



UNIVERSIDADE TÉCNICA DE LISBOA
INSTITUTO SUPERIOR TÉCNICO

Small Satellites Attitude Determination Methods

Sónia Maria Martinho Marques
(Licenciada)

**Dissertação para obtenção do Grau de Mestre em
Engenharia Electrotécnica e de Computadores**

DOCUMENTO PROVISÓRIO

Lisboa, Dezembro de 2000

Resumo

O conhecimento da orientação ou atitude de um satélite, relativamente a um dado referencial, é essencial para permitir o seu ajuste através de algoritmos de controlo em cadeia fechada. Este conhecimento da atitude torna-se ainda mais relevante quando os sistemas são autónomos e o sistema de estimação de atitude tem de ser mais preciso e robusto, de modo a que os algoritmos de controlo possam actuar sobre o satélite, sem intervenção humana a partir de estações terrestres.

Este trabalho foi motivado pela necessidade de obter um método de estimação de atitude e velocidade angular de um pequeno satélite em órbita a baixa altitude, em particular o satélite português PoSAT-1. Esta classe de satélites apresenta uma série de características tais como a limitação em massa e volume, a não linearidade da dinâmica e as medidas dos sensores de atitude que, para além de ruidosas, nem sempre estão disponíveis, originando desafios interessantes para a determinação de atitude.

Os métodos de determinação de atitude dividem-se em duas classes principais: os métodos ponto-a-ponto e os métodos recursivos. Os métodos ponto-a-ponto são baseados nas medidas de pelo menos dois sensores de atitude, num dado instante de tempo, enquanto a estimação recursiva usa informação de amostras sucessivas, bem como o conhecimento da dinâmica de atitude do satélite. Nesta dissertação, é estudada a possibilidade de aplicação do método ponto-a-ponto por decomposição em valores singulares (SVD), comparando-a com os resultados obtidos por um Filtro de Kalman Estendido (EKF). Resultados da aplicação dos métodos à simulação do PoSAT-1, utilizando o seu sensor de sol e os seus magnetómetros, bem como a dados reais do pequeno satélite PoSAT-1 são apresentados.

Palavras Chave: Estimação de atitude, Quaternões, Pequenos satélites, Filtro de Kalman Estendido, Métodos Ponto a Ponto.

Abstract

The determination of a satellite orientation or attitude with respect to a given coordinate system, has a major role in guidance, navigation and control, especially for autonomous systems which are less fault tolerant than ground-based systems.

This work was motivated by the need to develop a method to estimate both the attitude and the angular velocity of a low orbit small satellite, in particular the Portuguese satellite PoSAT-1. This class of satellites has a number of characteristics such as mass and volume constraints, non-linear equations of motion and noisy measurements from the attitude sensors which are not available along the whole orbit of the satellite.

Satellite attitude determination methods usually fall in one of two classes: point-to-point and recursive estimation algorithms. Point-to-point attitude determination is based on the measurements of two or more sensors in a single point in time, while recursive estimation uses information from successive time points, as well as knowledge about the spacecraft attitude dynamics model. In small satellites, only a single attitude sensor is often available, due to cost and space constraints, thus leading to the exploration of recursive estimation based solutions, such as the Kalman filter. In this work, the results of using a point-to-point Singular Value Decomposition (SVD) algorithm are compared to those obtained by an Extended Kalman Filter (EKF), when applied to a simulation of PoSAT-1. Results from the EKF, applied to the small satellite PoSAT-1 real data, is also presented.

Keywords: Attitude estimation, Quaternions, Small Satellites, Extended Kalman Filter, Point-to-point Methods.

Preface and Acknowledgments

I greatly acknowledge my thesis to my supervisor Professor Pedro Lima for his guidance, support and friendship during this work and for giving me the opportunity to participate in the ConSat project.

I would like to express my gratitude to my colleagues within the ConSat Project specially to Pedro Tavares and Paulo Tabuada for their friendship and support as well as to the Institute for Systems and Robotics (ISR) for providing me with excellent work conditions.

A especial thanks to Dr. Herman Steyn who shared with me his experience as Team Leader of Spacecraft Attitude Determination and Control System of the Surrey Space Center, and also to Prof. Thomas Bak from Aalborg University, Department of Control Engineering, for being so patient about my doubts, to Mark Pittelkau for his friendship and enlightened conversations about Attitude Determination System (ADS). To the PRAXIS XXI¹ program that financed my presentation in the American Institute of Aeronautics and Astronautics (AIAA) conference in Denver which gave me the opportunity to meet wonderful people like John Junkins, Daniele Mortari, Malcom Shuster, Landis Markley, Murty Challa, Gregory Natanson, Itzack Bar-Itzack, Y. Oshman, Reiner.

I would like to thank my family and my husband for its continued support and again to my supervisor Professor Pedro Lima, who read through the manuscript before I submitted it.

Finally, I would like to thank the EST/IPS (Escola Superior de Tecnologia - Polytechnic Institute of Setúbal) for encouraging and supporting this work.

¹This work is supported by PRAXIS XXI program project PRAXIS/3/3.1/CTAE/1942/95

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Historical Perspective	2
1.2.1	Iterative Methods	3
1.2.2	Point-to-point Methods	4
1.3	Goals and Contributions	6
1.4	Thesis Outline	7
2	Spacecraft Attitude Models	8
2.1	Coordinate Systems	8
2.2	Rotation Vector and Frame	10
2.2.1	Direction Cosine Matrix	10
2.2.2	Euler Axis/Angle	12
2.2.3	Euler Angles $roll(\psi)$, $pitch(\theta)$, $yaw(\phi)$	14
2.2.4	Gibbs Vector	15
2.2.5	Euler Symmetric Parameters - Quaternions	15
2.3	Attitude Parameterization Analysis	20
2.4	Spacecraft Equations of Motion	20
2.4.1	Dynamic Equations	21
2.4.2	Kinematic Equations	25
2.4.3	Small Satellite External Torques	26
3	Attitude Determination Methods	28
3.1	Recursive Estimation Methods	29
3.1.1	Kalman Filter Applied to Small Satellites	30
3.1.2	Extended Kalman Filter Intrinsic Problems	43
3.2	Point-to-point Approaches	45
4	Simulation Setup	50
4.1	SimSat Simulator	50
4.2	PoSAT-1 Experiments	51

4.2.1	PoSAT-1 Technical Data	52
4.2.2	PoSAT-1 Simulations	56
4.2.3	PoSAT-1 Real Data Experiments	63
4.3	Tests	70
4.4	Performance Criteria	72
5	Results	76
5.1	PoSAT-1 Simulation	76
5.1.1	Test A	76
5.1.2	Test D	77
5.1.3	Test G	82
5.1.4	Test J	82
5.1.5	Test B	84
5.1.6	Test E	84
5.1.7	Test H	87
5.1.8	Test K	91
5.1.9	Test C	91
5.1.10	Test F	96
5.2	PoSAT-1 Real Data	96
6	Conclusions and Future Work	101
A	Discrete Kalman Filter	108
A.1	Process Model	108
A.2	Measurement Model	109
A.3	Update estimate	109
A.4	Covariance Matrix and Optimal Gain Matrix	111
A.5	Propagation of the state variables	111
A.6	Propagation of the Error Covariance Matrix	112
A.7	KF Algorithm	112
B	Singular Value Decomposition (SVD)	114
B.1	SVD Solution to Wahba's Problem	115
B.1.1	Covariance matrix	116
C	Keplerian Orbital Elements	117
C.1	Orbit size and shape	117
C.2	Orbital orientation	118
C.3	Orbital position	120
C.4	Perturbations	120
D	NORAD Two-Line Element Set Format	121

List of Figures

2.1	Orbital Coordinate System (OCS) and Body Coordinate System (BCS).	9
2.2	Inertial Coordinate System.	10
2.3	Direction Cosines angle between triad \mathfrak{F}_b and \mathfrak{F}_d	11
2.4	The orientation of a vector \mathbf{r} , expressed in terms of the triad \mathfrak{F}_d	12
2.5	Angle/axis rotation about \mathbf{z}	13
2.6	The Euler angle 1 – 2 – 3 rotation or Roll - ψ , pitch - θ and yaw - φ .	14
2.7	Cross product of two vectors.	18
2.8	A rigid body with mass m has an acceleration \mathbf{a} due to external force \mathbf{F}	21
2.9	<i>Angular momentum</i>	22
2.10	Rate of change of a rotating vector.	23
2.11	Rate of change of the angular momentum of a rotating rigid body. . .	24
3.1	Execution Times for Robust Estimation Algorithms (reprinted from [35]).	46
4.1	PoSAT-1 design.	52
4.2	Expanded view of PoSat-1.	53
4.3	Motion of a nutating Spacecraft. The body cone rolls on the spacecraft cone for $I_1 = I_2 > I_3$. $\hat{\mathbf{p}}_3$ is the principal axis of inertia, $\boldsymbol{\omega}$ is the instantaneous rotation axis, θ is the <i>nutation angle</i> and \mathbf{H} is the angular momentum.	54
4.4	Single axis analogue Sun sensor.	54
4.5	The picture shows the quaternions obtained from the attitude matrix estimates using (4.15) - (4.18). The blue line is referred to the estimates and the red line is referred to the true data. For visualization of the SVD method, a signal indicating when Sun sensor measurements are available is high and low otherwise, originating a square wave in all plots.	62
4.6	From t_k to t_{k+1} the sign of the maximum value found can not change.	63
4.7	PoSAT-1 real data measured by the magnetometers [μT], at 2000/11/02, 00:00:34.	65

4.8	PoSAT-1 real vector magnetic moment m [Am ²], from all coils, starting at 2000/11/02, 00:00:34.	65
4.9	PoSAT-1 real attitude, Euler angles: roll, pitch and yaw, at 2000/11/02, 00:00:34.	66
4.10	PoSAT-1 IGRF 10 th order reference field components [μT], at 2000/11/02, 00:00:34, in orbital CS.	67
4.11	Orbital Coordinate System (OCS) and Body Coordinate System (BCS) for PoSAT-1 real data.	68
4.12	γ angle represented in the satellite.	73
5.1	Test A. RMS (black line) of the angular velocity error [rad/s] bounded by its standard deviation (blue line).	77
5.2	Test A. RMS (black line) of the Euler angles [o] error is bounded by its standard deviation (blue line).	78
5.3	Test A. Estimation results (blue line) and the true values of the angular velocities (red line). The y-axis label is the RMS for the angular velocity [$\omega_{x,y,z}$ err=rad/s].	78
5.4	Test A. Estimation results (blue line) and the true values of the <i>roll</i> (ψ), <i>pitch</i> (θ) and <i>yaw</i> (ϕ) Euler angles (red line). In the y-axis label is the RMS for the Euler angles [Roll/Pitch/Yaw err= o].	79
5.5	Test A. Euler angle error.	79
5.6	Test A. The γ evolution.	80
5.7	Test D. Estimation results (blue line) and the true values of the angular velocities (red line). The y-axis label is the RMS for the angular velocity [$\omega_{x,y,z}$ err=rad/s].	80
5.8	Test D. Estimation results (blue line) and the true values of the <i>roll</i> (ψ), <i>pitch</i> (θ) and <i>yaw</i> (ϕ) Euler angles (red line). In the y-axis label is the RMS for the Euler angles [Roll/Pitch/Yaw err= o].	81
5.9	Test D. Euler angle error.	81
5.10	Test D. The γ evolution.	82
5.11	Test G. Estimation results (blue line) and the true values of the angular velocities (red line). The y-axis label is the RMS for the angular velocity [$\omega_{x,y,z}$ err=rad/s].	83
5.12	Test G. Estimation results (blue line) and the true values of the <i>roll</i> (ψ), <i>pitch</i> (θ) and <i>yaw</i> (ϕ) Euler angles (red line). In the y-axis label is the RMS for the Euler angles [Roll/Pitch/Yaw err= o].	83
5.13	Test G. Euler angle error.	84
5.14	Test J. Estimation results (blue line) and the true values of the angular velocities (red line). The y-axis label is the RMS for the angular velocity [$\omega_{x,y,z}$ err=rad/s].	85

5.15	Test J. Estimation results (blue line) and the true values of the <i>roll</i> (ψ), <i>pitch</i> (θ) and <i>yaw</i> (ϕ) Euler angles (red line). In the <i>y</i> -axis label is the RMS for the Euler angles [Roll/Pitch/Yaw err= o].	85
5.16	Test J. Euler angle error.	86
5.17	Test B. Estimation results (blue line) and the true values of the angular velocities (red line). The <i>y</i> -axis label is the RMS for the angular velocity [$\omega_{x,y,z}$ err=rad/s].	86
5.18	Test B. Estimation results (blue line) and the true values of the <i>roll</i> (ψ), <i>pitch</i> (θ) and <i>yaw</i> (ϕ) Euler angles (red line). In the <i>y</i> -axis label is the RMS for the Euler angles [Roll/Pitch/Yaw err= o].	87
5.19	Test E. Estimation results (blue line) and the true values of the angular velocities (red line). The <i>y</i> -axis label is the RMS for the angular velocity [$\omega_{x,y,z}$ err=rad/s].	88
5.20	Test E. Estimation results (blue line) and the true values of the <i>roll</i> (ψ), <i>pitch</i> (θ) and <i>yaw</i> (ϕ) Euler angles (red line). In the <i>y</i> -axis label is the RMS for the Euler angles [Roll/Pitch/Yaw err= o].	88
5.21	Test E. Euler angle error.	89
5.22	Test H. Estimation results (blue line) and the true values of the angular velocities (red line). The <i>y</i> -axis label is the RMS for the angular velocity [$\omega_{x,y,z}$ err=rad/s].	89
5.23	Test H. Estimation results (blue line) and the true values of the <i>roll</i> (ψ), <i>pitch</i> (θ) and <i>yaw</i> (ϕ) Euler angles (red line). In the <i>y</i> -axis label is the RMS for the Euler angles [Roll/Pitch/Yaw err= o].	90
5.24	Test H. Euler angle error.	90
5.25	Test K. Estimation results (blue line) and the true values of the angular velocities (red line). The <i>y</i> -axis label is the RMS for the angular velocity [$\omega_{x,y,z}$ err=rad/s].	91
5.26	Test K. Estimation results (blue line) and the true values of the <i>roll</i> (ψ), <i>pitch</i> (θ) and <i>yaw</i> (ϕ) Euler angles (red line). In the <i>y</i> -axis label is the RMS for the Euler angles [Roll/Pitch/Yaw err= o].	92
5.27	Test K. Euler angle error.	92
5.28	Test C. RMS (black line) of the Euler angles [o] error is bounded by its standard deviation (blue line) which only is computed on SVD is running.	93
5.29	Test C. Estimation results (blue line) and the true values of the angular velocities (red line). The <i>y</i> -axis label is the RMS for the angular velocity [$\omega_{x,y,z}$ err=rad/s].	94
5.30	Test C. Estimation results (blue line) and the true values of the <i>roll</i> (ψ), <i>pitch</i> (θ) and <i>yaw</i> (ϕ) Euler angles (red line).	94
5.31	Test C. Euler angle error.	95
5.32	Test C. Euler angle error. Detail of Fig. 5.31.	95

5.33	Test C. The γ evolution. Estimated (blue line) and true (red line) values of γ .	96
5.34	Test F. Estimation results (blue line) and the true values of the angular velocities (red line). The y -axis label is the RMS for the angular velocity [$\omega_{x,y,z}$ err=rad/s].	97
5.35	Test F. Estimation results (blue line) and the true values of the <i>roll</i> (ψ), <i>pitch</i> (θ) and <i>yaw</i> (ϕ) Euler angles (red line).	97
5.36	Test ZP. EKF estimates, for aprox. 7.2 orbits, of the Euler angles (blue line) in comparison with the estimates of the EKF on-board of the PoSAT-1 satellite (red line). The magnetometers are the only data supllied to the filter.	98
5.37	Test ZP. EKF estimates of the angular velocity [rad/s] from PoSAT-1 real data for 14 orbits. Magnetometers measurements are the only data supplied to the filter.	99
A.1	Continuous process sampled at discrete times and an discrete measurement model.	109
A.2	Kalman filter loop.	113
C.1	Definition of the orbit size and shape parameters.	118
C.2	Definition of the orbital orientation parameters.	119
D.1	The description of the compact form of the two-line element (TLE) format. It consists of three lines, where the first is the satellites's name and the next two lines correspond to the data in a rigid format.	122

List of Tables

4.1	Two-Line Element of PoSAT-1 used in SimSat environment.	58
4.2	NORAD elements used of the PoSAT-1 simulation in Simsat.	58
4.3	The Two-Line Element of PoSAT-1 used for real data experiments. .	63
4.4	NORAD elements used of the PoSAT-1 real data.	64
4.5	The several tests performed to the data in this work. The shortname Init is referred to the initial conditions proposed.	70
4.6	The initial state conditions applied to the PoSAT-1 simulation. . . .	71
4.7	Initial orbit conditons for January, 1997.	71
4.8	Initial pitch and angular velocity.	71
4.9	Desired pitch and angular velocity for the closed loop tests.	72
5.1	RMS summary of the maximum Euler angles for all Tests.	99
5.2	RMS summary of angular velocity error for all Tests.	100

Nomenclature

Acronyms and Abbreviations

ACS	Attitude Control System
ADS	Attitude Determination System
CS	Coordinate System
COP	Classical Orbit Parameters
EKF	Extended Kalman Filter
ESOQ	Estimator of the Optimal Quaternion
FOAM	Fast Optimal Attitude Matrix
GPS	Global Positioning System
GEO	GEostationary satellite
IGRP	International Geomagnetic Reference Field
LEO	Low Earth Orbit
LVLH	Local Vertical Local Horizontal coordinate system
MME	Minimum Model Error
NORAD	Aerospace Defense Command North American
OCS	Orbital Coordinate System
BCS	Body Coordinate System
ICS	Inertial Coordinate System
ECEF	Earth-Centered Earth-Fixed coordinate system
REQUEST	REcursive QUartenion ESTimator
RTSF	Real Time Sequential Filter
QUEST	QUartenion ESTimator
SVD	Singular Value Decomposition
SGP4	especial General Perturbation model of fourth order
UTC	Universal Time Coordinated
Eq.	Equation
<i>h.o.t.</i>	higher order terms
<i>w.r.t.</i>	with respect to
<i>vd.</i>	from the Latin <i>videre</i> meaning look at
<i>i.e.</i>	from the Latin <i>id est</i> meaning that is
<i>e.g.</i>	from the Latin <i>exempli gratia</i> meaning for example

List of Symbols

$\Omega(\cdot)$	matrix left operator of a quaternionic multiplication
Ω	is the absolute angular velocity of a rotating frame
$\Xi(\cdot)$	matrix right operator of a quaternionic multiplication

q	Euler symmetric parameters or quaternion
\mathbf{q}	vector part of a quaternion, q_1, q_2, q_3
q_4	scalar part of a quaternion
q^*	conjugate of a quaternion
q^{-1}	inverse of a quaternion
A	direction cosine matrix
$A(\mathbf{q})$	direction cosine matrix expressed in terms of the quaternions or attitude matrix
\mathbf{H}	angular momentum
\mathbf{L}	linear momentum
I	principal moments of inertia matrix.
$\mathbf{1}$	eye matrix
\mathbf{N}_{ctrl}	control torque
\mathbf{N}_{gg}	gravity gradient torque
\mathbf{N}_{dist}	disturbance torque
\mathbf{N}_{wheel}	momentum wheel torque
$\mathbf{m}(t)$	magnetic moment generated by the satellite coils
\mathbf{b}_{meas}	geomagnetic field measured by on-board instruments
\mathbf{b}^o	geomagnetic field predicted by the IGRF model
\mathbf{y}_m^b	vector of observations taken by on-board instruments
\mathbf{y}^o	vector of measurements obtained from a model of the attitude sensors
R_{CM}	mean distance of the satellite to Earth center
R_s	position vector of the satellite center mass <i>w.r.t</i> Earth center
T	period of orbit
J	cost function
\mathbf{g}	Gibbs vector or Rodrigues parameters
ω_0	orbit angular rate
$\boldsymbol{\omega}$	angular velocity vector
$\boldsymbol{\omega}_{bi}^o$	angular velocity of body CS <i>w.r.t.</i> inertial CS expressed in orbit CS
$\omega_{x,y,z/b0}$	angular velocity components x,y,z in body CS <i>w.r.t.</i> inertial CS
γ	angle between the satellite's boom and the local vertical
μ	Earth's gravitational constant, $3.986005 \times 10^{14} \text{ m}^3\text{s}^{-2}$
ΔT	sampling time
\mathbf{x}	state vector
ψ, θ, ϕ	roll, pitch, yaw
$SO(n)$	special orthogonal Lie group of dimension n
$\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \hat{\mathbf{e}}_3$	constituent vectors or unit vectors that form a CS. Its directions have the positive axes of a CS.

Φ	Euler rotation angle
$\delta\omega$	angular velocity perturbation vector
$\delta\mathbf{q}$	quaternion perturbation vector
$\{F\}$	frame

Terminology

albedo fraction of reflected radiation from the globe due to all incident energy that falls on its surface

infrared (IR) total thermal radiation from both the surface and the atmosphere

Roll, Pitch, Yaw Euler angles describing the satellite attitude. Roll is the first rotation around the x -axis of the orbital coordinate system, pitch is the next rotation around the rotated y -axis, and yaw the final rotation around the rotated z -axis.

Universal Coordinated Time is a time scale determined using highly precise atomic clocks. The length of a UTC second is defined in terms of an atomic transition of the element cesium under specific conditions, and is not directly related to any astronomical phenomena.

Small Satellites in the spirit of the current small satellite world, the small or micro satellites are known by the slogan “Faster, Better, Smaller, Cheaper”. Small satellites are characterized by lighter systems designed inside smaller volumes weighing less than 500kg. Because of the budget restrictions they are built in few years leading to a quick turn-around instead years of planning when compared to the commercial space industry. So leading-edge technology is routinely included in order to provide innovative solutions. They are autonomous systems usually having low Earth orbit.

dextral right-handed.

Frame is an orthogonal right-handed triad whose constituent vectors are unit vectors.

Coordinate System is defined as a frame but used to denote special frames relating the satellite, its orbit and the Earth.

constituent vector versor or unit vector $\hat{\mathbf{e}}$.

Ecliptic is the mean plane of the Earth’s orbit about the Sun.

Precession of the equinoxes is the point where the ecliptic crosses the Earth equator from south to north.

Zenith is a unit vector in the control CS pointing to the direction opposite to the center of Earth.

Nadir is the direction opposite to the Zenith.

Geostationary Satellite is a satellite whose orbit has a period of one sideral day (1436.3 minutes) with zero inclination and eccentricity, *i.e.*, an equatorial and circular orbit.

Polar orbit is an orbit whose plane is inclined about 90° to the equatorial plane intersecting the North and South poles.

Chapter 1

Introduction

1.1 Motivation

Estimating the orientation of a spacecraft with respect to (*w.r.t.*) a Coordinate System (CS) is essential for most space missions in order to support an orbit manoeuvre. Without an efficient attitude determination algorithm, the mission may fail. This is even more important for autonomous spacecrafts, such as Low Earth Orbit (LEO) satellites, GEOstationary (GEO) satellites, or interplanetary satellites.

Due to their typically low budget, small satellites suffer structure restrictions. They have limited mass and volume, low cost components, low cost to launch, non-optimal orbit insertion. Through the years, small satellites have provided a platform for scientific research where new products or system concepts can be demonstrated, making possible for small satellites to benefit from leading-edge technology. Therefore, more and more small satellites can accomplish many functions of the conventional satellites without the huge costs associated with it. An example is the commercial Gemini geostationary minisatellite (to be launched in 2003 by Surrey Space Center, UK) that will provide reliable and accessible communications services for Nigeria and West African countries. Moreover, it is possible to have a quick turn-around instead a decade of planning and building as it is typical with conventional satellites.

Since small satellites are typically in LEO, the preferred attitude actuators are those which generate a magnetic momentum that interacts with the Earth geomagnetic field, thus generating a torque that rotates the satellite. These actuators are cheap to build and have long service life since no fuel is used except for electric power which is obtained from the solar panels. Also gas jets are becoming an option for this class of satellites but their life is limited by gas availability.

This work is included in the ConSat¹ project which aims at studying the dynam-

¹This work is supported by PRAXIS XXI program project PRAXIS/3/3.1/CTAE/1942/95,

ics of bodies under the influence of gravitational, aerodynamic and control torques in the particular case of small satellites. The work carried out in this dissertation is included in the development of new approaches to the attitude control where attitude determination is essential.

There are a reduced number of attitude sensors on-board small satellites: magnetometers; Sun sensors; GPS - Global Positioning System; Earth sensors; horizon sensors; star sensors. None of these sensors provides directly either the attitude or the angular velocity, except Star sensors that give the attitude of the camera *w.r.t.* inertial CS. The study of the estimation methods suitable for these reduced attitude sensors and coping with its noisy measurements and availability is part of this work. Often only a single attitude sensor, the magnetometer, is available during the whole orbit, leading to the usage of the recursive Kalman Filter (KF) to estimate the state vector. The point-to-point methods are studied as an alternative to Kalman Filters, to avoid their implementation complexity and the assumptions made on the process model and perturbations. This is particularly true for non-linear systems, as it is the case for small satellites, due to the need to linearize the KF around the system trajectory at each step - Extended Kalman Filter (EKF). However these methodologies just estimate the attitude but it is also essential to have estimates from angular velocity since usually small satellites do not have gyros on-board. Also, the fact that the point-to-point methods, in order to work, must have available at least measurements of two attitude sensors² in a single point in time, are also addressed in Chapter 4, and then applied to PoSAT-1 satellite.

PoSAT-1, a small satellite launched in 1993 by Surrey Space Center and owned by a Portuguese Consortium, offers challenging problems, with their sensor and actuator restrictions. The satellite has a gravity gradient boom for Earth stabilization using minimum energy. Due to the solar panels, placed on its 4 facets, the satellite is kept spinning slowly about the *z*-axis.

1.2 Historical Perspective

The main goal of the attitude determination system is to compute the attitude of a body fixed CS *w.r.t.* a CS of interest, as well as the angular velocity, based on noisy vector measurements taken in both systems.

There are two widely applied methodologies to the attitude determination problem: the Kalman filter and the point-to-point methods. Although both emerged in the sixties, only recently point-to-point methods have been subject to intensive research. In contrast, the Kalman filter has been applied successfully to many different estimation problems since it is well-suited to estimate state vectors of multi-

¹Stabilization and Control of small Satellites”.

²Except the TRIAD point-to-point algorithm.

input/output systems based in multi-sensor data. An historical perspective is made, based on these two methodologies and considering also the attitude parameterizations used.

1.2.1 Iterative Methods

Kalman filter's adequacy for real-time estimation is due to its recursive processing of noisy sensor measurements to determine the successive minimum variance state estimates. Assuming statistics of the system and measurements noise as known, and taking advantage of the system model, the filter propagates the estimated state from one time step to the next. Nevertheless, there are several problems associated with the application of Kalman filter to small satellites, explained in detail in Section 3.1.2.

Beyond the choice of the filter, the attitude representation is another problematic issue addressed by many authors and reviewed by Shuster [41]. Quaternion representation is the most commonly used satellite attitude estimation because it is not singular for any rotation. However it is subject to the constraint $\mathbf{q}^T \mathbf{q} = 1$ in order to maintain orthogonality in the estimated attitude. This constraint must be taken into account during the implementation of the EKF, otherwise the covariance matrix becomes singular. Lefferts *et al.* [27] described three different approaches to circumvent the covariance matrix singularity caused by the quaternion attitude representation. However, the angular velocities are obtained from gyroscopes, which are often not used in small spacecrafts because not only they are generally expensive but also they tend to fail. Also to overcome the quaternion problem, Bar-Itzhack *et al.* [20] showed that normalization improves filter convergence and accuracy. Applying the covariance modifications described by Lefferts *et al.* [27], Psiaki and Martel [31], using only magnetometer data, estimated the disturbance torques as well as the attitude, angular velocity and the vector part of the quaternion, $\mathbf{q} = [q_1 \ q_2 \ q_3]^T$. The scalar part of the quaternion q_4 is obtained from the estimated vector part using the quaternion constraint. However, their method offers coarse attitude information only due to the low accuracy of the observations and the inaccuracy of the knowledge of the Earth's magnetic field, as stressed by the author's.

Since the satellite attitude dynamics is a non-linear system with non-linear measurements, the application of the Kalman filter is only possible after linearization. This may lead to divergence problems in the error covariance matrix. Some solutions were presented by Brown and Hwang [18] in order to handle this problem. Also Bak [1] modelled some of the discrepancies between the process model and the actual behaviour of the spacecraft and gathered them in the covariance error matrix of an EKF using magnetic field measurements. Vathsal [53] expanded the process and measurement models to second order. However, this approach increases the

complexity of the filter and represents a computational burden. Usually, it is only worthwhile to go for second or higher order techniques in case of extreme system nonlinearities.

The process and measurement noise are assumed to be modelled by a zero-mean Gaussian stochastic process with known covariance, but the covariance matrices must be manually tuned, not necessarily to achieve optimal filter designs but sometimes to increase closed loop attitude control performance. Mook and Junkins [37] developed a new approach, designated as the Minimum Model Error Estimation (MME) method, where the model error is determined during the estimation process. Crassidis and Markley [7] used this approach to estimate the attitude of a real spacecraft without the utilization of gyro measurements. However, the MME filter is a batch estimator. Inspired in their paper, and also based on a predictive tracking scheme introduced by Lu [30], Crassidis and Markley [9] have proposed a predictive filter where the model error is estimated as part of the solution. The filter may take any form, even nonlinear, so the filter is free from covariance propagation, decreasing the computational burden associated with it. Moreover, it can be implemented on-line to filter noisy measurements, estimating quaternion attitude representation and rate trajectories.

The Kalman filter used in this work is based on the approach of Psiaki and Martel [31] but modified by Steyn [45] to be used for satellites with yaw spinning instead of assuming only a 3-axis nominal Earth pointing satellite as done by Psiaki and Martel. Also in this dissertation the Sun sensors are used to improve filter accuracy. To avoid the covariance singularity the filter estimates the vector part of the quaternion, reducing the rank of the covariance error matrix. The filter covariance matrix for measurement is based in the error artificially introduced in the SimSat³ simulator and is proposed an error covariance matrix for the process based in the quaternion constraint (see 3.1.2).

1.2.2 Point-to-point Methods

A different approach to the attitude estimation problem consists on determining the attitude based on a sequence of noisy vector measurements. Given a set of $n \geq 2$ vector measurements $\mathbf{b}_1, \dots, \mathbf{b}_n$ in the body system, and a set of reference vectors $\mathbf{r}_1, \dots, \mathbf{r}_n$ in the orbit system, there is an orthogonal matrix A (the attitude matrix or direction-cosine matrix) that transforms rotational vectors from the orbital to the body coordinates. The problem of finding the best estimate of the A matrix was posed by Wahba [54] who was the first to choose a least square criterion to define the best estimate, *i.e.*, to find the orthonormal matrix A that minimizes the loss

³SimSat is the simulation environment for LEO satellites implemented in Matlab/Simulink language under the ConSat project.

function

$$L(A) = \frac{1}{2} \sum_{i=1}^n w_i |\mathbf{b}_i - A\mathbf{r}_i|^2 \quad (1.1)$$

where w_i is a set of positive weights assigned to each measurement and $|\cdot|$ denotes the Euclidean norm. It was proved that the loss function can be rewritten as

$$L(A) = \lambda_0 - \text{tr}(AB)^T \quad (1.2)$$

with

$$\lambda_0 = \sum_{i=1}^n w_i \text{ and } B = \sum_{i=1}^n w_i \mathbf{b}_i \mathbf{r}_i^T \quad (1.3)$$

The loss function will be minimum when the trace of the matrix product AB^T is maximum, under the orthonormality constraint on A .

The \mathbf{q} method, introduced by Davenport [11], provides a quaternion-based solution for the Wahba problem, where the attitude quaternion \mathbf{q} which minimizes the loss function is the eigenvector of a matrix K , corresponding to K 's largest eigenvalue, λ_{\max} . Shuster [39] presented an implementation of the \mathbf{q} method, the QUaternion ESTimator (QUEST), which purpose is to determine λ_{\max} and the corresponding \mathbf{q} from the vector observations. This avoids solving the eigenvalue problem explicitly. The main disadvantage of this method is that the measurements are combined to provide an attitude estimate but the combination is not optimal in any statistical sense.

The Singular Value Decomposition (SVD) method, which computes the attitude matrix directly, is very simple and one of the most robust estimators minimizing Wahba's loss function, together with the \mathbf{q} method. However, the \mathbf{q} method is faster than the SVD when three or more measurements are available [35]. The Fast Optimal Attitude Matrix (FOAM), introduced by Markley [34], is a variation of the \mathbf{q} method which avoids the need to compute the eigenvectors and is faster than the \mathbf{q} method, though equally robust.

Shuster also derived a simple expression for the covariance matrix of the Three Axis Attitude Determination (TRIAD) algorithm deduced by Lerner [28]. In the latter, despite the simplicity of the attitude determination, the calculation of the covariance matrix was rather complicated because of the need to compute numerous partial derivatives as differences. Despite its popularity, the TRIAD algorithm⁴ can

⁴TRIAD was implemented in many missions, for instance the Small Astronomy Satellite (SAS) or the Atmospheric Explorer Missions (AEM).

only be solved for two different attitude measurements. This represents an important disadvantage when more measurements are available, since some accuracy is lost. Actually, it is possible to combine the attitude solutions of the various observation pairs. However, this solution tends to be too costly.

So, the point-to-point methods that use the vector measurements to obtain the attitude at a given time point require at least two vectors (except the TRIAD) to determine the attitude and they require weighting of the entire vector measurement. All point-to-point methods fail when only one set of vector measurements is available (e.g., magnetometer data only), which happens when a solar eclipse occurs. Moreover, they are single time point batch algorithm, where all measurements that are taken at a previous time are ignored. Bar-Itzack [25] presented a recursive routine derived from the QUEST which takes into account all the past measurements and because of that even one measurement is enough to update the attitude. In order to do so, the REQUEST algorithm uses the kinematic equation to propagate the quaternion obtained from the past measurements until time t_{k+1} and uses it together with the new measurements at time t_k . In spite of this, REQUEST requires exact knowledge of the angular velocity, relying on gyros measurements.

In general, all the point-to-point methods or deterministic methods compute the attitude matrix efficiently and with much less computational load than the EKF, because they do not use information from the dynamic and kinematic models, avoiding the modeling errors that arise in EKF. Therefore, they are very attractive to implement in small satellites with short computational resources. Nevertheless, they all require two vector measurements in order to estimate the attitude. Some researchers have used gyroscopes to obtain the angular velocity, but so far gyros are seldom used in small satellites because they are usually expensive and are often prone to failures, as referred before.

1.3 Goals and Contributions

The main goals and contributions of this work are:

- To analyse the several approaches to the problem of attitude estimation in small satellites, especially recursive versus point-to-point methods.
- To determine the most suitable representation for satellite attitude, to be used in the ADS and ACS.
- To implement an EKF and the SVD point-to-point method in the SimSat simulator, coping with PoSAT-1 characteristics.
- To introduce a method for the estimation of angular velocity when the SVD algorithm is used.

1.4 Thesis Outline

The present document is organized as follows:

Chapter 1, **Introduction** - an overview to the attitude determination of small satellites is presented as well as the motivation to this work. Also a historical perspective about the two main methodologies used (KF and point-to-point methods), considering the attitude parameterizations, is provided.

Chapter 2, **Spacecraft Attitude Models** - the Coordinate Systems relevant to the estimation problem are described, as well as the several ways to represent the attitude of a spacecraft, pointing out the most suitable to ADCS. A brief explanation of the dynamic equations of a spinning rigid body is provided and subsequently particularized to the small satellites external torques.

Chapter 3, **Attitude Determination Methods** - two methods representative of the two main approaches to the estimation problem applied to small satellites are explained. The algorithms used in this work are described, and the challenges imposed by the state estimation of small satellites are also addressed.

Chapter 4, **Simulation Setup** - PoSAT-1, the main case study, is technically described with attitude estimation in perspective. The attitude sensors availability and efficiency, actuators restrictions and the control model used are also described. The problems faced while implementing the EKF and SVD algorithms are described.

Chapter 5, **Results** - the results obtained from PoSAT-1 simulation with and without a controller in the loop are presented. A statistical analysis of the several tests performed to the simulation of PoSAT-1 and a comparison of performance of the estimator algorithms, EKF and SVD is obtained. Based on magnetometers and attitude estimate from a filter running on-board PoSAT-1 satellite, the EKF presented in this work is applied and comments made concerning the problems associated to the adjustments made to the initial KF formulation.

Chapter 6, **Conclusions and Future Work** - some conclusions are drawn, as well as remarks for future work.

Chapter 2

Spacecraft Attitude Models

The orientation of a spacecraft must be known *w.r.t.* the Earth, the Sun or the stars. Similarly, the sensor measurements, the local magnetic field, and the local Sun model must have its orientation vector described *w.r.t.* a reference frame. The Coordinate Systems relevant to this work as well as the parameterization of its relations are defined in Section 2.1. The fundamental quantity that represents the attitude of a spacecraft is the Direction Cosine Matrix (DCM), but there are other parameterizations used in aerospace applications as well, described in Section 2.2. For all those cases, the parameters used are related to the elements of the DCM. Each representation has its advantages/disadvantages. An analysis of those advantages/disadvantages is performed in Section 2.3 and the most commonly used parameterizations for the attitude determination problem is inferred.

Beginning with Newton's second law, the dynamic equations of motion for a spinning satellite *w.r.t* a fixed coordinate system is briefly obtained in Section 2.4, followed by an introduction to the kinematic equations.

2.1 Coordinate Systems

Five reference coordinate systems (dextral orthonormal triad) are used throughout this work. The most important ones are centered on the spacecraft and the others express the satellite orientation *w.r.t.* the Earth. Two of those are considered to be equal, control and body CS, since the only difference is in their axes of inertia, which is not relevant for the purpose of this work.

Orbital Coordinate System (OCS) $\{\hat{\mathbf{i}}_o, \hat{\mathbf{j}}_o, \hat{\mathbf{k}}_o\}$ - its origin is placed in the mass center of the spacecraft and it is attached to spacecraft orbit around the Earth. The $\hat{\mathbf{k}}_o$ is pointing zenith, $\hat{\mathbf{j}}_o$ is tangent to the orbit opposite to the orbital velocity and $\hat{\mathbf{i}}_o$ is orthogonal to the plane of orbit.

Body Coordinate System (BCS) $\{\hat{\mathbf{i}}_b, \hat{\mathbf{j}}_b, \hat{\mathbf{k}}_b\}$ - its origin is placed in the center

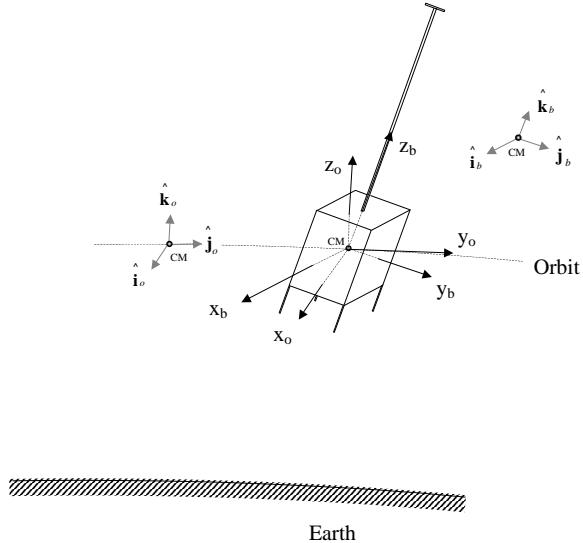


Figure 2.1: Orbital Coordinate System (OCS) and Body Coordinate System (BCS).

mass of the spacecraft and its axes are the principal axes of inertia. This referential frame is attached to the body of the satellite, rotating around the axis with the smallest moment of inertia, $\hat{\mathbf{k}}_b$, $\hat{\mathbf{i}}_b$ and $\hat{\mathbf{j}}_b$ are aligned with the two remaining principal moments of inertia of the satellite in order to form a right-handed triad. This is the reference CS for attitude measurements and magnetorquers.

Control Coordinate System (CCS) $\{\hat{\mathbf{i}}_c, \hat{\mathbf{j}}_c, \hat{\mathbf{k}}_c\}$ - When the coordinate system which origin is the satellite mass center does not include the principal axes of inertia, an additional coordinate system must be considered which is usually denoted as the control CS. For attitude estimation purposes this issue is not taken into account since, for simplicity of the estimator algorithm, the approximation of considering the principal axes of inertia along the body CS is made, in order to increase the computational efficiency of the algorithm. So in this work, the CCS and the BCS are considered to be the same.

Inertial Coordinate System (ICS) $\{\hat{\mathbf{i}}_i, \hat{\mathbf{j}}_i, \hat{\mathbf{k}}_i\}$ - its origin is placed on the Earth's mass centre and it does not rotate with the Earth. The $\hat{\mathbf{k}}_i$ is along the Earth spin vector and points from South to North. $\hat{\mathbf{i}}_i$ and $\hat{\mathbf{j}}_i$ form a plane parallel to the Earth's equatorial plane, where $\hat{\mathbf{i}}_i$ is along the vernal equinox and $\hat{\mathbf{j}}_i$ complements this right-handed triad.

Earth-Centered Earth-Fixed (ECEF) Coordinate System $\{\hat{\mathbf{i}}_t, \hat{\mathbf{j}}_t, \hat{\mathbf{k}}_t\}$ - its origin is placed on the Earth's mass centre rotating with Earth, and it coincides with

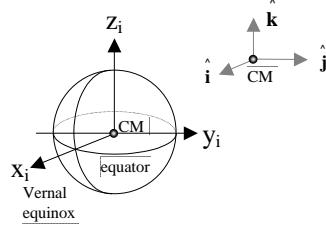


Figure 2.2: Inertial Coordinate System.

ICS on the vernal equinox. However, Earth's spin axis is not inertial but it rotates about the ecliptic pole, a movement known as the *precession of the equinoxes*, slow enough to have negligible effects on small spacecraft's.

2.2 Rotation Vector and Frame

There are several methods to represent the rotation between two coordinate systems:

- *Direction Cosine Matrix*
- *Euler Angle/Axis*
- *Euler Angles: roll, pitch, yaw*
- *Euler Symmetric Parameters or Quaternions*
- *Gibbs Vector*

The most general representation of a rotation between two triads is the DCM. It is also used to map vectors from one CS to another. Since the DCM is difficult to handle, different approaches to attitude vectors representation arise in spacecraft, and are used to parameterize the DCM. This is discussed in section 2.3.

2.2.1 Direction Cosine Matrix

Let \mathfrak{F}_d and \mathfrak{F}_b be two CS (dextral orthonormal triad) whose constituent vectors are respectively $\hat{\mathbf{d}}_1, \hat{\mathbf{d}}_2, \hat{\mathbf{d}}_3$, and $\hat{\mathbf{b}}_1, \hat{\mathbf{b}}_2, \hat{\mathbf{b}}_3$. The direction cosine between $\hat{\mathbf{b}}_i$ and $\hat{\mathbf{b}}_j$ is A_{ij} , (see Fig 2.3).

The rotation between the two triads \mathfrak{F}_b and \mathfrak{F}_d can be written in terms of the cosine of the angle between each of the positive \mathfrak{F}_b constituents and each of the positive \mathfrak{F}_d constituents.

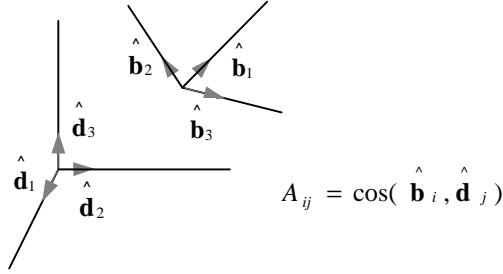


Figure 2.3: Direction Cosines angle between triad \mathfrak{F}_b and \mathfrak{F}_d .

$$\begin{cases} \hat{\mathbf{b}}_1 = \cos(\hat{\mathbf{b}}_1, \hat{\mathbf{d}}_1)\hat{\mathbf{d}}_1 + \cos(\hat{\mathbf{b}}_1, \hat{\mathbf{d}}_2)\hat{\mathbf{d}}_2 + \cos(\hat{\mathbf{b}}_1, \hat{\mathbf{d}}_3)\hat{\mathbf{d}}_3 \\ \hat{\mathbf{b}}_2 = \cos(\hat{\mathbf{b}}_2, \hat{\mathbf{d}}_1)\hat{\mathbf{d}}_1 + \cos(\hat{\mathbf{b}}_2, \hat{\mathbf{d}}_2)\hat{\mathbf{d}}_2 + \cos(\hat{\mathbf{b}}_2, \hat{\mathbf{d}}_3)\hat{\mathbf{d}}_3 \\ \hat{\mathbf{b}}_3 = \cos(\hat{\mathbf{b}}_3, \hat{\mathbf{d}}_1)\hat{\mathbf{d}}_1 + \cos(\hat{\mathbf{b}}_3, \hat{\mathbf{d}}_2)\hat{\mathbf{d}}_2 + \cos(\hat{\mathbf{b}}_3, \hat{\mathbf{d}}_3)\hat{\mathbf{d}}_3 \end{cases}$$

Since the nine cosines have different values let us express in terms of A_{ij} .

$$\begin{cases} \hat{\mathbf{b}}_1 = A_{11}\hat{\mathbf{d}}_1 + A_{12}\hat{\mathbf{d}}_2 + A_{13}\hat{\mathbf{d}}_3 \\ \hat{\mathbf{b}}_2 = A_{12}\hat{\mathbf{d}}_1 + A_{22}\hat{\mathbf{d}}_2 + A_{23}\hat{\mathbf{d}}_3 \\ \hat{\mathbf{b}}_3 = A_{13}\hat{\mathbf{d}}_1 + A_{32}\hat{\mathbf{d}}_2 + A_{33}\hat{\mathbf{d}}_3 \end{cases}$$

The a_{ij} are the elements of the *Direction Cosine Matrix* (DCM) A :

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{12} & A_{22} & A_{23} \\ A_{13} & A_{32} & A_{33} \end{bmatrix} \quad (2.1)$$

This rotation matrix describes the triad \mathfrak{F}_b in terms of the triad \mathfrak{F}_d , completely specifying the relative orientation between the two coordinate systems.

$$\mathfrak{F}_b = A_d^b \mathfrak{F}_d \quad (2.2)$$

So, the orientation of a vector \mathbf{r} , expressed in terms of the triad \mathfrak{F}_d , $\mathbf{r} = r_1\hat{\mathbf{d}}_1 + r_2\hat{\mathbf{d}}_2 + r_3\hat{\mathbf{d}}_3$, can be expressed in terms of triad \mathfrak{F}_b through the Attitude matrix,

$$\mathbf{r}^b = A_d^b \mathbf{r}^d \quad (2.3)$$

Direction Cosine Matrix Properties

Since the DCM represents the rotation between two orthonormal frames the elements of the DCM must satisfy the following conditions:

$$(A_d^b)^T A_d^b = I \text{ and } (A_d^b)^T = (A_d^b)^{-1} \quad (2.4)$$

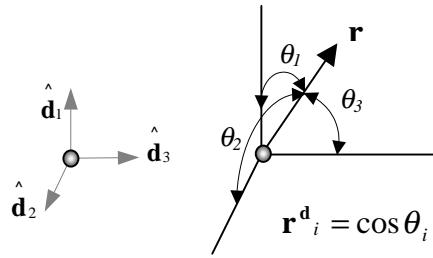


Figure 2.4: The orientation of a vector \mathbf{r} , expressed in terms of the triad \mathfrak{F}_d .

These properties are very useful, especially to map a vector \mathbf{r} , from a triad \mathfrak{F}_b to a triad \mathfrak{F}_d

$$\mathbf{r}^d = (A_d^b)^T \mathbf{r}^b = A_d^b \mathbf{r}^b \quad (2.5)$$

Also verified is

$$\det((A_d^b)^T A_d^b) = \det((A_d^b)^T) \det(A_d^b) = \det^2(A_d^b) = 1 \quad (2.6)$$

hence, $\det(A_d^b) = \pm 1$. To preserve the length of vectors and the angles between them, the determinant must be 1.

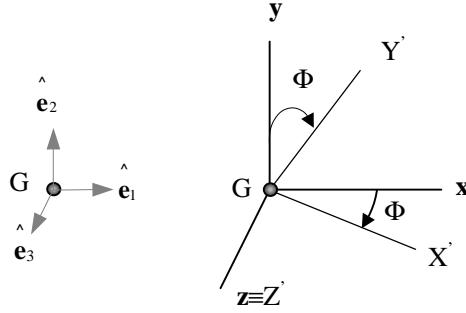
$$\det(A_d^b) = 1 \quad (2.7)$$

The orthonormal group of matrices that satisfies the previous relations is a special orthogonal Lie group of dimension $n(n-1)/2$, $SO(n)$. The attitude matrix A , belongs to the especial orthogonal group of dimension 3.

2.2.2 Euler Axis/Angle

One usual representation of rotations, due to its clear physical interpretation, is the *Euler axis/angle*. Using Euler's theorem - “*The most general displacement of a rigid body with one point fixed is equivalent to a single rotation of an angle about some axis through that point*” - and considering the angle as Φ and the unit vector $\hat{\mathbf{e}} = [\hat{\mathbf{e}}_1 \ \hat{\mathbf{e}}_2 \ \hat{\mathbf{e}}_3]^T$ about the \mathbf{z} axis of rotation (see Fig. 2.5), the DCM can be obtained for this rotation, as

$$A_z(\Phi) = \begin{bmatrix} \cos \Phi & \sin \Phi & 0 \\ -\sin \Phi & \cos \Phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$


 Figure 2.5: Angle/axis rotation about \mathbf{z} .

Similarly, the rotation matrices for rotations about the x and y axis are,

$$A_y(\Phi) = \begin{bmatrix} \cos \Phi & 0 & -\sin \Phi \\ 0 & 1 & 0 \\ \sin \Phi & 0 & \cos \Phi \end{bmatrix} \quad (2.9)$$

$$A_x(\Phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \Phi & \sin \Phi \\ 0 & -\sin \Phi & \cos \Phi \end{bmatrix} \quad (2.10)$$

where all matrices have their trace equal to,

$$\text{trace}(A_x) = \text{trace}(A_y) = \text{trace}(A_z) = 1 + 2 \cos \Phi \quad (2.11)$$

For a general axis $\hat{\mathbf{e}}$, $\hat{\mathbf{e}} = (e_1, e_2, e_3)^T$, the direction cosine matrix may be expressed in terms of the Euler axis/angle by,

$$A(\Phi, \hat{\mathbf{e}}) = \begin{bmatrix} \cos \Phi + e_1^2(1-\cos \Phi) & e_1 e_2(1-\cos \Phi) + e_3 \sin \Phi & e_1 e_3(1-\cos \Phi) - e_2 \sin \Phi \\ e_1 e_2(1-\cos \Phi) - e_3 \sin \Phi & \cos \Phi + e_2^2(1-\cos \Phi) & e_2 e_3(1-\cos \Phi) + e_1 \sin \Phi \\ e_1 e_3(1-\cos \Phi) + e_2 \sin \Phi & e_2 e_3(1-\cos \Phi) - e_1 \sin \Phi & \cos \Phi + e_3^2(1-\cos \Phi) \end{bmatrix} \quad (2.12)$$

$$= 1_{3 \times 3} \cos \Phi + (1 - \cos \Phi) \hat{\mathbf{e}} \hat{\mathbf{e}}^T - \sin \Phi [\hat{\mathbf{e}} \times] \quad (2.13)$$

where $[\hat{\mathbf{e}} \times]$ is defined by (2.31).

If $\sin \Phi \neq 0$, then the unit vector $\hat{\mathbf{e}}$ will be given by the following expression,

$$\begin{aligned} e_1 &= \frac{A_{23} - A_{32}}{2 \sin \Phi} \\ e_2 &= \frac{A_{31} - A_{13}}{2 \sin \Phi} \\ e_3 &= \frac{A_{12} - A_{21}}{2 \sin \Phi} \end{aligned} \quad (2.14)$$

Then, the Euler angle Φ and the unit vector $\hat{\mathbf{e}}$ can be obtained from the actual attitude matrix, Eq. (2.11) and (2.14).

2.2.3 Euler Angles $\text{roll}(\psi)$, $\text{pitch}(\theta)$, $\text{yaw}(\phi)$

Another set of parameters by which the orientation of a rigid body can be specified are the Euler Angles. Given two CS, a rotating xyz and a fixed XYZ system, the Euler angles give the orientation of xyz system relative to the fixed XYZ system. There are different sequences of Euler angles rotation. Only the sequence adopted in this dissertation is described here. The three rotations are presented in Fig. 2.6.

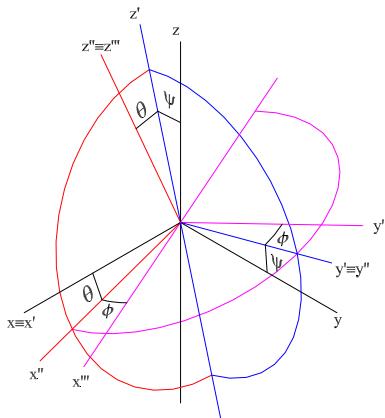


Figure 2.6: The Euler angle 1 – 2 – 3 rotation or Roll - ψ , pitch - θ and yaw - ϕ .

1. A positive rotation Ψ about the x axis, resulting in the $x'y'z'$ coordinate systems where $x' \equiv x$.
2. A positive rotation θ angle about the y' axis, resulting in the $x''y''z''$ coordinate systems where $y'' \equiv y'$.
3. A positive rotation ϕ about the z'' axis, resulting in the final $x'''y'''z'''$ system where $z''' \equiv z''$

This order of rotation is often referred as the roll (ψ), pitch (θ), yaw (ϕ) sequence and is related to the xyz rotation or 123 sequence.

Each rotation can be expressed in terms of the Euler Angle/Axis presented in section 2.2.2. So the roll, pitch, yaw sequence is the product of the three matrices from Eqs.(2.8) - (2.10), $A_x(\psi)$ $A_y(\theta)$ $A_z(\phi)$ where the Φ angle is different for each individual rotation. The final product matrix is,

$$A_{123}(\psi, \theta, \phi) = \begin{bmatrix} \cos \theta \cos \phi & \cos \phi \sin \theta \sin \psi + \sin \phi \cos \psi & -\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ -\sin \phi \cos \theta & -\sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \sin \phi \sin \theta \cos \psi + \cos \phi \sin \psi \\ \sin \theta & -\cos \theta \sin \psi & \cos \theta \cos \psi \end{bmatrix} \quad (2.15)$$

Any sequence of rotations can be expressed by a single rotation about the vector $\hat{\mathbf{e}}$ through an angle Φ .

2.2.4 Gibbs Vector

Using the Φ angle and the unit vector $\hat{\mathbf{e}}$ defined for Euler's Theorem in section 2.2.2, another attitude vector representation frequently used in a spacecraft, is defined by

$$\begin{aligned} g_1 &\equiv e_1 \tan \frac{\Phi}{2} \\ g_2 &\equiv e_2 \tan \frac{\Phi}{2} \\ g_3 &\equiv e_3 \tan \frac{\Phi}{2} \end{aligned} \quad (2.16)$$

The DCM is given in terms of the Gibbs vector by

$$A(\mathbf{g}) = \frac{1}{1 + g_1^2 + g_2^2 + g_3^2} \begin{bmatrix} 1 + g_1^2 + g_2^2 + g_3^2 & 2(g_1g_2 + g_3) & 2(g_1g_3 - g_2) \\ 2(g_1g_2 - g_3) & 1 - g_1^2 + g_2^2 - g_3^2 & 2(g_2g_3 + g_1) \\ 2(g_1g_3 + g_2) & 2(g_2g_3 - g_1) & 1 - g_1^2 - g_2^2 + g_3^2 \end{bmatrix} \quad (2.17)$$

The Gibbs' vector or the Rodrigues' parameters provide a computational advantage compared to the other representations, since it is a minimal parameterization not involving trigonometric functions as in the Euler angles representation.

2.2.5 Euler Symmetric Parameters - Quaternions

Quaternions were first introduced by Hamilton in 1843 and belong to the especial orthogonal Lie group $SO(3)$, [43] which can be identified with rotations in space. They are composed of a scalar part q_4 , and a vector part of three unit vectors, q_1 , q_2 and $q_3 \in \Re^3$. In particular, the **unit quaternions** form a unit sphere in \Re^4 .

$$\| q_1, q_2, q_3, q_4 \| = 1$$

Properties

A quaternion is represented by a **scalar** part and a **vector** part,

$$\mathbf{q} = q_s + \mathbf{q}_v = (q_s, \mathbf{q}_v) = q_4 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} \quad (2.18)$$

The quaternions represent a sphere in \Re^4 , where the radius is the module of the quaternion. In particular, the unit quaternions satisfy the following constraint,

$$q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1 \quad (2.19)$$

where $q_4, q_1, q_2, q_3 \in \Re$, and i, j, k satisfy the conditions

$$\begin{aligned} i^2 &= j^2 = k^2 = -1 \\ ij &= -ji = k \\ jk &= -kj = i \\ ki &= -ik = j \end{aligned} \tag{2.20}$$

Quaternions having zero scalar part are called **pure quaternions**:

$$q_4 = 0 \implies \mathbf{q} = 0 + \mathbf{q}_v = 0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}.$$

The **conjugate of a quaternion q** , denoted by \mathbf{q}^* , similarly to the inverse of the complex numbers, is defined by negating its vector part,

$$\mathbf{q}^* = q_s - \mathbf{q}_v = q_4 - q_1\mathbf{i} - q_2\mathbf{j} - q_3\mathbf{k} \tag{2.21}$$

The **inverse of a quaternion q** , denoted by \mathbf{q}^{-1} , is also obtained similarly to complex numbers,

$$\mathbf{q}^{-1} = \frac{\mathbf{q}^*}{\mathbf{q}\mathbf{q}^*} = \frac{q_s}{q_1^2 + q_2^2 + q_3^2 + q_4^2} - \frac{\mathbf{q}_v}{q_1^2 + q_2^2 + q_3^2 + q_4^2} \tag{2.22}$$

Since $q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1$, from Eq. (2.19), then

$$\mathbf{q}^{-1} = \mathbf{q}^* \tag{2.23}$$

It is convenient to express quaternions in terms of a 4×1 vector, for matrix algebra application, to simplify the manipulation of equations.

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} \mathbf{q} \\ q_4 \end{bmatrix} \tag{2.24}$$

Quaternionic Multiplication

Multiplication and division are new operations defined for quaternions. The product of two quaternions, \mathbf{p} and \mathbf{q} , is performed like polynomial multiplication considering them as hypercomplex numbers,

$$\begin{aligned} \mathbf{pq} &= (p_4 + p_1\mathbf{i} + p_2\mathbf{j} + p_3\mathbf{k})(q_4 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}) = \\ &= (p_s q_s - \mathbf{p}_v \bullet \mathbf{q}_v, p_s \mathbf{q}_v + q_s \mathbf{p}_v + \mathbf{p}_v \times \mathbf{q}_v) \end{aligned} \tag{2.25}$$

where (\bullet) denotes inner product and (\times) denotes cross product.

Taking the above relations into account,

$$\mathbf{pq} = \begin{bmatrix} p_4 & -p_3 & p_2 & p_1 \\ p_3 & p_4 & -p_1 & p_2 \\ -p_2 & p_1 & p_4 & p_3 \\ -p_1 & -p_2 & -p_3 & p_4 \end{bmatrix} \mathbf{q} \text{ or } \mathbf{pq} = \begin{bmatrix} q_4 & q_3 & -q_2 & q_1 \\ -q_3 & q_4 & q_1 & q_2 \\ q_2 & -q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix} \mathbf{p} \quad (2.26)$$

$$\mathbf{pq} \triangleq \Xi(\mathbf{p})\mathbf{q} \text{ or } \mathbf{pq} \triangleq \Omega(\mathbf{q})\mathbf{p} \quad (2.27)$$

where $\Xi(\mathbf{p})$ is the right operator and $\Omega(\mathbf{p})$ is the left operator.

It should be notice that quaternion composition, established by [15], is given by $A(\mathbf{q})A(\mathbf{p}) = A(\mathbf{pq})$. For the sake of simpler notation, it is more convenient to perform a direct multiplication $A(\mathbf{q})A(\mathbf{p}) = A(\mathbf{q} \otimes \mathbf{p})$. Hence,

$$\mathbf{q} \otimes \mathbf{p} = \mathbf{pq} \quad (2.28)$$

So the nomenclature used in this work is,

$$\boxed{\mathbf{q} \otimes \mathbf{p} = \Omega(\mathbf{q})\mathbf{p} = \Xi(\mathbf{p})\mathbf{q}} \quad (2.29)$$

For simplicity reasons the $\Omega(\mathbf{p})$ and $\Xi(\mathbf{p})$ matrices can be redefined as,

$$\boxed{\begin{aligned} \Omega(\mathbf{q}) &= \begin{bmatrix} q_4 \mathbf{1}_{3 \times 3} - [\mathbf{q} \times] & \mathbf{q} \\ -\mathbf{q}^T & q_4 \end{bmatrix} \\ \Xi(\mathbf{p}) &= \begin{bmatrix} p_4 \mathbf{1}_{3 \times 3} + [\mathbf{p} \times] & \mathbf{p} \\ -\mathbf{p}^T & p_4 \end{bmatrix} \end{aligned}} \quad (2.30)$$

where,

$$[\mathbf{b} \times] = \begin{bmatrix} 0 & -b_z & b_y \\ b_z & 0 & -b_x \\ -b_y & b_x & 0 \end{bmatrix} \quad (2.31)$$

is a skew-symmetric matrix representing the cross product of two vectors, (see Fig. 2.7), $\mathbf{c} = [c_x \ c_y \ c_z]^T$ and $\mathbf{b} = [b_x \ b_y \ b_z]^T$,

$$\mathbf{c} \times \mathbf{b} = \begin{bmatrix} c_y b_z - c_z b_y \\ c_z b_x - c_x b_z \\ c_x b_y - c_y b_x \end{bmatrix} = [\mathbf{c} \times] \mathbf{b} \quad (2.32)$$

$$[\mathbf{c} \times] \mathbf{b} = -[\mathbf{b} \times] \mathbf{c} \quad (2.33)$$

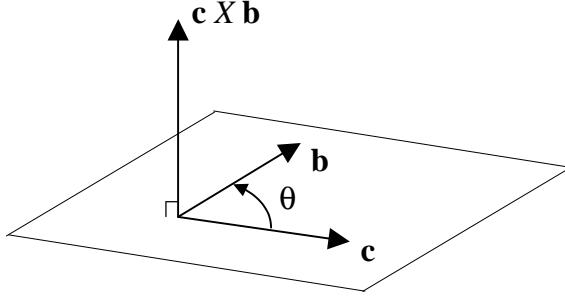


Figure 2.7: Cross product of two vectors.

Attitude Matrix - $A(\mathbf{q})$

Respecting the constraint defined by Eq.(2.19), the quaternions can be defined in terms of the Euler axis/angle by,

$$\begin{aligned} q_1 &\equiv e_1 \sin \frac{\Phi}{2} \\ q_2 &\equiv e_2 \sin \frac{\Phi}{2} \\ q_3 &\equiv e_3 \sin \frac{\Phi}{2} \\ q_4 &\equiv \cos \frac{\Phi}{2} \end{aligned} \tag{2.34}$$

Substituting into Eq.(2.12), the DCM is expressed in terms of the quaternions, denoted as Attitude Matrix, by

$$\begin{aligned} A(\mathbf{q}) &= \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1 q_2 + q_3 q_4) & 2(q_1 q_3 - q_2 q_4) \\ 2(q_1 q_2 - q_3 q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2 q_3 + q_1 q_4) \\ 2(q_1 q_3 + q_2 q_4) & 2(q_2 q_3 - q_1 q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \\ &= (q_4^2 - \|\mathbf{q}\|^2) \mathbf{1}_{3 \times 3} + 2\mathbf{q}\mathbf{q}^T - 2q_4 [\mathbf{q} \times] \end{aligned} \tag{2.35}$$

where $[\mathbf{q} \times]$ is the skew-symmetric matrix defined by Eq.(2.31).

However, there is a less cumbersome way to represent the rotation matrix. In fact, the quaternion multiplication can be used to represent attitude in a given CS.

Given two successive rotations represented by quaternions, the total rotation may be obtained through quaternionic multiplication. Let us consider two successive rotations represented by the attitude matrix $A(\mathbf{q}')$ and $A(\mathbf{q}'')$. The total rotation is given as follows:

$$A(\mathbf{q}') A(\mathbf{q}'') = A(\mathbf{q}_t) \tag{2.36}$$

This is equivalent to performing a direct quaternion multiplication,

$$A(\mathbf{q}')A(\mathbf{q}'') = A(\mathbf{q}' \otimes \mathbf{q}'') \quad (2.37)$$

Therefore,

$$\mathbf{q}_t = \mathbf{q}' \otimes \mathbf{q}'' \quad (2.38)$$

and from Eq.(2.29),

$$\begin{aligned} \mathbf{q}_t &= \begin{bmatrix} q'_4 & q'_3 & -q'_2 & q'_1 \\ -q'_3 & q'_4 & q'_1 & q'_2 \\ q'_2 & -q'_1 & q'_4 & q'_3 \\ -q'_1 & -q'_2 & -q'_3 & q'_4 \end{bmatrix} \mathbf{q}'' \\ &= \Omega(\mathbf{q}')\mathbf{q}'' \end{aligned} \quad (2.39)$$

This transformation matrix is easier to compute than the DCM since it only requires four parameters \mathbf{q}'_i , against the nine parameters A_{ij} , of the DCM, and does not involve trigonometric functions as in the DCM definition, Eq.(2.1).

So the quaternion multiplication provides a simple method to compute the total transformation from body to orbit to inertial CS given the quaternion representing the respective rotation,

$$\mathbf{q}_i^b = \mathbf{q}_o^b \otimes \mathbf{q}_i^o \quad (2.40)$$

$$= \Omega(\mathbf{q}_o^b)\mathbf{q}_i^o \quad (2.41)$$

$$= \Xi(\mathbf{q}_i^o)\mathbf{q}_o^b \quad (2.42)$$

Recalling the DCM definition in Section 2.2.1 and the CS of interest defined in Section 2.1, the rotation matrix that describes the orbit CS in terms of the body CS is given as

$$A(\mathbf{q}_o^b) = \begin{bmatrix} \hat{\mathbf{i}}_o^b & \hat{\mathbf{j}}_o^b & \hat{\mathbf{k}}_o^b \end{bmatrix} \quad (2.43)$$

where $\hat{\mathbf{i}}_o^b$, $\hat{\mathbf{j}}_o^b$, $\hat{\mathbf{k}}_o^b$, are the constituent vectors of the x , y and z -axes of the orbit CS, projected along the body CS. These unit vectors $\hat{\mathbf{i}}_o^b$, $\hat{\mathbf{j}}_o^b$, $\hat{\mathbf{k}}_o^b$, may be parameterized by the quaternions,

$$\begin{aligned} \hat{\mathbf{i}}_o^b &= [q_1^2 - q_2^2 - q_3^2 + q_4^2 \quad 2(q_1q_2 - q_3q_4) \quad 2(q_1q_3 + q_2q_4)]^T \\ \hat{\mathbf{j}}_o^b &= [2(q_1q_2 + q_3q_4) \quad -q_1^2 + q_2^2 - q_3^2 + q_4^2 \quad 2(q_2q_3 - q_1q_4)]^T \\ \hat{\mathbf{k}}_o^b &= [2(q_1q_3 - q_2q_4) \quad 2(q_2q_3 + q_1q_4) \quad -q_1^2 - q_2^2 + q_3^2 + q_4^2]^T \end{aligned} \quad (2.44)$$

2.3 Attitude Parameterization Analysis

Several ways to parameterize the attitude of a spacecraft were presented in the previous section. A more complete survey of attitude representations can be found in [41] or [55]. The most general representation of the orientation of a rigid body is given by the DCM, (2.1). As referred before, this representation is not very used since the orthogonality is difficult to maintain and the methods that provide optimal orthogonalization require computationally expensive matrices. Also, the redundancy of its nine parameters a_{ij} , turn it into an unfriendly representation. So, other ways to represent rotations arised, such as the *Euler Axis/Angle*, the *roll pitch* and *yaw Euler* angles, the *Gibbs* vector and *quaternions*. Among them, the quaternions are widely applied in attitude parameterization. This representation offers several advantages when compared to the others: it is not singular for any angle, as opposed to the Euler axis/angle, Gibbs vector or roll, pitch and yaw Euler angle representations. Another reason is the fact that the kinematic equations are linear and the computation of the associated matrix involves only algebra expressions, while the Euler angles representation involves nonlinear computations, due to the use of trigonometric functions, leading to additional computational burden. Moreover, the attitude matrix expressed in terms of quaternions can be easily computed by (2.35), while the Euler angles or Axis/Angle again involve trigonometric functions ((2.15) or (2.12)), and also the product of rotations is relatively simple since it is expressed in terms of quaternion multiplication.

However, the quaternions representation has the problem of not being a minimal representation (*i.e.* with 3 parameters only), which is a drawback in quaternions, leading to difficulties in maintaining the rotation matrix orthogonality. This issue has to be addressed for filtering algorithms, increasing the associated computational load. Also it has no clear physical interpretation, in opposition to the *Euler angle* representation. So, usually the quaternions are used in the ADS and then transformed into *Euler angle* for results analysis as well as in this dissertation. In this work, whenever the attitude matrix is referred, it means the DCM parameterized by the quaternions.

2.4 Spacecraft Equations of Motion

Mechanics has two main points of view: Lagrangian and Hamilton. The Lagrangian approach is based on variational principles and it is what generalizes most directly to the relativist content. The Hamilton mechanics is based directly on the energy concept and is more closely tied to quantum mechanics.

A short overview of the dynamics of a spinning spacecraft, *w.r.t.* a fix CS, is presented in the next section, based on the Lagragian approach.

2.4.1 Dynamic Equations

Dynamics is the study of the motions of a particle or a system of particles due to external forces acting in these systems. A spacecraft, and in this particular case a satellite, is a moving body in space. Therefore, it is a body animated with a certain velocity, ruled by the laws of Dynamics. The satellite is considered a rigid body since it is assumed that the distance between any two of its elements of mass remains fixed, not deformable or flexible among them but rigidly connected.

Newton's second law for the translational motion $\sum \mathbf{F} = m\mathbf{a}$ states that the sum of the external forces $\sum \mathbf{F}$ acting on a system is equal to the total mass m of the system times the absolute acceleration \mathbf{a} of its center of mass.

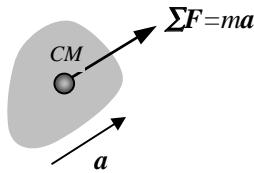


Figure 2.8: A rigid body with mass m has an acceleration \mathbf{a} due to external force \mathbf{F} .

Replacing the acceleration a by the derivative of the velocity, $\frac{d\mathbf{v}}{dt}$, Newton second law becomes

$$\sum \mathbf{F} = m \frac{d\mathbf{v}}{dt} \quad (2.45)$$

or

$$\sum \mathbf{F} = \frac{d}{dt} (m\mathbf{v}) \quad (2.46)$$

The vector $m\mathbf{v}$ is called the *linear impulse or momentum L*

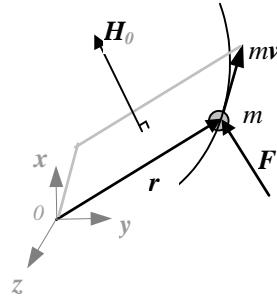
$$\boxed{\mathbf{L} = m\mathbf{v}} \quad (2.47)$$

Differentiating the linear impulse *w.r.t.* time t , and using the result in (2.45), Newton's second law becomes,

$$\sum \mathbf{F} = \dot{\mathbf{L}} \quad (2.48)$$

If there are no external forces, then the total momentum applied to the system of particles is constant,

$$\boxed{\text{Principle of conservation of linear momentum, } \sum \mathbf{F} = 0 \Rightarrow \mathbf{L} = \text{constant}} \quad (2.49)$$


 Figure 2.9: *Angular momentum.*

Considering a particle of mass m moving in a plane under the action of an external force \mathbf{F} (see Fig. 2.9), the external product between the position vector of a particle with mass m and the linear momentum *w.r.t.* a fixed reference point $\{o\}$, is called the *moment of the linear momentum* or *angular momentum* H ,

$$\mathbf{H} = \mathbf{r} \times m\mathbf{v} \quad (2.50)$$

$$= \mathbf{r} \times m\dot{\mathbf{r}} \quad (2.51)$$

Differentiating the angular momentum *w.r.t.* time, the angular momentum becomes,

$$\dot{\mathbf{H}} = \mathbf{r} \times m\ddot{\mathbf{r}} + \dot{\mathbf{r}} \times m\dot{\mathbf{r}} = \mathbf{r} \times m\ddot{\mathbf{r}} \quad (2.52)$$

Since $\sum \mathbf{F} = m\ddot{\mathbf{r}}$, $\dot{\mathbf{H}} = \mathbf{r} \times \sum \mathbf{F}$. As the *moment* \mathbf{M} of the force \mathbf{F} about the fixed point $\{o\}$ is the result of the vector cross product between that force and the position vector \mathbf{r} where the force is applied, the equivalent law for rotational motion is obtained,

$$\sum \mathbf{M} = \dot{\mathbf{H}} \quad (2.53)$$

and it states that the total moment $\sum \mathbf{M}$ of the external forces acting on the system is equal to the time rate of change of the angular momentum \mathbf{H} .

If the total of the external forces are parallel to the position vector \mathbf{r} then the angular momentum applied to the system of particles is constant,

Principle of conservation of angular momentum,
 $\sum \mathbf{M} = 0 \Rightarrow \mathbf{H} = \text{constant}$

(2.54)

For a body in a tridimensional plane rotating around a fixed axis AA' , the velocity is the cross product between the angular velocity $\boldsymbol{\omega}$ and the position of the body *w.r.t.* the frame, $\mathbf{v} = \frac{d\mathbf{r}}{dt} = \boldsymbol{\omega} \times \mathbf{r}$ (see Fig. 2.10).

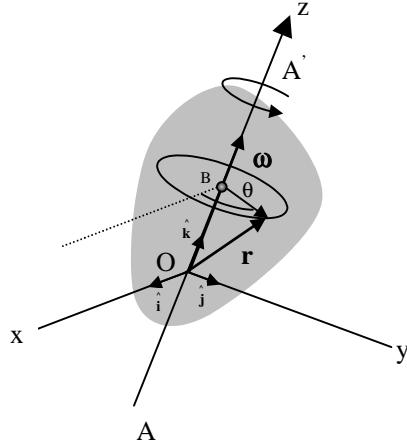


Figure 2.10: Rate of change of a rotating vector.

The angular velocity of the body is a vector parallel to the axis of rotation of the body, $\omega = \frac{d\theta}{dt}\hat{k}$. The angular momentum becomes,

$$\mathbf{H}^o = \mathbf{r} \times m(\omega \times \mathbf{r}) \quad (2.55)$$

Since in (2.55) \mathbf{H}^o is a vector with the same direction of ω with the module equal to $\sum mr^2$, and recalling that the *moment of inertia* $I = \sum mr^2$, the angular momentum becomes,

$$\mathbf{H}^o = I\omega \quad (2.56)$$

Using this result and Eq.(2.53), Newton's law for rotational motion becomes,

$$\sum \mathbf{M}^o = \dot{\mathbf{H}}^o = I\dot{\omega} \quad (2.57)$$

Consider now that there is a referential frame $\{o_{xyz}\}$ rotating with the body, both the $X'Y'Z$ and XYZ CS in Fig. 2.11 are inertial and the origins of $X'Y'Z$ and xyz are coincident. ω is the angular velocity of the rigid body *w.r.t.* the frame xyz . Then the absolute rate of change of the angular moment $\dot{\mathbf{H}}$ *w.r.t.* the frame $X'Y'Z$ is,

$$\dot{\mathbf{H}}^{o_{x'y'z'}} = \dot{\mathbf{H}}^{oxyz} + \Omega \times \mathbf{H}^o \quad (2.58)$$

and Ω is the absolute angular velocity of the rotating frame $\{o_{xyz}\}$ as viewed from the frame $\{G\}$. Since the frame

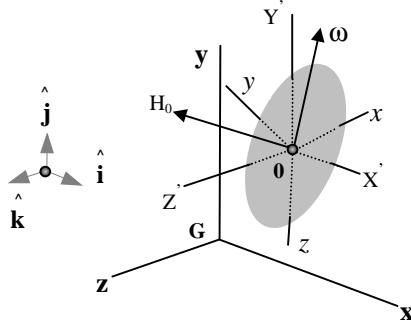


Figure 2.11: Rate of change of the angular momentum of a rotating rigid body.

$\{0_{xyz}\}$ rotates with the rigid body then

$$\Omega = \omega \quad (2.59)$$

Using (2.57) and the CS defined in the previous section the dynamic rotational equation of motion for an orbiting satellite is given by

$$\sum \mathbf{M}^G = \dot{\mathbf{H}}^{o_{xyz}} + \Omega \times \mathbf{H}^G \quad (2.60)$$

Expressed in the coordinate axes xyz , defined before as the principal axes at the center of mass of the rigid body, the products of inertia are all zero, $I_{xy} = I_{yx} = I_{yz} = I_{zy} = I_{xz} = I_{zx} = 0$ and the angular momentum \mathbf{H} is parallel to the angular velocity ω . Using (2.56), (2.60) reduces to

$$\begin{aligned} \sum M_x &= I_{xx}\dot{\omega}_x + (I_{zz} - I_{yy})\omega_y\omega_z \\ \sum M_y &= I_{yy}\dot{\omega}_y + (I_{xx} - I_{zz})\omega_z\omega_x \\ \sum M_z &= I_{zz}\dot{\omega}_z + (I_{yy} - I_{xx})\omega_x\omega_y \end{aligned} \quad (2.61)$$

known as the *Euler's equations of motion*.

Solving these equations in order to the rate of change of the angular velocity, the dynamic equation of motion becomes,

$$\boxed{\begin{aligned} \dot{\omega}_x &= I_{xx}^{-1} (\sum M_x - (I_{zz} - I_{yy})\omega_y\omega_z) \\ \dot{\omega}_y &= I_{yy}^{-1} (\sum M_y - (I_{xx} - I_{zz})\omega_z\omega_x) \\ \dot{\omega}_z &= I_{zz}^{-1} (\sum M_z - (I_{yy} - I_{xx})\omega_x\omega_y) \end{aligned}} \quad (2.62)$$

Considering the CS defined for this work in Section 2.1, *i.e.*, the angular velocity of body CS *w.r.t.* the inertial CS expressed in body CS, is expressed as ω_{bi}^b .

2.4.2 Kinematic Equations

Kinematics studies the motions of a particle or a system of particles, relating position, velocity, acceleration and time, disregarding the external forces that origins the motion.

So, given the angular velocity expressed as a function of the external moments, which are given by the dynamic equations, there is the need to relate the angular velocity with the derivative of the rotation vector. Since the attitude is parameterized by quaternions, presented in Section 2.3, the kinematic equation gives the mathematical relation between quaternions and angular velocity, [55],

$$\frac{d}{dt} \mathbf{q}_o^b(t) = \frac{1}{2} \Omega(\boldsymbol{\omega}_{bo}^b) \mathbf{q}_o^b \quad (2.63)$$

with,

$$\Omega(\boldsymbol{\omega}_{bo}^b) = \frac{1}{2} \begin{bmatrix} 0 & \omega_{z,bo}^b & -\omega_{y,bo}^b & \omega_{x,bo}^b \\ -\omega_{z,bo}^b & 0 & \omega_{x,bo}^b & \omega_{y,bo}^b \\ \omega_{y,bo}^b & -\omega_{x,bo}^b & 0 & \omega_{z,bo}^b \\ -\omega_{x,bo}^b & -\omega_{y,bo}^b & -\omega_{z,bo}^b & 0 \end{bmatrix} \quad (2.64)$$

where the angular rates components used are body CS *w.r.t.* orbit CS $\boldsymbol{\omega}_{bo}^b$, because the kinematic equations describe the rotation between the orbital axes and the satellite axes. As such, one must relate the orbital and the inertial references,

$$\boldsymbol{\omega}_{bi}^b = \boldsymbol{\omega}_{bo}^b + \boldsymbol{\omega}_{oi}^b \quad (2.65)$$

$$\boldsymbol{\omega}_{bo}^b = \boldsymbol{\omega}_{bi}^b - \boldsymbol{\omega}_{oi}^b \quad (2.66)$$

The vector $\boldsymbol{\omega}_{oi}^o = [\omega_0 \ 0 \ 0]^T$ is the angular velocity of the orbit CS *w.r.t.* inertial CS (see Fig 2.1) written in orbit CS. Hence, the unknown $\boldsymbol{\omega}_{oi}^o$ can be written in body CS through the transformation matrix $A_o^b(\mathbf{q})$, (2.43),

$$\boldsymbol{\omega}_{oi}^o = A_o^b(\mathbf{q}) \boldsymbol{\omega}_{oi}^o = \omega_0 \hat{\mathbf{i}}_o^b \quad (2.67)$$

Hence, the angular velocity used in the kinematic equations is computed according to,

$$\boldsymbol{\omega}_{bo}^b = \boldsymbol{\omega}_{bi}^b - \omega_0 \hat{\mathbf{i}}_o^b \quad (2.68)$$

Note that representing the angular velocity in terms of a quaternion, $\mathbf{w} = [\omega_x \ \omega_y \ \omega_z \ 0]^T$, the matrix $\Omega(\mathbf{w}_{bo}^b)$ from (2.64) is coincident with the left operator $\Omega(\mathbf{q})$ of a quaternionic multiplication defined in (2.30). Thus the kinematic equation (2.63) can be written as follows,

$$\frac{d}{dt} \mathbf{q}(t) = \frac{1}{2} \mathbf{w} \otimes \mathbf{q} \quad (2.69)$$

2.4.3 Small Satellite External Torques

There are several external moments, $\sum \mathbf{M}$, applied to the satellite body that will influence its dynamic movement (see Eq.(2.62)). The external torques acting on the spacecraft are: $\sum \mathbf{M} = \mathbf{N}_{dt} + \mathbf{N}_{gg} + \mathbf{N}_{ctrl} + \mathbf{N}_{wheel}$.

Gravity Gradient Torque, \mathbf{N}_{gg}

For satellites at low orbit the main contribution to the external torques is the gravity gradient torque, [55]

$$\mathbf{N}_{gg} = 3 \frac{\mu}{R_s^3} \left(\hat{\mathbf{k}}_b \times I \hat{\mathbf{k}}_b \right) \quad (2.70)$$

where μ is the Earth's gravitational constant; R_s is the position vector of the satellite mass center *w.r.t.* to the Earth center; $I = [I_{xx} \ I_{yy} \ I_{zz}]^T$ are the principal moments of inertia. Since the elliptic orbit of the satellite is simplified to circular for eccentricities close to zero, the R_s becomes the mean distance of the satellite to Earth center $R_s \approx R_{CM}$. The $\frac{\mu}{R_{CM}^3}$ element can be written as the square of the orbit angular rate,

$$\frac{\mu}{R^3} \approx \omega_0^2 \quad (2.71)$$

Since the DCM can be defined in terms of the unit vectors, $\hat{\mathbf{i}}_{ob}, \hat{\mathbf{j}}_{ob}, \hat{\mathbf{k}}_{ob}$, (2.44) of the x_b, y_b and z_b axes, and $A_o^b = [\hat{\mathbf{i}}_o^b \ \hat{\mathbf{j}}_o^b \ \hat{\mathbf{k}}_o^b]$, then the gravity gradient becomes,

$$\mathbf{N}_{gg} = 3\omega_0^2 \begin{bmatrix} (I_z - I_y)A_{23}(\mathbf{q})A_{33}(\mathbf{q}) \\ (I_x - I_z)A_{33}(\mathbf{q})A_{13}(\mathbf{q}) \\ (I_y - I_x)A_{13}(\mathbf{q})A_{23}(\mathbf{q}) \end{bmatrix} \quad (2.72)$$

Disturbance Torque, \mathbf{N}_{dt}

The \mathbf{N}_{dt} torque is the disturbance moment written in control CS due to aerodynamic drag and solar pressure, eccentricity of the orbit and other effects. The aerodynamic torque is very important for spacecraft orbiting below 400 km [55]. Recent studies were carried out [1] analysing the importance of these several types of contributions in the disturbance moment compared with the gravity gradient moment.

Control Torque, \mathbf{N}_{ctrl}

Another torque that must be considered is the control moment,

$$\mathbf{N}_{ctrl} = {}^c \mathbf{m}(t) \times {}^c \mathbf{b}_{meas} \quad (2.73)$$

This torque is expressed in the CCS and is generated by the cross coupling between the magnetic moment $\mathbf{m}(t)$ and the Earth's magnetic field \mathbf{b}_{meas} , measured by on-board instruments.

Momentum Wheel Torque, \mathbf{N}_{wheel}

Additionally, some satellites, like SUNSAT or China Tsinghua-1 microsatellite, have included a wheel or disk in order to change the attitude of the satellite. These can be included as an additional term in the dynamic equation of motion.

The wheel has an angular momentum, $\mathbf{L}_{wheel} = I\boldsymbol{\omega}_{wheel}$. From (2.57), the dynamic rotational equations, of the rigid body is,

$$\sum \mathbf{M}^o = I\dot{\boldsymbol{\omega}}_{body} + I\dot{\boldsymbol{\omega}}_{wheel} = \dot{\mathbf{H}}^o + \dot{\mathbf{h}}^o \quad (2.74)$$

Consider now two coordinate systems, one rotating with the rigid body and the other fixed, as seen in dynamics subsection. The term due to the wheel has to be taken into account in the dynamic equation (2.60), which results in

$$\sum \mathbf{M}^G = \dot{\mathbf{H}}^{oxyz} + \Omega \times \mathbf{H}_G + \dot{\mathbf{h}}^o + \Omega \times \mathbf{h}^G \quad (2.75)$$

Since $\dot{\mathbf{h}}_0$ is the spacecraft torque acting in the wheel, it is usually represented as \mathbf{N}_{wheel} and the Eq.(2.75) is rewritten as

$$\sum \mathbf{M}^G = \dot{\mathbf{H}}^{oxyz} + \Omega \times (\mathbf{H}^G + \mathbf{h}^G) + \mathbf{N}_{wheel} \quad (2.76)$$

Using Eq. (2.59), recalling that the products of inertia are zero (see Section 2.4.1), and solving *w.r.t.* the derivative of the angular velocity $\boldsymbol{\omega}$, the Euler equations of motion become,

$$\dot{\boldsymbol{\omega}}_{bi}^b = I^{-1} (\mathbf{N}_{gg} + \mathbf{N}_{ctrl} + \mathbf{N}_{dt} - \mathbf{N}_{wheel} - \boldsymbol{\omega}_{bi}^b \times (I\boldsymbol{\omega}_{bi}^b + I\boldsymbol{\omega}_{wheel})) \quad (2.77)$$

where \mathbf{N}_{wheel} is the momentum wheel.

Chapter 3

Attitude Determination Methods

A reliable, robust algorithm capable to estimate the satellite's state is essential in autonomous spacecrafts such as small satellites. A control algorithm that efficiently manoeuvres the satellite's attitude requires the knowledge of the angular velocity and quaternion estimates. The starting point for any attitude determination method is to analyse the attitude sensors on-board small satellites: magnetometers, Sun sensors, Earth sensors, horizon sensors, star sensors and Global Position System (GPS). Gyros are also used to measure the angular velocity directly, however they often fail to work, thus jeopardizing the ADS and the mission. Because of this, gyros are often not even included in small satellites such as PoSAT-1, Ørsted, Freja or SUNSAT.

At least one attitude sensor must be permanently available along the whole orbit for the estimator algorithm to work properly. One such sensor is the magnetometer. This made magnetometers the most used sensors for attitude determination. To avoid local measurability problems, the data from the magnetometers must be combined with other attitude sensors data or with information from the satellite model. This leads to two main methodologies to tackle the spacecraft attitude determination problem:

- *point-to-point or deterministic methods* - estimate only the attitude, based on vector observations from at least two attitude sensors, disregarding all the information from the past measurements and/or from the satellite's equations of motion.
- *recursive estimation methods* - estimate the elements of the state vector from magnetometers, taking advantage of the dynamic and kinematic models to propagate the estimated state vector between measurements.

An optimal state recursive estimator, in the sense that it minimizes the mean square estimation error when applied to linear dynamic systems is the Kalman filter.

However, as shown in Chapter 2, small satellite's equations of motion are non-linear and, in order to use a Kalman filter, its linearization must be made. Due to this, the Kalman filter optimality and stability properties are not guaranteed. So, there are other recursive non-linear estimators, like higher order filters, that try to handle inaccuracies or simplification errors resulting from the linearizations or to model the process and observation with non-Gaussian distributions (see Section 1.2). The recursive methodology used in this dissertation is the EKF which application to the small satellites is explained in next section.

The point-to-point methods are attractive for attitude estimation algorithms since these problems are avoided and also there is no need to initialize the filter or to guarantee symmetry and positive-definite state error covariance matrix. Since the point-to-point methods are exclusively based on a set of noisy vector measurements to determine the attitude matrix, there must be two sets of measurements from different attitude sensors. As stressed before, magnetometers are always available and Sun sensors which are less susceptible to errors than the Earth sensors, are the most common sensors used on-board LEO small satellites.

Among point-to-point solutions to Wahba's problem, (*e.g.* Quest, Davenport's \mathbf{q} , ESOQ or FOAM), the Singular Value Decomposition (SVD) was chosen for this work due to its robustness and low computational requirements when considering two attitude sensors. However, problems arise when just one sensor is available, *e.g.*, when the Sun is out of range of the Sun sensor or when the satellite is in the dark side of Earth, that must be handled when SVD is used. This is described in more detail in Section 3.2.

3.1 Recursive Estimation Methods

Recursive methods are suitable for nonlinear systems, especially for the small satellites' ADS in closed loop algorithms where the linearization is performed about the filter's estimates trajectory, that depends on the measurements data. Since the trajectory is continuously updated, the algorithm parameters can not be pre-computed once for the entire set of data as for the batch-algorithms. Also, the Kalman filter is well suited to real-time problems because it directly estimates the state vector at a single time, based on the measurements at that time and all measurements up to that time with a fading memory. However, the Kalman filter just guarantees optimal state estimate when applied to linear systems¹. The state space formulation for small satellites is non-linear both regarding the system and the observation models. Still, the Kalman filter can be used by linearization of the equations that describe the system, addressed in the next section.

¹See Appendix A for details.

The EKF presented is based in the work of Psiaki *et. al.* [38], using Steyn modifications [45], so that it can be used for spinning satellites and with Sun sensor measurements.

3.1.1 Kalman Filter Applied to Small Satellites

State Vector

In this work the state vector includes all the state variables required to ensure accurate spacecraft attitude determination. These variables are the angular velocity, ω , and the quaternions, q :

$$\mathbf{x}(t) = [\omega(t) \quad q(t)]^T \quad (3.1)$$

These state variables were chosen because: the quaternion, q , directly represents the satellite's attitude; the angular velocity, ω , significantly affects the equations of motion of the satellite and represents an important physical quantity. Other variables might be included, *e.g.* sensor misalignments, disturbance torque (see Section 2.4.3), noise, parameters that influence the sensor measurements or other orbital parameters besides attitude. However there are guidelines, [55] to restrict the choice of the state variables: these must be modelled for propagation which may be impossible to achieve or too complex to perform; non-observable state variables should not be considered or appropriate measurements should be added to make the system completely observable; some may be highly correlated and therefore redundant; whenever each parameter is added, the matrices rank increases thus requiring more computer memory, reducing the efficiency of the estimator algorithm. The criterion to choose the state vector depends on the attitude accuracy required to accomplish the satellite mission. For instance, for PoSAT-1, which is on an experimental mission, the minimum number of state parameters needed for the control algorithm, coping with one sensor permanently available on-board, are the quaternions and angular velocity.

System Model

The satellite is described by a continuous time non-linear model, defined as:

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t), t) + \mathbf{w}(t) \quad (3.2)$$

where $\mathbf{w}(t)$ is assumed to be zero mean Gaussian process noise with known covariance,

$$\mathbf{w}(t) \sim N\{\mathbf{0}, Q(t)\}$$

3.1. RECURSIVE ESTIMATION METHODS

$\mathbf{u}(t)$ is a deterministic control input vector and, $f(\mathbf{x}(t), \mathbf{u}(t), t)$ is a non-linear continuous function which describes the time changes of the satellite state, given by the dynamic and kinematic equations (2.62) and (2.63)

$$\begin{bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \begin{bmatrix} I_{xx}^{-1} \left(\sum M_x - (I_{zz} - I_{yy}) \omega_{y,bi}^b \omega_{z,bi}^b \right) \\ I_{yy}^{-1} \left(\sum M_y - (I_{xx} - I_{zz}) \omega_{z,bi}^b \omega_{x,bi}^b \right) \\ I_{zz}^{-1} \left(\sum M_z - (I_{yy} - I_{xx}) \omega_{x,bi}^b \omega_{y,bi}^b \right) \\ 0 & \omega_{z,bo}^b & -\omega_{y,bo}^b & \omega_{x,bo}^b \\ -\omega_{z,co}^b & 0 & \omega_{x,bo}^b & \omega_{y,bo}^b \\ \omega_{y,co}^b & -\omega_{x,bo}^b & 0 & \omega_{z,bo}^b \\ -\omega_{x,co}^b & -\omega_{y,bo}^b & -\omega_{z,bo}^b & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \quad (3.3)$$

Since the state differential equations as well as those relating the state and the measurements, (described ahead) the Kalman filter is not applicable unless this non-linear system is linearized about the current filter's estimates, resulting in the Extended Kalman Filter (EKF).

The linearization, by definition, is a small perturbation about an operational point, which is the estimated state. The actual state $\mathbf{x}(t)$ can be defined as the sum between the estimated state $\hat{\mathbf{x}}(t)$ and a perturbation of the state $\delta\mathbf{x}(t)$. Hence, the state variable is defined as,

$$\mathbf{x}(t) = \hat{\mathbf{x}}(t) + \delta\mathbf{x}(t) \quad (3.4)$$

The non-linear model, Eq.(3.2), becomes,

$$\dot{\hat{\mathbf{x}}}(t) + \delta\dot{\mathbf{x}}(t) = f(\hat{\mathbf{x}}(t) + \delta\mathbf{x}, \mathbf{u}(t), t) + \mathbf{w}(t) \quad (3.5)$$

Assuming that $\delta\mathbf{x}(t)$ is small and approximating the first term on the right by a Taylor's series expansions,

$$f(\hat{\mathbf{x}}(t) + \delta\mathbf{x}(t), \mathbf{u}(t), t) \approx f(\hat{\mathbf{x}}(t), \mathbf{u}(t), t) + \frac{\partial f}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\hat{\mathbf{x}}} \delta\mathbf{x}(t) + h.o.t \quad (3.6)$$

and retaining only the first-order terms,

$$\dot{\hat{\mathbf{x}}}(t) + \delta\dot{\mathbf{x}}(t) = f(\hat{\mathbf{x}}(t), \mathbf{u}(t), t) + \frac{\partial f}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\hat{\mathbf{x}}} \delta\mathbf{x}(t) + \mathbf{w}(t) \quad (3.7)$$

Since the $\hat{\mathbf{x}}(t)$ satisfies the deterministic differential equation

$$\dot{\hat{\mathbf{x}}}(t) = f(\hat{\mathbf{x}}(t), \mathbf{u}(t), t) \quad (3.8)$$

equation (3.7) becomes

$$\delta \dot{\mathbf{x}}(t) = \frac{\partial f}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\hat{\mathbf{x}}} \delta \mathbf{x}(t) + \mathbf{w}(t) \quad (3.9)$$

Defining $F(\hat{\mathbf{x}}(t), \mathbf{u}(t), t) = \frac{\partial f}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\hat{\mathbf{x}}}$ and replacing into Eq.(3.8) leads to the linearized model to be used by the EKF:

$$\delta \dot{\mathbf{x}}(t) = F(\hat{\mathbf{x}}(t), \mathbf{u}(t), t) + \mathbf{w}(t) \quad (3.10)$$

Linearization of State Vector

According to Eq.(3.4) the state vector can be defined as:

$$\begin{bmatrix} \boldsymbol{\omega}(t) \\ \mathbf{q}(t) \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{\omega}}(t) + \delta \boldsymbol{\omega}(t) \\ \hat{\mathbf{q}} + \delta \mathbf{q}(t) \end{bmatrix} \quad (3.11)$$

However some special care must be taken: the quaternion represents a rotation, so the sum of two quaternions is no longer a quaternion. Since the small perturbation is also a quaternion, the quaternion rules for multiplication (see Section 2.2.5), must be respected. Defining the quaternion as a function of the quaternion perturbation and the estimated quaternion, follows that:

$$\mathbf{q} = \delta \mathbf{q} \otimes \hat{\mathbf{q}} \quad (3.12)$$

and the linearized state vector becomes,

$$\begin{bmatrix} \boldsymbol{\omega} \\ \mathbf{q} \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{\omega}} + \delta \boldsymbol{\omega} \\ \delta \mathbf{q} \otimes \hat{\mathbf{q}} \end{bmatrix} \quad (3.13)$$

Although (3.13) is the only representation with a physical meaning, since a quaternion represents a rotation, (3.11) is also mathematically correct and works in practice with good results. Furthermore, it is more robust to large perturbations in the state vector, not causing the estimator filter to diverge. Divergence happens when (3.13) is used, as shown by Steyn [45].

The linearized state vector, from Eq.(3.13), has dimension seven. However, the unit constraint on the quaternion, Eq.(2.19), leads the state error covariance to be singular, as addressed by Lefferts *et.al.* [27] and demonstrated by Bak [1],

$$P_{[7x7]} = \begin{bmatrix} P_{\boldsymbol{\omega}\boldsymbol{\omega}} & 0_{3x4} \\ 0_{4x3} & \begin{bmatrix} P_{\mathbf{q}\mathbf{q}} & 0 \\ 0 & 0 \end{bmatrix} \end{bmatrix} \quad (3.14)$$

The singularity is necessary because it preserves the quaternion constraint (2.19) explicitly. However it is difficult to maintain due to the accumulation of round-off

3.1. RECURSIVE ESTIMATION METHODS

errors. The best way to maintain the singularity is to represent the P matrix by a smaller dimension matrix [27],

$$P_{[6x6]} = \begin{bmatrix} P_{\omega\omega} & 0_{3x3} \\ 0_{3x3} & P_{\mathbf{q}\mathbf{q}} \end{bmatrix} \quad (3.15)$$

as well as the Kalman gain, $K_{[6x6]}$, and the matrices involved in the EKF algorithm: (see Section 3.1.1) $H_{[3x6]}, Q_{[6x6]}$ and $\Phi_{[6x6]}$.

So, the state update, (A.7) $\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k \mathbf{e}_k$, of the EKF must be computed carefully. Instead of the seven elements, the perturbation state vector has been reduced to six elements,

$$\delta \mathbf{x} = [\delta \omega_x \ \delta \omega_y \ \delta \omega_z \ \delta q_1 \ \delta q_2 \ \delta q_3]^T \quad (3.16)$$

and $\delta \mathbf{x}$ will be updated according $\delta \mathbf{x} = K_k \mathbf{e}_k$.

Still, the perturbation of the scalar part of the quaternion must be computed. From the quaternion constraint (2.19),

$$\delta q_4 = \sqrt{1 - \|\delta \mathbf{q}\|} \quad (3.17)$$

as suggested by Psiaki *et. al.* [38], Bak [1], and Steyn [45].

The state update of the quaternion is rebuilt according to,

$$\begin{aligned} \hat{\mathbf{q}}_{k+1}^+ &= \delta \mathbf{q} \otimes \hat{\mathbf{q}}_{k+1}^- \\ &= [\delta \mathbf{q} \ \delta q_4]^T \otimes \hat{\mathbf{q}}_{k+1}^- \\ &= [\delta \mathbf{q} \ \sqrt{1 - \|\delta \mathbf{q}\|}]^T \otimes \hat{\mathbf{q}}_{k+1}^- \end{aligned} \quad (3.18)$$

and using (2.29),

$$\hat{\mathbf{q}}_{k+1}^+ = \Xi(\hat{\mathbf{q}}_{k+1}^-) [\delta \mathbf{q} \ \sqrt{1 - \|\delta \mathbf{q}\|}]^T \quad (3.19)$$

The state update of the angular velocity is computed as

$$\hat{\boldsymbol{\omega}}_{k+1}^+ = \hat{\boldsymbol{\omega}}_{k+1}^- + \delta \boldsymbol{\omega} \quad (3.20)$$

$$= [\hat{\omega}_{x/k+1}^- \ \hat{\omega}_{y/k+1}^- \ \hat{\omega}_{z/k+1}^-]^T + [\delta \omega_x \ \delta \omega_y \ \delta \omega_z]^T \quad (3.21)$$

Linearization of Dynamic Equations

The dynamic equations (2.62) must have the external moments explicit defined in order to be linearized. Since the small satellite studied in this work is the PoSAT-1 (see Section 4.2.2), the relevant external moments influence the dynamics as follows:

3.1. RECURSIVE ESTIMATION METHODS

$$\dot{\boldsymbol{\omega}} = I^{-1} (\mathbf{N}_{ctrl} + \mathbf{N}_{gg} - \boldsymbol{\omega} \times I\boldsymbol{\omega}), \quad (3.22)$$

where $\boldsymbol{\omega}$ is the angular velocity of the body CS *w.r.t.* inertial CS expressed in body CS, $\boldsymbol{\omega}_{bi}^b$, but here and henceforth denoted simply as $\boldsymbol{\omega}$, for notation convenience, unless referred otherwise.

The linearization of (3.22) leads to

$$\delta\dot{\boldsymbol{\omega}} = I^{-1} (\delta\mathbf{N}_{ctrl} + \delta\mathbf{N}_{gg} + ([I\boldsymbol{\omega}\times] - [\boldsymbol{\omega}\times]I)\delta\boldsymbol{\omega}) \quad (3.23)$$

where

$$[I\boldsymbol{\omega}\times] - [\boldsymbol{\omega}\times]I = \begin{bmatrix} 0 & (I_y - I_z)\hat{\omega}_z & (I_y - I_z)\hat{\omega}_y \\ (I_z - I_x)\hat{\omega}_z & 0 & (I_z - I_x)\hat{\omega}_y \\ (I_x - I_y)\hat{\omega}_y & (I_x - I_y)\hat{\omega}_x & 0 \end{bmatrix} \quad (3.24)$$

Linearization of Gravity Gradient Torque, $\delta\mathbf{N}_{gg}$ Recalling the gravity gradient equation (2.70)

$$\mathbf{N}_{gg} = 3\omega_0^2 (\hat{\mathbf{k}}_b \times I\hat{\mathbf{k}}_b)$$

the unit vector along the body, $\hat{\mathbf{k}}_b$, can be written as $\hat{\mathbf{k}}_b = A_o^b(\mathbf{q})\hat{\mathbf{k}}_o$ according to (2.3). The gravity moment (2.70) becomes,

$$\mathbf{N}_{gg} = 3\omega_0^2 (A(\mathbf{q})\hat{\mathbf{k}}_o \times IA(\mathbf{q})\hat{\mathbf{k}}_o) \quad (3.25)$$

According to the definition of quaternion from Eq.(3.12) and based on (2.37), the attitude matrix as function of the quaternion perturbation is

$$\begin{aligned} A(\mathbf{q}) &= A(\delta\mathbf{q} \otimes \hat{\mathbf{q}}) \\ &= A(\delta\mathbf{q})A(\hat{\mathbf{q}}) \end{aligned} \quad (3.26)$$

Then, the gravity gradient can be written as follows

$$\mathbf{N}_{gg} = 3\omega_0^2 (A(\delta\mathbf{q})A(\hat{\mathbf{q}})\hat{\mathbf{k}}_o \times IA(\delta\mathbf{q})A(\hat{\mathbf{q}})\hat{\mathbf{k}}_o) \quad (3.27)$$

which can be written,

$$\mathbf{N}_{gg} = 3\omega_0^2 (A(\delta\mathbf{q})\hat{\mathbf{k}}_b \times IA(\delta\mathbf{q})\hat{\mathbf{k}}_b) \quad (3.28)$$

3.1. RECURSIVE ESTIMATION METHODS

According to the definition of the quaternion in terms of the Euler axis/angle and since $\delta\mathbf{q}$ is a small perturbation of the attitude quaternion

$$\delta\mathbf{q} = \begin{bmatrix} e_1 \sin \frac{\delta\Phi}{2} \\ e_2 \sin \frac{\delta\Phi}{2} \\ e_3 \sin \frac{\delta\Phi}{2} \\ \cos \frac{\delta\Phi}{2} \end{bmatrix} \approx \begin{bmatrix} \delta q_1 \\ \delta q_2 \\ \delta q_3 \\ 1 \end{bmatrix} = \begin{bmatrix} \delta\mathbf{q} \\ 1 \end{bmatrix} \quad (3.29)$$

Note that the DCM can be expressed in terms of a quaternion (see Eq.(2.35)), $A(\mathbf{q}) = (q_4^2 - \|\mathbf{q}\|^2) \mathbf{1} + 2\mathbf{q}\mathbf{q}^T - 2q_4 [\mathbf{q}\times]$ and from (3.29), neglecting second and higher order terms, the DCM reduces to

$$A(\delta\mathbf{q}) = \mathbf{1} - 2[\delta\mathbf{q}\times] \quad (3.30)$$

Replacing into (3.28), it follows

$$\mathbf{N}_{gg} = 3\omega_0^2 \left[(\mathbf{1} - 2[\delta\mathbf{q}\times]) \hat{\mathbf{k}}_b \times I (\mathbf{1} - 2[\delta\mathbf{q}\times]) \hat{\mathbf{k}}_b \right] \quad (3.31)$$

and leaving out all the second and higher order terms, the linearized gravity gradient torque obtained is

$$\delta\mathbf{N}_{gg} = 6\omega_0^2 \left(-\hat{\mathbf{k}}_b \times I [\delta\mathbf{q}\times] \hat{\mathbf{k}}_b - [\delta\mathbf{q}\times] \hat{\mathbf{k}}_b \times I \hat{\mathbf{k}}_b \right) \quad (3.32)$$

and using (2.33), leads to,

$$\delta\mathbf{N}_{gg} = 6\omega_0^2 \left([\hat{\mathbf{k}}_b \times] I [\hat{\mathbf{k}}_b \times] \delta\mathbf{q} - [I \hat{\mathbf{k}}_b \times] [\hat{\mathbf{k}}_b \times] \delta\mathbf{q} \right) \quad (3.33)$$

or,

$$\delta\mathbf{N}_{gg} = 6\omega_0^2 \left([\hat{\mathbf{k}}_b \times] I [\hat{\mathbf{k}}_b \times] - [I \hat{\mathbf{k}}_b \times] [\hat{\mathbf{k}}_b \times] \right) \delta\mathbf{q} \quad (3.34)$$

Hence, the linearized gravity gradient torque is

$$\begin{aligned} \delta\mathbf{N}_{gg} &= 6\omega_0^2 \begin{bmatrix} (A_{23}^2 - A_{33}^2)(I_y - I_z) & -A_{13}A_{23}(I_y - I_z) & A_{13}A_{33}(I_y - I_z) \\ A_{12}A_{23}(I_z - I_x) & (A_{33}^2 - A_{13}^2)(I_z - I_x) & -A_{23}A_{33}(I_z - I_x) \\ -A_{13}A_{33}(I_x - I_y) & A_{23}A_{33}(I_x - I_y) & (A_{13}^2 - A_{23}^2)(I_x - I_y) \end{bmatrix} \delta\mathbf{q} \\ &= 6\omega_0^2 \mathbf{F}_{gg} \delta\mathbf{q} \end{aligned} \quad (3.35)$$

3.1. RECURSIVE ESTIMATION METHODS

Linearization of Control Moment $\delta\mathbf{N}_{ctrl}$ The control moment from (2.73) is,

$$\mathbf{N}_{ctrl} = \mathbf{m}(t) \times \mathbf{b}_{meas} \quad (3.36)$$

The \mathbf{b}_{meas} can be expressed as a function of the quaternion, $\mathbf{b}_{meas} = A_o^b(\mathbf{q})\mathbf{b}^o$. Hence, the control moment equation becomes,

$$\mathbf{N}_{ctrl} = \mathbf{m}(t) \times A_o^b(\mathbf{q})\mathbf{b}^o$$

The magnetic moment, which depends on the control law applied to the satellite, is also expressed in terms of the quaternion and of angular velocity, must also be partially differentiated *w.r.t.* the state. This means that each time there is a magnetic moment actuating in the satellite the EKF must have access to the control gains, which is inconvenient. For example, the *Predictive* stabilisation applied to PoSAT-1 satellite in the SimSat simulator is based in one of the available magnetic moments that minimize a cost function [47]. So there is no expression for the magnetic moment function. Besides, it is difficult to determine the control gain values from the satellite's real data, as happened with the real data used in this work from the PoSAT-1 satellite.

To handle this problem, one may either neglect the linearization of the control moment, assuming that the process noise covariance matrix, Q , will take it into account or use an accurate control moment model in order to fully linearize the magnetic moment. In this work, the linearization of the control moment was neglected.

Replacing the results from the linearization obtained before into (3.23), the linearized dynamic equations,

$$\delta\dot{\omega} = I^{-1} \left[[\mathbf{I}\boldsymbol{\omega} \times] - [\boldsymbol{\omega} \times]\mathbf{I} - 6\omega_0^2 F_{gg} \right] \begin{bmatrix} \delta\boldsymbol{\omega} \\ \delta\mathbf{q} \end{bmatrix} \quad (3.37)$$

Linearization of Kinematic Equation

From Equation (3.13), the quaternion is given as a function of the quaternion error and of the estimated quaternion by the following expression:

$$\mathbf{q} = \delta\mathbf{q} \otimes \hat{\mathbf{q}} \Rightarrow \delta\mathbf{q} = \mathbf{q} \otimes \hat{\mathbf{q}}^{-1} \quad (3.38)$$

Representing the angular velocity in terms of a quaternion $\mathbf{w} = [\omega_x \ \omega_y \ \omega_z \ 0]^T$ and from the definition of the quaternion product (2.29) used in this work, the kinematics equation (2.63) becomes:

$$\dot{\mathbf{q}} = \frac{1}{2}\Omega(\mathbf{w})\mathbf{q} = \frac{1}{2}\Xi(\mathbf{q})\mathbf{w} \quad (3.39)$$

$$= \frac{1}{2}\mathbf{w} \otimes \mathbf{q} \quad (3.40)$$

3.1. RECURSIVE ESTIMATION METHODS

From the application of the chain rule to (3.38) follows,

$$\dot{\mathbf{q}} = \frac{d}{dt} (\delta \mathbf{q} \otimes \hat{\mathbf{q}}) = \delta \dot{\mathbf{q}} \otimes \hat{\mathbf{q}} + \delta \mathbf{q} \otimes \dot{\hat{\mathbf{q}}} \quad (3.41)$$

Using the previous expression in the kinematic equations,

$$\delta \dot{\mathbf{q}} \otimes \hat{\mathbf{q}} + \delta \mathbf{q} \otimes \dot{\hat{\mathbf{q}}} = \frac{1}{2} \mathbf{w} \otimes \mathbf{q} \quad (3.42)$$

Right multiplying (3.42) by $\hat{\mathbf{q}}^{-1}$, it follows immediately

$$\delta \dot{\mathbf{q}} = \frac{1}{2} \mathbf{w} \otimes \mathbf{q} \otimes \hat{\mathbf{q}}^{-1} - \delta \mathbf{q} \otimes \dot{\hat{\mathbf{q}}} \otimes \hat{\mathbf{q}}^{-1} \quad (3.43)$$

From the Kinematics,

$$\dot{\hat{\mathbf{q}}} = \frac{1}{2} \hat{\mathbf{w}} \otimes \hat{\mathbf{q}} \quad (3.44)$$

$$\dot{\hat{\mathbf{q}}} \otimes \hat{\mathbf{q}}^{-1} = \frac{1}{2} \hat{\mathbf{w}} \quad (3.45)$$

and using (3.38), from expression (3.43) follows that,

$$\delta \dot{\mathbf{q}} = \frac{1}{2} \mathbf{w} \otimes \delta \mathbf{q} - \frac{1}{2} \delta \mathbf{q} \otimes \hat{\mathbf{w}} \quad (3.46)$$

Recalling that the angular velocity can be expressed in terms of a perturbation $\boldsymbol{\omega} = \hat{\boldsymbol{\omega}} + \delta \boldsymbol{\omega}$, follows that,

$$\delta \dot{\mathbf{q}} = \frac{1}{2} \hat{\mathbf{w}} \otimes \delta \mathbf{q} + \frac{1}{2} \delta \mathbf{w} \otimes \delta \mathbf{q} - \frac{1}{2} \delta \mathbf{q} \otimes \hat{\mathbf{w}} \quad (3.47)$$

Representing the quaternion multiplication with the help of the left and right operators defined in Section 2.2.5,

$$2\delta \dot{\mathbf{q}} = \Omega(\hat{\mathbf{w}})\delta \mathbf{q} + \Omega(\delta \mathbf{w})\delta \mathbf{q} - \Xi(\hat{\mathbf{w}})\delta \mathbf{q} \quad (3.48)$$

$$\begin{aligned} &= \begin{bmatrix} -[\hat{\boldsymbol{\omega}} \times] & \hat{\boldsymbol{\omega}} \\ -\hat{\boldsymbol{\omega}}^T & 0 \end{bmatrix} \delta \mathbf{q} + \begin{bmatrix} -[\delta \hat{\boldsymbol{\omega}} \times] & \delta \hat{\boldsymbol{\omega}} \\ -\delta \hat{\boldsymbol{\omega}}^T & 0 \end{bmatrix} \delta \mathbf{q} - \begin{bmatrix} [\hat{\boldsymbol{\omega}} \times] & \hat{\boldsymbol{\omega}} \\ -\hat{\boldsymbol{\omega}}^T & 0 \end{bmatrix} \delta \mathbf{q} \\ &= \begin{bmatrix} -2[\hat{\boldsymbol{\omega}} \times] - [\delta \hat{\boldsymbol{\omega}} \times] & \delta \hat{\boldsymbol{\omega}} \\ -\delta \hat{\boldsymbol{\omega}}^T & 0 \end{bmatrix} \delta \mathbf{q} \end{aligned} \quad (3.49)$$

From (3.49) and (3.29), $\delta \dot{\mathbf{q}} = \begin{bmatrix} (-2[\hat{\boldsymbol{\omega}} \times] - [\delta \hat{\boldsymbol{\omega}} \times])\delta \mathbf{q} + \delta \hat{\boldsymbol{\omega}} \\ -\delta \hat{\boldsymbol{\omega}}^T \delta \mathbf{q} \end{bmatrix}$ and neglecting second order terms ($-[\delta \hat{\boldsymbol{\omega}} \times] \delta \mathbf{q}$) and ($-\delta \hat{\boldsymbol{\omega}}^T \delta \mathbf{q}$), follows

3.1. RECURSIVE ESTIMATION METHODS

$$\delta \dot{\mathbf{q}} = -[\hat{\boldsymbol{\omega}} \times] \delta \mathbf{q} + \frac{1}{2} \delta \hat{\boldsymbol{\omega}}, \text{ with } \delta \dot{q}_4 = 0 \quad (3.50)$$

The angular velocity, used in the kinematics (see Section 2.4.2), is defined as in body CS *w.r.t* orbit coordinates expressed in terms of body CS,

$$\delta \dot{\mathbf{q}} = -[\hat{\boldsymbol{\omega}}_{bo}^b \times] \delta \mathbf{q} + \frac{1}{2} \delta \hat{\boldsymbol{\omega}}_{bo}^b, \text{ with } \delta \dot{q}_4 = 0 \quad (3.51)$$

However, the angular velocity must refer to body *w.r.t* inertial CS. So replacing (2.66) into the previous expression,

$$\delta \dot{\mathbf{q}} = -[\hat{\boldsymbol{\omega}}_{bi}^b \times] \delta \mathbf{q} + [\hat{\boldsymbol{\omega}}_{oi}^b \times] \delta \mathbf{q} + \frac{1}{2} \delta \hat{\boldsymbol{\omega}}_{bi}^b - \frac{1}{2} \delta \hat{\boldsymbol{\omega}}_{oi}^b \quad (3.52)$$

From (2.67) $\hat{\boldsymbol{\omega}}_{oi}^b = A_o^b(\mathbf{q}) \hat{\boldsymbol{\omega}}_{oi}^o$, resulting $\delta \hat{\boldsymbol{\omega}}_{oi}^b = A_o^b(\delta \mathbf{q}) A_o^b(\hat{\mathbf{q}}) \hat{\boldsymbol{\omega}}_{oi}^o = A_o^b(\delta \mathbf{q}) \hat{\boldsymbol{\omega}}_{oi}^b$. Using (3.30), the linearized angular velocity is $\delta \hat{\boldsymbol{\omega}}_{oi}^b = (\mathbf{1} - 2[\delta \mathbf{q} \times]) \hat{\boldsymbol{\omega}}_{oi}^b = \mathbf{1} \hat{\boldsymbol{\omega}}_{oi}^b + 2[\hat{\boldsymbol{\omega}}_{oi}^b \times] \delta \mathbf{q}$. Thus, (3.52) will become

$$\delta \dot{\mathbf{q}} = -[\hat{\boldsymbol{\omega}}_{bi}^b \times] \delta \mathbf{q} + [\hat{\boldsymbol{\omega}}_{oi}^b \times] \delta \mathbf{q} + \frac{1}{2} \delta \hat{\boldsymbol{\omega}}_{bi}^b - [\hat{\boldsymbol{\omega}}_{oi}^b \times] \delta \mathbf{q} - \mathbf{1} \hat{\boldsymbol{\omega}}_{oi}^b \quad (3.53)$$

Since $\hat{\boldsymbol{\omega}}_{oi}^b$ is constant for a circular orbit, then it is considered to be included in the noise, $w(t)$ from (3.10). Hence (3.53) simplified, leads to

$$\delta \dot{\mathbf{q}} = \frac{1}{2} \delta \hat{\boldsymbol{\omega}}_{bi}^b - [\hat{\boldsymbol{\omega}}_{bi}^b \times] \delta \mathbf{q} \quad (3.54)$$

The linearization of the dynamic equations, (3.37), and kinematic equations (3.54) are used in (3.10), leading to the following matrix,

$$F(\hat{\mathbf{x}}(t), t) = \begin{bmatrix} I^{-1} ([I \hat{\boldsymbol{\omega}}_{bi}^b \times] - [\hat{\boldsymbol{\omega}}_{bi}^b \times] I) & I^{-1} 6\omega_0 \mathbf{F}_{gg} \\ \frac{1}{2} \mathbf{1}_{3 \times 3} & -[\hat{\boldsymbol{\omega}}_{bi}^b \times] \end{bmatrix} \quad (3.55)$$

EKF Propagation

To propagate the state vector and the state error covariance matrix between measurements, from one time step, t_k to the next, t_{k+1} , the transition matrix, Φ_k , must be used (see Appendix A),

$$\hat{\mathbf{x}}_{k+1}^- = \Phi_k \hat{\mathbf{x}}_k^+ \quad (3.56)$$

$$P_{k+1}^- = \Phi_k P_k^+ \Phi_k + Q_k \quad (3.57)$$

3.1. RECURSIVE ESTIMATION METHODS

The transition matrix is obtained from the linearized dynamic and kinematic equations (3.55),

$$\Phi_k \approx \mathbf{1} + F(\hat{\mathbf{x}}(t), t)\Delta T + \frac{(F(\hat{\mathbf{x}}(t), t)\Delta T)^2}{2!} \quad (3.58)$$

Thus, a better way to propagate the state vector is to integrate numerically the nonlinear dynamics (3.23), and to use a closed-form of the kinematic equations with constant body rates, $\mathbf{q}_{k+1} = \exp\left[\frac{1}{2}\Omega_k\Delta T\right]\mathbf{q}_k$.

The most common one-step integration method used for spacecraft is the 4-stage Runge-Kutta method, according to Wertz [55],

$$\hat{\mathbf{x}}_{k+1}^- = \hat{\mathbf{x}}_k^+ + \frac{h}{6} (\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4) \quad (3.59)$$

with,

$$\begin{aligned} \mathbf{k}_1 &= f(\hat{\mathbf{x}}_k^+, k) \\ \mathbf{k}_2 &= f(\hat{\mathbf{x}}_k^+ + \frac{h}{2}\mathbf{k}_1, k + \frac{h}{2}) \\ \mathbf{k}_3 &= f(\hat{\mathbf{x}}_k^+ + \frac{h}{2}\mathbf{k}_2, k + \frac{h}{2}) \\ \mathbf{k}_4 &= f(\hat{\mathbf{x}}_k^+ + h\mathbf{k}_3, k + h) \end{aligned}$$

where $f(\hat{\mathbf{x}}_k^+, t) = f(\hat{\boldsymbol{\omega}}_k^+, t)$ is given by the dynamic equations of motion $\hat{\boldsymbol{\omega}}_{k+1}^- = \int I^{-1}(\mathbf{N}_{ctrl} + \mathbf{N}_{gg} - \boldsymbol{\omega} \times I\boldsymbol{\omega}) dt + \hat{\boldsymbol{\omega}}_k^+$. Hence,

$$f(\hat{\boldsymbol{\omega}}_k^+, t) = I^{-1}(\mathbf{N}_{ctrl} + \mathbf{N}_{gg} - \boldsymbol{\omega} \times I\boldsymbol{\omega}) \quad (3.60)$$

is used for discrete propagation of the angular velocity. The propagation of the quaternion, assuming that the angular velocity is constant over the sampling interval, is made according to the closed form solution [55],

$$\mathbf{q}_{k+1} = \exp\left[\frac{1}{2}\Omega_k\Delta T\right]\mathbf{q}_k \quad (3.61)$$

$$= \left[\cos\left(\frac{\bar{\omega}_k\Delta T}{2}\right) \mathbf{1} + \frac{1}{\bar{\omega}_k} \sin\left(\frac{\bar{\omega}_k\Delta T}{2}\right) \Omega_k \right] \mathbf{q}_k \quad (3.62)$$

where $\bar{\omega}_k = \sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2}$.

Observation Model

The discrete nonlinear observation model has the form

$$\mathbf{y}_k = h_k(\mathbf{x}(t_k), t_k) + \mathbf{v}_k \text{ with } k = 1, 2, \dots \quad (3.63)$$

3.1. RECURSIVE ESTIMATION METHODS

where the measurement noise covariance matrix , is given by

$$E [\mathbf{v}_k \mathbf{v}_k^T] = \begin{cases} R_k & , i = j \\ 0 & , i \neq j \end{cases} \quad (3.64)$$

where i is the row index and j is the column index.

Replacing the state variable definition (3.4) into the observation model

$$\mathbf{y}_k = h_k(\hat{\mathbf{x}}(t_k) + \delta\mathbf{x}(t), t_k) + \mathbf{v}_k \quad (3.65)$$

and assuming that $\delta x(t)$ is small and using the Taylor's series expansion of the first term retaining only the first-order terms, the following expression is obtained:

$$h_k(\hat{\mathbf{x}}(t_k) + \delta\mathbf{x}(t_k), t_k) \approx h_k(\hat{\mathbf{x}}(t_k), t_k) + \frac{\partial h}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\hat{\mathbf{x}}} \delta\mathbf{x}(t) \quad (3.66)$$

Defining $H_k(\hat{\mathbf{x}}_k) = \frac{\partial h}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\hat{\mathbf{x}}}$, the linearized measurement model is

$$\mathbf{y}_k - h_k(\hat{\mathbf{x}}(t_k), t_k) = H_k(\hat{\mathbf{x}}_k)\delta\mathbf{x} + \mathbf{v}_k \quad (3.67)$$

So the linearized innovation error model to be used is,

$$\mathbf{e}_k = H_k(\hat{\mathbf{x}}_k)\delta\mathbf{x} + \mathbf{v}_k \quad (3.68)$$

Since a cross product between parallel vectors is null, and from $\mathbf{y}_{meas} = A(\mathbf{q})\mathbf{y}^o$, the innovation error is defined as

$$\mathbf{e}_k = \mathbf{y}_{meas,k}^b \times A_o^b(\hat{\mathbf{q}}_k)\mathbf{y}_k^o \quad (3.69)$$

$$= \mathbf{y}_{meas,k}^b \times \hat{\mathbf{y}}_k^b \quad (3.70)$$

where \mathbf{y}_{meas}^b is a vector whose elements are the measurements obtained from the attitude sensors and \mathbf{y}^o is the vector computed from a model of the measurements, named as the reference vector.

For an error free measurement and reference vector,

$$\mathbf{y}_{meas,k}^b = A_o^b(\mathbf{q}_k)\mathbf{y}_k^o \quad (3.71)$$

Thus, the cross product between the normalized measurement and the normalized reference in body coordinates gives the innovation error,

$$\begin{aligned} \mathbf{e}_k &= \mathbf{y}_{meas,k}^b \times A_o^b(\mathbf{q}_k)\mathbf{y}_k^o \\ \mathbf{e}_k &= \mathbf{y}_{meas,k}^b \times \mathbf{y}_k^b \end{aligned} \quad (3.72)$$

3.1. RECURSIVE ESTIMATION METHODS

Considering that there is noise associated with the measurement and the reference vector, the expression becomes,

$$\mathbf{y}_{meas,k}^b = A_o^b(\mathbf{q}_k) \mathbf{y}_k^o + \mathbf{v}_k$$

Recalling the definition of quaternion (3.12)

$$\mathbf{y}_{meas,k}^b = A_o^b(\delta \mathbf{q}_k \otimes \hat{\mathbf{q}}_k) \mathbf{y}_k^o + \mathbf{v}_k \quad (3.73)$$

and using (2.37),

$$\mathbf{y}_{meas,k}^b = A_o^b(\delta \mathbf{q}_k) A_o^b(\hat{\mathbf{q}}_k) \mathbf{y}_k^o + \mathbf{v}_k \quad (3.74)$$

Knowing that $\hat{\mathbf{y}}^b = A_o^b(\hat{\mathbf{q}}) \mathbf{y}^o$, the previous expression becomes,

$$\mathbf{y}_{meas,k}^b = A_o^b(\delta \mathbf{q}_k) \hat{\mathbf{y}}_k^b + \mathbf{v}_k \quad (3.75)$$

The innovation error from (3.72) becomes,

$$\mathbf{e}_k = A_o^b(\delta \mathbf{q}_k) \hat{\mathbf{y}}_k^b \times \hat{\mathbf{y}}_k^b + \mathbf{v}_k \times \hat{\mathbf{y}}_k^b \quad (3.76)$$

Hence,

$$\mathbf{e}_k = A_o^b(\delta \mathbf{q}) \hat{\mathbf{y}}_k^b \times \hat{\mathbf{y}}_k^b + \tilde{\mathbf{v}}_k \quad (3.77)$$

where $\tilde{\mathbf{v}}_k = \mathbf{v}_k \times \hat{\mathbf{y}}_k^b$. Solving the previous expression, it follows that,

$$\mathbf{e}_k = 2 \begin{bmatrix} (\hat{y}_y^b)^2 + (\hat{y}_z^b)^2 & -\hat{y}_x^b \hat{y}_y^b & -\hat{y}_x^b \hat{y}_z^b \\ -\hat{y}_x^b \hat{y}_y^b & (\hat{y}_x^b)^2 + (\hat{y}_z^b)^2 & -\hat{y}_y^b \hat{y}_z^b \\ -\hat{y}_x^b \hat{y}_z^b & -\hat{y}_y^b \hat{y}_z^b & (\hat{y}_x^b)^2 + (\hat{y}_y^b)^2 \end{bmatrix}_k \delta \mathbf{q} + \tilde{\mathbf{v}}_k \quad (3.78)$$

Defining

$$\tilde{H}_k(\hat{\mathbf{y}}_k^b) = 2 \begin{bmatrix} (\hat{y}_y^b)^2 + (\hat{y}_z^b)^2 & -\hat{y}_x^b \hat{y}_y^b & -\hat{y}_x^b \hat{y}_z^b \\ -\hat{y}_x^b \hat{y}_y^b & (\hat{y}_x^b)^2 + (\hat{y}_z^b)^2 & -\hat{y}_y^b \hat{y}_z^b \\ -\hat{y}_x^b \hat{y}_z^b & -\hat{y}_y^b \hat{y}_z^b & (\hat{y}_x^b)^2 + (\hat{y}_y^b)^2 \end{bmatrix}_k \quad (3.79)$$

where $\hat{\mathbf{y}}_k^b = [\hat{y}_x^b \hat{y}_y^b \hat{y}_z^b]$. The innovation error may be written in terms of the state vector as follows,

$$\mathbf{e}_k = [0_{3 \times 3} \quad \tilde{H}_k(\hat{\mathbf{y}}_k^b)] \partial \mathbf{x}_k + \tilde{\mathbf{v}}_k \quad (3.80)$$

$$= H_k(\hat{\mathbf{x}}_k) \partial \mathbf{x}_k + \tilde{\mathbf{v}}_k \quad (3.81)$$

where,

$$H_k(\hat{\mathbf{x}}_k) = [0_{3 \times 3} \quad \tilde{H}_k(\hat{\mathbf{y}}_k^b)] \quad (3.82)$$

A summary of the EKF algorithm applied to small satellites, as presented before, follows:

EKF Algorithm

Between measurements

PROJECT AHEAD

- Propagation of the state variables $\mathbf{x}_{k+1}^- = [\hat{\omega}_{k+1}^- \quad \mathbf{q}_{k+1}]^T$ through Runge-Kutta methods (3.59), (3.60) and (3.61),

$$\hat{\omega}_{k+1}^- = \hat{\omega}_k^+ + \frac{h}{6} (\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4). \text{ where } \mathbf{k}_1 = f(\hat{\omega}_k^+, k), \mathbf{k}_2 = f(\hat{\omega}_k^+ + \frac{h}{2}\mathbf{k}_1, k + \frac{h}{2}), \mathbf{k}_3 = f(\hat{\omega}_k^+ + \frac{h}{2}\mathbf{k}_2, k + \frac{h}{2}), \mathbf{k}_4 = f(\hat{\omega}_k^+ + h\mathbf{k}_3, k + h) \text{ and } f(\hat{\omega}_k^+, t) = I^{-1}(\mathbf{N}_{ctrl} + \mathbf{N}_{gg} - \boldsymbol{\omega} \times I\boldsymbol{\omega})$$

$$\mathbf{q}_{k+1} = \left[\cos\left(\frac{\bar{\omega}_k \Delta T}{2}\right) \mathbf{1} + \frac{1}{\bar{\omega}_k} \sin\left(\frac{\bar{\omega}_k \Delta T}{2}\right) \Omega_k \right] \mathbf{q}_k, \text{ where } \bar{\omega}_k = \sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2}$$

- Compute $F(\hat{\mathbf{x}}(t), t)$ with (3.55).
- Compute transition matrix, $\Phi_k \approx \mathbf{1} + F(\hat{\mathbf{x}}(t), t)\Delta T + \frac{(F(\hat{\mathbf{x}}(t), t)\Delta T)^2}{2!}$
- Propagation of the state error covariance matrix (3.57), $P_{k+1}^- = \Phi_k P_k^+ \Phi_k^T + Q_{k+1}$

Across measurements $\mathbf{y}_{meas/k+1}$

- Compute $\hat{\mathbf{y}}_{k+1}^b = A_o^b(\hat{\mathbf{q}}_k) \mathbf{y}_{k+1}^o$
- Compute from (3.82) H_{k+1}^-
- Update the Kalman gain (see Eq.(A.7) in Appendix A),

$$K_{k+1} = P_{k+1}^- H_{k+1}^{-T} [H_{k+1}^- P_{k+1}^- H_{k+1}^{-T} + R_{k+1}]$$

- Compute the Innovation Error (3.69), $\mathbf{e}_{k+1} = \mathbf{y}_{meas,k+1}^b \times \hat{\mathbf{y}}_{k+1}^o$
- Compute the perturbation update (3.16), $\delta \mathbf{x}_{k+1} = K_{k+1} \mathbf{e}_{k+1}$
- Update estimate (3.20), $\hat{\omega}_{k+1}^+ = \hat{\omega}_{k+1}^- + \delta \boldsymbol{\omega}_{k+1}$
- Update estimate (3.19), $\hat{\mathbf{q}}_{k+1}^+ = \Xi(\hat{\mathbf{q}}_{k+1}^-) \delta \mathbf{q}_{k+1}$ where $\delta \mathbf{q}_{k+1} = \left[\begin{array}{c} \delta \mathbf{q}_{k+1} \\ \sqrt{1 - \|\delta \mathbf{q}_{k+1}\|} \end{array} \right]$
- Compute H_{k+1}^+ as H_{k+1}^- in step 1 but, $\hat{\mathbf{y}}_{body/k+1}$ is computed using the updated state, $\hat{\mathbf{q}}_{k+1}$.
- Compute the state error covariance matrix (see Eq.(A.7) in Appendix A),

$$P_{k+1}^+ = [\mathbf{1} - K_{k+1} H_{k+1}^+] P_{k+1}^- [\mathbf{1} - K_{k+1} H_{k+1}^+]^T + K_{k+1} R_{k+1} K_{k+1}^T$$

3.1.2 Extended Kalman Filter Intrinsic Problems

The Kalman filter is optimal only when applied to linear dynamic systems. Nevertheless, it can still be used for non-linear systems by linearizing equations that describe the system. However several problems arise:

- the linearization of the process and the measurement models may introduce significant errors;
- the properties of the Kalman filter can not be guaranteed for the EKF, which may lead to divergence of the state estimate error;
- the process noise is treated as Gaussian with known variance which is not usually an accurate description of the estimation errors;
- the process error matrix, Q , is manually tuned to reach confidence in the filter;
- the initial state, x_o , and state error covariance matrix, P_o , are required.

The initialization of the state vector and the P matrix, may be a problem. If the starting point is unknown, the initial state vector is in general very different from the true state, causing the covariance matrix to jump from a large value to a small value or close to zero in one step, often causing the algorithm to diverge. In practice some batch work must be done to obtain a suitable initial P matrix that will not cause the EKF to diverge. Also the symmetry of the state error covariance matrix must be assured while the EKF algorithm is running in order to avoid divergence problems due to numerical inaccuracies. For a real satellite which is supposed to work for long periods, it is desirable to implement algorithms that handle numerical instability with a special concern and are equivalent to the Kalman filter algorithm. According to [14] the most numerical stable algorithms for Kalman filter implementation are given by [4] and [51].

The modeling of a spacecraft is a complex issue because the external forces must be taken into account in the satellite equations of motion, according to Newton's law. LEO satellites have the influence of external torques (owing to solar pressure, solar heating, aerodynamic drag) which are non-linear and vary along the orbit and altitude. In practice they are disregarded due to its difficult modelling which would greatly increase the complexity of the process model. However, they influence the accuracy of the estimated angular velocity during the propagation (2.62) where they are explicitly expressed. Also, the linearization of the equations that describe the system is an approximation that is going to introduce errors in the estimates calculation.

Some of the problems discussed previously were addressed by different authors. Instead of neglecting second and higher order terms coming from the dynamics linearization, Vathsal [53] considered terms up to second order. This approach increases

3.1. RECURSIVE ESTIMATION METHODS

the complexity of the filter and its computational burden. It is only worthwhile to go for second or higher order techniques in case of extreme system nonlinearities.

Another approximation is to assume Gaussian noise with known variance for the observations and the process, which is not an accurate description of the estimation errors. In addition to this, the products of inertia are considered all zero because principal directions are assumed to be aligned with control CS due to the space-craft geometry [55] and there is some uncertainty in the calibration of the principal moments of inertia. In practice, these are systematic errors that cannot be modelled using Gaussian white noise, deviating the estimate from its true parameters. To avoid these problems, the state error and process noise covariance matrices are tuned with the filter in the attitude closed control loop, to compensate the resulting errors. So, a set of 6x6 elements, from P_o matrix, plus 6x6 elements from the Q matrix and $3^n \times 3^n$ for the R matrix, where n is the number of available sensors, must be determined before the EKF can be used. Since these are symmetric matrices, the number of elements is reduced to $\frac{6 \times 6 + 6}{2}$ for each P_o , Q matrices, and $\frac{3^n \times 3^n + 3^n}{2}$ for the R matrix.

Traditionally, all elements were manually tuned which is time consuming, prone to subjective error and complex, since the state variables affect each other. Taking advantage of the symmetry of the state error covariance matrix and of the PoSAT-1 geometry, a reduced group of parameters has to be tuned. Small satellites with gravity gradient stabilisation usually have a preferred spin axis, z axis. Thus, it is reasonable to assume identical covariance errors for the angular velocities *w.r.t.* x and y axis. Using this, the number of elements reduce once again to 17 elements for each matrix P_o and Q .

Instead of manual tuning, [56] used a genetic algorithm to tune the process noise covariance matrix Q , the measurement noise covariance matrix R and the state error covariance matrix P_o at the same time, with the EKF inside the closed control loop, and evaluating performance by measuring the accuracy of the resulting attitude control loop. The total number of elements to be computed is 13 for P_o and Q and $\frac{3^n \times 3^n + 3^n}{2}$ for the R matrix, where it was assumed that the q_1 and q_2 had identical covariance errors as well as q_3 and q_4 . It was necessary to group the parameters to reduce the number of parameters to be found and doing so making the tuning problem manageable. However, this still means many parameters to be tuned and a lot of batch work has to be done in order to find out all parameters needed to correctly run the EKF.

A different approach was taken by Bak [1], who defined the measurement noise covariance matrix, R , by,

$$R_{tot}(t_{k+1}) = R_m(t_{k+1}) + A(\hat{\mathbf{q}}_{k+1}^-)R_b(t_{k+1})A(\hat{\mathbf{q}}_{k+1}^-)$$

where R_m is the sensor noise and R_b is the geomagnetic field model errors. Also, the process noise covariance matrix was modelled taking into account some of the

discrepancies between the process model and the actual behaviour of the spacecraft and was gathered in the following process noise covariance matrix,

$$Q = \begin{bmatrix} Q_{wi} + Q_w + Q_{ws} & 0_{3 \times 3} \\ 0_{3 \times 3} & Q_{sq} \end{bmatrix} \quad (3.83)$$

where Q_{wi} reflects the uncertainty of the moments of inertia and the noise acting on the angular velocity rates which makes this matrix to depend on the actual state of the satellite. The Q_w indicates the influence of the aerodynamic drag. Q_{ws} and Q_{sq} are diagonal process noise parameters, empirically determined to compensate linearizations and inadequacies in the propagation of the state. This is a good approach to the problem of modelling a non-linear system to be used in the EKF since there are very few parameters to be determined. Thus, the process noise covariance matrix does not need to be tuned exhaustively. However these parameters are dependable of the satellite features like calibration of the inertias moments and it takes some time to find out the values suitable to the algorithm.

Also, Steyn [45] has suggested a process error covariance matrix, where the angular velocity is kept constant during the entire life of the satellite and the quaternion error is proportional to the other elements of the quaternion due to the quaternion constraint,

$$Q = \begin{bmatrix} diag(\sigma_{w_x}, \sigma_{w_y}, \sigma_{w_z}) & 0_{3 \times 3} \\ 0_{3 \times 3} & k_q diag(1 - q_1^2, 1 - q_2^2, 1 - q_3^2) \end{bmatrix} \quad (3.84)$$

where σ_{w_x} , σ_{w_y} , σ_{w_z} are tuned for a specific satellite. As explained before due to the geometry of the satellite usually $\sigma_{w_x} = \sigma_{w_y} \neq \sigma_{w_z}$ and k_q is empirically determined. Again the measurement noise covariance matrix was set as a diagonal matrix. For this noise covariance matrices there are 4 parameters to be determined, which is a reduced number comparing with the 32 elements of the covariance usually manually tuned. One may say that these are not the best covariances matrices to get good state estimates results but it is a very good starting point to have the EKF running properly.

3.2 Point-to-point Approaches

The deterministic or point-to point methods consists of determining the attitude matrix, A , on a set of noisy vector measurements. Given a set of minimal two, $n \geq 2$, vector measurements taken by the attitude sensors on-board the satellite, $\mathbf{y}_{meas,1}^b \dots \mathbf{y}_{meas,n}^b$, thus written in body CS and the correspondent set of reference vectors, $\mathbf{y}_1^o \dots \mathbf{y}_n^o$, obtained from the model of the same attitude sensors, thus written on orbit CS. There is a matrix denoted by A , that represents the rotation between

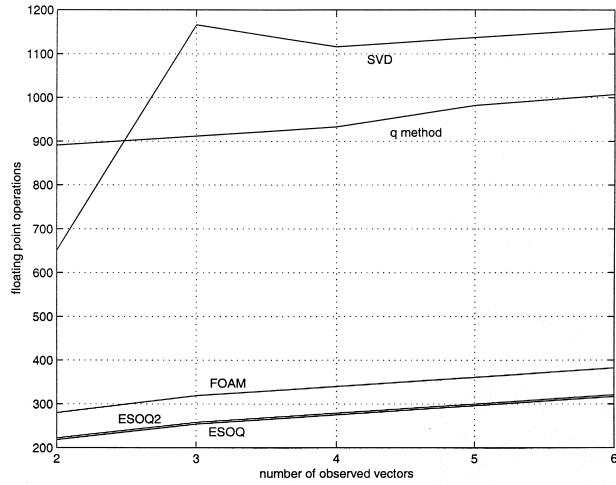


Figure 3.1: Execution Times for Robust Estimation Algorithms (reprinted from [35]).

both orbit and body CS, (in Section 2.2.1), hence relating both vectors,

$$\mathbf{y}_{meas}^b = A_o^b \mathbf{y}_o^o \quad (3.85)$$

Thus the problem resumes to find the orthogonal attitude matrix A , only based on the measurements. Wahba [54] was the first to choose a least square criterion to define the best estimate of A ,

$$L(A) = \frac{1}{2} \sum_{i=1}^n w_i^2 \| \mathbf{y}_{meas,i}^b - A \mathbf{y}_i^o \| \quad (3.86)$$

where $\| . \|$ denotes the Euclidean norm.

So the point-to-point methods consist of alternative approaches to the solution of the problem posed by Wahba in order to determine the best attitude matrix A . Many algorithms were introduced as, described in Section 1.2, but according to Markley and Mortari [35] the most robust, reliable and accurate point-to-point estimators are the q method and the SVD, because all, but these two, involve potentially numerically unreliable procedures [35].

Also, the Davenport solution has proven to be more efficient than the SVD method except for the two attitude sensors case, as depicted in Fig. 3.1 [35].

Taking that into account for the PoSAT-1 case study in this dissertation, where there are only two available attitude measurements, the SVD was chosen in this work.

The derivation of the SVD algorithm is given in appendix B, but the algorithm can be summarized in the following set of equations:

$$\begin{aligned} B &= \sum_{i=1}^n w_i \mathbf{y}_{meas,i}^b \mathbf{y}_i^{oT} \\ B &= USV^T \end{aligned} \quad (3.87)$$

The U and V matrices are orthogonal and

$$S = diag(s_1, s_2, s_3) \quad (3.88)$$

with B singular values s_1, s_2 and s_3 ordered as:

$$s_1 \geq s_2 \geq s_3 \geq 0 \quad (3.89)$$

The attitude matrix is given by,

$$A_{opt} = U \begin{bmatrix} \mathbf{1}_{2x2} & 0 \\ 0 & \det(U)\det(V) \end{bmatrix} V^T \quad (3.90)$$

For the computation of the state error covariance matrix

$$\begin{aligned} \sigma &= \det(U)\det(V)s_3 \\ P &= U \left[diag\left(\frac{1}{s_2+\sigma}, \frac{1}{s_1+\sigma}, \frac{1}{s_1+s_2}\right) \right] U^T \end{aligned} \quad (3.91)$$

The SVD algorithm is very simple to implement, compared to the EKF, since there is no need to give initial values to any state error covariance matrix and to the state vector, to tune any process noise covariance matrix, to linearize the dynamic and kinematic equations of motion, or to assume the process and measurements noises as Gaussian with known covariances. Therefore, the problems described previously are avoided as well as the time consumed to solve them. Also, for different satellites, the EKF covariance matrices must be re-tuned while the SVD algorithm does not. On the other hand, one may say that some useful information from the system model is lost, in the SVD method, but since the system model is non-linear and has to be linearized, it is better to ignore it than to introduce misleading information to the filter.

Nevertheless, a problem arises when the satellite is orbiting in the dark side of the Earth or when the Sun is outside the field of view of the Sun Sensor. At this point no attitude sensors are available except only for magnetometers, therefore the point-to-point methods are incapable of determining the satellite's attitude because they depend on the availability of two measurement vectors. So the propagation of the attitude must be done through the kinematics, but since these gaps are

small periods of time, the incurred error is very small and easily corrected when the SVD algorithm is operational again. The same applies when the estimator attempts to work without the Sun sensor. These issues are addressed in chapter 4, where the SVD algorithm is presented in detail for PoSAT-1. Another critical issue is to obtain estimates of the angular velocity which are needed for some control algorithms. Thus, the angular velocity, not given by any point-to-point method, is obtained through kinematic equations, using the quaternion estimates resulting from the SVD algorithm, as explained next.

Angular Velocity calculation

From the kinematics,

$$\dot{\mathbf{q}} = \frac{1}{2}\Omega(\boldsymbol{\omega}) \mathbf{q}$$

To manipulate the equation, the angular velocity is represented in terms of a quaternion $\boldsymbol{\omega} = [\omega_x \ \omega_y \ \omega_z \ 0]^T$, and using the definition of the quaternion product (2.29), the kinematics (2.63) becomes:

$$\dot{\mathbf{q}} = \frac{1}{2}\boldsymbol{\omega} \otimes \mathbf{q}$$

Multiplying both terms by the inverse of the quaternion, the expression can be written as follows,

$$\boldsymbol{\omega} = 2\dot{\mathbf{q}} \otimes \mathbf{q}^{-1} \quad (3.92)$$

where from (2.23) the inverse of the quaternion is given by $\mathbf{q}^{-1} = [-q_1 \ -q_2 \ -q_3 \ q_4]^T$. Using again the definition of the quaternion product (2.29),

$$\boldsymbol{\omega} = 2\Xi(\mathbf{q}^{-1}) \dot{\mathbf{q}} \quad (3.93a)$$

The discrete expression to be applied between measurements, t_k and t_{k+1} , with a sampling time of ΔT is given by,

$$\boldsymbol{\omega}_{t_{k+1}} = 2\Xi(\mathbf{q}_{t_{k+1}})\dot{\mathbf{q}}_{t_{k+1}} \quad (3.94)$$

The derivative of the quaternion, $\dot{\mathbf{q}}$, is obtained adding a pole, a , to the 'derivator' system that obtains $\dot{\mathbf{q}}$ from \mathbf{q} , to damp the high frequency noise. The transfer function $\frac{s}{s+a}$ is discretized using the bilinear transform, where $s \leftarrow \frac{z}{\Delta T} \frac{z-1}{z+1}$. The discrete expression to be applied between measurements with a sampling time of ΔT is given by the difference equation

$$\dot{\mathbf{q}}_{t_k} = (1 - a\Delta T) \dot{\mathbf{q}}_{t_{k-1}} - a\Delta T \dot{\mathbf{q}}_{t_{k-2}} + \mathbf{q}_{t_k} - \mathbf{q}_{t_{k-1}} \quad (3.95)$$

In addition to this, the propagation of the quaternion between measurements requires the knowledge of the angular velocity and since no gyros are on-board of many small satellites, it is imperative to have angular velocity estimated besides the attitude. These issues will increase the computational load of the SVD algorithm implementation but no additional errors are introduced in the algorithm that will affect its performance estimating the attitude. It can be seen as a complementary algorithm that can be used in small spacecrafts for which the knowledge of angular velocity is essential.

Chapter 4

Simulation Setup

To validate the estimation algorithms described in Chapter 3, a simulated LEO satellites environment, SimSat, is used, where all parameters are controlled and known, as described in Section 4.1. PoSAT-1 was used as the benchmark of the SimSat simulator since it is representative of the small satellite class with hardware constraints, reduced power supply, restrictive actuators and noisy attitude sensors that are not always available, common to most small satellites, and it is addressed in Section 4.2.1. Furthermore, all conditions needed to run the EKF and the SVD algorithms are described, as well as the disturbance variables added on purpose to increase the difficulties of the estimator. Afterwards, the real data obtained from PoSAT-1 satellite is presented, and explained in detail how to handle it, in Sections 4.2.3. Also, the actions performed over the real data to reduce errors and obtain better results are addressed.

The tests performed over the simulated PoSAT-1 are presented in 4.3 and the most important aspects and goals that the tests were meant to stress are explained. Also, the tests applied to the real data of PoSAT-1 satellite is explained in this section. Moreover, the statistical information used to analyse the performance of the estimators algorithms, EKF and SVD, for the simulation case and for the real data is defined and explained in Section 4.4.

4.1 SimSat Simulator

SimSat is a MATLAB/Simulink program developed under the ConSat¹ project which aims at reproducing a realistic environment, as perceived by the ADCS of spacecrafts, in the particular case of small satellites. Since the first year of the project, several researchers and students have been studying and modelling all entities which interact with small satellites in order to simulate the evolution of the

¹Supported by PRAXIS XXI program project PRAXIS/3/3.1/CTAE/1942/95.

satellite attitude motion and its time-varying behaviour [48], [49] [50]. The implementation of dynamic and kinematic equations of motion of a satellite is described in [44]. The SPG4 propagation model, suitable for prediction of near Earth satellites position and velocity was implemented. PoSAT-1 was used as a case study for the SimSat simulator, where PoSAT-1 sensors [48] and actuators were realistically simulated as well as the Earth's orbit geomagnetic field. One of the major problems that ADCS face is that, besides being noisy, the sensors are not always available depending on the relative position of the satellite, the Sun and Earth, [56]. Several control algorithms are available for use in the Consat simulator [46] in order to choose the most adequate to each mission phase for small satellites. The algorithm used in this work is the *Predictive Control* since it is an attitude stabilisation algorithm suitable to stabilise and control the spin of a small satellite using only magnetic actuation [47] and achieved very good performances [46] [47]. The ADS based in an EKF with the initial state error covariance matrix, process noise covariance matrix and the measurement noise covariance matrix tuned through genetic algorithms [6] and an SVD algorithm implementation as well as a comparison between both EKF and SVD were introduced in [36]. Also the SimSat graphical interface was implemented by Tavares [50] and improved by Clements *et. al.* [5]. There is still room for improvement in the SimSat, where new approaches to the attitude control algorithms as well as new attitude estimation algorithms, can be developed, validated and compared to other algorithms based on the simulations.

4.2 PoSAT-1 Experiments

A brief explanation of the PoSAT-1 small satellite in terms of hardware, structure, geometry and power supply is presented next. Also, the PoSAT-1 attitude sensors are described, since the estimation algorithms are based on their measurements, as well as PoSAT-1 actuators. To accomplish the controller requirements the attitude estimator system needs to estimate the full attitude and angular velocity [47]. The PoSAT-1 simulation environment is explained and the setup parameters presented, in Section 4.2.2, as well as the estimator algorithms EKF and SVD. The real data obtained to be used by the estimator is described in Section 4.2.3. Since it was only possible to get Sun sensor measurements from PoSAT-1 just the EKF, described in Appendix A.7, is applied to PoSAT-1 real data.

4.2.1 PoSAT-1 Technical Data

PoSAT-1² is the first Portuguese satellite in orbit, developed in a technology transfer program between the University of Surrey and a Portuguese industrial and educational consortium lead by INETI. It is an experimental small satellite for technology demonstrations, scientific and educational purpose, which has benefited from previous research on UoSAT satellites. It is an Earth Observation satellite which carries a wide and narrow angle CCD camera with a resolution of 200m/pixel to take Earth pictures.

PoSAT-1 was launched along with the primary payload, SPOT-3 satellite, by Ariane-4 ASAP-4 V59 launch vehicle from Korea, French Guiana on the 26th September 1993 at 01:45:00 UTC, along with six other micro-satellites, into an 822x800 km, with 98.6° inclination.

PoSAT-1's on board computer is based on an INTEL 80C186 processor running at 8 MHz with 512 Kbytes of RAM, interfacing with 16 Mbytes RAMDISK. Its secondary computer is based on an INTEL 80C188 processor running at 8 MHz with 512 Kbytes RAM. This limited on-board computing power justifies the quest for ADS algorithms which are moderate consumers of CPU power.

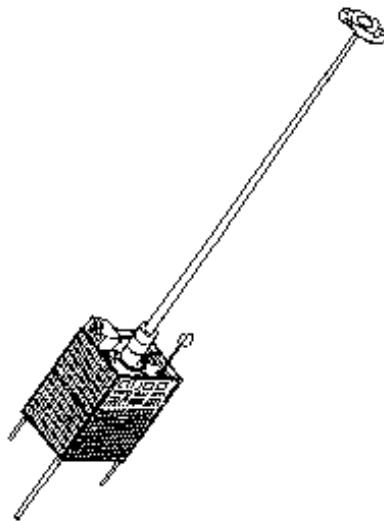


Figure 4.1: PoSAT-1 design.

²Some of the information was obtained from the INETI Aerospace Lab (*Instituto Nacional de Engenharia e Tecnologia Industrial*), at <http://www.laer.ineti.pt/posat> and Surrey University, at <http://www.ee.surrey.ac.uk>.

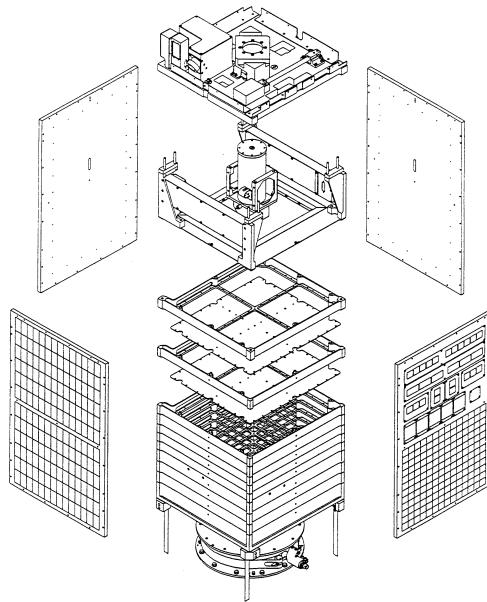


Figure 4.2: Expanded view of PoSat-1.

The satellite is parallelepiped in shape with dimensions $35 \times 35 \times 58 \text{ cm}^3$ and has a final mass just under 60 kg, with one mass tipped boom for passive stabilisation and an orbit in altitude around 800 km. It has solar panels mounted on the four sides. To assure the mean temperature of 10°C at the battery modules and a distribution of Sun irradiance onto the solar panels, in order to optimise their efficiency, PoSAT-1 has to perform a very slow spin about its z -axis. The magnetorquer used is composed of three wire coils around the satellite geometrical axes. During the sampling period each coil may be fired, one at a time in one direction, x_b , y_b or z_b , for up to three seconds with positive or negative polarity and 0.1 second resolution. PoSAT-1 has an internal and an external 3-axis fluxgate magnetometer. The internal magnetometer was used only until the deployment of the gravity gradient boom. PoSAT-1 has two Sun sensors with each sensor having two channels. The PoSAT-1 microsatellite has an experimental GPS Navigation Unit. It consists of one GPS antenna mounted on the face of the satellite and a receiver which interfaces with the transputer processing unit.

PoSAT-1 Attitude Sensors

The attitude estimation is supported by various data measurements. For very high attitude accuracy, it is important to use the most accurate attitude sources but there are advantages and disadvantages associated to each sensor, especially in small

satellites with so few available resources. PoSAT-1 has the following attitude sensors: two single-axis sun-sensors; two Sun detectors; one Earth underneath detector; two horizon sensors, two magnetometers and one currently non-functional star sensor.

Sun Sensor

This sensor reads the angle between the spin axis of the satellite and the Sun or the *Sun-angle* shown in, Fig. 4.3. PoSAT-1 has two single axis analogue Sun

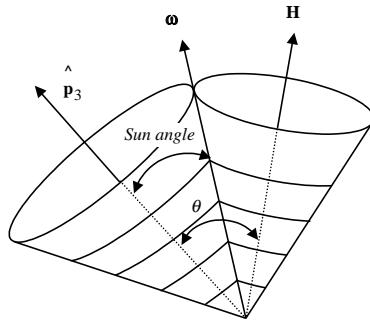


Figure 4.3: Motion of a nutating Spacecraft. The body cone rolls on the spacecraft cone for $I_1 = I_2 > I_3$. \hat{p}_3 is the principal axis of inertia, ω is the instantaneous rotation axis, θ is the *nutation angle* and \mathbf{H} is the angular momentum.

sensors. The exact design is unknown to the author, so the implemented model of the Sun sensors had to be based in common designs [55] and on telemetry from them [5]. The technical documentation states that each of the two cells per sensor has a total field view of 120° and based on the telemetry data the angle of inclination of the sensors, $\alpha = 30^\circ$ and the Field Of View (FOV) limits of each cell of the sensors is 60° , is derived in [5].

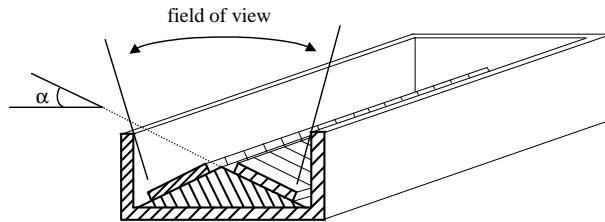


Figure 4.4: Single axis analogue Sun sensor.

Since the Sun is not visible during parts of the orbit, the Sun sensor information is not always available due to the satellite libration or the satellite orbiting in the dark

side of the Earth. This leads to the Sun sensor being the second sensor measurement option in the EKF.

Earth Horizon Sensor - EHS

This sensor determines the current view of the Earth, given the relative position of the sun, Earth and the satellite. The problem is that the camera has a narrow field of view, which difficulties Earth localization. Furthermore, it can be looking into space or the Earth's dark side. Moreover, the sensors may be *albedo* or *infrared (IR)* based sensors. These features make the sensors susceptible to errors due to strong variation in albedo for different refracting surfaces or by the temperature of the earth's surface and by the chemical composition of the atmosphere. These add errors to the EHS model, making it difficult to model. It was checked from telemetry data that the EHS is very sensitive to attitude and to atmospheric conditions, [5]. It is available in the light side of Earth and even so sometimes pointing into space.

GPS Antenna

When switched on, the Global Position System (GPS) is used mainly to time synchronisation and on-board orbital position determination, as well as velocity with high accuracy. In order to include GPS information in the EKF, the number of states has to increase causing a shortage in the program size, memory requirements and also processing time. This is the reason why GPS information has been left for future work.

Star Sensor

The Star sensor camera was not included in the simulator because the data rate is too low to be used in the control loop except for calibration purposes. So, the camera is not being used for any ADCS purpose but only to take pictures of Earth.

Magnetometers

PoSAT-1 has three external magnetometers sensors which, when combined, are used to measure the local geomagnetic field vector (magnitude and direction). This data is compared to a model of the geomagnetic field in order to determine the attitude. Using magnetometer information has the big advantage of being available throughout the whole orbit as well as the low power requirements, lightweight and inexpensive. However, is not very accurate due to errors in the IGRF models. There are factors that the models do not take into account like the influence of solar activity [55]. However, they are widely used for attitude determination as the main sensor due to its availability.

PoSAT-1 Actuators

The passive stabilisation of the satellite is based on the deployed gravity gradient boom, which generates a stabilising moment, (2.70) and magnetorquers (electromagnetic coils) on each of the satellites sides which interact with the Earth's geomagnetic field creating a magnetic moment, (2.73) to damp the libration motion. Earth's magnetic field was simulated using a spherical harmonic IGRF model [55]. The satellite design of PoSAT-1 and other satellites of the UoSAT class, have restricted the values of the control magnetic moment to only three different values of positive/negative polarity. Combining this restriction with the single-coil-actuation the available set of magnetic moments is reduced to only 18 different values (6 for each axis). In addition to this for each actuation on a coil there must be at least a back-off time of 100 seconds to recharge the power supplies. In practice it means that the actuators have a maximum duty cycle of 3%, since the maximum actuation time is only 3 seconds.

4.2.2 PoSAT-1 Simulations

PoSAT-1 is the case study simulated in the SimSat program with a sampling time of 1 second. All the details related to the implementation of PoSAT's environment, *e.g.* orbit parameters, attitude sensors, inertial parameters or systematic errors introduced on purpose to increase the difficulties of the estimators, are explained in the *SimSat Simulator Setup* Section. The simulation setup conditions used for both algorithms, EKF and SVD, which are presented in the *Estimation Algorithms Setup* Section, as well as other issues related to the estimation algorithm implementation. Also, a detailed explanation of how problems were solved in the implementation of the algorithms is included.

Attitude Sensors Simulation Setup

The PoSAT-1 attitude sensors simulated in SimSat are the magnetometers and Sun sensors. Magnetometers have its measurements always available. The magnetometer measurement is simulated as the rotation from OCS to BCS of the local magnetic field, given by the geomagnetic field at some point in orbit, mapped by the true attitude matrix. The Earth's magnetic field, is simulated according to the International Geomagnetic Reference field (IGRF) model [55]. The spherical harmonic IGRF model is calculated till 10th order. Results obtained from the simulator by comparison between the real data and simulated can be found in [48]. The Sun sensor is modelled according to the position of the Sun and Earth as seen from the satellite as described in [5].

Two non parallel directions are needed to uniquely determine the orientation of

a satellite from space, usually a known direction measured by the attitude sensors and a direction given by a model of the sensor based in the satellite's position (reference vector). The magnetometer reference vector is modelled as the IGRF model computed till 4th order and the magnetometer measurement is computed in the SimSat and also modelled as the IGRF model computed till 10th order. Thus, the difference between the measurement given by the magnetometer sensor and the magnetometer reference vector, simulates the real error. For the Sun sensor model a random noise with a normal distribution with a mean of 0.0 and a standard deviation of 1.0×10^{-5} is added, to the measurements of the simulated Sun sensor [5].

Equations of Motion Simulation Setup

The attitude motion of the satellite is modelled according to the Euler equations of motion (2.63), (2.62). Where, the external torques simulated for the PoSAT-1 are the gravity moment (2.70) written in control CS and the control moment which results from the interaction between the geomagnetic field and the satellite's magnetic field (2.73). Besides these, other important external torques acting on a spacecraft are solar radiation pressure and aerodynamic drag. According to the discussion carried out in 2.4.3, these torques are neglectable. Thus, for PoSAT-1 satellite, the dynamic equations reduce to the following expression,

$$\dot{\boldsymbol{\omega}}_{bi}^b = I^{-1} (\mathbf{N}_{gg} + \mathbf{N}_{ctrl} - \boldsymbol{\omega}_{bi}^b \times I\boldsymbol{\omega}_{bi}^b) \quad (4.1)$$

A spacecraft orbiting around Earth has an elliptic orbit, according to Kepler's first law because the satellite continuously interacts gravitationally with the Earth. For LEO satellites, where the satellite's mass is neglected compared to that of Earth, the orbit eccentricity is very close to circular orbit, $e = 0$. Because of that, the orbit simulated in SimSat is circular.

Due to PoSAT-1 axially symmetric geometry, the body CS axes are along the principal moments of inertia axes resulting in null products of inertia. The principal moments of inertia tensor has the following values:

$$I_{xx} = I_{yy} = 119.1 \neq I_{zz} = 0.784 \quad (4.2)$$

$$I_{xy} = I_{yx} = I_{xz} = I_{zx} = I_{yz} = I_{zy} = 0 \quad (4.3)$$

Since these are approximately values and the deployment of the boom along the z_b -axis causes changes on it and to introduce some uncertainty, the moment of inertia tensor was simulated in SimSat using the following inertia matrix ,

$$I = \begin{bmatrix} 119.14 \pm 0.05 & \pm 0.0005 & \pm 0.0005 \\ \pm 0.0005 & 119.06 \pm 0.05 & \pm 0.0005 \\ \pm 0.0005 & \pm 0.0005 & 0.78 \pm 0.05 \end{bmatrix} \text{kg.m}^2 \quad (4.4)$$

Orbital Parameters Simulation Setup

The orbit generation in the SimSat was done with the United States Space Command (USSPACECOM) SGP4 mathematical model for prediction of satellite position and velocity. The orbit determination of the PoSAT-1 was obtained from a NORAD Two-Line Element (TLE) (Appendix D) which has the format shown in Table 4.1.

PoSAT-1									
1	22829U	93061G	98051.65721957	+.00000069	+00000-0	+44725-4	0	6120	
2	22829	98.5167	125.5480	0009163	216.4411	143.6151	14.28203542229593		

Table 4.1: Two-Line Element of PoSAT-1 used in SimSat environment.

The TLE gives the following Keplerian orbital elements, (Appendix C), valid for a certain time (Epoch), Table 4.2.

<i>Epoch</i>	1998/02/2015:46:23.7709 UTC
<i>i</i>	98.5167°
<i>Bstar</i>	0.000044725
Ω	125.5480°
<i>e</i>	0.0009163
ω_p	216.4411°
M_o	143.6151°
<i>MM</i>	14.28203542 revs/day

Table 4.2: NORAD elements used of the PoSAT-1 simulation in Simsat.

Since the eccentricity is very close to zero the orbital geometry of the satellite is considered to be circular. Thus the angular orbital rate, ω_o of PoSAT-1 for this epoch, is approximately constant and given by,

$$\omega_0 = MM \cdot \frac{2\pi}{24 \cdot 60 \cdot 60} rad/s \quad (4.5)$$

$$= 1.038618925 \times 10^{-3} rad/s \quad (4.6)$$

and the period,

$$T = \frac{1}{\omega_0 [revs/s]} \approx 6050 \text{ sec}$$

Extended Kalman Filter Simulation Setup

The EKF is implemented according to the algorithm described in Section 3.1.1. The only remarks concern the principal moments of inertia used,

$$I = \text{diag} (I_{xx} \ I_{yy} \ I_{zz}) = \text{diag} (119.1 \ 119.1 \ 0.784) \quad (4.7)$$

according to (4.2), and the calculation of H matrix (3.82) which is expanded to a rank of 6x6 in order to incorporate also the Sun sensor measurement, s_{meas} , when this sensor is available

$$H_k^+ (\hat{\mathbf{x}}_k) = \begin{bmatrix} 0_{3 \times 3} & \tilde{H}_k(\hat{\mathbf{b}}_k^b) \\ 0_{3 \times 3} & \tilde{H}_k(\hat{\mathbf{s}}_k^b) \end{bmatrix} \quad (4.8)$$

$\tilde{H}_k(\hat{\mathbf{y}}_k^b)$ is computed according to (3.79).

EKF Initial Conditions Setup

According to the discussion in 3.1.2, and after some batch work in the simulated environment it was concluded that the initial state error covariance values suitable to the angular velocity are 10^{-3} and for the quaternions 10. Thus, the P_o matrix is diagonal with elements:

$$P_o = \text{diag} (10^{-3}, 10^{-3}, 10^{-3}, 10, 10, 10) \quad (4.9)$$

The initial attitude variable has been set up for two different situations, see Section 4.3. An error is added to the true initial state of the angular velocity

$$\boldsymbol{\omega}_{oi}^b = A_o^b(q) [\omega_0 \ 0 \ 0]^T \quad (4.10)$$

and to the initial roll, pitch and yaw true angles. The other situation is for the case that the angular velocity and Euler angles were kept equal to its true value.

EKF Measurement noise Covariance Matrix Setup

The measurement noise covariance matrix (3.64) for magnetometers was calculated based in the root mean square error between the measurements taken with the IGRF 10th order model and 4th order model in 20 simulations of 16 orbits each. Thus, the measurement noise covariance matrix obtained, assuming non-correlated parameters is the following,

$$R_b = \text{diag} (10^{-2}, 10^{-2}, 10^{-2}) \quad (4.11)$$

When the Sun sensor measurements are available the measurement covariance matrix is expanded to a 6x6 matrix, $R_{tot} = \begin{bmatrix} R_b & 0_{3 \times 3} \\ 0_{3 \times 3} & R_s \end{bmatrix}$. The measurement noise covariance matrix for Sun measurements was calculated based on the root mean square error between the measurements taken from the Sun sensor model and its measurements for 20 simulations of 16 orbits each, leading to the following values,

$$R_s = \text{diag}(1.0, 1.0, 1.0) \quad (4.12)$$

This variance values are worse than the magnetometers, since the Sun sensors data is considered to be less precise than the magnetometers, as results from comparing the PoSAT-1 telemetry and the Sun sensor model data in [5].

EKF Process noise Covariance Matrix Setup

The process noise covariance matrix used is the one adopted in [45] and explained in Section 3.1.2,

$$Q = \text{diag}(10^{-5}, 10^{-5}, 10^{-3}, 10^{-6}(1 - q_1^2), 10^{-6}(1 - q_2^2), 10^{-6}(1 - q_3^2)) \quad (4.13)$$

SVD Magnetometers and Sun sensors Setup

Determining A_{opt} - The SVD method needs at least two noisy vector measurements to determine the estimate of the Attitude matrix. Assuming that the Sun sensor is available, the observation vector is composed by the measurements of both the magnetometers and the Sun sensor, $\mathbf{y}_{meas}^b = [\mathbf{b}_{meas} \ \mathbf{s}_{meas}]$. Thus, the Attitude matrix is obtained, according to the SVD algorithm presented in Section (3.2), as follows: from (3.87) computing the B matrix with the measurements of the sensors and the respective reference vectors,

$$B = \mathbf{b}_{meas} (\mathbf{b}^o)^T + \mathbf{s}_{meas} (\mathbf{s}^o)^T \quad (4.14)$$

then, the B matrix is transformed, through the singular value decomposition method, in the U_b , S_b and V_b orthogonal matrices which will allow the computation of the attitude matrix (3.90) and the state error covariance matrix (3.91).

Determining q - the quaternion estimate must be obtained from the attitude matrix. A set of four solutions can be determined from (2.35) considering for the first set that q_4 is the first to be found; q_1 for the second set; q_2 for the third set and q_3 for the last set. As for the first set,

$$\begin{aligned} q_4^1 &= \pm 0.5\sqrt{1 + A_{11} + A_{22} + A_{33}} \\ q_1^1 &= 0.25(A_{23} - A_{32})/q_4^1 \\ q_2^1 &= 0.25(A_{31} - A_{13})/q_4^1 \\ q_3^1 &= 0.25(A_{12} - A_{21})/q_4^1 \end{aligned} \quad (4.15)$$

However numerical inaccuracies may arise when q_4 is very small. One way to overcome this is to compute the maximum of $1 + A_{11} - A_{22} - A_{33}$, $1 - A_{11} + A_{22} - A_{33}$, and $1 - A_{11} - A_{22} + A_{33}$ and based on this, switch among solutions, as suggested by Sidi [42]. The three other solutions are,

$$\begin{aligned} q_3^4 &= \pm 0.5\sqrt{1 - A_{11} - A_{22} + A_{33}} \\ q_1^4 &= 0.25(A_{13} + A_{31})/q_3^4 \\ q_2^4 &= 0.25(A_{23} + A_{32})/q_3^4 \\ q_4^4 &= 0.25(A_{12} - A_{21})/q_3^4 \end{aligned} \quad (4.16)$$

$$\begin{aligned} q_2^3 &= \pm 0.5\sqrt{1 - A_{11} + A_{22} - A_{33}} \\ q_1^3 &= 0.25(A_{12} + A_{21})/q_2^3 \\ q_3^3 &= 0.25(A_{23} + A_{32})/q_2^3 \\ q_4^3 &= 0.25(A_{31} - A_{13})/q_2^3 \end{aligned} \quad (4.17)$$

$$\begin{aligned} q_1^2 &= \pm 0.5\sqrt{1 + A_{11} - A_{22} - A_{33}} \\ q_2^2 &= 0.25(A_{12} + A_{21})/q_1^2 \\ q_3^2 &= 0.25(A_{13} + A_{31})/q_1^2 \\ q_4^2 &= 0.25(A_{23} - A_{32})/q_1^2 \end{aligned} \quad (4.18)$$

However due to the inaccuracies of the attitude matrix, the sign of the quaternions changes suddenly when the algorithm is running, as can be seen from Fig. 4.5.

In order to avoid this problem, a constraint is added to the algorithm results that converts the attitude matrix into the quaternions, forcing the sign of the maximum (q_1 , q_2 , q_3 , q_4) to be the same as the previous. For instance from Fig.4.6 shows that the vector element of the quaternion q_3 , which is the maximum quaternion component, changes its sign from one step time to the next.

Determining ω - Since some algorithms need the angular velocity and to achieve proper quaternion propagation between two sampling times, when the SVD does not have two attitude sensors available, the angular velocity is computed, using as initial condition the inverse of the kinematic equation, in Section 3.2 - Eq. (3.93a),

$$\mathbf{w} = 2\Xi(\mathbf{q}^{-1}) \dot{\mathbf{q}}$$

where $\dot{\mathbf{q}}$ is obtained from Eq. (3.95), in Section 3.2.

SVD in the absence of Sun sensors

If the satellite is hidden behind the Earth not seeing the Sun or if the attitude of the satellite does not permit the Sun sensor to have any sunlight passing through the

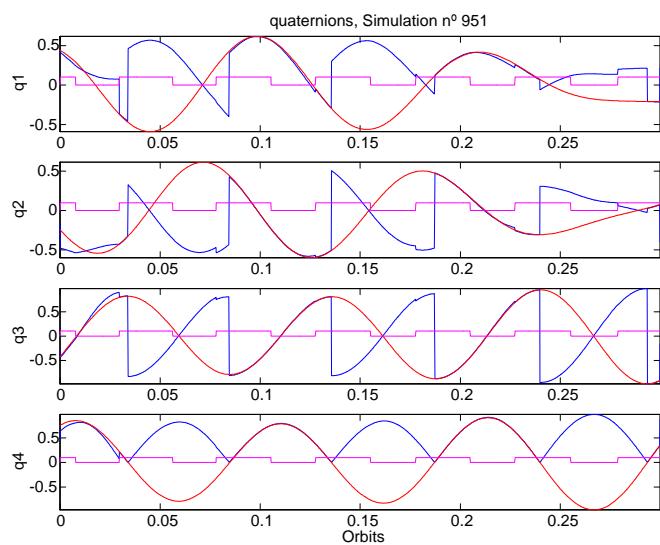


Figure 4.5: The picture shows the quaternions obtained from the attitude matrix estimates using (4.15) - (4.18). The blue line is referred to the estimates and the red line is referred to the true data. For visualization of the SVD method, a signal indicating when Sun sensor measurements are available is high and low otherwise, originating a square wave in all plots.

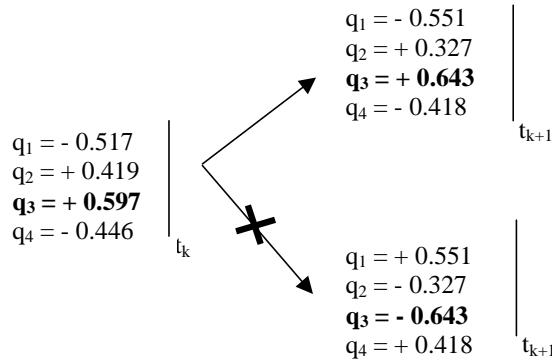


Figure 4.6: From t_k to t_{k+1} the sign of the maximum value found can not change.

PoSAT
 1 22829U 93061G 00307.69679095 +.00000744 +00000-0 +31149-3 0 8437
 2 22829 98.3895 0.6205 0009063 194.5985 165.4938 14.28865027370367

Table 4.3: The Two-Line Element of PoSAT-1 used for real data experiments.

lit of the sensor, then the SVD method is not able to estimate the Attitude matrix. However the estimates from the angular velocity and the attitude are needed for the ADCS to carry on. Thus the choice is to propagate the state variables through the equations of motion (4.1) and (2.63).

4.2.3 PoSAT-1 Real Data Experiments

In this section the real data obtained from the PoSAT-1 satellite is presented. The Keplerian orbital elements used to generate the reference vectors needed for the estimator algorithm and other relevant parameters are addressed. Since only magnetometer measurements were obtained from PoSAT-1, only the EKF is applied to the data. Practical issues related with the way the real data was treated and the EKF algorithm setup are also explained.

Orbital Parameters

The orbit determination of PoSAT-1 has been obtained from a NORAD TLE (Appendix D) which has the following format,

The TLE gives the following Keplerian orbital elements, (Appendix C), valid for a certain time (Epoch),

<i>Epoch</i>	2000/11/02 16:43:23 UTC
<i>i</i>	98.3895°
<i>Bstar</i>	0.00031149
Ω	0.6205°
<i>e</i>	0.0009163
ω_p	194.5885°
M_o	165.4938°
<i>MM</i>	14.28865027 revs/day

Table 4.4: NORAD elements used of the PoSAT-1 real data.

The angular orbital rate, ω_o of PoSAT-1 at these epoch, is approximately given by,

$$\omega_0 = MM \cdot \frac{2\pi}{24 \cdot 60 \cdot 60} rad/s \quad (4.19)$$

$$\omega_0 \approx 0.00104 rad/s \quad (4.20)$$

and the period,

$$T = \frac{1}{\omega_0 [\text{revs/s}]} \approx 6046.75 \text{ sec} \quad (4.21)$$

Real Data

The real data from PoSAT-1 starts at UTC 2000/11/02, 00:00:34, and lasts for 84960 seconds or about 14 orbits, with a sampling time of 20 seconds. The non-normalized data obtained is the magnetometer measurements plotted in Fig. 4.7 and the average magnetic moment in Fig. 4.8. The electromagnetic coils are actuated, \mathbf{m}_{\max} , for a period of time (firing duration), different from to the sampling time. Thus, the magnetic moment used to compute the control moment in Eq. (3.36), must be computed as an average of magnetic moment fired by the coils plus its firing duration per sampling time,

$$\mathbf{m}_{aveg}(t) = \frac{\text{firing duration}}{\text{sampling time}} \mathbf{m}_{\max}(t) = \frac{1}{20} \mathbf{m}_{\max}(t) \quad (4.22)$$

Gyros are not available on-board PoSAT-1 so, there is no reference vector concerning the angular velocity and by which the estimated angular velocity can be compared with.

4.2. POSAT-1 EXPERIMENTS

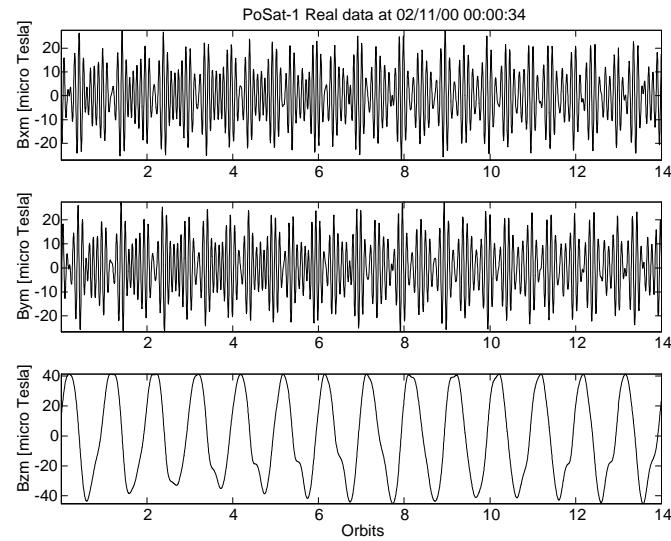


Figure 4.7: PoSAT-1 real data measured by the magnetometers [μT], at 2000/11/02, 00:00:34.

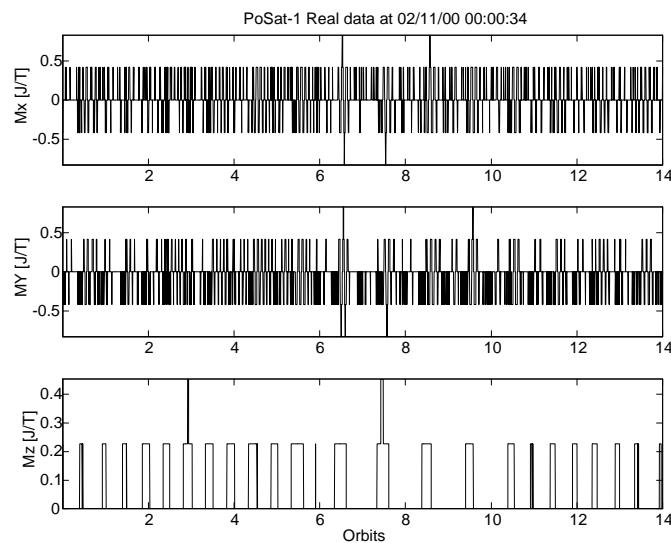


Figure 4.8: PoSAT-1 real vector magnetic moment m [Am^2], from all coils, starting at 2000/11/02, 00:00:34.

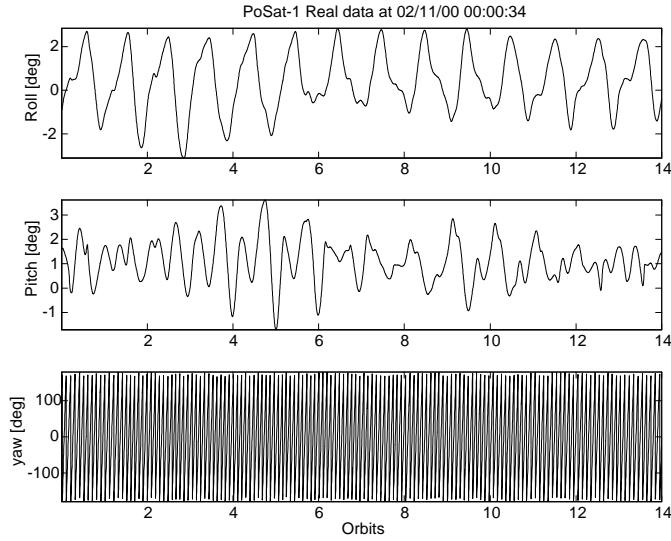


Figure 4.9: PoSAT-1 real attitude, Euler angles: roll, pitch and yaw, at 2000/11/02, 00:00:34.

The orbit determination of the PoSAT-1 was computed 60168.7 seconds before NORAD epoch and 84960 seconds afterwards in order to obtain the IGRF 10th order reference field components starting at the same time as the rest of the data, 2000/11/02, 00:00:34 for 14 orbits, see Fig. 4.10.

The star sensor is not being used to compute the satellite attitude, however the PoSAT-1 has an on-board ADCS, [16]. It is implemented an EKF based only in magnetometers data. Its design assumes that the *roll* (ψ) and *pitch* (θ) angles, according to the CS defined in Section (4.2.3), are small and the equations are only valid for LMZ body axes. This CS is defined as a non-spinning CS and it is identical to the two sequence Euler rotation 21 where the first rotation is the *pitch* (θ) angle around the y -axis and the second rotation, the *roll* (ψ) angle, is around the x -axis. The state vector estimated is $\mathbf{x} = [\psi \ \theta \ \phi \ \dot{\psi} \ \dot{\theta} \ \dot{\phi}]^T$ and for an axially symmetric satellite which experiences small pitch and roll angles the equations of motion are set to the following:

$$\begin{aligned}\ddot{\psi} &= -4(1-k)\omega_o^2\psi + k\omega_o\dot{\phi} + \frac{N_x^{MT}}{I_t} + w_x \\ \ddot{\theta} &= -3(1-k)\omega_o^2\theta + k\omega_o\dot{\theta} + \frac{N_y^{MT}}{I_t} + w_y \\ \ddot{\phi} &= -\omega_o\dot{\psi} + \frac{N_z^{MT}}{I_t} + w_z\end{aligned}\quad (4.23)$$

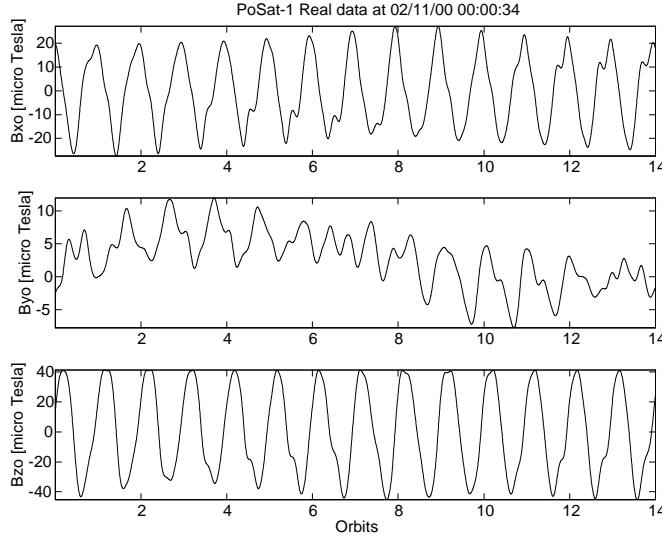


Figure 4.10: PoSAT-1 IGRF 10th order reference field components [μT], at 2000/11/02, 00:00:34, in orbital CS.

where $I_t \equiv I_x \equiv I_y \neq I_z$, $k = \frac{I_z}{I_t}$, N_j^{MT} is a torque induced by the magnetorquers only, and $\mathbf{w} = [w_x \ w_y \ w_z]^T$ is a zero mean noise vector.

Thus, estimation of the roll, pitch and yaw angles were supplied to be used as comparison with the EKF estimates described in this work (Fig.4.9). The data supplied has a different CS from the one defined in Section 2.1 which is presented in Fig. 4.11. The Euler angles roll (ψ), pitch (θ) and yaw (ϕ) are rotations about the x -axis, followed by a rotation about the y -axis and end with the z -axis like in Section (2.2.3). However, the sequence of rotations used is Euler-213. This means a sequence of positive rotations about the y -axis followed by the x -axis and end with the z -axis in the new CS. Thus, the DCM matrix is,

$$A_{213}(\theta, \psi, \phi) = \begin{bmatrix} \cos \phi \cos \theta + \sin \phi \sin \psi \sin \theta & \cos \phi \sin \psi & -\cos \phi \sin \theta + \sin \phi \sin \psi \cos \theta \\ -\sin \phi \cos \theta + \cos \phi \sin \psi \sin \theta & \cos \phi \cos \psi & \sin \phi \sin \theta + \cos \phi \sin \psi \cos \theta \\ \cos \psi \sin \theta & -\sin \psi & \cos \psi \cos \theta \end{bmatrix} \quad (4.24)$$

EKF Changes

The average magnetic moment is used, from (4.22), together with the magnetometers sensor measurements to obtain the control moment (2.73) to be used in the EKF,

$$\mathbf{N}_{ctrl} = \mathbf{m}_{aveg}(t) \times \mathbf{b}_{meas}^b 10^{-6} \quad (4.25)$$

The magnetometer measurements and the respective reference vector are then

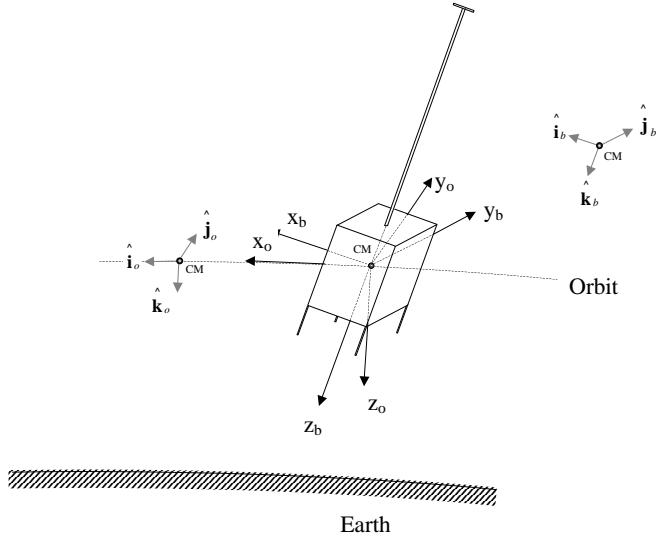


Figure 4.11: Orbital Coordinate System (OCS) and Body Coordinate System (BCS) for PoSAT-1 real data.

normalized,

$$\mathbf{b}_{meas}^b = \frac{\mathbf{b}_{meas}^b}{\|\mathbf{b}_{meas}^b\|}, \quad \mathbf{b}^o = \frac{\mathbf{b}^o}{\|\mathbf{b}^o\|} \quad (4.26)$$

to be used in the innovation step of the EKF algorithm (3.69).

The CS defined in Section 2.1 has been changed to the following: the inertial z axes is pointing in *nadir* direction, the x axis is tangent to the orbit with the same direction as the orbital velocity and y completes the orthogonal CS. The angular velocity of the orbit CS *w.r.t.* inertial CS written in body CS, (2.67) for this CS definition is

$$\boldsymbol{\omega}_{oi}^b = A_o^b(\mathbf{q}) \boldsymbol{\omega}_{oi}^o = A_o^b(\mathbf{q}) \begin{bmatrix} 0 & -\omega_o & 0 \end{bmatrix} = -\omega_0 \hat{j}_o \quad (4.27)$$

hence, the angular velocity used in the kinematic equations is computed according to (2.68),

$$\boldsymbol{\omega}_{bo}^b = \boldsymbol{\omega}_{bi}^b + \omega_0 \hat{j}_o \quad (4.28)$$

For a circular orbit the $\frac{\mu}{R_s^3}$ element from the gravity gradient torque (2.70) can be approximated to a constant denoted by square of the orbit angular rate, ω_0^2 . For

elliptical orbit and because we are dealing with real data from a satellite the position vector of the satellite mass center *w.r.t.* Earth center, R_s is computed based in the propagation of the NORAD elements of the satellite for the time by which the data is collected. Thus, instead of having ω_0^2 in the gravity gradient, the R_s is computed for each time step through the SGP4 model.

EKF Setup

The principal moments of inertia used are equal to those presented in Eq. (4.7).

Initial Conditions

According to the discussion in (3.1.2), the initial state error covariance matrix is diagonal with elements:

$$P_o = \text{diag} (10^{-2}, 10^{-2}, 10^{-1}, 10^2(1 - q_1^2), 10^2(1 - q_2^2), 10^2(1 - q_3^2)) \quad (4.29)$$

The initial attitude angular velocity has been set up to

$$\boldsymbol{\omega}_{oi}^b = [0 \ 0 \ 0]^T \quad (4.30)$$

while the quaternion was added an error of 50% to the first value given by the roll/pitch/yaw angles,

$$\psi_o = -0.91^\circ \cdot 1.5, \theta_o = 1.6^\circ \cdot 1.5, \phi_o = 136.1^\circ \cdot 1.5 \quad (4.31)$$

and from (4.24) and (2.35) the quaternions have the following values,

$$\mathbf{q} = [-0.0230 \ -0.0073 \ -0.9775 \ 0.2094]^T \quad (4.32)$$

Measurement noise Covariance Matrix

The measurement noise covariance matrix (3.64) for magnetometers was used as in Eq. (4.11).

Process noise covariance matrix

The process noise covariance matrix used is the one adopted by [45] and explained in Section 3.1.2, but with σ_{ω_x} , σ_{ω_y} , σ_{ω_z} tuned for the PoSAT-1 real data and k_q parameters,

$$Q = \text{diag} (10^{-7}, 10^{-7}, 10^{-5}, 10^{-4}(1 - q_1^2), 10^{-4}(1 - q_2^2), 10^{-4}(1 - q_3^2)) \quad (4.33)$$

Estimator only			Estimator in the Loop			} SimSat Simulation	
Without/SS	With/SS		Without/SS	With/SS			
EKF	EKF	SVD	EKF	EKF	SVD		
Init 1	A	B	C	D	E	F	
Init 2	G	H		J	K		
PoSAT-1	ZP						

Table 4.5: The several tests performed to the data in this work. The shortname Init is referred to the initial conditions proposed.

4.3 Tests

A set of simulation tests were set up to obtain the attitude estimation performance in different scenarios for the EKF and SVD point-to-point methods and to compare them. The tests are divided in two main groups: open loop and closed loop. In open loop, the satellite has no control moment actuating on it and it experiences a motion ruled by the equations of motion. In this case the estimators attempt to follow the real attitude and angular velocity. In closed loop the estimator feeds the control with estimates of the satellite state. The Predictive Controller, *vd.* (4.1), which is influencing the state of the satellite, aims to damp or eliminate the satellite libration disturbed by a large initial angle *w.r.t.* the local vertical, see Tables 4.8 and 4.9. For this stability test, the satellite already has its boom deployed in the *zenith* direction but has libration movement and spins. For the estimators the closed loop is more demanding since the control algorithm is not aware of the real state but fed by the estimator algorithm results. From the estimation algorithm standpoint the goal is to follow the true state as accurately as possible with or without the control influence.

For each of these groups described before, there is the need to analyse when both magnetometers and Sun sensors are available because it restricts the applicability of the estimators. With both attitude sensors available, the SVD and EKF algorithms are able to work, but when only magnetometers measurements are available the range of estimators algorithms is reduced to the EKF. Thus, the EKF based on both attitude sensors have to be compared with the SVD and also be compared with the EKF based only on magnetometer measurements, as summarized in Table 4.5.

The EKF tests were also divided according to different initial conditions of the state vector. Two initial conditions were considered for them: initial values of the state variables without error and with an error added to the angular velocity and

roll/pitch/yaw at the same time, in order to test the capability of the estimator algorithms to follow the true state, in Table 4.6.

Init. 1 : $\omega_o = \omega_{\text{true}}$ and $\psi_o, \theta_o, \phi_o = \psi_{\text{true}}, \theta_{\text{true}}, \phi_{\text{true}}$
Init. 2 : $\omega_o = \text{Error}_{\omega} \cdot \omega_{\text{true}}$ and $\psi_o, \theta_o, \phi_o = \text{Error}_{\text{Euler}} \cdot (\psi_{\text{true}}, \theta_{\text{true}}, \phi_{\text{true}})$

Table 4.6: The initial state conditions applied to the PoSAT-1 simulation.

The values for the errors, $\text{Error}_{\text{Euler}}$ and Error_{ω} , will be presented in the next Chapter. The SVD algorithm does not need to be tested with an error added to the initial state vector since the state vector does not need to be initialized.

Also, common to each simulation test, a batch of ten different initial conditions for orbit propagation are used and also initial values for roll angle were randomly obtained in the interval $[0^\circ, 360^\circ]$, while the yaw angle is chosen to be zero, as shown in Table 4.7. The starting conditions for pitch angle and for angular velocity were

Orbit n°	1	2	3	4	5	6	7	8	9	10
Day	1	1	1	1	1	1	1	1	1	1
Hours	1	8	19	0	3	4	4	14	6	4
Minutes	23	28	30	14	20	51	46	29	31	46
Seconds	22	8	28	12	0	59	10	28	57	18
Roll - ψ	301.7	7.0	245.3	136.6	299.5	181.0	255.4	154.4	109.7	68.3

Table 4.7: Initial orbit conditons for January, 1997.

set up as in Table 4.8, and when the Predictive control is being used the desired

Initial Condition	
$\gamma = 60^\circ$	
$\omega_{bo}^b = [0.001037 \ 0 \ 0.02] \text{ rad/s}$	

Table 4.8: Initial pitch and angular velocity.

angle γ and angular velocity are set up as in Table 4.9.

Desired Condition
$\gamma = 0^\circ$
$\omega_{bo}^b = [\begin{array}{ccc} 0 & 0 & 0.02 \end{array}] \text{ rad/s}$

Table 4.9: Desired pitch and angular velocity for the closed loop tests.

4.4 Performance Criteria

Since the quaternions have no clear physical meaning, as opposite to the Euler angles, the quaternions are then transformed to Euler angles (4.43), according to the sequence of rotations considered, for this work and for the simulation case, defined in Section 2.2.3.

There are several ways to analyse the state estimates coming from the algorithms. The state vector considered is the angular velocity and the quaternions, in Section 3.1.1. As explained before, the attitude estimates are presented in terms of the Euler angles, because they have a clear physical representation, instead of the quaternions. The Euler angles are obtained through the quaternions in several ways, and the one adopted in this work for the simulation case is described in Section *Quaternions to Euler Angles*. For the SimSat simulation, the performance of the EKF and SVD filters is evaluated according to the RMS of the error between the true angular velocity and the true roll, pitch and yaw Euler angles obtained from the equations of motion and the estimates computed by the estimator algorithms. For the EKF a standard deviation for both angular velocity estimates and roll, pitch and yaw Euler angles are obtained from the state error covariance matrix. However, the SVD algorithm gives only the error covariance for the attitude matrix so, in this case, only the roll, pitch and yaw Euler angle errors are bounded by the standard deviation.

Another parameter used to analyse the accuracy of the algorithms is the Euler angle error plot, which is based in the computation of the attitude matrix. This information is needed to better show the error incurred by the SVD, since it does not run continuously during a whole orbit.

For the EKF, results obtained from real PoSAT-1 satellite data, the true attitude and angular velocity is unknown so it is not possible to obtain neither the Euler angle nor the angular velocity and Euler roll, pitch and yaw errors but only a comparison with the EKF running on-board of the PoSAT-1 [16].

Pointing Accuracy

The angle γ , or pointing accuracy of the satellite several times plotted in the results, is the angle between the local vertical and the boom axis, Fig. 4.12.

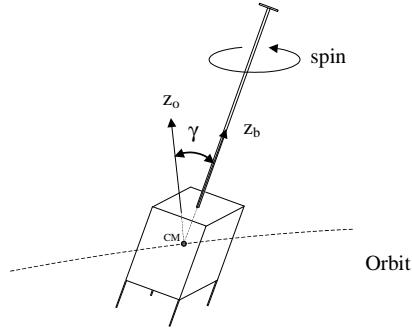


Figure 4.12: γ angle represented in the satellite.

Small Angle Approximation

For a small angle approximation, the Euler angles may be obtained directly from the quaternions,

$$\begin{aligned} q_1 &\simeq \frac{\psi}{2} \\ q_2 &\simeq \frac{\theta}{2} \\ q_3 &\simeq \frac{\phi}{2} \\ q_4 &\simeq 1 \end{aligned} \quad (4.34)$$

Angular Velocity Error

The error in the angular velocity is calculated as the root mean square error (RMS),

$$\begin{aligned} E_{\omega_x} &\simeq \lim_{m \rightarrow \infty} \sqrt{\frac{1}{m} \sum_{t=1}^m (\omega_x(t) - \hat{\omega}_x(t))^2} \\ E_{\omega_y} &\simeq \lim_{m \rightarrow \infty} \sqrt{\frac{1}{m} \sum_{t=1}^m (\omega_y(t) - \hat{\omega}_y(t))^2} \\ E_{\omega_z} &\simeq \lim_{m \rightarrow \infty} \sqrt{\frac{1}{m} \sum_{t=1}^m (\omega_z(t) - \hat{\omega}_z(t))^2} \end{aligned} \quad (4.35)$$

Euler Roll, Pitch and Yaw Error

The Euler angle errors are calculated considering the small angle approximation (4.34) and taking the RMS which is twice the difference between the real and the estimated quaternion,

$$\begin{aligned} E_\psi &\simeq \lim_{m \rightarrow \infty} \sqrt{\frac{1}{m} \sum_{t=1}^m (\psi(t) - \hat{\psi}(t))^2} \\ E_\theta &\simeq \lim_{m \rightarrow \infty} \sqrt{\frac{1}{m} \sum_{t=1}^m (\theta(t) - \hat{\theta}(t))^2} \\ E_\phi &\simeq \lim_{m \rightarrow \infty} \sqrt{\frac{1}{m} \sum_{t=1}^m (\phi(t) - \hat{\phi}(t))^2} \end{aligned} \quad (4.36)$$

Euler Angle Error

Given the true attitude matrix, A_{true} , and the estimated attitude matrix obtained from the estimation algorithms, \hat{A} , an error matrix, \tilde{A} , denotes the error between estimated matrix and the true one,

$$\begin{aligned} A_{true} &= \tilde{A}\hat{A} \\ \tilde{A} &= A_{true}\hat{A}^T \end{aligned}$$

From the error matrix, the Euler angle error can be computed according to (2.11),

$$trace(\tilde{A}) = 1 + 2\cos(\chi) \quad (4.37)$$

$$\chi = a \cos \left(\frac{trace(\tilde{A}) - 1}{2} \right) \quad (4.38)$$

Recalling that the attitude matrix represents a rotation between two CS, this statistical information tells how far the estimated attitude matrix is from the right solution, i.e. the correct transformation.

Standard Deviation

The angular velocity and Euler angle errors defined before can be bounded by a standard deviation, σ_j which provides realistic error estimates assuming that the measurements are uncorrelated. The standard deviation is calculated from the corresponding diagonal elements of the state error covariance matrix, P ,

$$\sigma_j = \sqrt{P_{jj}} \quad (4.39)$$

Since the diagonal of the error covariance matrix is defined as $P_{jj} = E[e_j e_j^T]$ where $e = x - \hat{x}$ is the error of the corresponding parameter, for the angular velocity,

$$\begin{aligned} P_{\omega_{xx}, yy, zz} &= E[(\omega_{xx}, yy, zz - \hat{\omega}_{xx}, yy, zz)(\omega_{xx}, yy, zz - \hat{\omega}_{xx}, yy, zz)^T] \\ &= E[e_{\omega_{xx}, yy, zz} e_{\omega_{xx}, yy, zz}^T] \\ \sigma_{\omega_{xx}, yy, zz} &= \sqrt{P_{\omega_{xx}, yy, zz}} \end{aligned} \quad (4.40)$$

and for the Euler angles,

$$P_{q_1, q_2, q_3} = E[(q_1, q_2, q_3 - \hat{q}_1, \hat{q}_2, \hat{q}_3)(q_1, q_2, q_3 - \hat{q}_1, \hat{q}_2, \hat{q}_3)^T] \quad (4.41)$$

Using the small angle approximation from (4.34),

$$\begin{aligned} P_{q_1, q_2, q_3} &= E\left[\left(\frac{\psi, \theta, \phi}{2} - \frac{\hat{\psi}, \hat{\theta}, \hat{\phi}}{2}\right)\left(\frac{\psi, \theta, \phi}{2} - \frac{\hat{\psi}, \hat{\theta}, \hat{\phi}}{2}\right)^T\right] \\ &= \frac{1}{4}E[e_{\psi, \theta, \phi}e_{\psi, \theta, \phi}^T] = \frac{1}{4}P_{\psi, \theta, \phi} \\ \sigma_{\psi, \theta, \phi} &= \sqrt{P_{\psi, \theta, \phi}} = \sqrt{4P_{q_1, q_2, q_3}} = 2\sqrt{P_{q_1, q_2, q_3}} \end{aligned} \quad (4.42)$$

The Euler angles are computed in terms of the elements of the attitude matrix that maps any vector from the OCS to the BCS, Eq. (2.35), or in terms of the Euler angles depending on the sequence of rotations used. For the 123 sequence, from Eq. (2.15) the following relation can be established between quaternions $\mathbf{q} = [q_1 \ q_2 \ q_3 \ q_4]^T$ and Euler angles, ψ, θ, ϕ :

$$\begin{aligned} \psi &= \text{atan2}\left(-\frac{A_{32}}{A_{33}}\right), \quad 0^\circ \leq \psi \leq 360^\circ \\ \theta &= \text{atan2}(A_{31}), \quad 90^\circ \leq \theta \leq 90^\circ \\ \phi &= \text{atan2}\left(-\frac{A_{21}}{A_{11}}\right), \quad 0^\circ \leq \phi \leq 360^\circ \end{aligned} \quad (4.43)$$

where $\text{atan2}(Y, X)$ is a Matlab functionis the four quadrant arctangent of the real parts of the elements of X and Y. $-\pi \leq \text{atan2}(Y, X) \leq \pi$.

For the PoSAT-1 real data the sequence of Euler rotations 213 was considered. Thus, a different relation can be established between the attitude matrix (2.35) and the Euler angles from (4.24),

$$\theta = \arctan 2\left(\frac{A_{31}}{A_{33}}\right), \quad 0^\circ \leq \theta \leq 360^\circ \quad (4.44)$$

$$\psi = \arcsin(-A_{32}), \quad -90^\circ \leq \psi \leq 90^\circ \quad (4.45)$$

$$\phi = \arctan 2\left(\frac{A_{12}}{A_{22}}\right), \quad 0^\circ \leq \phi \leq 360^\circ \quad (4.46)$$

Considering other elements in the attitude matrix, different expressions can be obtained relating quaternions and Euler angles.

Chapter 5

Results

In this chapter, the results obtained for the conditions set up in the previous chapter are presented. The results are described according to the Tests description done in Section 4.3, summarized in Table 4.5. For each test, ten experiences were performed according to Table 4.7 and numbered. The results of the most significant experiences are presented.

Each test is analysed according to the performance criteria defined in Section 4.4. Besides accuracy and computational load, one of the goals of the EKF tests is to study the effect of the covariance matrix parameters tuning on the estimator results.

5.1 PoSAT-1 Simulation

5.1.1 Test A

The RMS was obtained from a set of three orbits for each of the ten different orbital conditions described in Table 4.7. The three orbits were established because it was verified during the several experiences that a minimum number of three orbits was required for the control loop to achieve its goal. So, the RMS obtained for this test, according to Eq. (4.35), is for ω_x and ω_y , 9.71×10^{-5} rad/s and 9.63×10^{-5} rad/s, respectively. These values are very similar, as expected, due to the satellite geometry. For the ω_z , the RMS obtained is 1.58×10^{-4} rad/s. For the Euler angles the RMS obtained according to Eq. (4.36), is 2.89° , 1.99° , 3.02° for *roll* (ψ), *pitch* (θ) and *yaw* (ϕ) respectively. Since the EKF computes the process error covariance matrix for the angular velocity and for the quaternions then the RMS can be bounded by the standard deviation computed by Eqs. (4.40) and (4.42).

In Fig. 5.1 the angular velocity and in Fig. 5.2 the *roll* (ψ), *pitch* (θ) and *yaw* (ϕ) angles are plotted for one of the ten experiments. Also in Fig. 5.3 and 5.4, the evolution of the angular velocity and *roll* (ψ) *pitch* (θ) and *yaw* (ϕ) Euler angles

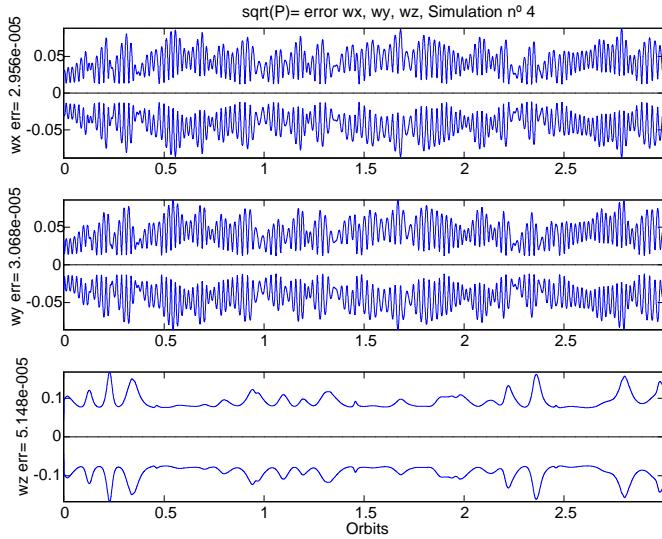


Figure 5.1: Test A. RMS (black line) of the angular velocity error [rad/s] bounded by its standard deviation (blue line).

can be seen for three orbits. From the graphics, the estimates (blue line) almost coincide with the true value (red line) all the time. Moreover, the evolution of the Euler angle error as defined in Eq. (4.38) is shown in Fig. 5.5. Notice that the estimator and system initial conditions are exactly the same in Test A, hence the initial error is zero and the maximum error is about the same for the Euler angles RMS previously referred. The evolution of the angle γ in Fig. 5.6, shows the absence of control leading the satellite to a libration movement around the local vertical.

5.1.2 Test D

In closed loop, the EKF results are, as expected, worse than those in open loop, since the closed loop is a more demanding situation. These results can be compared through the RMS obtained for the angular velocity: ω_x and ω_y is 3.77×10^{-4} rad/s and for $\omega_z = 4.456 \times 10^{-4}$ rad/s. The RMS for the Euler angles is: $\psi = 7.65^\circ$, $\theta = 3.89^\circ$ and $\phi = 6.62^\circ$. This test is exemplified by the experience plotted in Figs. 5.7 - 5.10. To improve the results of the estimator, all subsystems, estimator and control algorithm must be tuned in closed loop.

From Fig. 5.10 it is evident that the control algorithm is working since the libration of the satellite is being reduced over time.

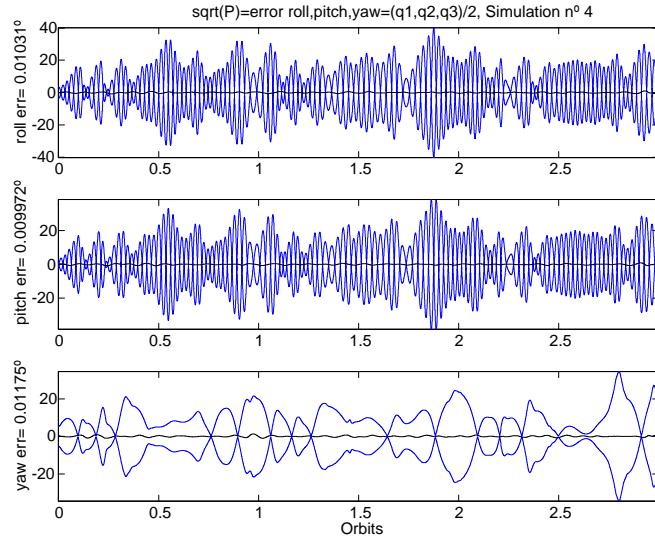


Figure 5.2: Test A. RMS (black line) of the Euler angles [°] error is bounded by its standard deviation (blue line).

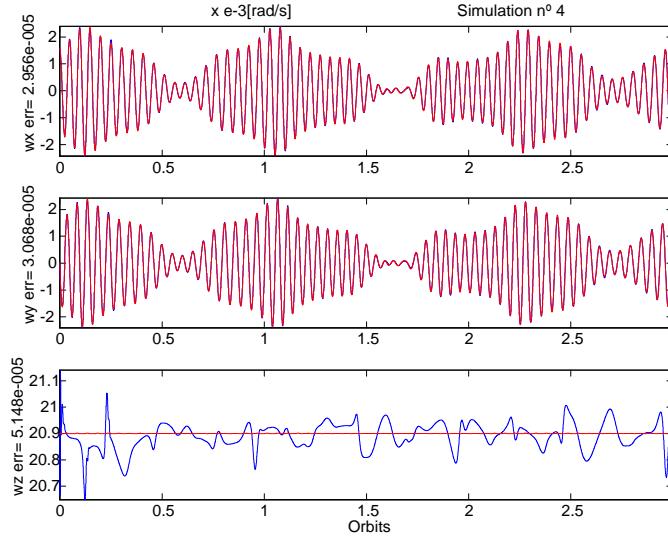


Figure 5.3: Test A. Estimation results (blue line) and the true values of the angular velocities (red line). The y -axis label is the RMS for the angular velocity [$\omega_{x,y,z}$ err=rad/s].

5.1. POSAT-1 SIMULATION

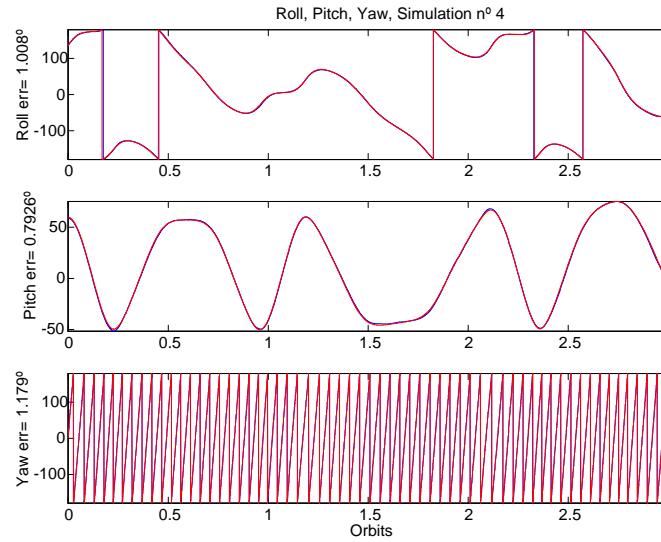


Figure 5.4: Test A. Estimation results (blue line) and the true values of the *roll* (ψ), *pitch* (θ) and *yaw* (ϕ) Euler angles (red line). In the y-axis label is the RMS for the Euler angles [Roll/Pitch/Yaw err= $^{\circ}$].

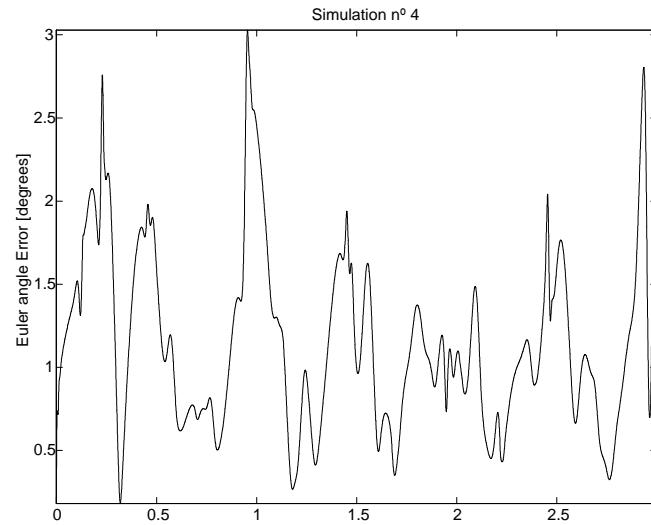


Figure 5.5: Test A. Euler angle error.

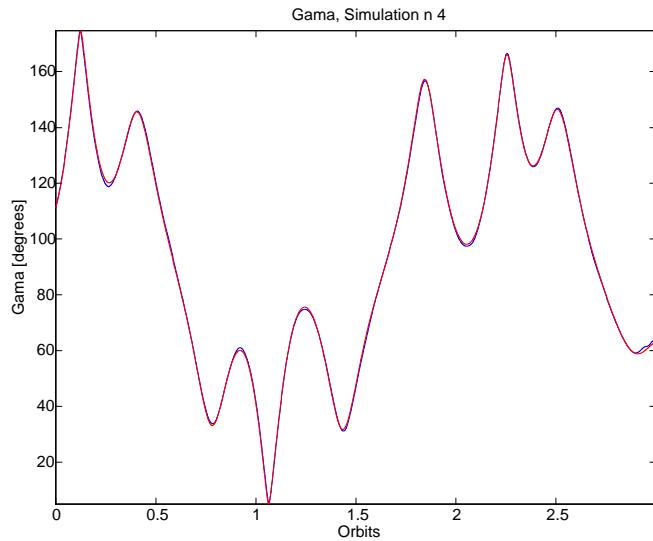


Figure 5.6: Test A. The γ evolution.

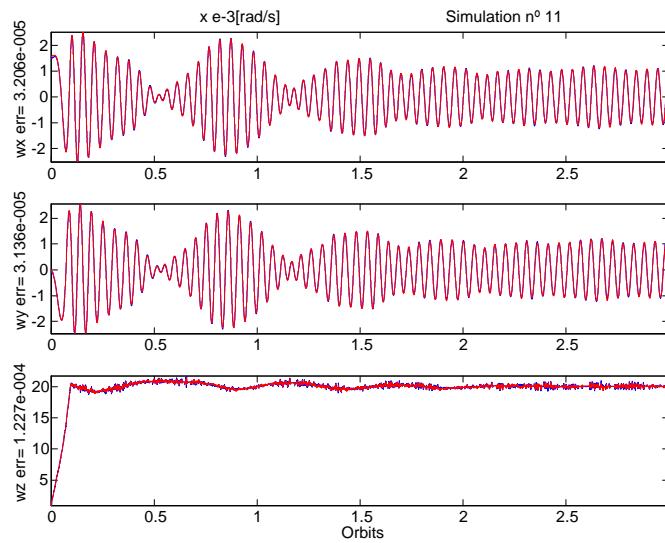


Figure 5.7: Test D. Estimation results (blue line) and the true values of the angular velocities (red line). The y -axis label is the RMS for the angular velocity [$\omega_{x,y,z}$ err=rad/s].

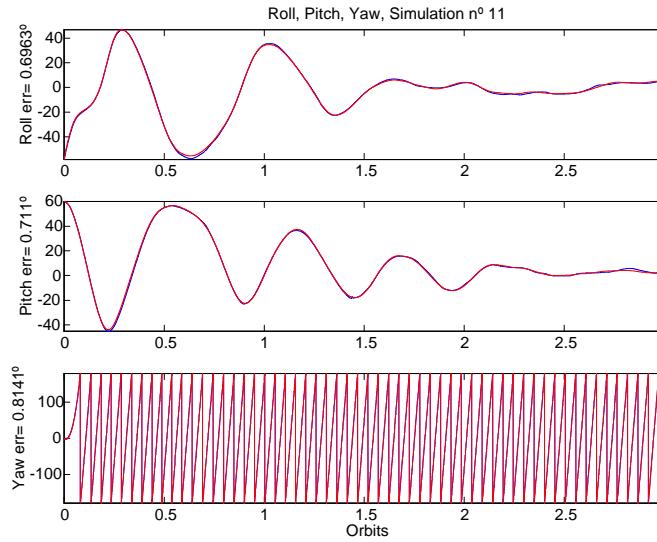


Figure 5.8: Test D. Estimation results (blue line) and the true values of the *roll* (ψ), *pitch* (θ) and *yaw* (ϕ) Euler angles (red line). In the y-axis label is the RMS for the Euler angles [Roll/Pitch/Yaw err= $^{\circ}$].

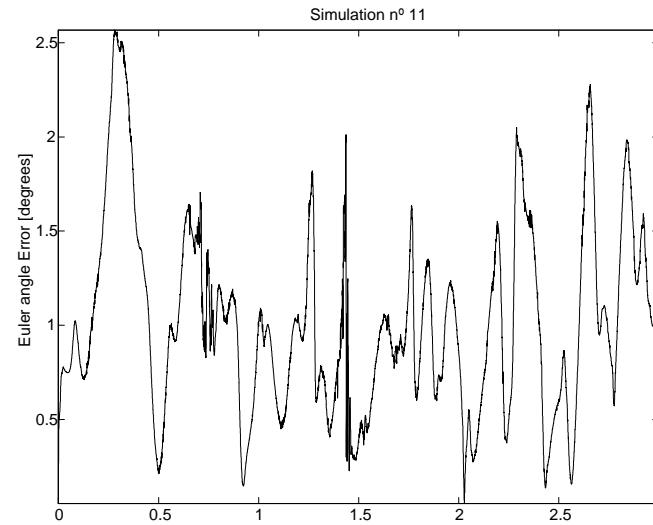
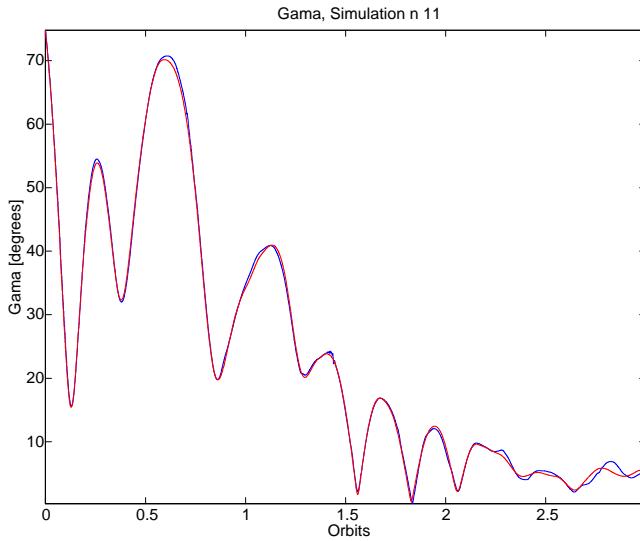


Figure 5.9: Test D. Euler angle error.

Figure 5.10: Test D. The γ evolution.

5.1.3 Test G

An error of 50% was added to the initial angular velocity and to the initial ψ , θ and ϕ . The initial conditions of error covariance matrix, P_o , are changed, *i.e.*, it is maintained exactly equal to the one used in Test A. Despite the 50% error in the initial conditions, the EKF was able to converge but the results of the estimator algorithm are poor compared with the results from Test A. The RMS for the angular velocity is: $\omega_x \equiv \omega_y = 1.05 \times 10^{-3}$ rad/s and $\omega_z = 2.51 \times 10^{-3}$ rad/s and for the Euler angles is: $\psi = 25.27^\circ$, $\theta = 12.85^\circ$, $\phi = 29.56^\circ$. This test is exemplified by the experiences plotted in Figs 5.11 - 5.13, that stress the initial error and the estimated state variables quick convergence to the true values.

However, this is a sensitive issue because both the convergence and the estimator result quality are dependable of the initial P_o . If the values of its diagonal are decreased it can cause the algorithm to diverge or to take longer to converge. On the other hand, the results are also influenced by the initial error for some time, as can be checked by comparing the accuracy obtained between Test A and Test G. This is due to the fading memory characteristic of the EKF.

5.1.4 Test J

As in Test G, an error of 50% had been added to the initial angular velocity and to the initial ψ , θ and ϕ . If the RMS error increased comparing Test G to Test A, then for closed loop the RMS error increased even more. As explained before the

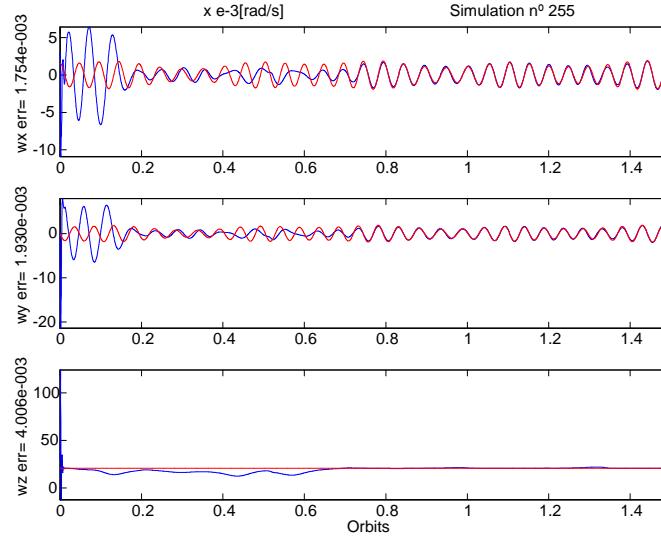


Figure 5.11: Test G. Estimation results (blue line) and the true values of the angular velocities (red line). The y -axis label is the RMS for the angular velocity [$\omega_{x,y,z}$ err=rad/s].

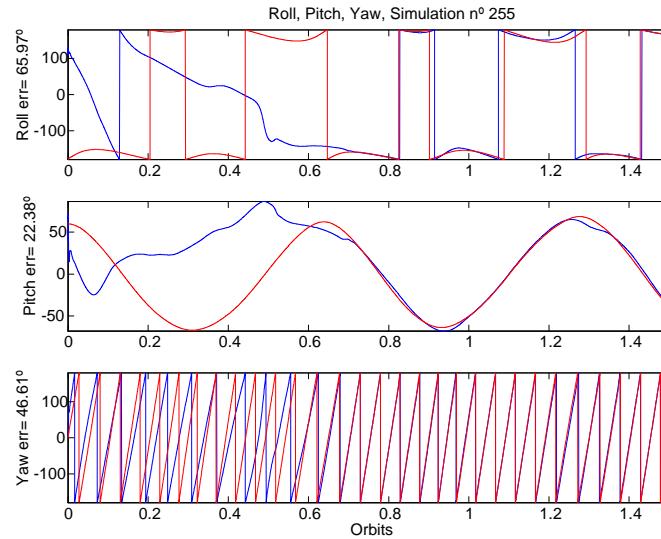


Figure 5.12: Test G. Estimation results (blue line) and the true values of the *roll* (ψ), *pitch* (θ) and *yaw* (ϕ) Euler angles (red line). In the y -axis label is the RMS for the Euler angles [Roll/Pitch/Yaw err=°].

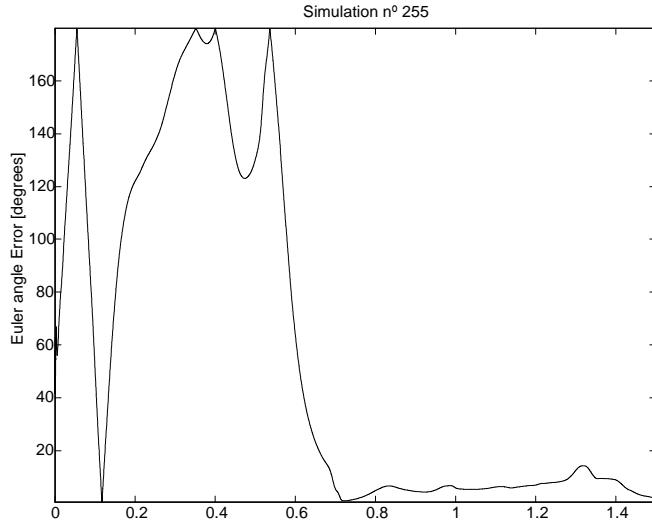


Figure 5.13: Test G. Euler angle error.

RMS error increased when error is added to the initial state, comparing Test G to Test A, and for closed loop too, comparing Test A to Test D. Thus, for this Test is expected a worse performance than for Test G or even for Test D, due to fact of closing the loop and adding error to the initial state variables, as stressed before. The RMS for the angular velocity is: $\omega_x \equiv \omega_y = 1.1 \times 10^{-3}$ rad/s and $\omega_z = 2.7 \times 10^{-3}$ rad/s and for the Euler angles is: $\psi = 28.46^\circ$, $\theta = 13.71^\circ$, $\phi = 24.46^\circ$. In Figs. 5.14 - 5.16 are plotted, for one of the ten experiments, an experiment that exemplifies this Test.

5.1.5 Test B

The EKF results adding the Sun sensor measurements are about the same as for the Test A. In fact, it is known that the more measurements available, the better are the estimates obtained by the Kalman filter. It must be stressed that for this Test the initial state vector is known for the filter. This can be visualized comparing Figs. 5.3 - 5.4 with Figs. 5.17 - 5.18.

5.1.6 Test E

The EKF results improve with the presence of the Sun sensors data in comparison with the Test D. This results can be compared by the RMS obtained for the angular velocity: ω_x and ω_y is 3.4×10^{-4} rad/s and for $\omega_z = 3.0 \times 10^{-4}$ rad/s. The Euler angles: $\psi = 5.63^\circ$, $\theta = 6.45^\circ$ and $\phi = 3.20^\circ$. In Figs. 5.19 - 5.21 are plotted,

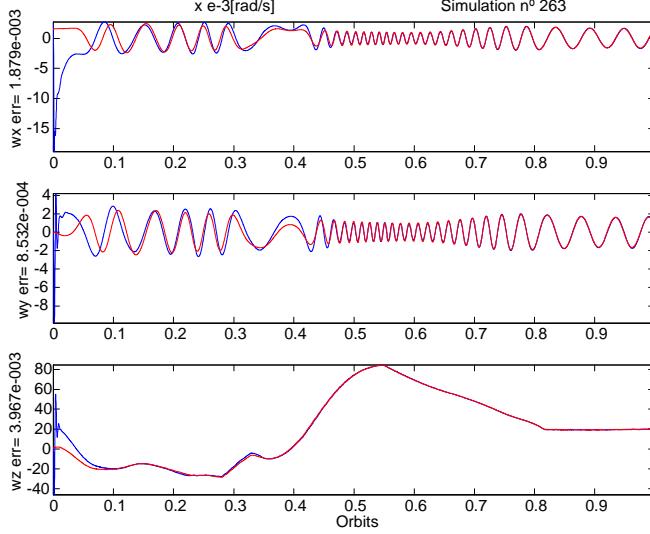


Figure 5.14: Test J. Estimation results (blue line) and the true values of the angular velocities (red line). The y -axis label is the RMS for the angular velocity [$\omega_{x,y,z}$ err=rad/s].

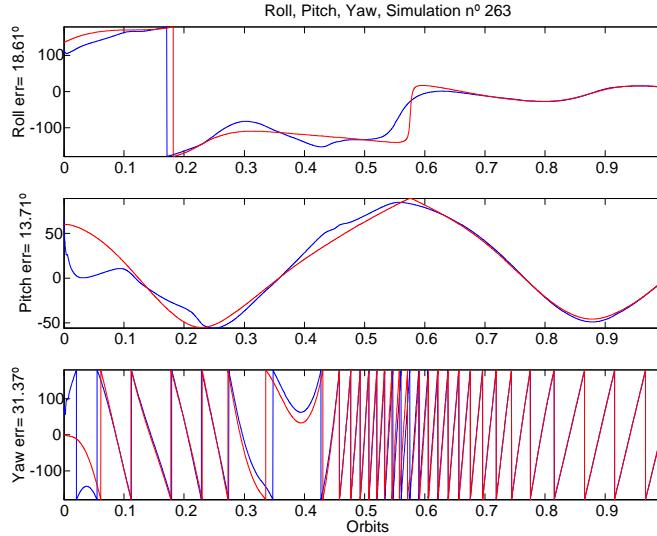


Figure 5.15: Test J. Estimation results (blue line) and the true values of the *roll* (ψ), *pitch* (θ) and *yaw* (ϕ) Euler angles (red line). In the y -axis label is the RMS for the Euler angles [Roll/Pitch/Yaw err=°].

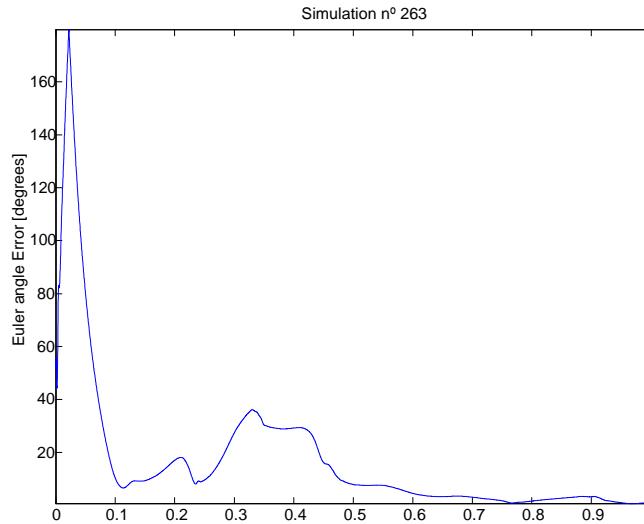


Figure 5.16: Test J. Euler angle error.

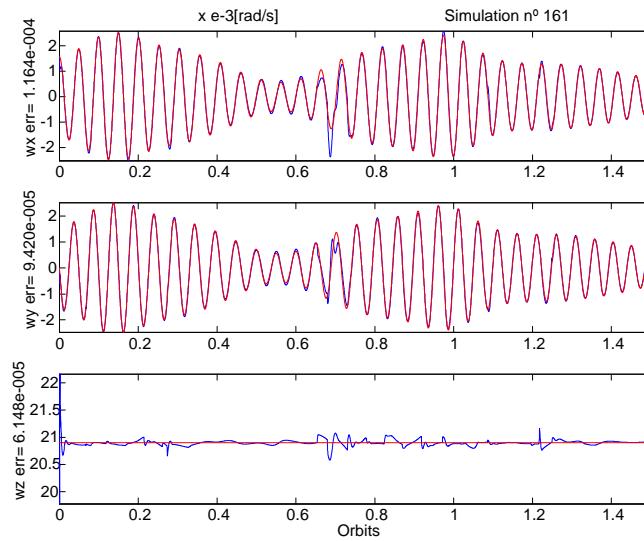


Figure 5.17: Test B. Estimation results (blue line) and the true values of the angular velocities (red line). The y -axis label is the RMS for the angular velocity [$\omega_{x,y,z}$ err=rad/s].

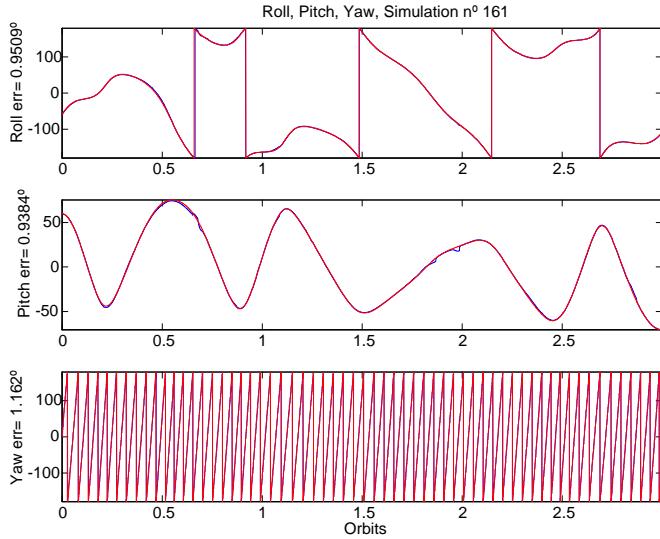


Figure 5.18: Test B. Estimation results (blue line) and the true values of the *roll* (ψ), *pitch* (θ) and *yaw* (ϕ) Euler angles (red line). In the y-axis label is the RMS for the Euler angles [Roll/Pitch/Yaw err= $^{\circ}$].

for one of the ten experiments, an experiment that exemplifies this Test.

5.1.7 Test H

When the Sun sensor measurements are available for the EKF besides the magnetometers, the results improved considerably comparing with those obtained from Test G. This is evident in the RMS obtained: $\omega_x = 9.13 \times 10^{-4}$ rad/s, $\omega_y = 9.56 \times 10^{-4}$ rad/s and $\omega_z = 1.76 \times 10^{-3}$ rad/s and for the Euler angles is: $\psi = 6.46^{\circ}$, $\theta = 3.77^{\circ}$, $\phi = 6.24^{\circ}$. The experiment that exemplify is in Figs. 5.22 - 5.24 comparing with Test G in Figs 5.11 - 5.13. As stressed before, the statistical characteristics of the Kalman filter improve with more measurements available. This is not so evident when Test A is compared with Test B but the improvement in Test G is compared to Test H is more evident due to the error in the initial conditions. The filter recovers from the error added to the initial state vector in Test H considerably, than in Test G, as can be compared by Fig. 5.24 and Fig. 5.13. It is to notice that Test G and H have exactly the same orbit parameters, defined in Table 4.7.

An error in the initial state variables is more likely to happen in real situations and this initial error is reflected for a long time in the estimator algorithm results because of the EKF fading memory, hence two or more measurements help to improve the filter performance.

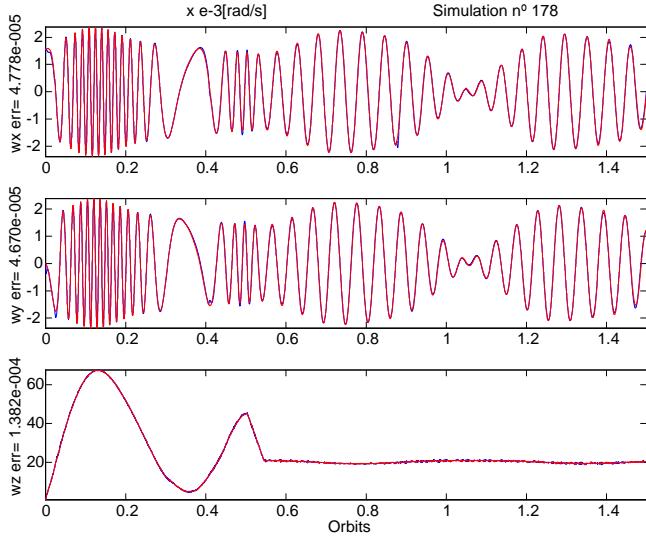


Figure 5.19: Test E. Estimation results (blue line) and the true values of the angular velocities (red line). The y -axis label is the RMS for the angular velocity [$\omega_{x,y,z}$ err=rad/s].

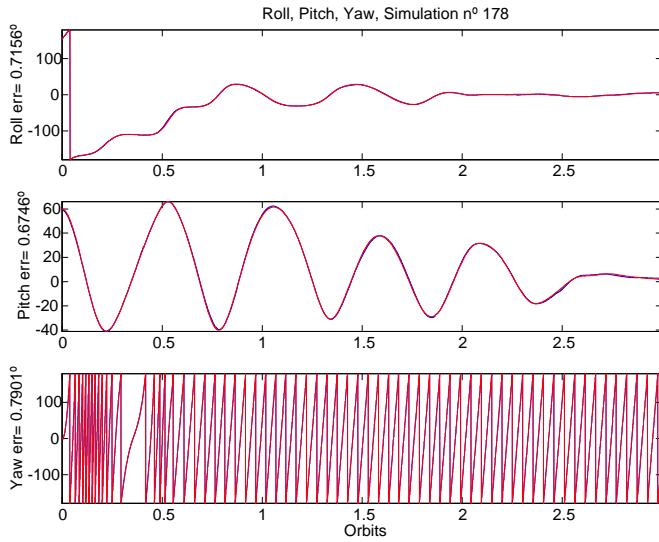


Figure 5.20: Test E. Estimation results (blue line) and the true values of the *roll* (ψ), *pitch* (θ) and *yaw* (ϕ) Euler angles (red line). In the y -axis label is the RMS for the Euler angles [Roll/Pitch/Yaw err=°].

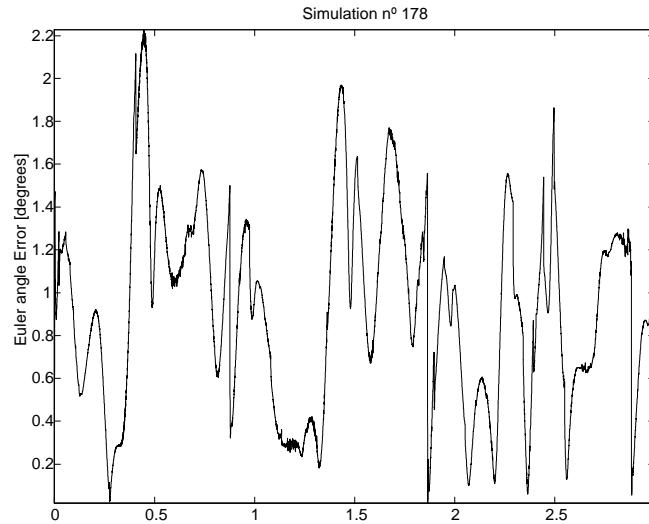
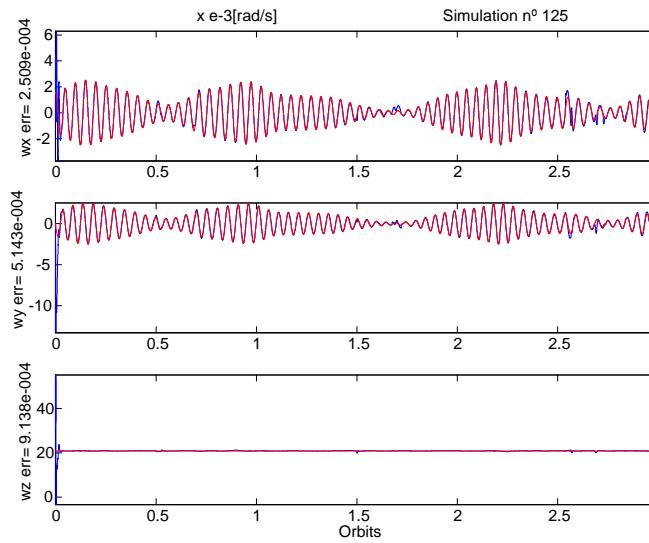


Figure 5.21: Test E. Euler angle error.


 Figure 5.22: Test H. Estimation results (blue line) and the true values of the angular velocities (red line). The y -axis label is the RMS for the angular velocity [$\omega_{x,y,z}$ err=rad/s].

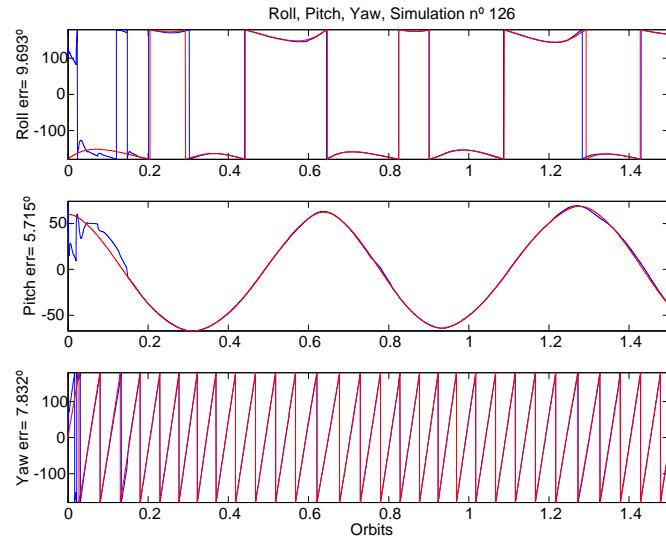


Figure 5.23: Test H. Estimation results (blue line) and the true values of the *roll* (ψ), *pitch* (θ) and *yaw* (ϕ) Euler angles (red line). In the y-axis label is the RMS for the Euler angles [Roll/Pitch/Yaw err= o].

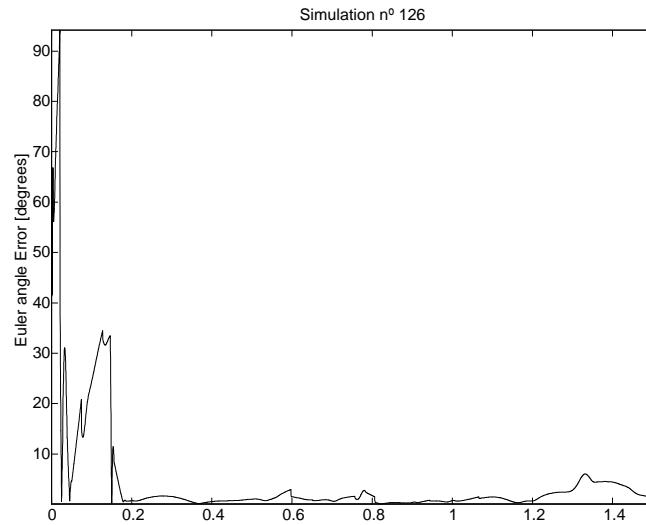


Figure 5.24: Test H. Euler angle error.

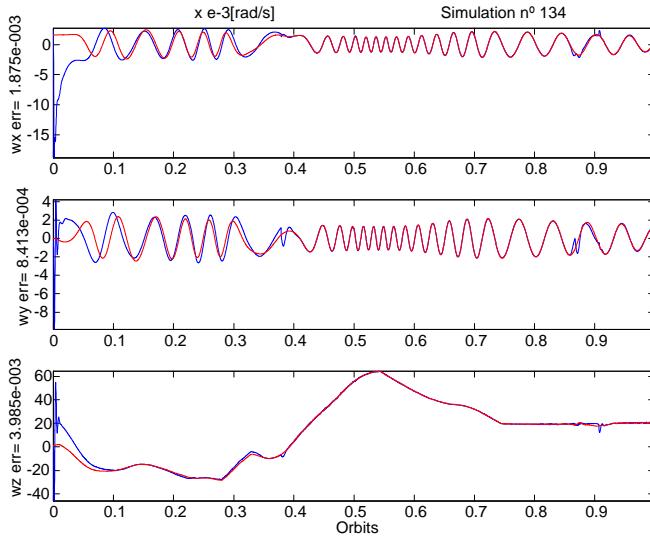


Figure 5.25: Test K. Estimation results (blue line) and the true values of the angular velocities (red line). The y -axis label is the RMS for the angular velocity [$\omega_{x,y,z}$ err=rad/s].

5.1.8 Test K

In the same way that Test H have better results than Test G, the Test K present better results than Test J, due the presence of the Sun sensor measurements. However the errors increased then those obtained from Test H, because the estimator is in the loop and the control algorithm depends on the estimator results, which is a more demanding situation. Once again this results are supported by the RMS of the variables states: $\omega_x = 9.51 \times 10^{-4}$ rad/s, $\omega_y = 1.05 \times 10^{-3}$ rad/s and $\omega_z = 1.37 \times 10^{-3}$ rad/s and for the Euler angles is: $\psi = 6.97^\circ$, $\theta = 3.89^\circ$, $\phi = 6.58^\circ$. In Figs. 5.25 - 5.27 are plotted, for one of the ten experiments, an experiment that exemplifies this Test. It is to notice that Test K and J have exactly the same orbit parameters, defined in Table 4.7.

5.1.9 Test C

Just for visualization of the SVD method, a signal indicating when Sun sensor measurements are available is high and low otherwise, originating a square wave in all plots. Figures 5.29 - 5.31 show that, when the satellite is hidden from the Sun, "behind" the Earth, the attitude estimation and angular velocity are obtained exclusively from propagation of the satellite attitude dynamics. In these orbit segments, the error of the quaternion estimate raises dramatically. This is so because

5.1. POSAT-1 SIMULATION

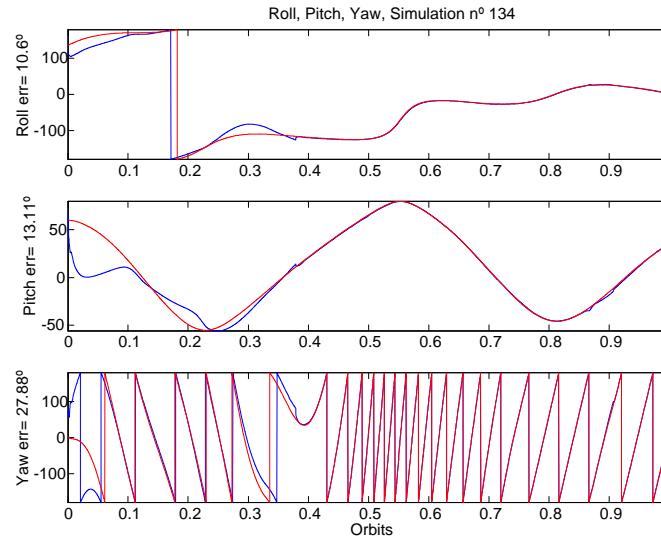


Figure 5.26: Test K. Estimation results (blue line) and the true values of the *roll* (ψ), *pitch* (θ) and *yaw* (ϕ) Euler angles (red line). In the y-axis label is the RMS for the Euler angles [Roll/Pitch/Yaw err= o].

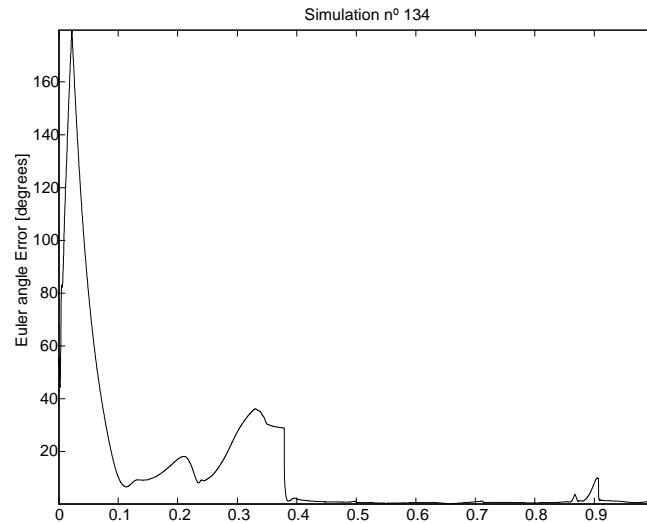


Figure 5.27: Test K. Euler angle error.

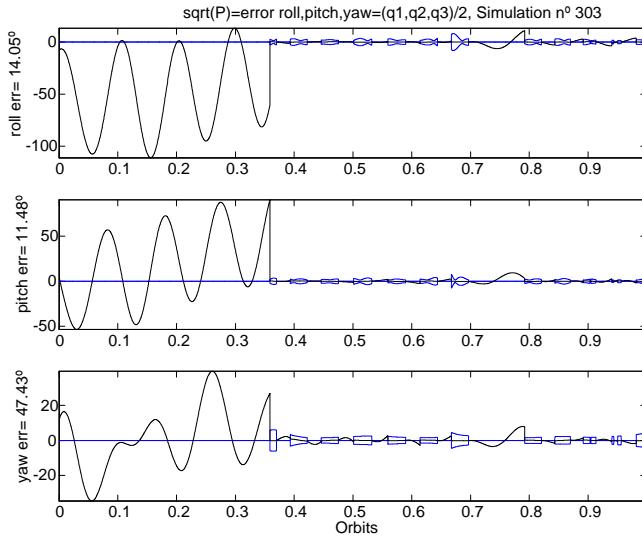


Figure 5.28: Test C. RMS (black line) of the Euler angles [°] error is bounded by its standard deviation (blue line) which only is computed on SVD is running.

the model of the Sun sensor is based on the estimates of the previous attitude which already has error due to propagation. So the attitude obtained by the algorithm is also influenced by the error. This is also due to the fact of the error of the angular velocity, since as can be seen by the kinematics equation, the quaternion is influenced directly by the angular velocity.

However, when the Sun is in the FOV of the sensor the quaternion is becoming close to the true value and the error incurred by the SVD it is very low (Fig. 5.32 and a detail of the Fig. 5.31) when compared to the EKF results in Fig. 5.5 - Test A. If the RMS is computed for the whole orbit, with the SVD running only when the two sensors are available, the RMS errors will be: $\omega_{x,y} = 1.09 \times 10^{-3}$ rad/s and $\omega_z = 9.46 \times 10^{-4}$ rad/s and, for the Euler angles, $\psi = 40.1^\circ$, $\theta = 15.6^\circ$, $\phi = 38.3^\circ$. Still, the angular velocity has a very good accuracy compared with the EKF. Hence, if the RMS is only computed for when the SVD is running the results for the Euler angles improve considerable, approx. $\psi = 0.52^\circ$, $\theta = 0.55^\circ$, $\phi = 0.52^\circ$.

There are some points in orbit, when the Sun sensor measurements is available, where the angular velocity error is very high. This is so because the angular velocity is calculated based in the derivatives of the quaternions and if the quaternions have a smal error it will be amplified when the angular velocity is being computed. However, the angular velocity estimate along the z-axis is very good since it is a constant.

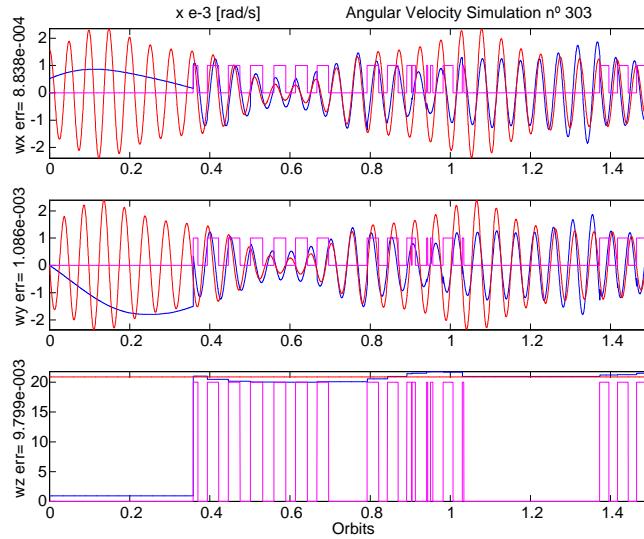


Figure 5.29: Test C. Estimation results (blue line) and the true values of the angular velocities (red line). The y -axis label is the RMS for the angular velocity [$\omega_{x,y,z}$ err=rad/s].

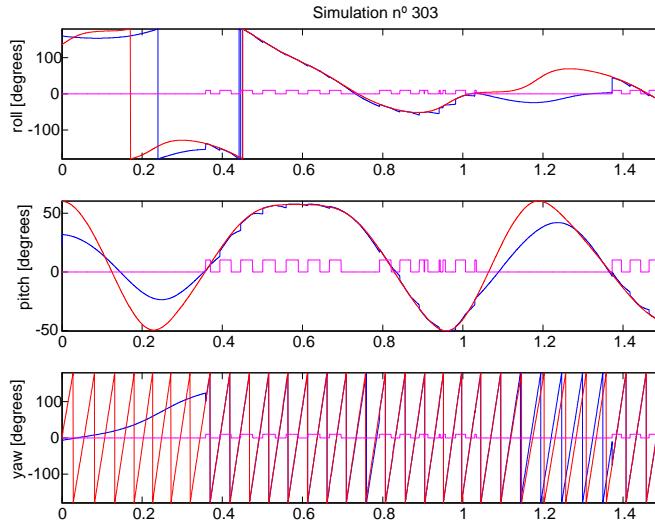


Figure 5.30: Test C. Estimation results (blue line) and the true values of the $roll$ (ψ), $pitch$ (θ) and yaw (ϕ) Euler angles (red line).

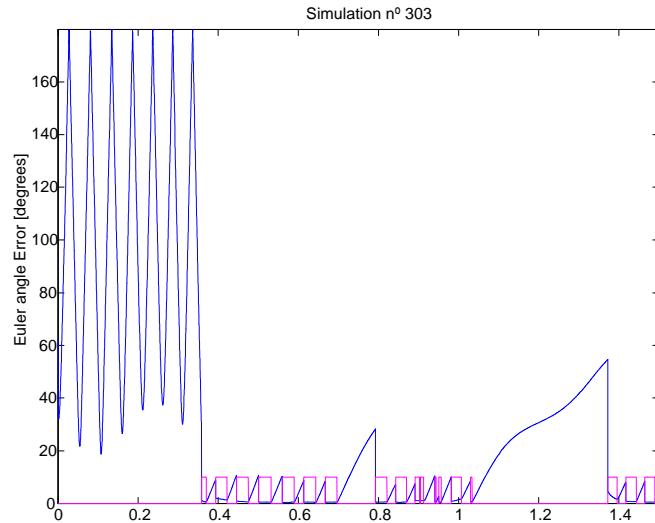


Figure 5.31: Test C. Euler angle error.

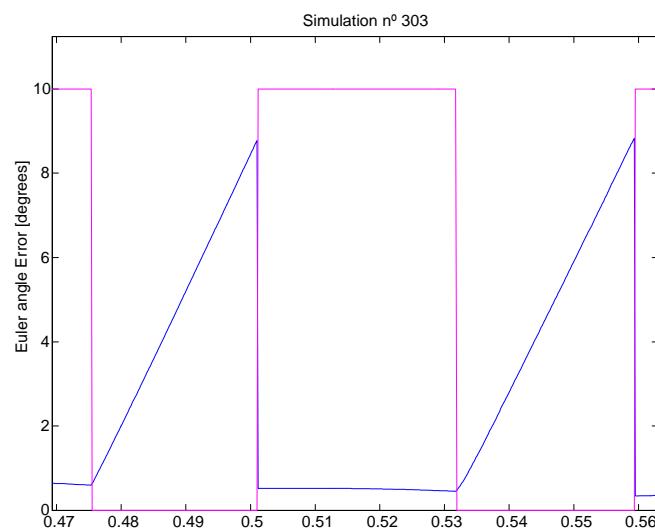


Figure 5.32: Test C. Euler angle error. Detail of Fig. 5.31.

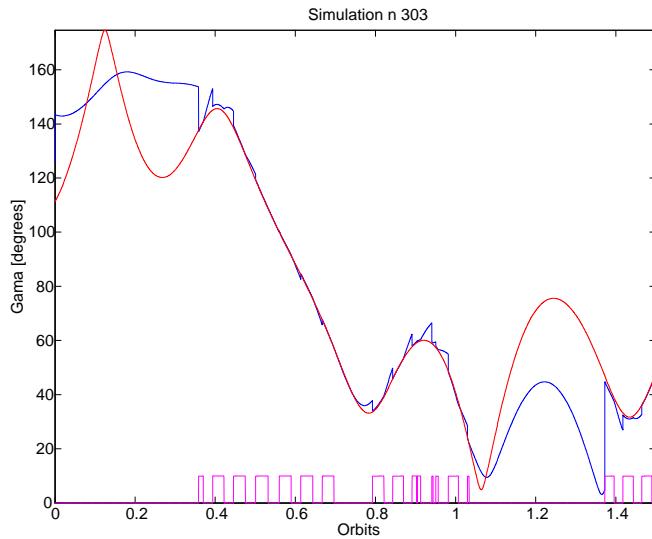


Figure 5.33: Test C. The γ evolution. Estimated (blue line) and true (red line) values of γ .

5.1.10 Test F

For Test F the results are similar to those from Test C since the SVD is not recursive but relies on the attitude sensor measurements on each time point to compute the estimates to that point disregarding all past information. This Test is exemplified by the experiences plotted in Figs. 5.34 - 5.35. It is evident the angular velocity correction when the SVD algorithm is available, about the 2.5 orbits in Fig. 5.34.

5.2 PoSAT-1 Real Data

The results obtained using PoSAT-1 real data presented in Section (4.2.3) are represented in the Figs 5.37 and 5.36 for angular velocity and Euler angles respectively. The estimator algorithm used was the EKF defined in Chapter 3 with the modifications explained in Chapter 4. The results are compared with the results from the EKF implemented on-board PoSAT-1.

The RMS obtained between the estimates from the EKF and the estimates from the estimator on-board are for the Euler ψ and θ angles are $5^\circ - 6^\circ$, which is about the results obtained for EKF applied to the simulation of PoSAT-1 in SimSat. However, for the yaw angle the difference between the two algorithms is bigger due to the spinning of the satellite, $\phi = 40^\circ$. The EKF presented in this work does not

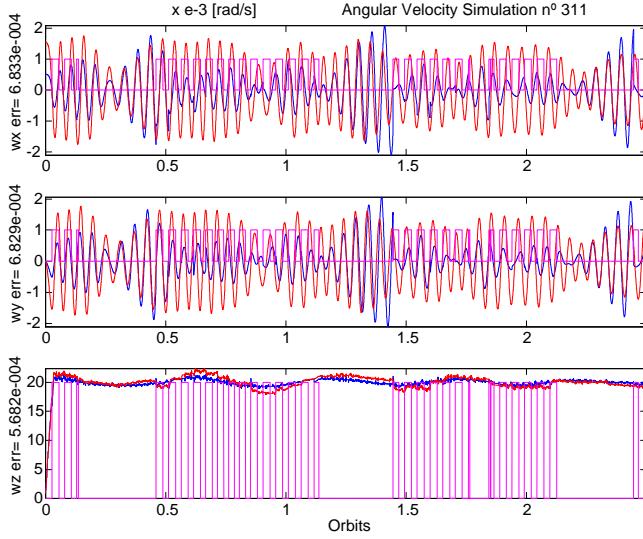


Figure 5.34: Test F. Estimation results (blue line) and the true values of the angular velocities (red line). The y -axis label is the RMS for the angular velocity [$\omega_{x,y,z}$ err=rad/s].

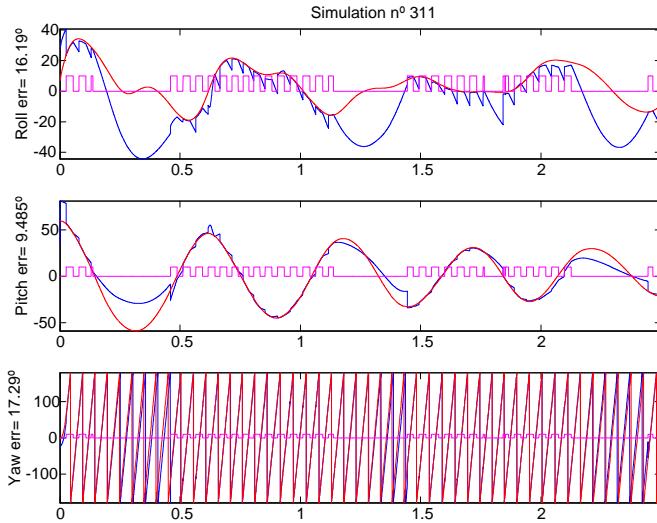


Figure 5.35: Test F. Estimation results (blue line) and the true values of the roll (ψ), pitch (θ) and yaw (ϕ) Euler angles (red line).

assume small Euler angles approximation as opposed to the EKF implemented on-board PoSAT-1. The great advantage of this simplified algorithm, according to [16], is to significantly reduce the matrix computation demand to the computer on-board.

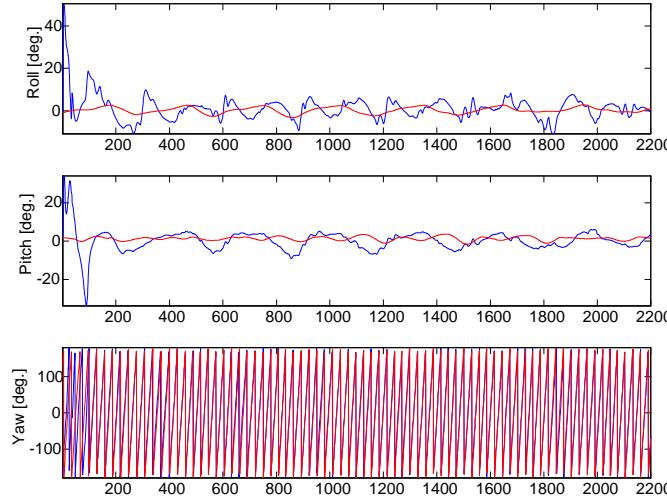


Figure 5.36: Test ZP. EKF estimates, for approx. 7.2 orbits, of the Euler angles (blue line) in comparison with the estimates of the EKF on-board of the PoSAT-1 satellite (red line). The magnetometers are the only data supplied to the filter.

5.2. POSAT-1 REAL DATA

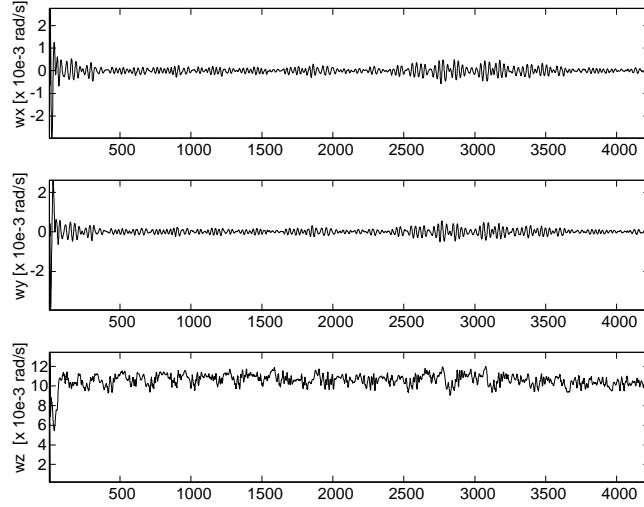


Figure 5.37: Test ZP. EKF estimates of the angular velocity [rad/s] from PoSAT-1 real data for 14 orbits. Magnetometers measurements are the only data supplied to the filter.

$\psi/\theta/\phi$	Estimator only			Estimator in the Loop		
	Without/SS		With/SS	Without/SS		With/SS
	EKF	EKF	SVD	EKF	EKF	SVD
Init 1	2.9°/1.99°/3.0°	$\cong \mathbf{A}$	0.52°/0.55°/0.52°	7.7°/3.9°/6.6°	5.6°/6.5°/3.2°	1.2°/1.4°/0.64°
Init 2	25.3°/12.8°/29.6°	6.5°/3.8°/6.3°		28.5°/13.7°/24.5°	6.9°/3.9°/6.6°	
PoSAT-1	6°/5°/40°					

Table 5.1: RMS summary of the maximum Euler angles for all Tests.

[rad/s]		Estimator only			Estimator in the Loop		
		Without/SS		With/SS	Without/SS		With/SS
		EKF	EKF	SVD	EKF	EKF	SVD
Init 1	ω_x	9.7E-5	$\cong A$	1.1E-3	3.8E-4	3.4E-4	6.0E-4
	ω_y	9.6E-5	$\cong A$	1.0E-3	3.8E-4	3.4E-4	8.1E-4
	ω_z	1.6E-5	$\cong A$	9.5E-4	4.5E-4	3.0E-4	7.7E-4
Init 2	ω_x	1.1E-3	9.2E-4		1.1E-3	9.5E-4	
	ω_y	1.1E-3	9.6E-4		1.1E-3	1.1E-3	
	ω_z	2.51E-3	1.8E-3		2.7E-3	1.4E-3	
PoSAT-1		-					

Table 5.2: RMS summary of angular velocity error for all Tests.

Chapter 6

Conclusions and Future Work

In this work a point-by-point (SVD) and a recursive estimation (EKF) methods for attitude determination were tested both on simulated and real data from small satellites. In order to analyse the accuracy of attitude and angular velocity and be able to compare between both algorithms, a set of perform criteria was set up in chapter 4. Even so, the comparison had to be performed with special care because the SVD is not always running due to the temporary unavailability of the Sun sensor.

The EKF produces, as expected, the most accurate results, at the cost of increased use of computational resources, due to the computation of the linearized dynamics at each orbital location and due to the manipulation of matrices with high rank. The SVD requires the permanent availability of two sensors, but only magnetometers and Sun sensors are available on-board PoSAT-1, thus leading to the requirement of propagating the attitude dynamics and kinematics while the Sun sensor is unavailable. This produces worse results than if SVD had two sensors available along the whole orbit as shown in Chapter 5. Even so a more carefull analysis of the SVD results shows that the error obtained is very small and just about the same as the EKF, plus the advantage of the SVD algorithm is not influenced by initial errors and it's algorithm does not depend of the control moment applied to the satellite, since, disregards the equations of motion. On the other hand, the EKF algorithm is influenced by the initial state error and it is strongly influenced by the tuning of the covariances matrices and by the control.

The EKF was applied to the real data of PoSAT-1satellite. As expected, the results obtained from the PoSAT-1 simulation were better compared to those coming from the PoSAT-1 real data for two main reasons. First, the attitude sensor measurements are sampled at a larger rate in PoSAT-1 satellite, $\Delta T = 20$ sec compared to the $\Delta T = 1$ sec sampling period for the SimSat simulation. This allows the PoSAT-1 to change its movement considerably from one sample to the next while for the simulated PoSAT-1 having the data sampled in a short period gives almost a continuous function of the PoSAT-1 behaviour through time. Also the results of

the PoSAT-1 are compared to a very simplified EKF, valid only for small pitch and roll angles, while the results from the EKF applied to the PoSAT-1 simulation are compared to a simulated attitude.

The choice of either EKF or SVD algorithms for application in a real satellite depends mainly on the accuracy required to accomplish the satellite mission. Overall, the EKF and the SVD (whenever both sensors are available) have very similar results, so if attitude sensors are available along a whole orbit a good choice would be the SVD (or another point-to-point algorithm depending on the number of attitude sensors available). The SVD does not have problems of divergence and does not require covariances matrices tuning. It also does not require initial conditions to start running the algorithm. However, the lack of attitude sensors available is a considerable disadvantage. One may say that the EKF accuracy presented in this work could be much better if the covariance matrices were better tuned. But, as stressed before, the covariance matrices presented, especially the process noise covariance matrix, are very simple, without exhaustive tuning, and still the results are good.

Future work includes:

- to improve the performance of the SVD algorithm during the periods of orbit for which only magnetometers are available.
- to validate the SVD algorithm with real data from a small satellite.
- the use of measurements from another sensor always available in space for small satellites, GPS combined with the magnetometer measurements [52] or [10], to improve the SVD performance and to obtain a more fair comparison of the two methods in the way that the SVD would be running the whole orbit and not just parts of it.
- regarding the previous suggestion, to analyse the possibility of using other point-to-point method besides SVD, concerning when more than two attitude sensors are available.
- to implement a equivalent Kalman filter that handles numerical instability as described in Section 3.1.2, in order to upload the algorithim to the real satellite.

Bibliography

- [1] Bak, T., "Spacecraft Attitude Determination - a Magnetometer Approach", Ph.D thesis 1999, Aalborg University.
- [2] Bak, T., "on-board Models for Autonomous Attitude Determination", TN T03-06-021, 1995, Aalborg University.
- [3] Battin, R. H., "An Introduction to the Mathematics and Methods of Astrodynamics, Revised Edition", AIAA Educational Series.
- [4] Bierman, G. J., "Factorization Methods for discrete Sequential Estimation", Mathematics in Science and Engineering, Vol. 129, Academic Press 1997.
- [5] Clements, R., and Tavares, P., "Update on status of the Small Satellite Simulator", RT-402-99, Instituto de Sistemas e Robótica (ISR), Instituto Superior Técnico (IST).
- [6] Clements, R., Tavares, P., Lima, P., "Small Satellite Attitude Closed Loped Control using a Kalman Filter Estimator", Proc. of IEEE Int. Symp. on Intelligent Control 2000.
- [7] Crassidis, J. L., and Markley, F. L., "An MME-Based Attitude Estimator Using Vector Observations", Proceedings of the Flight Mechanics/Estimination Theory Symposium, NASA Goddard Space Flight Center, Greenbelt, MD, 1995, pp. 137-151.
- [8] Crassidis, J. L., and Markley, F. L., "Attitude Estimation Using Modified Rodrigues Parameters", Proceedings of the Flight Mechanics/Estimination Theory Symposium, NASA Goddard Space Flight Center, Greenbelt, MD, 1996, pp. 71 -83.
- [9] Crassidis, J. L. and Markley, F. L. "Predictive Filtering for Attitude Estimation Without Rate Sensors", Journal of Guidance, Control and Dynamics, Vol. 20, No. 3, May-June, 1997. pp. 522-527.

BIBLIOGRAPHY

- [10] Crassidis, John L., Markley, F. Landis, "New Algorithm for Attitude Determination Using Global Positioning System Signals", Journal of Guidance, Control and Dynamics, Vol.20, No. 5, September-October 1997.
- [11] Davenport, P., "A Vector Approach to the Algebra of Rotations with Applications", NASA Technical Note TN D-4696, August 1968.
- [12] Farrell, J. L., "Attitude Determination by Kalman Filtering", Automatica, Vol. 6, No 3, 1970, pp 419-430.
- [13] Gelb, A., "Applied Optimal Estimation", MIT Press, Cambridge, MA, 1974.
- [14] Grewal,M. S., Andrews, A. P., "Kalman Filtering Theory and Practice", Information and System Science Series, Prentice Hall.
- [15] Hamilton, W.R., "Elements of Quaternions", Longmans, Green and Co., London 1866
- [16] Hashida, Y., Surrey Satellite Technology Ltd, University of Surrey, U. K., Internal Report, 1997
- [17] Hildebrand, F. B., "Advanced Calculus for Applications", Prentice-Hall, Inc.,Englewood Cliffs, N.J.,1962.
- [18] Hwang, Robert G. B. P. Y., "Introduction to Random Signals and Applied Kalman Filtering-with Matlab Exercises and Solutions", Third-edition.
- [19] Idan, M., "Estimation of the Rodrigues Parameters from Vector Observations", IEEE Transactions on Aerospace and Electronic Systems, Vol. AES 32 No. 2, 1996, pp. 578-586.
- [20] Itzhack Y. Bar-Itzhack, Markley, L. and Detchmann, J. K. "Quaternion Normalization in Additive EKF for Spacecraft Attitude Determination", In proc.:Flight Mechanics/Estimation Symposium.
- [21] Itzhack Y. Bar-Itzhack, Reiner, J., "Recursive Attitude Determination from Vector Observations: Direction Cosine matrix Identification", Journal of Guidance, Control and Dynamics, Vol. 7, No. 1, January-February 1984, pp. 51-56.
- [22] Itzhack Y. Bar-Itzhack, Oshman, Y., "Attitude Determination from Vector Observations: Quaternion Estimation", IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-21, No. 1, January 1985.
- [23] Itzhack Y., Bar-Itzhack. and Idan M., "Recursive Attitude Determination from vector Observations: Euler Angle Estimation", Journal of Guidance, Control and Dynamics, Vol. 10, No. 2, 1987, pp. 152-157.

BIBLIOGRAPHY

- [24] Itzhack, Y., Bar-Itzhack and Deutchmann, J. K., "Extended Kalman Filter for Attitude Estimation of the Earth radiation Budget Satellite", Proceedings of the AAS Astrodynamics Conference (Portland, OR), American Astronautical Society, Springfield, VA, 1990, pp. 786-796 (AAS Paper 90-2964).
- [25] Itzhack Y., Bar-Itzhack, "REQUEST: A Recursive QUEST Algorithm for Sequential Attitude Determination", Journal of Guidance, Control and Dynamics, Vol. 19, No. 5, September-October 1996, pp. 1034-1038.
- [26] Keat, J., "Analysis of Least-Squares Attitude Determination Routine DOAOP", CSC Report Csc/TM-77/6034, February 1977.
- [27] Lefferts, E. J., Markley, F. L., and Shuster, M. D., "Kalman Filtering for Sapacecraft Attitude Estimation", Journal of Guidance, Control and Dynamics, Vol. 5, No. 5, 1982, pp. 417-429.
- [28] Lerner, G. M., "Three-Axis Attitude Determination", Spacecraft Attitude Determination and Control, edited by J. R. Wertz, Reidel, Dordrecht, The Netherlands, 1978, pp. 420-428.
- [29] Lu, P., "Nonlinear Predictive Controllers for continuous Systems", Journal of Guidance, Control and Dynamics, Vol. 17, No. 3, 1994, pp. 553-560.
- [30] Lu, P., "Nonlinear Predictive Controllers for Continuous Systems", Journal of Guidance, Control, and Dynamics, Vol. 17, No. 3, May-June, 1994, pp. 553-560.
- [31] Martel F., Pal P.K. and Psiaki M.L., "Three-Axis Attitude Determination via Kalman Filtering of Magnetometer Data", Journal of Guidance, Control and Dynamics, Vol. 13, No. 3, 1989, pp. 506-514.
- [32] Marandi, S. R., and Modi, V. J., "A Preferred Coordinate System and Associated Orientation Representation in Attitude Dynamics", Acta Astronautica, Vol. 15, No. 11, 1987, pp. 833-843.
- [33] Markley, F. L., "Attitude Determination using Vector Observations and Singular Value Decomposition", The Journal of the Astronautical Sciences, Vol. 36, No. 3, July-September 1988, pp.245-258
- [34] Markley, F. L., "Attitude Determination from Vector Observations: A Fast Optimal matrix Algorithm", Journal of the Astronautical Sciences, Vol. 41, No. 2, 1993, pp. 261-280.
- [35] Markley, F. ,L., Mortari, D., "How To Estimate Attitude From Vector Observations", AIAA/AAS Astrodynamics Specialist Conference, Girdwood, Alaska,16-19 August 1999.

BIBLIOGRAPHY

- [36] Marques, Sónia, Clements, R., Lima, P. "Comparison of Small Satellite Attitude Determination Methods", in Proceedings of 2000 AIAA Conference on Navigation, Guidance and Control, 14-17 August 2000, Colorado, USA.
- [37] Mook, D. J., and Junkins, J. LL., "Minimum Model Error Estimation for Poorly Modelled Dynamics Systems", Journal of Guidance, Control and Dynamics, Vol. 3, No. 4, 1998, pp. 367-375.
- [38] Psiaki M. L., Martel, F. and Pal, P., "Three-Axis Attitude Determination via Kalman Filtering of Magnetometer Data", Journal of Guidance, Control and Dynamics, Vol.13, No3, May-June 1990, pp. 506-514.
- [39] Shuster, M. D., and Oh, S. D., "Three-Axis Attitude Determination from vector Observations", Journal of Guidance and Control, Vol. 4, No. 1, 1981, pp 70-77.
- [40] Shuster, M. D., "A Simple Kalman Filter and smoother for Spacecraft Attitude", Journal of the Astronautical Sciences Corp., CSC/TM-78/6056, Silver Spring, MD, April 1978.
- [41] Shuster, M. D. "A Survey of Attitude Representations", Journal of the Astronautical Sciences, Vol. 41, No. 4, 1993, pp. 439-517.
- [42] Sidi, Marcel J. "Spacecraft Dynamics and Control-A Practical Engineering Approach", Cambridge Aerospace Series 7, Cambridge University Press, 1997.
- [43] Sontag, E., "Mathematical Control Theory - Deterministic Finite Dimensional Systems", 2nd ed., 1998 Springer-Verlag New-York, Inc.
- [44] Sousa, B."Simulador da Dinâmica de Atitude de Micro-Satélites", TFC - Engenharia Aeroespacial, Instituto Superior Técnico, 1997.
- [45] Steyn, W.H., "A Multi-mode Attitude Determination and Control System for Small Satellites", University of Stellenbosch, December 1995.
- [46] Tabuada, P., Alves, P., Tavares, Lima P. "Attitude Control Strategies for Small Satellites", ISR Internal Report RT-404-98 (62 pages).
- [47] Tabuada, P., Alves, P., Tavares, P., Lima, P., "A Predictive Algorithm for Attitude Stabilisation and Spin Control of Small Satellites", European Control Conference (ECC'99), Karlsruhe - Germany.
- [48] Tavares, P., Sousa, B. and Lima, P., "A Simulator of Satellite Attitude Dynamics", Proc. of CONTROLO 98, Coimbra, Portugal, 1998.

BIBLIOGRAPHY

- [49] Tavares, P., Clements R., "Update on Status of the Small Satellite Simulator", ISR Internal Report RT-402-99 April 1999 (16 pages).
- [50] Tavares, P, "SimSat User's Manual", ISR Internal Report RT-405-99 September 1999 (20 pages).
- [51] Thornton, C. L., "Triangular Covariance Factorization for Kalman Filtering", PhD. thesis, University of California at Los Angeles, 1976
- [52] Unwin, MJ, Hashida, Y., Pasetti, A., "A New Receiver for Small Satellite Positioning and Attitude Determination Experiment in Orbit", Proceedings Third International Conference on Spacecraft Guidance, Navigation and Control Systems, ESTEC, Noordwijk, The Netherlands, 26-29 November 1996, ESA SP-3817
- [53] Vathsal, S., "Spacecraft Attitude Determination Using a Second-Order Non-linear Filter", Journal of Guidance, Control and Dynamics, Vol. 10, No. 6, Nov.-Dec., 1987.
- [54] Wahba, G., "A Least-Squares Estimate of Satellite Attitude", SIAM Review, Vol. 7, No. 3, 1965, p.409 (problem 65-1).
- [55] Wertz, J. R., "Spacecraft Attitude Determination and Control", Astrophysics and Space Science Library Vol. 73, Kluwer Academic Publishers.
- [56] Wisniewski, R., Astolfi, A., Bak, T., Blanke, M., Lima, P., Spindler, K., Tabuada, P., Tavares, P., "Satellite Attitude Control Problem", chapter in Control of Complex Systems (COSY), Springer-Verlag, Berlin, 2000.
- [57] "Small Satellites Home Page", Surrey Space Centre, "<http://www.ee.surrey.ac.uk/CSER/UOSAT/SSHP/>"

Appendix A

Discrete Kalman Filter

The Kalman filter is the optimal (in the sense of minimizing the estimate variance) linear estimator of the state of a multiple-input/output linear system using a dynamical model of the process and a model of the sensor measurements.

In this appendix, the discrete-time version of the filter is summarized, following the derivation in [18] and [13].

A.1 Process Model

Given a linear system model, on a priori estimate, $\hat{\mathbf{x}}_k^-$, of the system state at time t_k , the measurements, \mathbf{y}_k , assumed statistics of the system noise and measurement errors, plus initial condition information, the Kalman filter processes the measurements data at each step in order to update the state estimate $\hat{\mathbf{x}}_k^+$. Given a continuous process described by

$$\dot{\mathbf{x}} = F\mathbf{x} + Gu \quad (\text{A.1})$$

where the u is a forcing function whose elements are white noise, and assuming that the continuous process is sampled at discrete times $t_0, t_1, \dots, t_{k-1}, t_k, \dots$, an equation relating the samples of \mathbf{x} may be written using state space methods

$$\mathbf{x}(t_{k+1}) = \Phi(t_{k+1}, t_k)\mathbf{x}(t_k) + \int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \tau)G(\tau)u(\tau)d\tau \quad (\text{A.2})$$

or, in an abbreviated notation,

$$\mathbf{x}_{k+1} = \Phi_k \mathbf{x}_k + \mathbf{w}_k \quad (\text{A.3})$$

where Φ_k is the transition matrix and the w_k is a white sequence with known covariance,

$$E [\mathbf{w}_k \mathbf{w}_k^T] = \begin{cases} Q_k & , i = j \\ 0 & , i \neq j \end{cases} \quad (\text{A.4})$$

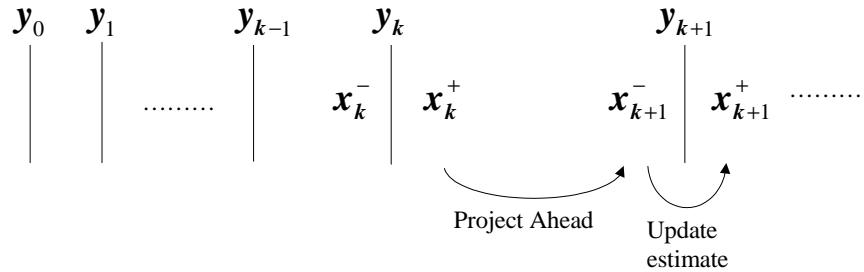


Figure A.1: Continuous process sampled at discrete times and an discrete measurement model.

A.2 Measurement Model

The process measurements are taken as linear combinations of the system state variables and occur at discrete points in time

$$\mathbf{y}_k = H_k \mathbf{x}_k + \mathbf{v}_k \quad (\text{A.5})$$

where the covariance matrix for the w_k and v_k are given by

$$E[\mathbf{v}_k \mathbf{v}_k^T] = \begin{cases} R_k, & i = j \\ 0, & i \neq j \end{cases} \quad (\text{A.6})$$

\mathbf{w}_k and \mathbf{v}_k are assumed to uncorrelated: $E[\mathbf{w}_k \mathbf{v}_k^T] = 0$ for all j and i . (A.7)

A.3 Update estimate

Defining the estimation error to be

$$\begin{aligned} \mathbf{e}_k^- &= \mathbf{x}_k - \hat{\mathbf{x}}_k^- \\ \mathbf{e}_k^+ &= \mathbf{x}_k - \hat{\mathbf{x}}_k^+ \end{aligned} \quad (\text{A.8})$$

A.3. UPDATE ESTIMATE

where the $\hat{\mathbf{x}}_k^-$ denotes the *a priori* estimate of the system state at time t_k and $\hat{\mathbf{x}}_k^+$ denotes the *a posteriori* estimate.

The *a posteriori* estimate, $\hat{\mathbf{x}}_k^+$ can be updated based on the use of the measurements \mathbf{y}_k and given *a priori* estimate, $\hat{\mathbf{x}}_k^-$ at time t_k .

$$\hat{\mathbf{x}}_k^+ = K'_k \hat{\mathbf{x}}_k^- + K_k \mathbf{y}_k \quad (\text{A.9})$$

where K'_k and K_k are time varying gain matrices, yet unspecified.

Substituting Eq. (A.8) into Eq. (A.9) and using Eq. (A.5) the result is,

$$\mathbf{e}_k^+ = \mathbf{x}_k \left[\mathbf{1} - K'_k - K_k H_k \right] + K'_k \mathbf{e}_k^- - K_k \mathbf{v}_k \quad (\text{A.10})$$

Calculating the mean value of the previous equation,

$$E[\mathbf{e}_k^+] = E \left[\mathbf{x}_k \left[\mathbf{1} - K'_k - K_k H_k \right] \right] + E \left[K'_k \mathbf{e}_k^- \right] - E \left[K_k \mathbf{v}_k \right] \quad (\text{A.11})$$

By definition $E[\mathbf{v}_k] = 0$. If $E[\mathbf{e}_k^-] = 0$, the estimator will be unbiased (*i.e.*, $E[\mathbf{e}_k^+] = 0$) for any state vector \mathbf{x}_k only if $K'_k = \mathbf{1} - K_k H_k$. So the updating estimate equation (A.9) is given by

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k (\mathbf{y}_k - H_k \hat{\mathbf{x}}_k^-) \quad (\text{A.12})$$

where the optimal gain matrix K_k is chosen in order to minimize the elements along the major diagonal of the error covariance matrix P_k^+ , because the error covariance matrix provides an indication of the accuracy of the estimate produced by the filter. The expression $(\mathbf{y}_k - H_k \hat{\mathbf{x}}_k^-)$ is known as the error innovation,

$$\mathbf{e}_k = \mathbf{y}_k - H_k \mathbf{x}_k^- \quad (\text{A.13})$$

So the previous expression follows,

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k \mathbf{e}_k \quad (\text{A.14})$$

The state vector can also be computed in terms of the perturbation $\delta \mathbf{x}_k$, defined as the term $K_k \mathbf{e}_k$,

$$\hat{\mathbf{x}}_{k+1}^+ = \hat{\mathbf{x}}_{k+1}^- + \delta \mathbf{x}_{k+1} \quad (\text{A.15})$$

A.4 Covariance Matrix and Optimal Gain Matrix

From the definition of covariance of the error,

$$P_k^+ = E[\mathbf{e}_k^+ \mathbf{e}_k^{+T}] = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^+)(\mathbf{x}_k - \hat{\mathbf{x}}_k^+)^T] \quad (\text{A.16})$$

Replacing the Eq. (A.5) into the Eq. (A.12) and substituting the resulting expression for $\hat{\mathbf{x}}_k^+$ in the Eq. (A.16), the equation becomes

$$P_k^+ = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^- - K_k(H_k \mathbf{x}_k + v_k - H_k \mathbf{x}_k^-))(\mathbf{x}_k - \hat{\mathbf{x}}_k^- - K_k(H_k \mathbf{x}_k + v_k - H_k \mathbf{x}_k^-))^T] \quad (\text{A.17})$$

Since \mathbf{e}_k^- is uncorrelated with the measurement error \mathbf{v}_{k+1} , the previous equation becomes,

$$P_k^+ = (\mathbf{1} - K_k H_k) P_k^- (\mathbf{1} - K_k H_k)^T + K_k R K_k^T \quad (\text{A.18})$$

Since the elements along the major diagonal of P_k represent the variance of the estimation error for the state variables, the trace of P_k has to be differentiated to obtain the updating equation for the optimal gain matrix.

$$\frac{\partial(\text{trace } P_k)}{\partial K_k} = -2(H_k P_k)^T + 2K_k(H_k P_k H_k + R_k) \quad (\text{A.19})$$

and be equal to zero to obtain the optimal gain matrix

$$K_k = P_k^- H_k^{-T} [H_k^- P_k^- H_k^{-T} + R_k]^{-1} \quad (\text{A.20})$$

Substituting the optimal gain equation into Eq. (A.18), the covariance equation becomes,

$$P_k^+ = [\mathbf{1} - K_k H_k^-] P_k^- \quad (\text{A.21})$$

The expression from Eq. (A.18), is called **the Joseph Form**, and is usually used in the algorithms instead of Eq. (A.21), because it has a natural symmetry avoiding divergence problems in P matrix and is the only one valid for optimal or suboptimal gain matrix. However when the gain is optimal the results of both expressions are identical.

A.5 Propagation of the state variables

Between measurements it is desirable to obtain an estimate and the error covariance matrix, since the optimal gain depends on the error covariance matrix, at

a later time, t_{k+1} , in order to make optimal use of \mathbf{y}_k in Eq. . The updated estimate is projected ahead via the transition matrix which relates the state-vector at time t_k with the state vector at t_{k+1} ,

$$\hat{\mathbf{x}}_k^- = \Phi_{k-1} \hat{\mathbf{x}}_{k-1}^- \quad (\text{A.22})$$

assuming that the w_k is zero mean and is not correlated with any of its terms.

Assuming that F is constant over an interval of interest ($k, k + 1$), then the transition matrix is simply the matrix exponential of $F\Delta T$, where ΔT is the sampling period,

$$\Phi_k = e^{F\Delta T} = \mathbf{1} + F\Delta T + \frac{(F\Delta T)^2}{2!} + \frac{(F\Delta T)^3}{3!} + \dots \quad (\text{A.23})$$

A.6 Propagation of the Error Covariance Matrix

To project the error covariance matrix, P_k^- , let us write P_k^- by definition

$$P_k^- = E[\mathbf{e}_k^- \mathbf{e}_k^{-T}] \quad (\text{A.24})$$

Substituting Eq. (A.8) into the previous,

$$P_k^- = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T] \quad (\text{A.25})$$

and using Eq. (A.22) the error covariance matrix becomes

$$P_{k+1}^- = E[(\Phi_k \mathbf{x}_k + \mathbf{w}_k - \Phi_k \hat{\mathbf{x}}_k^+)(\Phi_k \mathbf{x}_k + \mathbf{w}_k - \Phi_k \hat{\mathbf{x}}_k^+)^T] \quad (\text{A.26})$$

$$\boxed{P_{k+1}^- = \Phi_k P_k^- \Phi_k^T + Q_k} \quad (\text{A.27})$$

A.7 KF Algorithm

Between measurements

PROJECT AHEAD

1. Propagation of the state variable (A.22), $\hat{\mathbf{x}}_{k+1}^- = \Phi_k \hat{\mathbf{x}}_k^+$
2. Compute transition matrix (A.23), $\Phi_k \approx \mathbf{1} + F_k \Delta T + \frac{(F_k \Delta T)^2}{2!}$
3. Propagation of the error covariance matrix (A.27), $P_{k+1}^- = \Phi_k P_k^+ \Phi_k^T + Q_{k+1}$

Across measurements \mathbf{y}_{k+1}

1. Update Kalman gain (A.20), $K_{k+1} = P_{k+1}^- H_{k+1}^T [H_{k+1} P_{k+1}^- H_{k+1}^T + R_{k+1}]^{-1}$
2. Innovation Error (A.13), $\mathbf{e}_{k+1} = \mathbf{y}_{k+1} - H_{k+1} \hat{\mathbf{x}}_{k+1}^-$
3. Perturbation update (A.14), $\delta \mathbf{x}_{k+1} = K_{k+1} \mathbf{e}_{k+1}$
4. Update estimate (A.15), $\hat{\mathbf{x}}_{k+1}^+ = \hat{\mathbf{x}}_{k+1}^- + \delta \mathbf{x}_{k+1}$
5. Update the error covariance matrix (A.18),

$$P_{k+1}^+ = [\mathbf{1} - K_{k+1} H_{k+1}] P_{k+1}^- [\mathbf{1} - K_{k+1} H_{k+1}]^T + K_{k+1} R_{k+1} K_{k+1}^T$$

Follow to the project ahead $\hat{\mathbf{x}}_{k+1}^+ \equiv \hat{\mathbf{x}}_k^+$ and $P_{k+1}^+ = P_k^+$

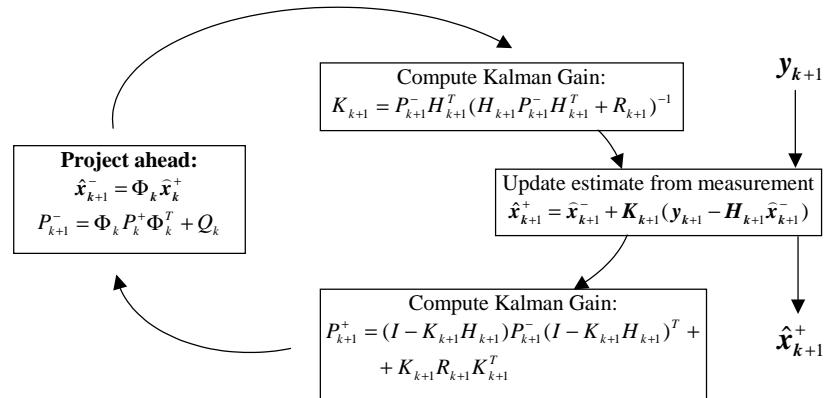


Figure A.2: Kalman filter loop.

Appendix B

Singular Value Decomposition (SVD)

The deterministic or point-to-point methods consist of determining the attitude based on a set of noisy vector measurements. So, given a set of $n \geq 2$ vector measurements $\mathbf{b}_1, \dots, \mathbf{b}_n$ in the body system and a set of reference vectors $\mathbf{r}_1, \dots, \mathbf{r}_n$ in the orbit system, there exists an orthogonal matrix A that transforms rotational vectors from the orbital to the body coordinates. The problem of finding the best estimate of the A matrix was posed by Wahba [54] who was the first to choose a least square criterion to define the best estimate, *i.e.*, find the orthonormal matrix A that minimizes the loss function

$$L(A) = \frac{1}{2} \sum_{i=1}^n w_i^2 \| \mathbf{b}_i - A \mathbf{r}_i \| \quad (\text{B.1})$$

where $\| \cdot \|$ denotes the Euclidean norm and w_i are a set of positive weights assigned to each measurement and assuming that at least one observation is available, the weights can be normalized yielding to

$$\sum_{i=1}^n w_i = 1 \quad (\text{B.2})$$

Then after some manipulation the loss function can be written as following,

$$L(A) = 1 - \sum_{i=1}^n w_i \mathbf{b}_i^T A \mathbf{r}_i = 1 - \text{TR}(AB^T) \quad (\text{B.3})$$

where

$$B \equiv \sum_{i=1}^n w_i \mathbf{b}_i \mathbf{r}_i^T \quad (\text{B.4})$$

B.1 SVD Solution to Wahba's Problem

In this section we follow the description of the method in [33]. There always exist the matrices U , S and V such that

$$B = USV^T \quad (\text{B.5})$$

The matrices U and V are orthogonal matrices and S a diagonal matrix with singular values of Y

$$S = \text{diag}(s_1, s_2, s_3) \quad (\text{B.6})$$

with

$$s_1 \geq s_2 \geq s_3 \geq 0 \quad (\text{B.7})$$

It has the equivalent to write the proper orthogonal matrices

$$U_+ \equiv U \left[\text{diag} \begin{pmatrix} 1 & 1 & \det U \end{pmatrix} \right] \quad (\text{B.8})$$

$$V_+ \equiv V \left[\text{diag} \begin{pmatrix} 1 & 1 & \det V \end{pmatrix} \right] \quad (\text{B.9})$$

and

$$W \equiv U_+^T A V_+^T \quad (\text{B.10})$$

It is possible for any proper orthogonal matrix to be expressed in terms of the Euler axis/angle representation, from (2.12)

$$W = \mathbf{1}_{3 \times 3} \cos \Phi + (1 - \cos \Phi) \hat{\mathbf{e}} \hat{\mathbf{e}}^T - \sin \Phi [\hat{\mathbf{e}} \times] \quad (\text{B.11})$$

where Φ is the rotation angle, $\hat{\mathbf{e}}$ the unit vector and $[\hat{\mathbf{e}} \times]$ the skew-symmetric matrix (2.31). Knowing that $\det(U)\det(V) = \pm 1$ the matrix S' can also be define,

$$S' \equiv \left[\text{diag} \begin{pmatrix} s_1 & s_2 & ds_3 \end{pmatrix} \right] \quad (\text{B.12})$$

In terms of these matrices, Eq (B.5) can be written as follows,

$$B = U_+ S' V_+^T \quad (\text{B.13})$$

Substituting this into Eq (B.3) and using the cyclic invariance of the trace and Eq. (B.11) gives

$$L(A) = 1 - \text{TR}(S' W) \quad (\text{B.14})$$

$$= 1 - \text{TR}(S') + (1 - \cos \Phi) [s_2 + ds_3 + (s_1 - s_2) \hat{\mathbf{e}}_2^2 + (s_1 - ds_3) \hat{\mathbf{e}}_3^2] \quad (\text{B.15})$$

Remembering (B.7), the loss function is minimized for $\Phi = 0$, which gives $W = 1$

$$L(A_{opt}) = 1 - TR(S') = 1 - s_1 - s_2 - s_3 - ds_3 \quad (\text{B.16})$$

Thus from (B.10)

$$A_{opt} = U_+^T V_+ \quad (\text{B.17})$$

$$= U [diag(1, 1, \det U \det V)] V^T \quad (\text{B.18})$$

Hence, the SVD solution to Wahba's problem is,

$$A_{opt} = U \begin{bmatrix} \mathbf{1}_{2x2} & 0_{2x2} \\ 0_{2x2} & \det(U) \det(V) \end{bmatrix} V^T \quad (\text{B.19})$$

B.1.1 Covariance matrix

It is of statistical interest to compute the covariance of the attitude estimate, which is a measure of the estimation errors coming from the measurement and reference vectors,

$$P = U \begin{bmatrix} \frac{1}{s_1 + \sigma} & 0 & 0 \\ 0 & \frac{1}{s_2 + \sigma} & 0 \\ 0 & 0 & \frac{1}{s_3 + \sigma} \end{bmatrix} U^T \quad (\text{B.20})$$

where

$$\sigma = \det(U) \det(V) S_3 \quad (\text{B.21})$$

As stressed by [33] the uniqueness of the solution is closely related to the rank of the matrix Y , which is equal to the number of non-zero singular values (number of linearly independent reference vectors). If the rank of Y is less than two, $s_2 = s_3 = 0$, and the attitude matrix is not unique.

Appendix C

Keplerian Orbital Elements

A set of five constants and one quantity which varies with time, are needed to accurately describe the size, shape and orientation of an orbit and the position of a satellite in that orbit at a given *epoch*¹ (time), called Keplerian elements² or Classical Orbital Elements which are:

a	:	semimajor axis	
e	:	eccentricity	
i	:	inclination	(C.1)
Ω	:	right ascension of the ascending node	
ω_p	:	argument of perigee	
θ	:	true anomaly	

In the Keplerian model, the satellites orbit in an ellipse with the center of the Earth at one of the ellipses's focus points. The position in the orbit closest to the Earth is called *perigee* and the position furthest from the Earth is called *apogee*.

C.1 Orbit size and shape

a - semimajor axis describes the size of the ellipse and it is half the distance between apogee and perigee. By Kepleer's third law of planetary motion, the orbital period is computed from the semimajor axis. Sometimes, instead of

¹**Epoch** is a number indicating the exactly time in which the Two-Line Orbital Element Set Format (Appendix D) is obtained. This element gives the time for which the orbital elements are correct.

²For a better understanding of this issue, see [3] and [55].

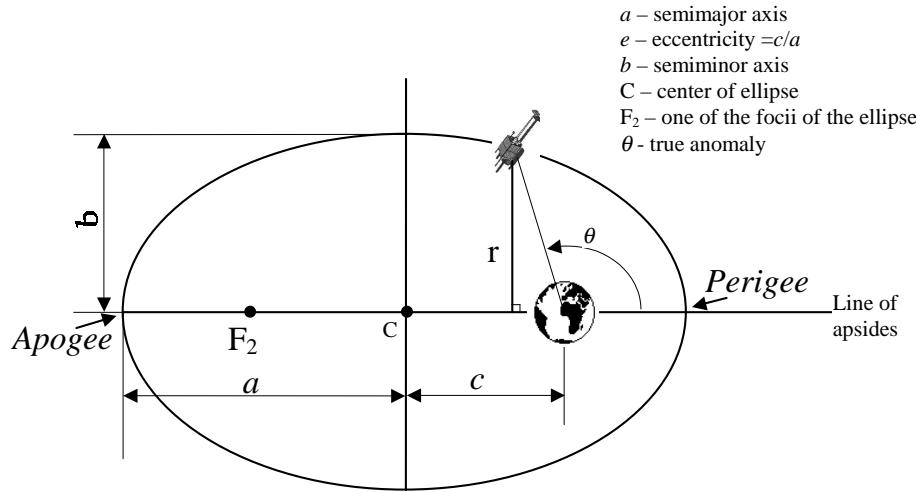


Figure C.1: Definition of the orbit size and shape parameters.

the semimajor axis the **mean motion** (*MM*) is given, which is the orbital frequency and the reciprocal of the period.

The mean motion is the average rate of satellite motion. The mean motion is usually given in revolutions per day for Earth satellites.

e - Eccentricity determines the shape of the orbital ellipse. Formally, this is ratio between distance from the center of the ellipse (which isn't the center of the Earth) to the focus of the ellipse (which is the center of the Earth) and the semi-major axis. An ellipse with an eccentricity of 0 is a circle, and when *e* is near 1, the ellipse is long and skinny.

C.2 Orbital orientation

i - Inclination describes the orbit ellipse plane's tilt angle *w.r.t.* the plane of the equator and it varies from 0° to 180° . Inclinations of near 0° are called equatorial orbits, and those near 90° are called polar orbits. By convention, orbits that go the same way as the Earth rotates (prograde or counter-clockwise from above) have inclinations of 0° to 90° . Satellites that orbit retrograde, opposite to the rotation of the Earth, have inclinations great than 90° . For example, a satellite with an inclination of 180° is in an equatorial orbit going east to west.

Ω - Right Ascension of Ascending Node is another angle that aligns the orbit ellipse in space. The intersection of the orbit plane and the equatorial plane is called

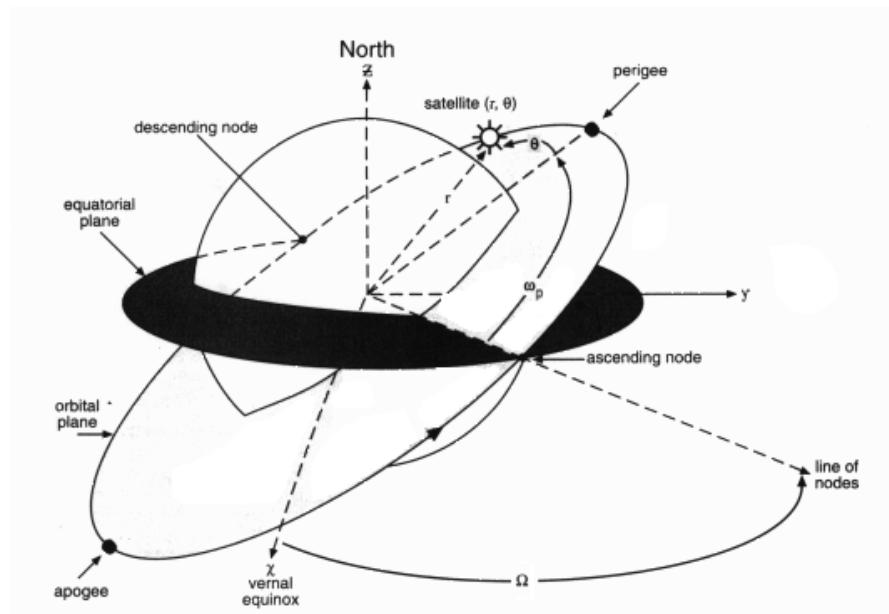


Figure C.2: Definition of the orbital orientation parameters.

the **lines of nodes**. The point where the satellite's orbit crosses the equator going south to north is called the **ascending node**. The one on the opposite side of the Earth, where the satellite passes into the Southern Hemisphere is called the descending node. Since the orbit is fixed relative to the stars and not to the surface of the Earth, the astronomical coordinate system of right ascension and declination is needed to measure the position of the ascending node. Right ascension is an angle measured in the equatorial plane from a fixed point in space, called the point of Ares (which is also the point of the vernal equinox, where the Sun crosses the equator in the spring). **Declination** is the angle measured up from the equatorial plane, just like inclination.

- θ - True Anomaly is the angle measured in the direction of motion from the perigee to the satellite's position at the given *epoch* time. **Mean anomaly** (M_o) describes what the satellite's true anomaly would be if it were in a circular orbit where the satellite moves at a constant rate around the orbit. In this case the angle would point directly at the satellite. A satellite in a non-circular orbit doesn't move at a constant velocity so this relation does not hold. However, no matter what eccentricity, the M_o at the perigee will be 0° and at the apogee 180° . Kepler's equation relates the true anomaly and the mean anomaly for an eccentric orbit.

C.3 Orbital position

ω_p - Argument of Perigee, determines the position of the orbital ellipse in the orbital plane. The angle between the major axis and the line of nodes is the argument of perigee. This is measured in the plane of the orbit. It ranges from 0° to 360° , and is 0° when the perigee is at the ascending node and 180° when the satellite is farthest from the Earth when rises up over the equator.

C.4 Peturbations

Reality is more complex than the Keplerian model, so the models used to determine the satellites orbit in space include small corrections called perturbations which add extra parameters. These parameters, besides the keplerian elements are stored in various different formats where the Two Line Element (TLE) is the most common, (Appendix D). The peturbations are due to the lumpiness of the gravitational field and to the drag (B_{star}) on the satellite due to the atmosphere.

B_{star} - is the drag caused by the Earth's atmosphere causing the satellites to spiral downward. This element is one half the first time derivative of the Mean Motion and tells us the rate at which the Mean Motion is changing due to drag or other related effects.

Appendix D

NORAD Two-Line Element Set Format

Two-line Orbital Element Set Format is a file with a special format, shown in Table D.1, generated by NASA and distributed by NORAD, to store the general elements to obtain the location of any resident space object.

These element sets are periodically refined so as to maintain a reasonable prediction capability, but it must be reconstructed by the United States Space Command (USSPACECOM) SPG4 mathematical model in order to obtain good predictions.

The element data are the six classical orbit parameters (COP), (Appendix C): e — eccentricity; i — inclination, Ω — right ascension of the ascending node, ω_p — argument of the perigee, M_o — mean anomaly, MM — Mean Motion and an extra orbital parameter, B_{star} — drag related parameter. All the orbital parameters are correct for a certain Epoch time. Data for each satellite consists of three lines in the following format:

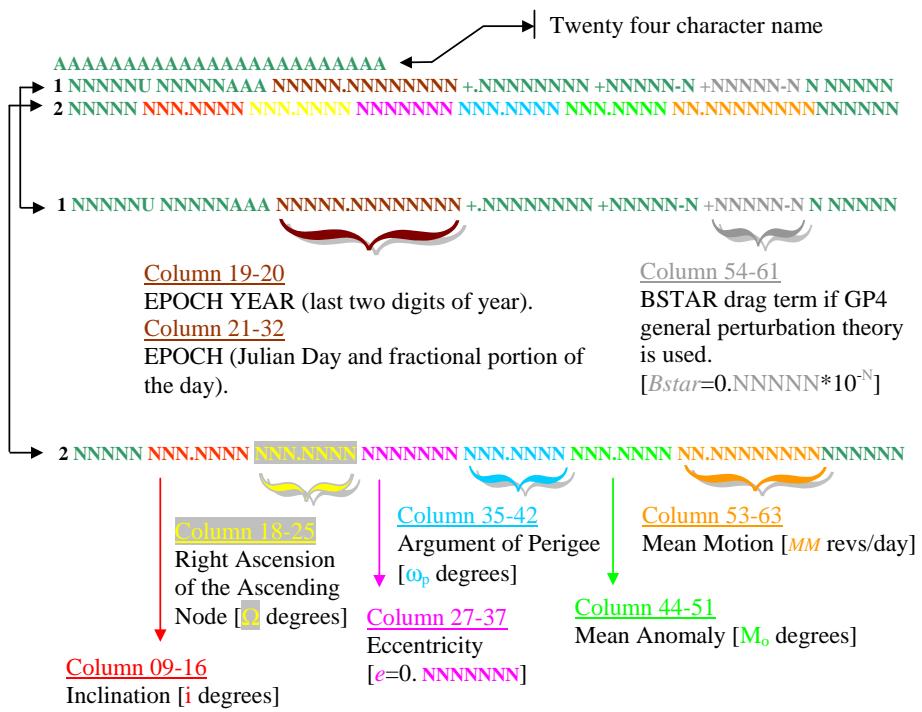


Figure D.1: The description of the compact form of the two-line element (TLE) format. It consists of three lines, where the first is the satellite's name and the next two lines correspond to the data in a rigid format.