
SUN_TETA	: double;	*{Sun sensor angular measurement (± 60°)}*
HORX,HORY	: boolean;	*{Flag to indicate a valid X or Y-horizon meas.}*
HX_TETA,HY_TETA	: double;	*{X or Y-Horizon sensor angular meas. (± 15°)}*

Outputs:

Same as for the magnetometer EKF

Constants:

Same as for the magnetometer EKF but,

TS = 1.0; *{1 second Sampling Period}*

Matrix Toolbox:

Same as for the magnetometer EKF

Pascal Code:

```
procedure INIT_SH_EKF;                    {Initialize Sun/Horizon sensor EKF}
begin
{Initialized state variables from magnetometer EKF}
  KFillM(PMAT,0.0,8,8,8);
  PMAT[1,1] := 1e-2;  PMAT[2,2] := 1e-2;  PMAT[3,3] := 1e-2;
  PMAT[4,4] := 2e4;  PMAT[5,5] := 2e4;  PMAT[6,6] := 2e4;  PMAT[7,7] := 2e4;
  PMAT[8,8] := 5e-6;
  KFillM(QMAT,0.0,8,8,8);
  QMAT[1,1] := 2e-5;  QMAT[2,2] := 2e-5;  QMAT[3,3] := 2e-5;
  QMAT[4,4] := 5e-1;  QMAT[5,5] := 5e-1;  QMAT[6,6] := 5e-1;  QMAT[7,7] := 5e-1;
  QMAT[8,8] := 5e-9;
end;{INIT_SH_EKF}

procedure UPDATE_SH;                    {Update the estim. state vector and cov. matrix}
var
  XME,YME,ZME,XMO,YMO,ZMO,VVX,VVY,VVZ : double;
  HMAT : array[1..3,1..8] of double;

  procedure OBTAINH;                    {Compute measurement matrix, Eqs. 5.40,5.41}
  begin
    KFillM(HMAT,0.0,3,8,8);
    HMAT[2,4] :=  EQQ[2]*XMO - EQQ[1]*YMO + EQQ[4]*ZMO;
    HMAT[2,5] :=  EQQ[1]*XMO + EQQ[2]*YMO + EQQ[3]*ZMO;
    HMAT[2,6] := -EQQ[4]*XMO - EQQ[3]*YMO + EQQ[2]*ZMO;
    HMAT[2,7] := -EQQ[3]*XMO + EQQ[4]*YMO + EQQ[1]*ZMO;
    HMAT[1,4] :=  EQQ[1]*XMO + EQQ[2]*YMO + EQQ[3]*ZMO;
    HMAT[1,5] := -EQQ[2]*XMO + EQQ[1]*YMO - EQQ[4]*ZMO;
    HMAT[1,6] := -EQQ[3]*XMO + EQQ[4]*YMO + EQQ[1]*ZMO;
    HMAT[1,7] :=  EQQ[4]*XMO + EQQ[3]*YMO - EQQ[2]*ZMO;
    HMAT[3,4] :=  EQQ[3]*XMO - EQQ[4]*YMO - EQQ[1]*ZMO;
    HMAT[3,5] :=  EQQ[4]*XMO + EQQ[3]*YMO - EQQ[2]*ZMO;
    HMAT[3,6] :=  EQQ[1]*XMO + EQQ[2]*YMO + EQQ[3]*ZMO;
    HMAT[3,7] :=  EQQ[2]*XMO - EQQ[1]*YMO + EQQ[4]*ZMO;
    KxM(HMAT,2.0,HMAT,3,8,8,8);
  end;{OBTAINH}

  procedure INNOVATION;                 {Obtain the innovation, Eq. 5.39}
  var
    EXM,EYM,EZM : double;
  begin
    EXM := XMO*EAA[1,1] + YMO*EAA[1,2] + ZMO*EAA[1,3];
    EYM := XMO*EAA[2,1] + YMO*EAA[2,2] + ZMO*EAA[2,3];
    EZM := XMO*EAA[3,1] + YMO*EAA[3,2] + ZMO*EAA[3,3];
    VVX := XME - EXM;
    VVY := YME - EYM;
    VVZ := ZME - EZM;
  end;{INNOVATION}
```

```
procedure UPDATE;                         {Update the EKF}
var
  DUM2             : array[1..3,1..3] of double;
  DUM3,KMAT        : array[1..8,1..3] of double;
  DUM,DUM1         : array[1..8,1..8] of double;
  II         : integer;
  ABS        : double;
begin
{Gain Update}
  OBTAINH;                                                          {Eq.5.43c}
  MxMxMT(DUM2,HMAT,PMAT,3,8,3,8,8);
  for II := 1 to 3 do DUM2[II,II] := RMAT + DUM2[II,II];
  InvMatr(DUM2,3,3,ABS);
  MxMT(DUM3,PMAT,HMAT,8,8,3,3,8,8);
  MxM(KMAT,DUM3,DUM2,8,3,3,3,3,3);
  INNOVATION;                                                       {Eq.5.43d}
{State Update}                                                      {Eq.5.43f}
  EWXI := EWXI + KMAT[1,1]*VVX + KMAT[1,2]*VVY + KMAT[1,3]*VVZ;
  EWYI := EWYI + KMAT[2,1]*VVX + KMAT[2,2]*VVY + KMAT[2,2]*VVZ;
  EWZI := EWZI + KMAT[3,1]*VVX + KMAT[3,2]*VVY + KMAT[3,3]*VVZ;
  EQQ[1] := EQQ[1] + KMAT[4,1]*VVX + KMAT[4,2]*VVY + KMAT[4,3]*VVZ;
  EQQ[2] := EQQ[2] + KMAT[5,1]*VVX + KMAT[5,2]*VVY + KMAT[5,3]*VVZ;
  EQQ[3] := EQQ[3] + KMAT[6,1]*VVX + KMAT[6,2]*VVY + KMAT[6,3]*VVZ;
  EQQ[4] := EQQ[4] + KMAT[7,1]*VVX + KMAT[7,2]*VVY + KMAT[7,3]*VVZ;
{Normalize quaternion, Eq. 5.44}
  ABS := sqrt(EQQ[1]*EQQ[1]+EQQ[2]*EQQ[2]+EQQ[3]*EQQ[3]+EQQ[4]*EQQ[4]);
  EQQ[1] := EQQ[1]/ABS;  EQQ[2] := EQQ[2]/ABS;
  EQQ[3] := EQQ[3]/ABS;  EQQ[4] := EQQ[4]/ABS;
  EDY0 := EDY0 + KMAT[8,1]*VVX + KMAT[8,2]*VVY + KMAT[8,3]*VVZ;
{Covariance matrix Update}
  OBTAINH;                                                          {Eq.5.43g}
  KFillM(DUM,0.0,8,8,8);
  for II := 1 to 8 do DUM[II,II] := 1.0 + DUM[II,II];
  MxM(DUM1,KMAT,HMAT,8,3,8,8,3,8);
  MSubM(DUM,DUM,DUM1,8,8,8,8,8);
  MxMxMT(DUM1,DUM,PMAT,8,8,8,8,8);
  MxMT(DUM,KMAT,KMAT,8,3,8,8,3,3);
  KxM(DUM,RMAT,DUM,8,8,8,8);
  MAddM(PMAT,DUM1,DUM,8,8,8,8,8);
end;{UPDATE}
```

```
function GET_ELEV(YAW:double):double;          {Compute the elevation to the horizon}
const
  f = 0.00335281;             {Flattening factor of the earth}
 a2 = 40680669.86;            {Square of the equatorial radius (km^2)}
var
 DUM1,DUM2,R2,D2,PSI,RHO : double;
begin
 DUM1 := (2 - f)*f*sqr(cos(LAT));                                   {Eq.5.64}
 DUM2 := a2*sqr(1 - f);
 R2 := DUM2/(1 - DUM1);
 D2 := RADIUS*RADIUS;                                               {Eq.5.63}
 PSI := sin(X0EAST - YAW);
 RHO := sqrt(((D2 - R2)/a2)*(1 + DUM1*R2*PSI*PSI/DUM2))
             + (2 - f)*f*R2*sin(2*LAT)*PSI/(2*DUM2);
 GET_ELEV := 0.5*pi-arctan(1/RHO);
end;{GET_ELEV}

procedure XHOR_UPDATE;                 {Do the X-horizon sensor update}
const
  HORIZON = 26.85;          {Average horizon angle (°) for the SUNSAT orbit}
var
 TETA,AZIM,ELEV,ABS : double;
begin
 TETA := (HORIZON+HX_TETA)/DEG;                                     {Eq.5.59}
 XME := cos(TETA);  YME := 0.0;  ZME := sin(TETA);
 XMO := XME*EAA[1,1] + ZME*EAA[3,1];                                {Eq.5.61}
 YMO := XME*EAA[1,2] + ZME*EAA[3,2];
 if XMO = 0.0 then XMO := 1e-6;
 AZIM := arctan(YMO/XMO);                                           {Eq.5.60}
 if XMO < 0.0 then
   if YMO > 0.0 then AZIM := AZIM + pi
   else AZIM := AZIM - pi;
 ELEV := GET_ELEV(AZIM);                                            {Eq.5.62}
 ZMO := sin(ELEV);  XMO := cos(ELEV)*cos(AZIM);  YMO := cos(ELEV)*sin(AZIM);
 UPDATE;
end;{XHOR_UPDATE}

procedure YHOR_UPDATE;                 {Do the Y-horizon sensor update}
const
  HORIZON = 26.85;          {Average horizon angle (°) for the SUNSAT orbit}
var
 TETA,AZIM,ELEV,ABS : double;
```

```
  begin
    TETA := (HORIZON+HY_TETA)/DEG;                                    {Eq.5.65}
    XME := 0.0; YME := -cos(TETA); ZME := sin(TETA);
    XMO := YME*EAA[2,1] + ZME*EAA[3,1];
    YMO := YME*EAA[2,2] + ZME*EAA[3,2];
    if XMO = 0.0 then XMO := 1e-6;
    AZIM := arctan(YMO/XMO);
    if XMO < 0.0 then
      if YMO > 0.0 then AZIM := AZIM + pi
      else AZIM := AZIM - pi;
    ELEV := GET_ELEV(AZIM);
    ZMO := sin(ELEV);  XMO := cos(ELEV)*cos(AZIM);  YMO := cos(ELEV)*sin(AZIM);
    UPDATE;
  end;{YHOR_UPDATE}

  procedure SUN_UPDATE;                    {Do the sun sensor update}
  var
    TETA,AZIM,ABS : double;
  begin
    XMO := -sin(BETA)*cos(OMEGA);                                     {Eq.5.66}
    YMO := -sin(BETA)*sin(OMEGA);
    ZMO := -cos(BETA);
    ZME := XMO*EAA[3,1] + YMO*EAA[3,2] + ZMO*EAA[3,3];               {Eq.5.69}
    AZIM := (SUN_TETA - 90.0)/DEG;                                    {Eq.5.68}
    ABS := sqrt(1.0 - ZME*ZME);                                       {Eq.5.67}
    XME := ABS*cos(AZIM);  YME := ABS*sin(AZIM);
    UPDATE;
  end;{SUN_UPDATE}

{Main part UPDATE_SH}
begin
  if SUNOK then SUN_UPDATE;
  if HORX then XHOR_UPDATE;
  if HORY then YHOR_UPDATE;
end;{UPDATE_SH}


procedure EKF_SUNHORIZON;
begin
  EMODEL;
  PREDICTION;
  UPDATE_SH;
end;{EKF_SUNHORIZON}
```

Appendix F. Magnetometer Calibration

F.1 LMS Algorithm

This magnetometer calibration method is presented in Paragraph 6.2. It can be used during the initial phase of a space mission when the satellite's attitude has not been determined accurately yet. The LMS algorithm only requires the modelled magnitude of the geomagnetic field and the raw magnetometer measurement vector during each orbit at 10 second sampling intervals. The procedure INIT_LMS is called initially to set up the default scaling and bias calibration parameters. Procedure LMSMAGCAL is then called every 10 seconds to update these calibration parameters and to compute the calibrated magnetometer vector.

Software Code

Calibration Method:

Least Mean Square algorithm

Inputs:

```
      BXM,BYM,BZM       : double;        {Measured magnetometer vector (μT)}
      BX0,BY0,BZ0       : double;        {IGRF orbit referenced vector (μT)}
      LOFFS : array[1..3] of double;     {LMS bias calibration vector (μT)}
      LGAIN: array[1..3] of double;      {LMS scale factor calibration vector}
```

Outputs:

```
      BXC,BYC,BZC       : double;        {Calibrated magnetometer vector (μT)}
```

Pascal Code:

```
procedure INIT_LMS;                      {Initialize LMS parameters}
var
  I : integer;
begin
  for I := 1 to 3 do begin
    LOFFS[I] := 0.0;
    LGAIN[I] := 1.0;
  end;
end; {INIT_LMS}

procedure LMSMAGCAL;
var
  DUM,DUMA,ERR,STEPA,STEPB,DUMX,DUMY,DUMZ : double;
```

```
begin
  DUMX := LGAIN[1]*BXM + LOFFS[1];                              {Eq. 6.3}
  DUMY := LGAIN[2]*BYM + LOFFS[2];
  DUMZ := LGAIN[3]*BZM + LOFFS[3];
  DUMA := DUMX*DUMX + DUMY*DUMY + DUMZ*DUMZ;                    {Eq. 6.4}
  ERR := sqrt(BX0*BX0+BY0*BY0+BZ0*BZ0) - sqrt(DUMA);
  ERR := ERR/(1.0 + 0.1*abs(ERR));                              {Eq. 6.15}
  DUM := -2.0*ERR*DUMA;
  ERR := sqr(ERR);                                              {Eq. 6.8}
  STEPA := 1e-9*(1.0 - exp(-ERR));
  STEPB := 1e-6*(1.0 - exp(-ERR));
  LGAIN[1] := LGAIN[1] - STEPA*DUM*DUMX*BXM;                    {Eq. 6.6}
  LGAIN[2] := LGAIN[2] - STEPA*DUM*DUMY*BYM;
  LGAIN[3] := LGAIN[3] - STEPA*DUM*DUMZ*BZM;
  LOFFS[1] := LOFFS[1] - STEPB*DUM*DUMX;
  LOFFS[2] := LOFFS[2] - STEPB*DUM*DUMY;
  LOFFS[3] := LOFFS[3] - STEPB*DUM*DUMZ;
  BXC := LGAIN[1]*BXM + LOFFS[1];                               {Eq. 6.3}
  BYC := LGAIN[2]*BYM + LOFFS[2];
  BZC := LGAIN[3]*BZM + LOFFS[3];
end; {LMSMAGCAL}
```

F.2 RLS Algorithm

This new magnetometer calibration method is presented in Paragraph 6.3. It can be used when accurate attitude knowledge of the satellite exists. The RLS algorithm requires the modelled geomagnetic field vector in body coordinates and the raw magnetometer measurement vector during each orbit at 10 second sampling intervals. The procedure INIT_RLS is called initially to set up the default scaling and bias calibration parameters. Procedure RLSMAGCAL is then called every 10 seconds to update these calibration parameters and to compute the calibrated magnetometer vector.

Software Code

Calibration Method:

Recursive Least Squares algorithm

Inputs:

```
BXM,BYM,BZM     : double;     {Measured magnetometer vector (μT)}
BX0,BY0,BZ0     : double;     {IGRF orbit referenced vector (μT)}
```

```
      AA : array[1..3,1..3] of double;        {DCM to transform from orbit to body axes}
      ROFFS : array[1..3] of double;          {RLS bias calibration vector (μT)}
      RGAIN: array[1..3,1..3] of double;      {RLS scale factor calibration matrix}
      P1,P2,P3 : array[1..4,1..4] of double; {Parameter covariance matrices}
```

Outputs:

```
      BXC,BYC,BZC        : double;            {Calibrated magnetometer vector (μT)}
```

Pascal Code:

```
procedure INIT_RLS;                          {Initialize the RLS parameters}
var
 I,J    : integer;
begin
 for I := 1 to 4 do begin
  for J := 1 to 4 do begin
   P1[I,J] := 0.0;  P2[I,J] := 0.0;  P3[I,J] := 0.0;
  end;
  P1[I,I] := 10.0;  P2[I,I] := 10.0;  P3[I,I] := 10.0;
 end;
 for I := 1 to 3 do begin
  ROFFS[I]:=0.0;
  for J := 1 to 3 do RGAIN[I,J] := 0.0;
 end;
 for J := 1 to 3 do RGAIN[J,J] := 1.0;
end; {INIT_RLS}

procedure RLSMAGCAL;
var
 YY : array[1..3] of double;

 procedure UPDATE(var PP:array[1..4,1..4] of double; I:integer);
 const
  LAMBDA = 0.99;          {Forgetting factor}
 var
  ERR,DEN  : double;
  WW       : array[1..4] of double;
 begin                                                                   {Eq. 6.10}
  ERR := YY[I] - BXM*RGAIN[I,1] - BYM*RGAIN[I,2] - BZM*RGAIN[I,3] - ROFFS[I];
  ERR := ERR/(1.0 + 0.1*abs(ERR));                                       {Eq. 6.15}
  WW[1] := PP[1,1]*BXM + PP[1,2]*BYM + PP[1,3]*BZM + PP[1,4];            {Eq. 6.12}
  WW[2] := PP[2,1]*BXM + PP[2,2]*BYM + PP[2,3]*BZM + PP[2,4];
  WW[3] := PP[3,1]*BXM + PP[3,2]*BYM + PP[3,3]*BZM + PP[3,4];
  WW[4] := PP[4,1]*BXM + PP[4,2]*BYM + PP[4,3]*BZM + PP[4,4];
```

```
    DEN := WW[1]*BXM + WW[2]*BYM + WW[3]*BZM + WW[4] + LAMBDA;
    RGAIN[I,1] := RGAIN[I,1] + WW[1]*ERR/DEN;                        {Eq. 6.13}
    RGAIN[I,2] := RGAIN[I,2] + WW[2]*ERR/DEN;
    RGAIN[I,3] := RGAIN[I,3] + WW[3]*ERR/DEN;
    ROFFS[I] := ROFFS[I] + WW[4]*ERR/DEN;
    PP[1,1] := (PP[1,1] - WW[1]*WW[1]/DEN)/LAMBDA;                    {Eq. 6.14}
    PP[1,2] := (PP[1,2] - WW[1]*WW[2]/DEN)/LAMBDA;
    PP[1,3] := (PP[1,3] - WW[1]*WW[3]/DEN)/LAMBDA;
    PP[1,4] := (PP[1,4] - WW[1]*WW[4]/DEN)/LAMBDA;
    PP[2,2] := (PP[2,2] - WW[2]*WW[2]/DEN)/LAMBDA;
    PP[2,3] := (PP[2,3] - WW[2]*WW[3]/DEN)/LAMBDA;
    PP[2,4] := (PP[2,4] - WW[2]*WW[4]/DEN)/LAMBDA;
    PP[3,3] := (PP[3,3] - WW[3]*WW[3]/DEN)/LAMBDA;
    PP[3,4] := (PP[3,4] - WW[3]*WW[4]/DEN)/LAMBDA;
    PP[4,4] := (PP[4,4] - WW[4]*WW[4]/DEN)/LAMBDA;
    PP[2,1] := PP[1,2];  PP[3,1] := PP[1,3];  PP[4,1] := PP[1,4];
    PP[3,2] := PP[2,3];  PP[4,2] := PP[2,4];  PP[4,3] := PP[3,4];
  end; {UPDATE}

{Main part of RLSMAGCAL}
begin
  YY[1] := BX0*AA[1,1] + BY0*AA[1,2] + BZ0*AA[1,3];                  {Eq.6.10a}
  YY[2] := BX0*AA[2,1] + BY0*AA[2,2] + BZ0*AA[2,3];
  YY[3] := BX0*AA[3,1] + BY0*AA[3,2] + BZ0*AA[3,3];
  UPDATE(P1,1);             {3× scalar RLS estimation problems}
  UPDATE(P2,2);
  UPDATE(P3,3);                                                      {Eq. 6.3}
  BXC := BXM*RGAIN[1,1] + BYM*RGAIN[1,2] + BZM*RGAIN[1,3] + ROFFS[1];
  BYC := BXM*RGAIN[2,1] + BYM*RGAIN[2,2] + BZM*RGAIN[2,3] + ROFFS[2];
  BZC := BXM*RGAIN[3,1] + BYM*RGAIN[3,2] + BZM*RGAIN[3,3] + ROFFS[3];
end; {RLSMAGCAL}
```

Appendix G. ADCS Software Specification

G.1 Tasks triggered by asynchronous events

1. ICP Communication Input (ICP_COMMS_IN)
Function:

- Service the UART interface from the ICP.
- Generate a time-out indicating an ICP error.

Input:

- Sensor data message interrupt every second.

Output:

- Sensor data to SENSOR_CALIB.

2. OBC Communication Input (OBC_COMMS_IN)
Function:

- Service the Transputer link from the OBC.

Inputs:

- Attitude control commands as generated by the OBC.
- Parameter updates for the environment models, as received from the uplink.
- Telemetery sensor data when the ICP is down.

Outputs:

- Attitude control commands to ADCS_MANAGER.
- Parameter updates to ADCS_MODELS.
- Telemetery sensor data to SR_INTERFACE.

G.2 Tasks driven by timers

1. Environment Modelling (ADCS_MODELS)
Timer:

1 second.

Function:

- Implements various models required by die ADCS control algorithms,
  1. Satellite orbit
  2. Sun orbit
  3. Sun terminator
  4. Horizon angle
  5. Geomagnetic field (IGRF)

Input:

- Model parameter updates from OBC_COMMS_IN.

Outputs:

- Model data updates every second to ATTITUDE_VALID.
- Satellite orbit data every 10 seconds to OBC_COMMS_OUT.
- IGRF model data every second to SENSOR_CALIB.

2. Direct Reaction Wheel Interface (RW_INTERFACE)

Timer:

100 ms.

Function:

- Generate the reaction wheel speed reference commands from the control vector.

Input:

- Control vector update every second from CONTROL_ALGO.

Output:

- Speed reference commands every 100 ms. to the reaction wheels.

3. Direct Magnetorquer Interface (MT_INTERFACE)

Timer:

100 ms.

Function:

- Generate the switching pulses for the magnetorquers from the control vector.

Input:

- Control vector update every 10 seconds from CONTROL_ALGO.

Output:

- Switching commands at 100 ms. resolution to the magnetorquer coils.

4. Direct Sensor Interface (SR_INTERFACE)

Timer:

100 ms.

Function:

- Obtain the various sensor measurements.
- Low pass filter some sensor data.

Input:

- Direct measurements from the sensor hardware every 100 ms.
  1. Magnetometer.
  2. Reaction wheel tachos and angular position counters.
  3. Star camera (every second).
- Telemetery data of sensors every second from OBC_COMMS_IN,
  4. X/Y Horizon sensors.
  5. Fine sun sensor.
  6. Coarse sun sensor.

Output:

- Sensor datu message every second to SENSOR_CALIB.

G.3 Data (message) driven tasks

1. **ICP Communication Output** (ICP_COMMS_OUT)

Function:

- Service the UART interface to the ICP.
- Generate a time-out indicating an ICP error.

Input:

- Reaction wheel control vector every second from CONTRCL_ALGO.
- Magnetorquer control vector every 10 seconds from CONTROL_ALGO.

Output:

- Control vector(s) every second to the ICP.

2. **Sensor Data Calibration** (SENSOR_CALIB)

Function:

- Convert the raw sensor measurements to pre-calibrated standard units.
- Do on-line calibration of the magnetometer (Chapter 6).

Input:

- IGRF model data every second from ADCS_MODELS.
- Calibration constants and sensor status from ADCS_MANAGER.
- Sensor measurement data every second from ICP_COMMS_IN.

or

- Sensor measurement data every second from SR_INTERFACE.

Output:

- Calibrated sensor data every second to ATTITUDE_VALID.
    1. Horizon sensor.
    2. Fine sun sensor.
- Calibrated sensor and IGRF model data every second to ATTITUDE_ESTIM.
    3. Magnetometer.
    4. IGRF model.
    5. Reaction wheel tacho.
    6. Star sensor.
- Calibrated reaction wheel tacho data to ADCS_MANAGER.

3. **Attitude Test and Validation** (ATTITUDE_VALID)

Function:

- Verify horizon and sun sensor measurements against model data.
- Generate horizon and sun model vectors in the orbit reference frame.

Input:

- Environment model data every second from ADCS_MODELS.
- Horizon and sun sensor measured data every second from SENSOR_CALIB.
- Estimated attitude DCM every second from ATTITUDE_ESTIM.

Output:

- Measured and modelled vectors pairs every second to ATTITUDE_ESTIM.
- Environment status of the horizon and sun sensors every second to ADCS_MANAGER.

4. Attitude Estimators (ATTITUDE_ESTIM)

Function:

- Robust angular rate Kalman filter (Chapter 5).
- Magnetometer full state EKF (Chapter 5).
- Sun/Horizon/Star full state accurate EKF (Chapter 5).

Input:

- Sensor and IGRF model data every second from SENSOR_CALIB.
- Sun/Horizon vector pairs every second from ATTITUDE_VALID.
- Actuator control torque vectors every second from CONTROL_ALGO.
- Estimator supervision signals from ADCS_MANAGER.

Output:

- Estimated attitude DCM every second to ATTITUDE_VALID.
- Full estimated state vector every second to CONTROL_ALGO.
- Full estimated state vector plus state errors every second to ADCS_MANAGER.
- Estimated attitude plus rate data every 10 seconds to OBC_COMMS_OUT.

5. Control Algorithms (CONTROL_ALGO)

Function:

- Detumbling and pre-boom deployment control (Chapter 2).
- Libration and Z-spin rate control (Chapter 2).
- Pointing and tracking control (Chapter 3).
- Large angular slew maneuvers (Chapter 3).
- Momentum dumping (Chapter 4).

Input:

- Full estimated state vector every second from ATTITUDE_ESTIM.
- Reference attitude and rate vectors every second from ADCS_MANAGER.
- Controller supervision signals from ADCS_MANAGER.

Output:

- Reaction wheel control command vector every second to ICP_COMMS_OUT.

or

- Reaction wheel control command vector every second to RW_INTERFACE.
- Magnetorquer control command vector every 10 seconds to ICP_COMMS_OUT.

or

- Magnetorquer control command vector every 10 seconds to MT_INTERFACE.
- Actuator control torque vectors every second to ATTITUDE_ESTIM.

- Actuator control torque vectors every second to ADCS_MANAGER.

6. ADCS Manager (ADCS_MANAGER)

Function:

- Schedule the various control modes.
- Oversee and initialize the estimators.
- Generate the attitude and rate reference commands.
- Dynamically reconfigure the tasks during errors (e.g. ICP time-out).
- Interpret ADCS commands from the OBC (e.g. for camera pointing).
- Hardware reconfiguration (on/off) of the attitude sensors.

Input:

- Attitude control commands from OBC_COMMS_IN.
- Reaction wheel tacho data every second from SENSOR_CALIB.
- Full estimated state vector plus errors every second from ATTITUDE_ESTIM.
- Actuator control torque vectors every second from CONTROL_ALGO.
- Environment status of the horizon and sun sensors every second from ATTITUDE_VALID.

Output:

- Calibration constants and sensor status to SENSOR_CALIB.
- Estimator supervision signals to ATTITUDE_ESTIM.
- Controller supervision signals to CONTROL_ALGO.
- Reference attitude and rate vectors every second to CONTROL_ALGO.
- ADCS hardware reconfigure commands for the telecommand system to OBC_COMMS_OUT.

7. OBC Communications Output (OBC_COMMS_OUT)

Function:

- Service the Transputer link output to the OBC.

Input:

- Satellite orbit data every 10 seconds from ADCS_MODELS.
- Estimated attitude plus rate data every 10 seconds from ATTITUDE_ESTIM.
- ADCS hardware reconfigure commands for the telecommand system from ADCS_MANAGER.

Output:

- Link messages to the OBC.