ATTITUDE CONTROL BY MEANS OF EXPLICIT MODEL PREDICTIVE CONTROL, VIA MULTI-PARAMETRIC QUADRATIC PROGRAMMING

MASTER THESIS Spring 2004

ØYVIND HEGRENÆS

DEPARTMENT OF ENGINEERING CYBERNETICS NORWEGIAN UNIVERSITY OF SCIENCE AND TECHNOLOGY

NTNU Norges teknisk-naturvitenskapelige universitet

Fakultet for informasjonsteknologi, matematikk og elektroteknikk Institutt for teknisk kybernetikk



MASTEROPPGAVE

Kandidatens navn:	Øyvind Hegrenæs
Fag:	Teknisk Kybernetikk
Oppgavens tittel (norsk):	Styring av orientering ved bruk av eksplisitt modellprediktiv regulering, via multi-parametrisk kvadratisk programmering
Oppgavens tittel (engelsk):	Attitude Control by means of Explicit Model Predictive Control, via Multi-Parametric Quadratic Programming

Oppgavens tekst:

Dette prosjektet hører inn under ESAs Student Space Exploration & Technology Initiative (SSETI). Institutt for teknisk kybernetikk har ansvar for design og simulering av AOCS. Aktuatorene for satelitten vil være et reaksjonshjul og thrustere.

- 1. Sett deg inn i fagområdet styring av orientering for romfartøy, og da med fokus på lignende konfigurasjoner som for SSETI. Gi en presentasjon av SSETI prosjektet og AOCS for dette.
- Gi en oversikt over ulike forstyrrelser og aktuatorer, og utled en matematisk modell for orienteringen til ESEO, samt en baneestimator. Modellen skal implementeres i Simulink. Denne bør være så modulær som mulig, slik at det er enkelt å skifte ut komponenter. Denne delen av oppgaven utføres i samarbeid med MSc-student Morten Topland.
- 3. Sett deg inn i, og presenter, teorien om eksplisitt MPC (Model Predictive Control)
- 4. Design, ved hjelp av metoder fra pkt 3, regulatorer for styring av orientering i "nominal mode". Sammenlign ytelsen til regulatoren med andre kjente regulator strukturer.
- 5. Ta hensyn til at thrusterene gir et begrenset pådrag som kun kan settes ut i pulser.

Oppgaven gitt: 19. januar 2004

Besvarelsen leveres: 7. juni 2004

Besvarelsen levert:

Utført ved Institutt for teknisk kybernetikk

Trondheim, den 12. januar 2004

Jan Tommy Gravdahl Faglærer

Preface

The work presented in the following is submitted in partial satisfaction of the requirements for the degree of Master of Science in Engineering Cybernetics. It has been carried out at the Norwegian University of Science and Technology, Department of Engineering Cybernetics.

I would like to thank my supervisor Associate Professor Jan Tommy Gravdahl and advisor Postdoctoral Fellow Petter Tøndel for their support, valuable advice and interesting discussions during this work. I would also like to thank Professor David Auslander¹ for a great year at University of California at Berkeley, and for his willingness to always answer my questions.

Finally I would like thank all my fellow students for all the fun and hard work during one year in Berkeley and four years in Trondheim. Also, part of the work on establishing the satellite model was done together with MSc. student Morten Topland. Thank you for creative discussions at the office.

Øyvind Hegrenæs Trondheim 2004-06-07

¹University of California at Berkeley, Department of Mechanical Engineering

Abstract

The topic of this thesis is attitude control of a micro-satellite. Due to their physical size, microsatellites have fairly limited power supply, data storage, and computational resources, hence making it crucial to have a simple and reliable control system. A reasonable approach is to evaluate the controller off-line, and consequently reducing the need for CPU speed drastically as real-time effort in space can be restricted to a table-lookup. In many circumstances it may also be desirable to prevent the different components from operating near their thresholds, either the design focus is to keep power consumption within some limits or to keep the rate of wear as low as possible. It is shown in this thesis that all these concerns can be dealt with by formulating a Model Predictive Control problem (MPC).

It is shown that explicit solutions to constrained linear MPC problems can be computed by solving multi-parametric quadratic programs (mpQP), where the parameters are the components of the state vector. The solution to the mpQP is a piecewise affine (PWA) function, which can be evaluated at each sample to obtain the optimal control law. In addition to reducing the on-line effort, the controller can be implemented on inexpensive hardware as fixed-point arithmetics can be used. A simple second order system is included to illustrate the procedure.

An explicit MPC (eMPC) controller is derived for the SSETI/ESEO micro-satellite, initiated by the European Space Agency (esa). The structural data is based on the the latter, and the spacecraft is modelled as an ideal rigid body. Various topics within spacecraft and astrodynamics are included to provide a thorough insight in how the model is derived. To represent its attitude, the well known Euler parameters are utilized, while the dynamic equations are based on the Newton-Euler formulation. An important thing to keep in mind is that the thrusters on the satellite are on-off by nature. An attempt to solve this problem is done using a preliminary bang-bang modulation scheme.

The controller is connected in closed-loop with the nonlinear plant, and the effectiveness is demonstrated through simulations. With purpose of comparing the performance to other control schemes, we also derive and simulate PD-control and stationary LQR. As familiar to the author, the eMPC approach has not yet been applied for attitude control of satellites. A paper based on the work in this thesis is to be submitted to the American Control Conference (ACC) 2005. A preprint is given in the appendix.

Keywords: Predictive control; Constraints; Piecewise linear controllers; Linear quadratic regulators; PD-control; Spacecraft and astrodynamics; Attitude Control; Bang-bang modulation;

Contents

Pr	eface		i
Al	ostrac	t	iv
1	Intro	oduction	1
	1.1	Student Space Exploration & Technology Initiative	1
		1.1.1 SSETI/ESEO-AOCS	3
	1.2	Previous work	6
	1.3	Outline of the thesis	8
2	Atti	tude parametrization	9
	2.1	The rotation matrix	9
		2.1.1 Kinematic differential equation	0
		2.1.2 Attitude deviation	0
	2.2	Euler angles	0
	2.3	Euler parameters	1
	2.4	Rodrigues parameters	2
		2.4.1 The classical Rodrigues parameters	12
		2.4.2 The modified Rodrigues parameters	3
	2.5	The (w, z) parametrization	13
	2.6	Discussion	4
3	Spac	cecraft and astrodynamics	15
	3.1	Coordinate reference systems	15
		3.1.1 Transformations between reference frames	17
	3.2	Attitude dynamics	20
		3.2.1 Newton-Euler equations of motion for rigid bodies	20
		3.2.2 Disturbance torques	22
		3.2.3 Control torques and actuators	26
	3.3	Celestial mechanics	29
		3.3.1 A simple orbit propagator	30

vi CONTENTS

4	Mode	el and control properties	31
	4.1	Satellite model	31
		4.1.1 Linearized model	34
	4.2	Controllability	35
	4.3	Stabilizability	35
	4.4	Summary	36
5	Attitu	ide control and simulations	37
	5.1	SSETI/ESEO parameters	37
	5.2	Open-loop simulations	38
	5.3	Control design	38
		5.3.1 PD-control	39
		5.3.2 LOR	41
		5.3.3 Explicit MPC	42
	5.4	Signal modulation	50
	5.5	Simulations of the SSETI/ESEO satellite	51
		5.5.1 Controller settings	52
		5.5.2 eMPC compared to LOR and PD-control	53
		5.5.3 eMPC when exposed to extreme initial conditions	61
		5.5.4 eMPC in combination with bang-bang modulation	65
		5.5.5 Discussion	67
_			
6	Conc	lusions	73
	6.1	Further work	74
Bi	bliogra	aphy	75
A	Pape	r submitted to the 24th American Control Conference (ACC)	79
B	Com	munication and Workshops associated to ESA	85
С	MAT	LAB scripts and Simulink block diagrams	89
D	Newt	on-Euler equations of motion	107
	D.1	Translational motion	107
	D.2	Angular motion	108
	D.3	Model summary	109
Е	Orbit	tal mechanics	111
	E.1	The Two-Body problem	111
	E.2	Classical Orbital Elements COE	112
	. —	E.2.1 The orbital period	113
		E 2.2 Conversion from COEs to the ECI frame	113
	E.3	Perturbed Satellite Motion	114
F	The	- a) nonemetrization	115
r	nne (w, 2) parametrization	112

Chapter 1 Introduction

Orientation control of rigid bodies has attracted many researchers in recent years. This is due to the broad range of applications and the theoretical challenges it offer. Many real-life mechanical systems, including aircrafts, helicopters, spacecrafts, underwater vehicles, surface vessels and robots are examples of such systems. In the case of spacecrafts, we formally say that we deal with the problem of *attitude control*. More precisely, this is the process of orienting the spacecraft in a specified, predetermined direction. The problem consists of two areas called attitude stabilization and attitude maneuver (tracking) control. The first is the process of main-taining an exciting orientation, while the latter has to do with controlling the reorientation of the spacecraft from one attitude to another. The two areas are not totally distinct however, and in many cases the problem to be solved deals with both.

The purpose of this thesis is to establish and investigate a reasonable model of a micro-satellite, and then finally propose a strategy to solve the problem of attitude control. However, unlike preceding work, typically solved using PD- or LQ-control, Lyapunov design procedures, sliding mode, adaptive- or quaternion feedback techniques, \mathcal{H}_{∞} or $\mathcal{H}_2/\mathcal{H}_{\infty}$, the work in this thesis will be on explicit Model Predictive Control. As familiar to the author, this approach has not yet been applied for attitude control of spacecrafts. When doing implementation, an important thing to keep in mind is that the actuating thrusters are on-off by nature. An attempt to solve this problem is done using a preliminary bang-bang scheme.

The satellite model and technical data is based on the SSETI project initiated by the European Space Agency (esa), and the thesis is a contribution to this project. Note that whenever referred to, the Phase B documents can be found on the official SSETI/ESEO homepage¹.

1.1 Student Space Exploration & Technology Initiative

The Student Space Exploration & Technology Initiative, from now on called SSETI, was founded by esa's education office in hope of improving young people's interest in space related technology. Through the project, students from different European universities participate in designing, building and operating a micro satellite. The European Student Earth Orbiter, ESEO, is to be suitable to circle and take pictures of the Earth. Having reached this aim, further assignments are intended.

¹References and general information can be found on the official website - http://www.sseti.org

2 Introduction

The project is structured into different missions, each with different complexity level. One reason for this layered structure is that students as well as teachers, professors and space professionals can gain more experience and knowledge of the requirements for the *distributed development*² concept by taking one step after another. The missions can be further divided into different phases, ranging from Phase 0 to Phase F. The latter contains mission operations after launch, while the first phases can be described as design phases which are necessary for further development and manufacture of the satellite. The overall SSETI project structure is illustrated in Figure 1.1, where Mission 1, Mission 2 and Mission 3 are Earth Orbiter (ESEO), Moon Orbiter (ESMO) and Moon Rover (ESMR), respectively.





In addition to the satellite, the whole system consists of the payload carried by the spacecraft and the associated ground systems. The different universities have one or several teams, each responsible for one particular subsystem. Also, for some of the subsystems there are numerous teams working together. At the time of writing there are nineteen subsystems, which are summarized in Table 1.1. Since the main focus in this thesis will be on attitude control, further details on the ESEO-AOCS subsystem will be given shortly.

²By the term distributed development we denote a new and unconventional kind of communication, where all the teams communicate using modern communication tools such as the internet, chat sessions, e-mails and telephone conferences. In the end, all the different contributions are put together to make up the final product.

Attitude and Orbit Control System [AOCS]	Propulsion [PROP]
Mechanisms [MECH]	Mission analysis [MIAS]
On-board data handling [OBDH]	Electrical Power Supply [EPS]
Thermal Control [TCS]	Structures-Configuration [CONF]
SSETI Infrastructure [INFR]	Systems [SYS]
Communication [COMM]	Structures-Calculations [STRU]
Harness [HARN]	Public Relations [PR]
Simulations [SIMU]	Ground stations [GROU]
Risks analysis [RISK]	Payload [PAY]
Operations [OPER]	

Table	1.1:	SSET	ΓI/ESE	O su	lbsyst	ems

1.1.1 SSETI/ESEO-AOCS

The following information is taken from the Phase B report, which gives the major design and performance requirements for all the subsystems, including ESEO-AOCS. Some technical specifications, such as specific structural data, will be omitted at this point as they will be repeated in Chapter 5. Also, for more in-depth details the reader should refer to the report.

ESEO mission objectives

In addition to the human aspects of the SSETI project, the high level operational goals or mission objectives for ESEO can be summarized as

- 1) Acquire and uphold the required attitude during operational phases.
- 2) Perform required attitude maneuvers.
- 3) Perform required orbit maneuvers.
- 4) Compensate attitude perturbations created by orbit maneuver operations.

The satellite should be able to execute these tasks through its whole lifetime. The nominal mission duration is set to be 28 days, while the orbit maneuver window is 18 days.

In terms of attitude and control requirements, the above mentioned goals are specified according to the diversified situations the satellite is expected to face during its lifetime. We denote these situations as *attitude mission modes*, and they include launch-, initialization-, stabilization-, nominal-, safe-, failure- and transfer mode. Further details will be given shortly, when discussing AOCS requirements.

ESEO-AOCS architecture

The AOCS is one of the subsystems needed to fulfill the high level objectives mentioned above. It makes up a complex system which comprises the attitude sensors, a dedicated processor and the software for attitude and orbit determination and control. All actuators belong to the AOCS subsystem with exception of the thrusters, which are components of the propulsion subsystem.

ESEO subsystems

4 Introduction

In terms of controlling the satellite it was decided to have 8 attitude thrusters, using a cold gas system. Four of these are to be used for the *Attitude Control System*³ (ACS) and the other four for the *Reaction Control System*⁴ (RCS). A reaction wheel will also be included in the ESEO satellite. The orbital thruster, which makes up part of the *Orbit Control System* (OCS), is the only mean of achieving the orbital transfer.

A short summary of the sensors and actuators onboard ESEO is given as

Sensors
1 Magnetometer (MAG)
1 Star tracker (STAR)
2 Horizon sensors (EHS) 4 Sun sensors (SUN1, SUN2)

Table 1.2: ESEO actuators and sensors

Further details on actuator placement and constructional aspects will be given later.

Performance requirements

Closely related to the mission objectives, we state the high level AOCS requirements as

- 1) The AOCS should control the satellite in order to allow the solar panels to get maximum energy.
- 2) The AOCS should allow communication to Earth via the high gain antenna.
- 2) The AOCS should be able to perform a transfer maneuver.
- 3) The AOCS should be able to perform attitude maneuvers in order to take pictures, using either wide angle camera (WAC) or narrow angle camera (NAC).
- 4) The AOCS should have redundancy at the system level, which means that no component failure should be critical.

As mentioned earlier we have attitude mission modes associated with the performance requirements. Even though they are all relevant, only the nominal mode will be considered in this thesis.

The nominal mode refers to general spacecraft operations when in orbit. It includes the task of maintaining a stable attitude, i.e. the satellite is nadir⁵ pointing, as well as pointing the onboard camera towards a desired location. The latter is controlled by the OBDH subsystem. During the nominal phase, attitude estimation must be within 0.1° accuracy (2σ) .

³Is to be used for general attitude control operations. System is redundant to RCS.

⁴Is to be used for reaction control on orbit control perturbations. System is redundant to ACS.

⁵According to Wertz (1999), nadir is the direction towards the center of the Earth

The accuracy depends mainly on the status of the sensors. As for the control algorithms, there will be a need for precise maneuvering and the algorithms used will consequently have to guarantee low errors. At the time of writing, the choice of control scheme is not definite for the general ESEO-AOCS. However, it is most likely that a nonlinear controller would be desirable for large angle maneuvers and tracking, while for precision and set-point control a linearized model, and consequently also a linear controller, could be adequate. This will be investigated further, when deriving the controllers in later chapters.

For details on the remaining operation modes, the reader should refer to the Phase B report.

We now give a summary of some specific pointing and stability requirements. In our case the values in Table 1.5 are to be used as a reference for the remainder of this thesis.

	Axis of rotation	
	x, y (roll, pitch)	z (yaw)
Absolute pointing error	16°	No constrain
Absolute measurement error	No constrain	No constrain
Stability error	$0.12^{\circ}/\mathrm{s}$	$0.0055^{\circ}/\mathrm{s}$
Absolute rate error	$0.01^{\circ}/\mathrm{s}$	$0.0006^{\circ}/\mathrm{s}$

Table 1.3: Satellite pointing requirements during WAC operation

	Axis of rotation	
	x, y (roll, pitch)	z (yaw)
Absolute pointing error	6°	No constrain
Absolute measurement error	3°	No constrain
Stability error	$0.12^{\circ}/\mathrm{s}$	$0.0055^{\circ}/\mathrm{s}$
Absolute rate error	$0.01^{\circ}/\mathrm{s}$	$0.0006^{\circ}/\mathrm{s}$

Table 1.4: Satellite pointing requirements during NAC operation

	Axis of rotation	
	x, y (roll, pitch)	z (yaw)
Absolute pointing error	1°	5°
Absolute measurement error	0.1°	1°
Stability error	No constrain	No constrain
Absolute rate error	No constrain	No constrain

Table 1.5: Satellite pointing requirements during normal operations

6 Introduction

	Axis of rotation	
	x, y (roll, pitch)	z (yaw)
Absolute pointing error	1°	1°
Absolute measurement error	0.1°	0.1°
Stability error	No constrain	No constrain
Absolute rate error	No constrain	No constrain

Table 1.6: Satellite pointing requirements during orbit maneuvers

	Axis of rotation	
	x, y (roll, pitch)	z (yaw)
Absolute pointing error	1°	No constrain
Absolute measurement error	No constrain	No constrain
Stability error	No constrain	No constrain
Absolute rate error	No constrain	No constrain

Table 1.7: Satellite pointing requirements during communications

Based on the high level requirements and the specific requirements, it is possible to calculate torque requirements for the attitude thrusters. We also note that ACS failure is considered a worst-case scenario. This is due to the fact that the RCS consumes more fuel, and it does not offer the same accuracy as ACS. ACS and RCS torque values and boundaries are given in Table 1.8, which are the values for the thrusters manufactured by the propulsion team.

	$ au_x [\mathrm{Nm}]$	$\tau_y [\mathrm{Nm}]$	$\tau_z [\mathrm{Nm}]$
ACS			
Min	0.0306	0.0306	0.0252
Nominal	0.0484	0.0484	0.0398
Max	0.0510	0.0510	0.0420
RCS			
Min	0.1000	0.1000	0.1000
Nominal	0.1580	0.1580	0.1195
Max	0.1661	0.1661	0.1256

Table 1.8: Required torque supply from attitude thrusters

1.2 Previous work

There exist numerous research articles on the problem of attitude control for spacecrafts. Most of these deals with the case of complete control actuation using either momentum exchange devices, thrusters or magnetic actuators, while some also deal with the underactuated case. For the latter the reader should refer to Tsiotras and Doumtchenko (2000) and references therein.

The following summary gives an overview of some of the work that has been done on fully actuated attitude control for spacecrafts. Since the model to be derived later utilizes thrusters and a momentum exchange device, focus will be on similar configurations.

Controllability criterions of a rigid body equipped with thrusters and momentum wheels are addressed in Crouch (1984), while a general analytic framework for the stability analysis for a large family of globally stable tracking control laws is presented in Wen and Kreutz-Delgado (1991). For the same paper, various errors are corrected in Fjellstad and Fossen (1994).

A new class of globally asymptotically stabilizing feedback control laws is presented in Tsiotras (1994). By utilizing a non-quadratic Lyapunov function, together with stereographic projection, a linear controller is proposed for the case of three kinematic parameters. According to the author, only nonlinear controllers were known prior to this work. In Hall (1995) attention is given to the spinup dynamics of gyrostats containing a single axisymmetric rotor. It is also indicated how to apply the theory for multiple rotor gyrostats. This theory is later utilized in Hall et al. (1998), when developing tracking control laws for a rigid spacecraft using N-rotor gyrostats and thrusters. In Hall (2000) a subset of the equilibrium attitudes of a satellite with N rotors in a central gravitational field is studied. This is done using a non-canonical Hamiltonian formulation. Local stability for the equilibriums is also discussed.

Robust attitude control is addressed by means of a wide range of different schemes. In Chen and Lo (1993) this is implemented for multiaxial tracking, the actuators being pairs of opposing thrusters, using a sliding-mode design technique. Similar techniques are used together with singular perturbations and nonlinear quaternion feedback in Cavallo and Maria (1996). Nonlinear large angle robust attitude control is also obtained using \mathcal{H}_{∞} methods. This is the case in Show et al. (2001), where the satellite has fully coupled body fixed cantered thrusters. A mixed $\mathcal{H}_2/\mathcal{H}_{\infty}$ approach, together with LMI (linear matrix inequality) design, is utilized in Sun and Yang (2002) to overcome inertia matrix uncertainty and unbalanced thruster torques. In Show et al. (2003) the \mathcal{H}_{∞} controller contains linear terms for stabilization and nonlinear terms for performance enhancement. The nonlinear controller parameters are designed using a LMI method. The proposed controller is applicable to wheel controlled systems as well as thruster controlled systems.

Some work has also been done on integrated power/attitude control systems (IPACS). In Hall (1997) a cluster of four or more high-speed flywheels was used to provide large angle attitude control and energy storage, by using a singular value decomposition to decouple the wheels. Integrated attitude control and power tracking was also addressed in Tsiotras et al. (2001). For the latter, four or more energy/momentum wheels in a non-coplanar configuration and a set of three thrusters were used to implement the torque inputs. Prior to this work, exact non-linear equations of motion had not been considered in sense of IPACS, according to the authors.

The above mentioned contributions were done using rigid body models. In the case of flexible spacecrafts, the complexity increases. This could for instance be the case for satellites with large flexible solar arrays. In Wie and Plescia (1984) a nonlinear microprocessor-based reaction jet controller is presented. To reduce vibrations, and the fact that thrusters are on-off by nature, makes the use of signal modulation necessary. The aid of both static and dynamic Pulse-Width Pulse-Frequency (PWPF) modulators is also addressed in this paper. The same issues are discussed in Wie and Barba (1985), as well as comparing PWPF with the more traditional bang-bang control scheme. Some stability and control analysis are also given, by means of quaternion feedback. In addition to PWPF, Song and Agrawal (1999) use smart materials for active vibration suppression.

1.3 Outline of the thesis

The thesis is organized as follows:

- **Chapter 2:** Different parametrizations of the attitude and their properties are described. Our choice of attitude parametrization is discussed in the end.
- Chapter 3: An introduction to spacecraft dynamics, actuators and astrodynamics is given. The satellite model to be used later is derived based on the theory in this chapter.
- Chapter 4: A complete model of the satellite is presented and some important control properties are discussed.
- **Chapter 5:** A review on PD-control and LQR, together with an introduction to explicit Model Predictive Control, is given. Based on this theory, and the model in Chapter 4, attitude controllers are designed. The performance of the explicit MPC approach is compared with the other schemes, which are both well known in the literature. This is done by means of simulations in MATLAB and Simulink. Since the actuating thrusters in the model are on-off by nature, a continuous control sequence will not be applicable. A suggestion on how to solve this problem, using bang-bang modulation, is given.
- Chapter 6: Conclusions and recommendations for further work are given.
- Appendix A: Based on the work done in this thesis, a paper is to be submitted to the 24th American Control Conference (ACC) 2005, Portland, Oregon. A preprint is given in this appendix.
- Appendix B: During the process of this thesis extensive communication and information flow have been done within the SSETI project and esa. Some examples can be found in this appendix.
- Appendix C: Includes different printouts of source code and block diagrams from MAT-LAB and Simulink, respectively.
- Appendix D: Newton-Euler equations of motion for a rigid body are derived.
- **Appendix E:** An overview of important topics within celestial mechanics is given. Celestial mechanics underlies all the dynamical aspects of the orbital motion of a spacecraft and the motion of the mass center.
- Appendix F: The relatively new (w, z) parametrization is derived in detail.

Chapter 2 Attitude parametrization

Rigid body dynamics is important for a wide range of control applications, and is essential in robot control, ship control, control of aircrafts and satellites, and vehicle control in automotive systems. In the case of describing the dynamics of a rigid spacecraft, the Newton-Euler equations of motion are commonly used to provide a complete and well defined framework. For the kinematics the situation is different, due to the fact that the rotation matrix, which describes the relative orientation between two reference frames, can be parameterized in more than one way. Which parametrization to use is clearly dependent on the problem to be solved. This chapter gives an overview of different attitude parametrizations, including the relatively new (w, z) parametrization. A brief discussion can be found in the end, explaining our choice of parametrization.

2.1 The rotation matrix

The rotation matrix, also called the *direction cosine matrix*, has three interpretations;

- Describes the mutual orientation between two coordinate frames, where the column vector are cosines of the angles between the two frames.
- Transforms vectors represented in one reference frame to another.
- Rotates a vector within a reference frame.

The rotation matrix **R** from frame *a* to frame *b* is denoted \mathbf{R}_b^a . A matrix **R** is a rotation matrix if and only if it is an element of the set denoted by SO(3), that is,

$$SO(3) = \left\{ \mathbf{R} \in \mathbb{R}^{3 \times 3} : \mathbf{R}^{\mathrm{T}} \mathbf{R} = \mathbf{I} \text{ and } \det \mathbf{R} = 1 \right\},$$
(2.1)

where **I** is the 3×3 identity matrix.

A useful parametrization of the rotation matrix is the angle-axis parametrization corresponding to a rotation $\theta \in \mathbb{R}$ about a unit vector $\mathbf{k} \in \mathbb{R}^3$. *Rodrigues' formula* (Murray et al., 1994) gives

$$\mathbf{R}(\mathbf{k},\theta) = \mathbf{I} + \mathbf{S}(\mathbf{k})\sin\theta + \mathbf{S}^{2}(\mathbf{k})(1 - \cos\theta)$$
(2.2)

where $S(\cdot)$ is a skew-symmetric matrix operator and member of the set denoted by so(3);

$$so(3) = \left\{ \mathbf{S} \in \mathbb{R}^{3 \times 3} : \mathbf{S}^{\mathrm{T}} = -\mathbf{S} \right\}$$
(2.3)

In coordinate vector notation we introduce the *skew-symmetric form* of the vector $\boldsymbol{\omega}$ defined by

$$\boldsymbol{\omega}^{\times} \triangleq \mathbf{S}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}, \qquad \boldsymbol{\omega} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$
(2.4)

2.1.1 Kinematic differential equation

From the properties of SO(3), it can be shown that the kinematic differential equation for the rotation matrix can be given by the two alternative forms (Egeland and Gravdahl, 2002)

$$\dot{\mathbf{R}}^a_b = (\boldsymbol{\omega}^a_{ab})^{\times} \mathbf{R}^a_b$$
 (2.5a)

$$\dot{\mathbf{R}}^a_b = \mathbf{R}^a_b (\boldsymbol{\omega}^b_{ab})^{\times}$$
 (2.5b)

where ω_{ab}^a is the instantaneous angular velocity of frame *b* relative to frame *a* as seen from the *a* frame. Similar, ω_{ab}^b is the angular velocity of frame *b* relative to frame *a* as seen from the *b* frame. Using (2.4) we can rewrite (2.5) as

$$\dot{\mathbf{R}}^{a}_{b} = \mathbf{S}(\boldsymbol{\omega}^{a}_{ab})\mathbf{R}^{a}_{b} = \mathbf{R}^{a}_{b}\mathbf{S}(\boldsymbol{\omega}^{b}_{ab})$$
(2.6)

2.1.2 Attitude deviation

Let the frame *a* define a reference orientation and let frame *b* be a body fixed frame. Then the rotation matrix $\mathbf{R} \triangleq \mathbf{R}_b^a$ will describe the orientation of the body. Suppose that the desired orientation of the body is given by a rotation matrix \mathbf{R}_d .

In the case of rotation matrices it does not make sense to subtract \mathbf{R}_d from \mathbf{R} as the result would not be a valid rotation matrix. Instead the deviation between the desired and the actual orientation is described by the rotation matrix $\tilde{\mathbf{R}}_b \in SO(3)$ defined by

$$\tilde{\mathbf{R}} \triangleq \mathbf{R}_d^{\mathrm{T}} \mathbf{R} \tag{2.7}$$

It can be shown (Egeland and Gravdahl, 2002), by using composite rotations, that the kinematic differential equations for the attitude deviation can be calculated as

$$\tilde{\boldsymbol{\omega}}_b = \boldsymbol{\omega}_b - \boldsymbol{\omega}_d^b$$
 (2.8a)

$$\frac{d}{dt}\tilde{\mathbf{R}} = \tilde{\mathbf{R}}\mathbf{S}(\tilde{\boldsymbol{\omega}}^b)$$
(2.8b)

Clearly, from (2.7) we se that when $\mathbf{R} \equiv \mathbf{R}_d \Rightarrow \tilde{\mathbf{R}} = \mathbf{I}$.

2.2 Euler angles

The Euler angles parametrization gives a rather physical interpretation of the orientation of one coordinate frame relative to another. When describing the motion of rigid bodies that move freely, like aeroplanes and satellites, the yaw-pitch-roll (ZYX) type is commonly used. The rotation matrix from a to b can be found by post-multiplying three composite rotation matrices, which are all obtained from simple rotations about the axes fixed in the b-system. The resulting rotation matrix is given as

$$\mathbf{R}_{b}^{a} = \mathbf{R}_{z}(\psi)\mathbf{R}_{y}(\theta)\mathbf{R}_{x}(\phi)$$
(2.9)

where

$$\mathbf{R}_{x}(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\phi) & -s(\phi) \\ 0 & s(\phi) & c(\phi) \end{bmatrix}, \ \mathbf{R}_{y}(\theta) = \begin{bmatrix} c(\theta) & 0 & s(\theta) \\ 0 & 1 & 0 \\ -s(\theta) & 0 & c(\theta) \end{bmatrix}, \ \mathbf{R}_{z}(\psi) = \begin{bmatrix} c(\psi) & -s(\psi) & 0 \\ s(\psi) & c(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and we have used that $c(\cdot)$ and $s(\cdot)$ denotes $\cos(\cdot)$ and $\sin(\cdot)$, respectively. The final rotation matrix from a to b can then be found from (2.9) as

$$\mathbf{R}_{b}^{a} = \begin{bmatrix} c(\psi)c(\theta) & c(\psi)s(\theta)s(\phi) - s(\psi)c(\phi) & c(\psi)c(\phi)s(\theta) + s(\psi)s(\phi) \\ s(\psi)c(\theta) & s(\psi)s(\theta)s(\phi) + c(\psi)c(\phi) & s(\psi)s(\theta)c(\phi) - c(\psi)s(\phi) \\ -s(\theta) & c(\theta)s(\phi) & c(\theta)c(\phi) \end{bmatrix}$$
(2.10)

Without any further explanation we conclude that we get a singularity in (2.10) for $\theta = \pm \frac{\pi}{2}$.

Note that many possible permutations of the Euler angles parametrization exist, depending on how the composite rotations are done. An overview can be found in Kane et al. (1983).

2.3 Euler parameters

The Euler parameters, also called *unit quaternions*, are attractive due to their nonsingular parametrization and linear kinematic differential equations if the angular velocities are known. The quaternion representation requires less computations than for instance the Euler angles representation, and is therefore useful in applications where computer resources are limited.

The Euler parameters are defined in terms of the angle-axis parameters θ and **k**, briefly discussed in regards to (2.2). The mapping is defined as

$$\eta = \cos\frac{\theta}{2}, \quad \epsilon = \mathbf{k}\sin\frac{\theta}{2}$$
 (2.11)

which gives the corresponding rotation matrix

$$\mathbf{R}(\eta, \boldsymbol{\epsilon}) = \mathbf{I} + 2\eta \mathbf{S}(\boldsymbol{\epsilon}) + 2\mathbf{S}^2(\boldsymbol{\epsilon})$$
(2.12)

By defining $\boldsymbol{\epsilon} \triangleq [\epsilon_1, \epsilon_2, \epsilon_3]^T$ we can write the rotation matrix from *a* to *b* by means of (2.12) as

$$\mathbf{R}_{b}^{a} = \begin{bmatrix} 1 - 2(\epsilon_{2}^{2} + \epsilon_{3}^{2}) & 2(\epsilon_{1}\epsilon_{2} - \eta\epsilon_{3}) & 2(\epsilon_{1}\epsilon_{3} + \eta\epsilon_{2}) \\ 2(\epsilon_{1}\epsilon_{2} + \eta\epsilon_{3}) & 1 - 2(\epsilon_{1}^{2} + \epsilon_{3}^{2}) & 2(\epsilon_{2}\epsilon_{3} - \eta\epsilon_{1}) \\ 2(\epsilon_{1}\epsilon_{3} - \eta\epsilon_{2}) & 2(\epsilon_{2}\epsilon_{3} + \eta\epsilon_{1}) & 1 - 2(\epsilon_{1}^{2} + \epsilon_{2}^{2}) \end{bmatrix}$$
(2.13)

It is found in Egeland and Gravdahl (2002) that the derivatives of the Euler parameters can be given as functions of the angular velocity, which gives the kinematic differential equations

$$\dot{\eta} = -\frac{1}{2} \boldsymbol{\epsilon}^{\mathrm{T}} \boldsymbol{\omega}_{ab}^{b}$$
(2.14a)

$$\dot{\boldsymbol{\epsilon}} = \frac{1}{2} \left[\eta \mathbf{I} + \mathbf{S}(\boldsymbol{\epsilon}) \right] \boldsymbol{\omega}_{ab}^{b}$$
 (2.14b)

By defining $\boldsymbol{\omega}_{ab}^{b} \triangleq [\omega_1, \omega_2, \omega_3]^{\mathrm{T}}$, (2.14) can be written in component form as

$$\dot{\eta} = -\frac{1}{2}(\varepsilon_1\omega_1 + \varepsilon_2\omega_2 + \varepsilon_3\omega_3)$$
 (2.15a)

$$\dot{\varepsilon}_1 = \frac{1}{2} (\eta \omega_1 - \varepsilon_3 \omega_2 + \varepsilon_2 \omega_3)$$
(2.15b)

$$\dot{\varepsilon}_2 = \frac{1}{2}(\varepsilon_3\omega_1 + \eta\omega_2 - \varepsilon_1\omega_3)$$
 (2.15c)

$$\dot{\varepsilon}_3 = \frac{1}{2}(-\varepsilon_2\omega_1 + \varepsilon_1\omega_2 + \eta\omega_3)$$
 (2.15d)

Finally, we note that $\eta^2 + \epsilon^T \epsilon = 1$. The latter is known to be a redundancy.

2.4 Rodrigues parameters

The classical and modified Rodrigues parameters can be interpreted as the coordinates resulting from a stereographic projection of the four-dimensional Euler parameter hypersphere onto a three-dimensional hyperplane (Schaub et al., 1995). The difference between them is how the projection point and mapping hyperplane is chosen.

2.4.1 The classical Rodrigues parameters

The classical Rodrigues parameters can be derived from the Euler parameters with the transformation

$$\mathbf{q} = \frac{\boldsymbol{\epsilon}}{\eta} \tag{2.16}$$

Combining (2.16) and (2.11) yields

$$\mathbf{q} = \mathbf{k} \tan \frac{\theta}{2} \tag{2.17}$$

Clearly, the classical Rodrigues parameters have a singular condition for $\theta = \pm \pi$, where $|\mathbf{q}| \rightarrow \infty$. The kinematic differential equation is derived from (2.14)

$$\dot{\mathbf{q}} = \frac{d}{dt}\frac{\boldsymbol{\epsilon}}{\eta} = \frac{\eta \dot{\boldsymbol{\epsilon}} - \dot{\eta} \boldsymbol{\epsilon}}{\eta^2} \tag{2.18}$$

which gives the quadratic nonlinear differential equation for the kinematics, that is

$$\dot{\mathbf{q}} = \frac{1}{2} \left[\mathbf{I} + \mathbf{S}(\mathbf{q}) + \mathbf{q}\mathbf{q}^{\mathrm{T}} \right] \boldsymbol{\omega}_{ab}^{b}$$
(2.19)

By defining $\boldsymbol{\omega}_{ab}^{b} \triangleq [\omega_1, \omega_2, \omega_3]^{\mathrm{T}}$, (2.19) can be written in component form as

$$\dot{q}_1 = \frac{1}{2} \left[(1+q_1^2)\omega_1 + (q_1q_2 - q_3)\omega_2 + (q_1q_3 + q_2)\omega_3 \right]$$
 (2.20a)

$$\dot{q}_2 = \frac{1}{2} \left[(q_1 q_2 + q_3)\omega_1 + (1 + q_2^2)\omega_2 + (q_2 q_3 - q_1)\omega_3 \right]$$
 (2.20b)

$$\dot{q}_3 = \frac{1}{2} \left[(q_3 q_1 - q_2)\omega_1 + (q_3 q_2 + q_1)\omega_2 + (1 + q_3^2)\omega_3 \right]$$
 (2.20c)

Unlike the Euler parameters, the Rodrigues parameters are numerically unique. They uniquely define a rotation on the open range of $(-\pi, \pi)$. As is evident in (2.16), reversing the sign of the Euler parameters has no effect on **q**.

2.4.2 The modified Rodrigues parameters

The modified Rodrigues parameters can be derived from the Euler parameters with the transformation

$$\boldsymbol{\sigma} = \frac{\boldsymbol{\epsilon}}{1+\eta} \tag{2.21}$$

Combining (2.16) and (2.21) yields

$$\boldsymbol{\sigma} = \mathbf{k} \tan \frac{\theta}{4}.\tag{2.22}$$

Clearly, the the modified Rodrigues parameters have a singular condition for $\theta = \pm 2\pi$, which allow twice the principal rotation angle compared to the classical Rodrigues parameters. From (2.16) and in (2.14) we get the differential kinematic equations

$$\dot{\boldsymbol{\sigma}} = \frac{1}{4} \left[(1 - \boldsymbol{\sigma}^{\mathrm{T}} \boldsymbol{\sigma}) \mathbf{I} + 2 \mathbf{S}(\boldsymbol{\sigma}) + 2 \boldsymbol{\sigma} \boldsymbol{\sigma}^{\mathrm{T}} \right] \boldsymbol{\omega}_{ab}^{b}$$
(2.23)

By defining $\boldsymbol{\omega}_{ab}^{b} \triangleq [\omega_1, \omega_2, \omega_3]^{\mathrm{T}}$, (2.23) can be written in component form as

$$\dot{\sigma}_1 = \frac{1}{4} (1 + \sigma_1^2 - \sigma_2^2 - \sigma_3^2) \omega_1 + \frac{1}{2} (\sigma_1 \sigma_2 - \sigma_3) \omega_2 + \frac{1}{2} (\sigma_1 \sigma_3 + \sigma_2) \omega_3 \quad (2.24a)$$

$$\dot{\sigma}_2 = \frac{1}{2}(\sigma_2\sigma_1 + \sigma_3)\omega_1 + \frac{1}{4}(1 - \sigma_1^2 + \sigma_2^2 - \sigma_3^2)\omega_2 + \frac{1}{2}(\sigma_2\sigma_3 - \sigma_1)\omega_3 \quad (2.24b)$$

$$\dot{\sigma}_3 = \frac{1}{2}(\sigma_3\sigma_1 - \sigma_2)\omega_1 + \frac{1}{2}(\sigma_3\sigma_2 + \sigma_1)\omega_2 + \frac{1}{4}(1 - \sigma_1^2 - \sigma_2^2 + \sigma_3^2)\omega_3 \quad (2.24c)$$

The equations display a similar degree of nonlinearity as do the corresponding equations in terms of the classical Rodrigues parameters. However, unlike the classical Rodrigues parameters, the modified Rodrigues parameters are not unique. This can be seen in (2.21).

2.5 The (w, z) parametrization

The three-dimensional (w, z) parametrization is a relatively new formulation for describing the relative orientation of two reference frames using two perpendicular rotations. Although it uses three parameters to describe the motion, two of the parameters can be combined to a single complex variable. The complex variable is used to designate the second of the two rotations and it is derived using stereographic projection (Conway, 1978). Since this parametrization is quite new and not very well known it is derived in detail in Appendix F. The main results are given in the following.

The differential kinematic equations for the (w, z) parametrization are

$$\dot{\mathbf{w}}_1 = \omega_3 \mathbf{w}_2 + \omega_2 \mathbf{w}_1 \mathbf{w}_2 + \frac{\omega_1}{2} (1 + \mathbf{w}_1^2 - \mathbf{w}_2^2)$$
 (2.25a)

$$\dot{\mathbf{w}}_2 = -\omega_3 \mathbf{w}_1 + \omega_1 \mathbf{w}_1 \mathbf{w}_2 + \frac{\omega_2}{2} (1 + \mathbf{w}_2^2 - \mathbf{w}_1^2)$$
 (2.25b)

$$\dot{z} = \omega_3 - \omega_1 \mathbf{w}_2 + \omega_2 \mathbf{w}_1 \tag{2.25c}$$

Alternatively they can be written more compactly as

$$\dot{\mathbf{w}} = -i\omega_3 \mathbf{w} + \frac{\omega}{2} + \frac{\bar{\omega}}{2} \mathbf{w}^2, \qquad (2.26a)$$

$$\dot{z} = \omega_3 + \frac{i}{2}(\bar{\omega}w - \omega\bar{w})$$
 (2.26b)

14 Attitude parametrization

Properties

The (w, z) parametrization has some unique properties that makes it useful in attitude control problems.

- The kinematic equations are compact and have a clear physical interpretation. It can be realized using two rotations about perpendicular axes.
- The z parameter does not appear in (2.25a) and (2.25b). This means that in some applications the control problem can be decomposed into one of controlling only w and one of controlling z.
- A three dimensional parametrization will always involve singularities. A singularity appears in the (w, z) parametrization when the body is upside down and consequently w → ∞. The equilibrium (w₁, w₂, z) = (0,0,0) is nevertheless as far away from the singularity as possible.
- The (w, z) parametrization can easily be connected to other well known parametrizations like Euler angles, Rodrigues parameters and angle-axis parametrization.

2.6 Discussion

The previous sections have shown that there are many attitude parametrizations to choose from. Some differences exist however, and which parametrization to use is clearly dependent on the problem to be solved.

The Euler angles may seem intuitive but they can introduce complicated nonlinear expression with inherent singularities. To avoid the singularities we would have to stay in the open range $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$. The Rodrigues parametrizations also suffer from singularities. It has been shown however (Schaub et al., 1996), in the case of modified Rodrigues parameters, that the singularities can be dealt with using a kind of switching technique. Nevertheless, in the general case it is a well known fact that a three dimensional parametrization always will involve singularities. This is also the case for the (w, z) parametrization. The equilibrium $(w_1, w_2, z) = (0, 0, 0)$ is nevertheless as far away from the singularity as possible, which occurs when the body is in upside down configuration. The parametrization using Euler parameters on the hand, is the only four dimensional parametrization discussed sofar, and consequently it does not suffer from any singularities. This makes it a popular choice in many applications and in the literature.

Despite the singularity problem, the (w, z) parametrization has some unique properties that would make it useful in attitude control problems. The reason why we choose the Euler parameters instead is partly due to singularity avoidance, but mostly because the latter is more known in literature, and also due to the fact that the Euler parameters have been used extensively in earlier phases of the SSETI project.

Chapter 3 Spacecraft and astrodynamics

The study of the dynamics of objects in interplanetary or interstellar space is called astrodynamics and has two major divisions. *Celestial mechanics* or orbit dynamics is concerned with the motion of the center of mass of objects in space, whereas *attitude dynamics* is concerned with the motion about the center of mass. Since the attitude dynamics is the main focus of this thesis, only a brief review will be given on the celestial mechanics. For the interested reader, Sellers (2000) is highly recommended as an introduction to astrodynamics. For more in-depth information see Hughes (1986), Wertz (1978, 1999) and Vallado (1997). The work in this chapter is based on these references.

3.1 Coordinate reference systems

As described in Chapter 2, we have already established the mathematics needed to describe the relative orientation between to coordinate reference frames. Obviously, to utilize these strong results it is necessary to have well defined reference frames. In the following we define suitable coordinate systems, which will be used throughout this thesis. We adopt the notation in Hughes (1986), by letting a reference frame be denoted by \mathcal{F}_a , where the index *a* denotes which system we consider. The three unit vectors forming the basis, are given the same index. Furthermore, all the systems to be defined are right-handed, and described by their origins, fundamental planes, and their preferred directions. Positive rotation in a system is when rotating counter clockwise about positive axis, when seen towards the origin.

Earth-centered inertial frame

This system originates at the center of the Earth, and is designated by the letters ECI. The fundamental plane is the Earth's equator. The \mathbf{x}_i axis points towards *vernal equinox*, Υ , the \mathbf{y}_i axis is 90° east in the equatorial plane, and the \mathbf{z}_i axis extends through the North Pole, as shown in Figure 3.1 This coordinate system is non-rotating, and assumed fixed in space, hence we call it an inertial reference frame. For the remainder we use \mathcal{F}_i to denote the ECI system.

Remark 3.1.1. The *geocentric equatorial coordinate system*, denoted IJK, is often used interchangeably with the ECI system. This is often confusing, and in fact, the IJK system is usually what people mean when they use the term ECI. The IJK is the *real* inertial system, while the ECI system is slightly moving by means of *precession* and *nutation*. However, we will not pursue this problem any further, and for our purposes we assume that the ECI system coincide with the IJK system, at all time.



Figure 3.1: The ECI frame, \mathcal{F}_i

Earth-centered Earth-fixed frame

As for the ECI system, the Earth-centered Earth-fixed system, denoted by ECEF, \mathcal{F}_e , originates at the center of the Earth, and the fundamental plane is the Earth's equator. The basis is spanned by the unit vectors \mathbf{x}_e , \mathbf{y}_e and \mathbf{z}_e , where \mathbf{x}_e is fixed and aligned with the particular meridian. A brief discussion on the rotation of the Earth, and consequently the rotation of \mathcal{F}_e relative to \mathcal{F}_i will be given shortly.

Satellite orbit-fixed frame

The orbit frame has its origin located in the satellite's center of mass. Its basis is defined by the unit vectors \mathbf{x}_o , \mathbf{y}_o and \mathbf{z}_o . The \mathbf{z}_o axis is *always* nadir pointing (center of the Earth), while the \mathbf{x}_o axis is perpendicular to \mathbf{z}_o and pointing in the direction of the velocity. Both vectors lie in the orbit plane. The \mathbf{y}_o axis completes the right-hand coordinate system. An illustration can be seen in Figure 3.2. For the remainder we let \mathcal{F}_o denote the orbit frame.

Remark 3.1.2. In the SSETI/ESEO phase B report they use the term *attitude reference frame* (ARF), apposed to our orbit-fixed frame.



Figure 3.2: Satellite orbit frame, \mathcal{F}_o

Satellite body-fixed frame

The body frame, \mathcal{F}_b , has its origin located in the satellite's center of mass, and is spanned by the unit vectors \mathbf{x}_b , \mathbf{y}_b , \mathbf{z}_b , respectively denoted as roll, pitch and yaw axis. The \mathbf{z}_b axis is such that it points in the direction of positive orbit maneuver thrust. The fundamental plane is then the plane perpendicular to \mathbf{z}_b . The \mathbf{y}_b axis lies in the fundamental plane, and points parallel to both solar panels. The \mathbf{x}_b axis completes the coordinate system. Unlike \mathcal{F}_o , \mathcal{F}_b is fixed to the satellite body, hence rotating with the satellite.

Remark 3.1.3. In the SSETI/ESEO phase B report they use the term *satellite reference frame* (SRF), apposed to our body-fixed frame.



Figure 3.3: Satellite body frame, \mathcal{F}_b

3.1.1 Transformations between reference frames

In the following we give rotation matrices between some of the different coordinate reference frames defined above. We note that if $\{\mathbf{R}_b^a: \mathcal{F}_b \to \mathcal{F}_a\} \Leftrightarrow \{(\mathbf{R}_a^b) = (\mathbf{R}_b^a)^{-1}: \mathcal{F}_a \to \mathcal{F}_b\}.$

Transformation from $\mathcal{F}_i \to \mathcal{F}_e$

The relative rotation and movement of the Earth is very complex, and a lot of the prediction is based on empiric data.

In the most general form, thus also the most complex, it is useful to consider the transformation of position of a stellar object in \mathcal{F}_i into \mathcal{F}_e as two rotations, that is

$$\mathbf{r}_e = \mathbf{T}_{3\times 3} \mathbf{U}_{3\times 3} \,\mathbf{r}_i \tag{3.1}$$

where U includes the rotations of the Earth caused by external torques, and \mathbf{T} the rotations to which the Earth would be subjected to if all external torques would be removed. This formulation is approximately realized in practice by the transformation consisting of the 9 rotations

$$\mathbf{r}_{e} = \left[\mathbf{R}_{y}(x_{p})\mathbf{R}_{x}(y_{p})\mathbf{R}_{z}(-GAST) \dots \\ \mathbf{R}_{x}(\varepsilon + \Delta\varepsilon)\mathbf{R}_{z}(\Delta\psi)\mathbf{R}_{x}(-\varepsilon)\mathbf{R}_{z}(z_{A})\mathbf{R}_{y}(-\theta_{A})\mathbf{R}_{z}(\zeta_{A}) \right] \mathbf{r}_{i}$$
(3.2)

Without any further explanations of the terms in (3.2), we easily see that the equation would involve rigorous calculations. For more details on (3.2), refer to Teunissen (1998).

When ignoring nutation, precession, polar motion and change in vernal equinox, the equatorial plane is identical for \mathcal{F}_i and \mathcal{F}_e . It is then assumed that \mathcal{F}_e rotates about \mathbf{z}_i with a constant angular velocity. In terms of the rotation matrices given in Section 2.2, we can write this as

$$\mathbf{R}_{i}^{e} = \mathbf{R}_{z}(-\alpha) = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0\\ -\sin(\alpha) & \cos(\alpha) & 0\\ & 0 & 1 \end{bmatrix}, \quad \text{where } \alpha = \omega_{ie}^{i} \cdot t$$
(3.3)

The mean angular velocity to the Earth is equal to $\omega_{ie}^i \triangleq \omega_{\oplus} = 7.2921158 \cdot 10^{-5} [rad/s]$, while t in (3.3) is the elapsed time since a defined $epoch^1$.

Transformation from \mathcal{F}_i or $\mathcal{F}_e \rightarrow$ Latitude and Longitude

Representing the position of the target in either \mathcal{F}_i or \mathcal{F}_e gives a distinct description, but it is often convenient to convert to latitude and longitude, and sometimes also the height of the target above the Earth reference ellipsoid. Given the cartesian coordinates there are many different algorithms in performing this task. Figure 3.4 shows the definitions and parameters needed for the transformation. μ_c and μ are the geocentric and geodetic latitudes respectively.



Figure 3.4: Definitions of the ellipsoidal parameters

Furthermore r is the geocentric radius, r_0 is the geocentric radius of the user position projected onto the surface of the Earth, h is the ellipsoidal height, and N is the radius of curvature in the prime vertical obtained from

$$N = \frac{r_e^2}{\sqrt{r_e^2 \cos^2(\mu) + r_p^2 \sin^2(\mu)}}$$

where the equatorial and polar radii, r_e and r_p , are the semiaxes of the Earth ellipsoid.

¹The epoch designates a particular instant, or moment of occurrence.

The longitude λ is easily computed as

$$\lambda = \arctan(y/x)$$

while latitude μ and height h above the Earth's surface are implicitly defined by

$$\tan(\mu) = \frac{z}{p} \left(1 - e^2 \frac{N}{N+h} \right)^{-1}, \quad h = \frac{p}{\cos(\mu)} - N$$

Finally we note that p and e are given as

$$p = \sqrt{x^2 + y^2}, \quad e = \sqrt{1 - (r_p/r_e)^2}$$

Without pursuing these equations any further, we see that they can be solved iteratively for λ , μ and h. The input parameters would be the radius vector of the target, i.e. $\mathbf{r_t} = [x, y, z]^{\mathrm{T}}$.

Transformation from $\mathcal{F}_o \to \mathcal{F}_i$

In Vallado (1997) the transformation from what he calls the RSW frame to the ECI frame is given in terms of classical orbit elements or *Keplerian orbit elements (COE)*. More information about COE can be found in Appendix E. However, the RSW frame does not coincide with \mathcal{F}_o as we define it, hence we have to include the rotation from RSW $\rightarrow \mathcal{F}_o$, i.e. we have to find the rotation matrix \mathbf{R}_o^{RSW} . The latter is easily found to be given as

$$\mathbf{R}_{o}^{RSW} = \mathbf{R}_{z}(\pi/2)\mathbf{R}_{x}(-\pi/2) \tag{3.4}$$

where $\mathbf{R}_{z}(\cdot)$ and $\mathbf{R}_{x}(\cdot)$ are similar to the Euler angle rotation matrices given in Section 2.2.

The transformation from $\mathcal{F}_o \to \mathcal{F}_i$ can then be written as

$$\mathbf{r}_i = \mathbf{R}_{RSW}^i \mathbf{R}_o^{RSW} \mathbf{r}_o = \mathbf{R}_o^i \mathbf{r}_o$$

where \mathbf{R}_{RSW}^{i} is given in Vallado (1997). The rotation matrix \mathbf{R}_{o}^{i} is given in terms of COE as

$$\mathbf{R}_{o}^{i} = \begin{bmatrix} -c(i)s(\Omega)c(u) - c(\Omega)s(u) & -s(i)s(\Omega) & -c(\Omega)c(u) + c(i)s(\Omega)s(u) \\ c(i)c(\Omega)c(u) - s(\Omega)s(u) & s(i)c(\Omega) & -s(\Omega)c(u) - c(i)c(\Omega)s(u) \\ s(i)c(u) & -c(i) & -s(i)s(u) \end{bmatrix}$$
(3.5)

where $u = \omega + \nu$ and we have used that $c(\cdot)$ and $s(\cdot)$ denote $\cos(\cdot)$ and $\sin(\cdot)$, respectively.

Transformation from $\mathcal{F}_b \to \mathcal{F}_o$

Based on the discussion in Chapter 2, we chose to represent this rotation by means of Euler parameters. As in (2.13), we can write the rotation matrix from $\mathcal{F}_b \to \mathcal{F}_o$ as

$$\mathbf{R}_{b}^{o} = \begin{bmatrix} 1 - 2(\epsilon_{2}^{2} + \epsilon_{3}^{2}) & 2(\epsilon_{1}\epsilon_{2} - \eta\epsilon_{3}) & 2(\epsilon_{1}\epsilon_{3} + \eta\epsilon_{2}) \\ 2(\epsilon_{1}\epsilon_{2} + \eta\epsilon_{3}) & 1 - 2(\epsilon_{1}^{2} + \epsilon_{3}^{2}) & 2(\epsilon_{2}\epsilon_{3} - \eta\epsilon_{1}) \\ 2(\epsilon_{1}\epsilon_{3} - \eta\epsilon_{2}) & 2(\epsilon_{2}\epsilon_{3} + \eta\epsilon_{1}) & 1 - 2(\epsilon_{1}^{2} + \epsilon_{2}^{2}) \end{bmatrix}$$
(3.6)

3.2 Attitude dynamics

The following section considers Newton-Euler equations of motion for rigid bodies, as well as giving an overview of disturbance and control torques. Some of the equations found in this chapter, together with the differential kinematic equations that were found earlier, make up the system to be investigated in later chapters. Unless otherwise is stated, the principal axes in the equations will coincide with the body reference system.

3.2.1 Newton-Euler equations of motion for rigid bodies

The angular motion of a spacecraft can be modelled as an ideal rigid body. However, most spacecrafts have flexible parts like for instance antennas and solar panels. Also, internal effects like fuel sloshing and thermal deformations are not accounted for using a rigid body model. Nevertheless, for many problems the rigid body model is a good approximation, especially for small spacecrafts.

Since only rotational motion will be considered throughout, the translational case will not be taken into account at this stage. However, a detailed derivation of the equations of motion for both cases is given in Appendix D. For the rotational case, when referred to the center of mass and the body reference system, the well known equations for a rigid body can be written as

$$\mathbf{I}\dot{\boldsymbol{\omega}}_{ib}^{b} + \mathbf{S}(\boldsymbol{\omega}_{ib}^{b})\mathbf{I}\boldsymbol{\omega}_{ib}^{b} = \boldsymbol{\tau} = \sum_{k} \boldsymbol{\tau}_{k}$$
(3.7)

where I is the inertia matrix for the rigid body, referred to the center of mass, $\tau \triangleq [\tau_x, \tau_y, \tau_z]^T$ is the total torque acting on the body, and $\omega_{ib}^b \triangleq [\omega_1, \omega_2, \omega_3]^T$ is the angular velocity as explained in Chapter 2. The torques τ_k , acting on the individual mass elements in the body, are due to both forces between individual mass elements and externally applied forces. Usually the internal torques sum to zero and the resultant torque is simply the torques due to external forces. The external torques τ_e can be divided into two groups, called *disturbance torques* and *control torques*. The first case is caused by environmental effects such as aerodynamic drag and gravity gradient torque, while the latter is deliberately applied torques from devices such as thrusters, wheels, or magnetic coils. Both cases will be discussed in the following.

Assuming a diagonal inertia matrix, $I = \text{diag}\{i_{11}, i_{22}, i_{33}\}$, the dynamics in (3.7) can easily be found to be given in component form as

$$i_{11}\dot{\omega}_1 + (i_{33} - i_{22})\omega_2\omega_3 = \tau_x \tag{3.8a}$$

$$i_{22}\dot{\omega}_2 + (i_{11} - i_{33})\omega_3\omega_1 = \tau_y \tag{3.8b}$$

$$i_{33}\dot{\omega}_3 + (i_{22} - i_{11})\omega_1\omega_2 = \tau_z \tag{3.8c}$$

Remark 3.2.1. By defining the angular momentum $\mathbf{h} \triangleq \mathbf{I}\boldsymbol{\omega}_{ib}^{b}$, and assuming only external torques, (3.7) can be rewritten as

$$\frac{d\mathbf{h}}{dt} = \boldsymbol{\tau}_e - \mathbf{S}(\boldsymbol{\omega}_{ib}^b)\mathbf{h}$$
(3.9)

From this equation it can easily be seen that the angular momentum **h**, and hence ω_{ib}^b , is not constant in the body frame, even when the external torque τ_e is equal to zero. The resulting motion is called *nutation*. Rotational motion without nutation only occurs when $\omega_{ib}^b \parallel \mathbf{h}$, that is, only if the rotation is about a principle axis of the rigid body.

Remark 3.2.2. A spacecraft equipped with reaction or momentum wheels is not a rigid body in the sense that they cause a redistribution of the angular momentum between the wheels and the spacecraft body. The wheels do not change the total angular momentum of the spacecraft, hence they can not be external torques.

However, in the case of using reduction or momentum wheels in the spacecraft body, the equations above can still be used with one minor modification. To encounter for the angular momentum of the wheels, we redefine the total angular momentum for the spacecraft, that is

$$\mathbf{h}_b = \mathbf{I}\boldsymbol{\omega}_{ib}^b + \mathbf{h}_w \tag{3.10}$$

where the inertia matrix I includes the mass of the wheels and the vector $\mathbf{h}_w \triangleq [h_1, h_2, h_3]^{\mathrm{T}}$ is the net angular momentum due to the rotation of the wheels relative to the body. Using a similar procedure as when deriving (3.7) we get the following equation of motion

$$\mathbf{I}\dot{\boldsymbol{\omega}}_{ib}^{b} + \mathbf{S}(\boldsymbol{\omega}_{ib}^{b})(\mathbf{I}\boldsymbol{\omega}_{ib}^{b} + \mathbf{h}_{w}) = \boldsymbol{\tau} - \frac{d\mathbf{h}_{w}}{dt}$$
(3.11)

The quantity $d\mathbf{h}_w/dt$ is the net torque applied to the wheels from the spacecraft body, so by Newton's 3rd law of motion, $-d\mathbf{h}_w/dt \triangleq [\tau_{wx}, \tau_{wy}, \tau_{wz}]^T$ is the torque applied to the spacecraft body by the wheels. Writing (3.11) in component form in the body system, referred to the center of mass, yields

$$i_{11}\dot{\omega}_1 + (i_{33} - i_{22})\omega_2\omega_3 + h_3\omega_2 - h_2\omega_3 = \tau_x + \tau_{wx}$$
(3.12a)

$$i_{22}\dot{\omega}_2 + (i_{11} - i_{33})\omega_3\omega_1 + h_1\omega_3 - h_3\omega_1 = \tau_y + \tau_{wy}$$
(3.12b)

$$i_{33}\dot{\omega}_3 + (i_{22} - i_{11})\omega_1\omega_2 + h_2\omega_1 - h_1\omega_2 = \tau_z + \tau_{wz}$$
(3.12c)

Remark 3.2.3. A rigid body with one or more spinning wheels is commonly called a gyrostat.

An alternative representation of the multi-spin system described in (3.11) is derived in Hughes (1986), in the case of using only one wheel. In Hall (1995) the representation is expanded to include any number of wheels, which makes it quite practical to use. The rotational equations of motion for a N-wheel gyrostat can be written as

$$\dot{\mathbf{h}}_b = \boldsymbol{\tau} - \left[\mathbf{J}^{-1} (\mathbf{h}_b - \mathbf{A} \mathbf{h}_a) \right] \times \mathbf{h}_b$$
 (3.13a)

$$\dot{\mathbf{h}}_a = \boldsymbol{\tau}_a$$
 (3.13b)

where \mathbf{h}_a is the $N \times 1$ vector of the axial angular momenta of the wheels, $\boldsymbol{\tau}$ is the 3×1 vector of the total torque acting on the body, not including wheel torques, $\boldsymbol{\tau}_a$ is the $N \times 1$ vector of the internal axial torques applied by the platform to the wheels, and \mathbf{A} is the $3 \times N$ matrix whose columns contain the axial unit vectors of the N momentum exchange wheels. The vector \mathbf{h}_b is the total angular momentum for the spacecraft in the body frame, given by

$$\mathbf{h}_b = \mathbf{J}\boldsymbol{\omega}_{ib}^b + \mathbf{A}\mathbf{h}_a \tag{3.14}$$

J is the inertialike matrix defined as

$$\mathbf{J} \triangleq \mathbf{I} - \mathbf{A}\mathbf{I}_s \mathbf{A}^{\mathrm{T}}$$
(3.15)

where **I** is the moment of inertia matrix for the spacecraft, including wheels, and the matrix $\mathbf{I}_s = \text{diag}\{\mathbf{I}_{s1}, \mathbf{I}_{s2}, ..., \mathbf{I}_{sN}\}$ contains the axial moments of inertia of the wheels on the diagonal. The axial angular momenta of the wheels can be written in terms of the body angular velocity and the wheels' axial angular velocities relative to the body, $\boldsymbol{\omega}_s$, as

$$\mathbf{h}_a = \mathbf{I}_s \mathbf{A}^{\mathrm{T}} \boldsymbol{\omega}_{ib}^b + \mathbf{I}_s \boldsymbol{\omega}_s \tag{3.16}$$

Note that $\boldsymbol{\omega}_s = [\boldsymbol{\omega}_{s1}, \boldsymbol{\omega}_{s2}, ..., \boldsymbol{\omega}_{sN}]^{\mathrm{T}}$ is a $N \times 1$ vector, and that these relative angular velocities are those that would for instance be measured by tachometers fixed to the platform. We denote the resultant axial angular velocity of the wheels, relative to the inertial frame as $\boldsymbol{\omega}_c = [\boldsymbol{\omega}_{c1}, \boldsymbol{\omega}_{c2}, ..., \boldsymbol{\omega}_{cN}]^{\mathrm{T}}$. Using this notation we can write (3.16) as

$$\mathbf{h}_a = \mathbf{I}_s \boldsymbol{\omega}_c \tag{3.17}$$

where $\omega_c = \omega_s + \mathbf{A}^{\mathrm{T}} \omega_{ib}^b$. Note that because typically the wheels spin at a much higher speed than the spacecraft itself, $\omega_s \gg \omega_{ib}^b$ and we have that $\omega_c \approx \omega_s$.

It is also possible to write (3.13) in terms of angular velocities. By defining $\boldsymbol{\mu} \triangleq [\mathbf{h}_b, \mathbf{h}_a]^{\mathrm{T}}$ and $\boldsymbol{\upsilon} \triangleq [\boldsymbol{\omega}_{ib}^b, \boldsymbol{\omega}_s]^{\mathrm{T}}$, we can write (3.14) and (3.16) in the compact form

$$\mu = \Lambda v$$
, where $\Lambda = \begin{bmatrix} \mathbf{I} & \mathbf{A}\mathbf{I}_s \\ \mathbf{I}_s \mathbf{A}^{\mathrm{T}} & \mathbf{I}_s \end{bmatrix}$ (3.18)

Clearly, we can find ω_{ib}^b and ω_s from $v = \Lambda^{-1} \mu$, or equally, we can write $\dot{v} = \Lambda^{-1} \dot{\mu}$.

Lemma 3.1 (Matrix inversion lemma). Suppose A and D are square, D invertible, B, C compatible dimensions. If $(A - BD^{-1}C)$ is invertible then

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}^{-1} = \begin{bmatrix} (\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} & -(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1}\mathbf{B}\mathbf{D}^{-1} \\ -\mathbf{D}^{-1}\mathbf{C}(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} & \mathbf{D}^{-1}\mathbf{C}(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1}\mathbf{B}\mathbf{D}^{-1} + \mathbf{D}^{-1} \end{bmatrix}$$

By utilizing Lemma 3.1 together with (3.18), we get that

$$\begin{bmatrix} \dot{\boldsymbol{\omega}}_{ib}^{b} \\ \dot{\boldsymbol{\omega}}_{s} \end{bmatrix} = \begin{bmatrix} \mathbf{J}^{-1} & -\mathbf{J}^{-1}\mathbf{A} \\ -\mathbf{A}^{\mathrm{T}}\mathbf{J}^{-1} & \mathbf{A}^{\mathrm{T}}\mathbf{J}^{-1}\mathbf{A} + \mathbf{I}_{s}^{-1} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{h}}_{b} \\ \dot{\mathbf{h}}_{a} \end{bmatrix}$$
(3.19)

3.2.2 Disturbance torques

As mentioned above, the environmental disturbances contribute as torques on the spacecraft body, making them noneligible when doing attitude prediction in real life. If they were to be added to the equations of motion they would have to be modelled as functions of time as well as of the orientation of the spacecraft. Worth noticing is that external torques will change the angular momentum of the spacecraft, while the internal torques only will affect the distribution of momentum between the moving parts. Some disturbance torques are given in Table 3.1, indicating at what height above the surface they are most likely to dominate. In the following an overview of the most important disturbance torques is given. This could contribute as a good starting point, if advanced models or robustness issues are to be considered.

External torque source	Region of space where dominant*	
Aerodynamic Gravity gradient	< 500km [†] 500km to 35000km	
Magnetic Solar pressure	500km to 35000km > 700 km [†]	
Thrust misalignment	all heights	
Internal torque source		
Mechanical and electrical devices Fuel sloshing		
General mass movement Flexible appendages		

Table 5.1. Environmental distarbance torques	Table 3.1:	Environmental	disturbance	torques
--	------------	---------------	-------------	---------

*The specific altitude at which the various torques dominate are highly spacecraft dependent [†]Value depends upon the level of solar activity

Aerodynamic torque

The interaction of the upper atmosphere with a spacecraft's surface produces a torque about the center of mass. The effect is clearly dependent on the area and shape of the exposed surface. In general the impact of the atmospheric molecules can be modelled as an elastic impact without reflection. For low orbit spacecrafts the air density is high enough to influence the attitude dynamics of the body. Calculating the aerodynamic torques and forces can be done in several ways, and more or less all approaches lead to rather complicated expressions. Use of empiric data is also common.

If the spacecraft surface comprises a collection of small incremental areas dA, each with outwards unit normal $\hat{\mathbf{n}}$, then the force on a surface element is given by

$$d\mathbf{f}_{aero} = -\frac{1}{2}\rho v^2 C_D(\hat{\mathbf{n}} \cdot \hat{\mathbf{v}})\hat{\mathbf{v}}dA$$
(3.20)

where \mathbf{v} is the translational velocity of the surface element relative to the incident stream, and $\hat{\mathbf{v}}$ is the unit vector in the same direction. The coefficients ρ and C_D are the atmospheric density and the drag coefficient, respectively. Performing a summation over all such areas gives the simplified expression for the aerodynamic torque, that is

$$\tau_{aero} = \int \mathbf{r}_s \times d\mathbf{f}_{aero} \tag{3.21}$$

where \mathbf{r}_s is the position vector from the center of mass of the body to the surface element dA. Usually this integral is not amenable to simple solutions for a surface associated with a complex structure. A commonly used alternative is to represent the spacecraft as a collection

of simple geometrical elements. The torque about the center of mass of the spacecraft is then the vector sum of the individual torques for each of these geometrical simplifications, that is

$$\bar{\tau}_{aero} = \sum_{k} \mathbf{r}_{k} \times \mathbf{F}_{aero,k} \tag{3.22}$$

The vector \mathbf{r}_k is in this case the vector distance from the center of mass of the spacecraft to the center of pressure of the the specific geometric shape and $\mathbf{F}_{aero,k}$ is the force acting on the component. Aerodynamic forces for some simple geometric figures are listed in Table 3.2.

Geometric figures	$\mathbf{F}_{aero,k}$
Sphere of radius R	$-\frac{1}{2}\rho v^2 C_D \pi R^2 \hat{\mathbf{v}}$
Plane with surface area A and normal unit vector $\hat{\mathbf{n}}$	$-\frac{1}{2}\rho v^2 C_D A(\hat{\mathbf{n}}\cdot\hat{\mathbf{v}})\hat{\mathbf{v}}$

Table 3.2: Aerodynamic force for some simple geometric figures

Gravity gradient torque

As a result of the nonuniform gravitational field surrounding the Earth, any nonsymmetrical object in orbit is subject to a gravitational torque. It is important to emphasize that this can only occur as long as there are variations in the specific gravitational force over the spacecraft.

The unit vector \mathbf{z}_o that appears in (3.23) is called the *local vertical* and by definition it is always nadir pointing. We also defined it when describing \mathcal{F}_o .

$$\boldsymbol{\tau}_{g} = \frac{3\mu}{R_{c}^{3}} \left[\mathbf{z}_{o} \times (\mathbf{I} \, \mathbf{z}_{o}) \right]$$
(3.23)

The parameters in (3.23) are summarized in the following table

Symbol	Explanation
μ	Gravitational coefficient, $\mu = 3.986 \cdot 10^{14} \mathrm{Nm^2/kg}$
R_c	Distance to center of the Earth (m)
Ι	Spacecraft inertia matrix
\mathbf{Z}_O	Unit vector toward nadir

Remark 3.2.4. The expression in (3.23) is rather simplified due to the four assumptions

- a) Only one celestial primary is considered. In most cases the primary will be the Earth.
- b) The primary possesses a spherically symmetrical mass distribution.
- c) The spacecraft is small compared to its distance from the mass center of the primary.
- d) The spacecraft consists of a single body.

Note that in most spacecraft situations these are realistic assumptions.

Remark 3.2.5. There is no gravitational torque about the local vertical, that is

$$\boldsymbol{\tau}_g \cdot \mathbf{z}_o = 0$$

As can be seen from (3.23), the vector equation is given in the local orbiting frame \mathcal{F}_o . However, in most cases it would be useful to represent the torque in the body fixed reference frame. By letting the rotation matrix $\mathbf{R}_o^b = (\mathbf{R}_b^o)^{-1}$ represent the rotation matrix from the body frame to the orbit frame, we get the following expression for the gravitational torque, as referred to the body frame

$$\boldsymbol{\tau}_{g}^{b} = \frac{3\mu}{R_{c}^{3}} \left[\mathbf{c}_{3} \times (\mathbf{I}\mathbf{c}_{3}) \right]$$
(3.24)

where $\mathbf{c}_3 \triangleq [c_{13}, c_{23}, c_{33}]^{\mathrm{T}}$ is the third column in the rotation matrix \mathbf{R}_o^b . Independent of the attitude parametrization we use to represent this rotation matrix, and assuming a diagonal inertia matrix $\mathbf{I} = \text{diag}\{i_{11}, i_{22}, i_{33}\}$, the gravitational torque in (3.24) simplifies to

$$\boldsymbol{\tau}_{g}^{b} = \frac{3\mu}{R_{c}^{3}} \begin{bmatrix} (i_{33} - i_{22})c_{23}c_{33}\\ (i_{11} - i_{33})c_{33}c_{13}\\ (i_{22} - i_{11})c_{13}c_{23} \end{bmatrix}$$
(3.25)

Solar pressure torque

Solar radiation pressure produces a force on a surface, which depends upon its distance to the sun. Since light carries momentum, it represents an exchange of momentum with the surface when it is reflected. For most applications, the forces may be modelled adequately by assuming that the incident radiation is either absorbed, reflected specularly, reflected diffusely, or in some combination of these.

If the spacecraft surface comprises a collection of small incremental areas dA, each with outwards unit normal $\hat{\mathbf{n}}$, and $\hat{\mathbf{s}}$ is the unit vector from the spacecraft to the sun, then the force on a surface element due to solar radiation is given by

$$d\mathbf{f}_{solar} = -P\cos\theta \left[(1 - f_s)\hat{\mathbf{s}} + 2(f_s\cos\theta + \frac{1}{3}f_d)\hat{\mathbf{n}} \right] dA$$
(3.26)

where P is the mean momentum flux ($\sim 4.67 \cdot 10^{-6} Nm^2$ at the Earth), and f_s and f_d are the coefficients of specular and diffuse reflection, respectively. The angle of incidence radiation is given as $\theta = \cos^{-1}(\hat{\mathbf{s}} \cdot \hat{\mathbf{n}})$. Performing a summation over all such areas gives the expression for the solar pressure torque, that is

$$\boldsymbol{\tau}_{solar} = \int \mathbf{r}_s \times d\mathbf{f}_{solar} \tag{3.27}$$

where \mathbf{r}_s is the position vector from the center of mass of the body to the surface element dA. This integration is in general difficult to solve for a surface associated with a complex structure. However, as for the aerodynamic torque, a commonly used alternative is to represent the spacecraft as a collection of simple geometrical elements. The total torque about the center of mass of the spacecraft is then the vector sum of the individual torques for each of these geometrical simplifications.

$$\bar{\tau}_{solar} = \sum_{k} \mathbf{r}_{k} \times \mathbf{F}_{solar,k}$$
(3.28)

The vector \mathbf{r}_k is in this case the vector distance from the center of mass of the spacecraft to the center of pressure of the the specific geometric shape and $\mathbf{F}_{solar,k}$ is the force acting on the component. Solar radiation forces for some simple geometric figures are listed in Table 3.3.

Geometric figures	$\mathbf{F}_{solar,k}$
Sphere of radius R	$-P(4\pi R^2)\left(rac{1}{4}+rac{f_d}{9} ight)\hat{f s}$
Plane with surface area A and normal unit vector $\hat{\mathbf{n}}$: $\theta = \cos^{-1}(\hat{\mathbf{s}} \cdot \hat{\mathbf{n}})$	$-PA\cos\theta\left[(1-f_s)\hat{\mathbf{s}}+2\left(f_s\cos\theta+\frac{f_d}{3}\right)\hat{\mathbf{n}}\right]$

Table 3.3: Solar radiation force for some simple geometric figures

Magnetic torque

The residual magnetic field generated by a spacecraft interacts with the local field from the Earth and thereby exerts a couple on the body. The effect of the magnetic torque is altitude dependent and strongest at low altitudes. The instantaneous magnetic disturbance torque τ_{mag} due to the spacecraft effective magnetic moment **m** is given by

$$\boldsymbol{\tau}_{mag} = \mathbf{m} \times \mathbf{B} \tag{3.29}$$

where \mathbf{B} is the geocentric magnetic flux density and \mathbf{m} is the sum of the individual magnetic moments caused by permanent and induced magnetism and spacecraft generated current loops.

Internal torques

Internal torques are defined as torques exerted on the spacecraft body by such internal moving parts as reaction and momentum wheels, fuels or liquids inside partially filled containers or flexible constructions. The reaction and momentum wheels are usually not to be considered as disturbances, but are included in this section for consistency. In a highly hypothetical scenario, the internal torques would also exist if the spacecraft was to be removed entirely from all external influences in space.

As mentioned earlier the external torques will change the angular momentum of the spacecraft, while the internal torques only will affect the distribution of momentum between the moving parts. However, even though the angular momentum remains constant in the absence of external torques, the kinetic energy for the body could change and in most cases would the redistribution of the angular momentum between the moving parts lead to a change in the dynamic characteristics.

In general the internal disturbance torques are undesired, hence must be encountered for using external torques. Also, when designing a spacecraft the effects of internal disturbances could be tried encountered for in the sense of placing the different devices in a clever manner.

3.2.3 Control torques and actuators

Deliberately applied torques can be generated using various approaches, and commonly as a combination of several different actuators. The types of actuators that can be used to control the orientation of a spacecraft can usually be divided into three categories. These are thrusters, momentum exchange devices and magnetic actuators. The actuator can also be categorized to
Туре	Advantages	Disadvantages
External torque	Control momentum build-up	
Thrusters	Insensitive to altitude Suit any orbit Create torque about any axis	Requires fuel On-Off operation only Minimum impulse Exhaust plume contaminants
Gravity gradient	No fuel or energy needed	No torque about the local vertical Low accuracy Low torque, altitude sensitive Libration mode needs damping
Magnetic	No fuel required Control torque magnitude	No torque about local field direction Altitude and latitude sensitive Can cause magnetic interference
Solar radiation	No fuel required	Needs controllable panels Very low torque
Internal torque	No fuel required Can store momentum Control torque magnitude	Uncontrollable momentum build-up
Reaction wheels	Continuous Fine-pointing capability	Nonlinearity at zero speed
Momentum wheels Control moment gyroscope	Provide momentum bias Suitable for three-axis control Provide momentum bias	Complicated Potential reliability problems

Table 3.4: Control torques

be either active or passive. Some control torques, and their properties, are given in Table 3.4. A short explanation of some of the different designs is also given subsequently.

An important part of the design of a spacecraft is to decide the size and properties of the actuators since it is crucial to have enough control to overcome the disturbances discussed above, as well as getting the spacecraft into its desired configuration. However, as the topic is not discussed any further in this thesis, the reader should consult to Wertz (1999) for details.

In later chapters the control actuation for the system is achieved using thrusters and a reaction wheel. More details about this matter will be given shortly.

Gravity gradient

As mentioned earlier any nonsymmetrical object in orbit is subject to a gravitational torque. Although this effect often is considered as a disturbance, it can also be utilized as a passive control torque. This is commonly done using a gravitational boom. However, in the sense of stabilizing a spacecraft, the body will only be in a stable equilibrium if its axis of minimum inertia is aligned with the local vertical. Due to low accuracy and the need for damping makes the use of other control torques necessary as well. In the Danish satellite Ørsted (Wiśniewski and Blanke, 1999) the gravity gradient was utilized together with magnetic actuators to archive complete three-axis stabilization.

Solar radiation

In the previous section on disturbance torques, it was explained how the solar radiation causes a passive torque on an exposed spacecraft. This can be utilized with controllable panels or solar sails. The torques achieved are nevertheless low.

Thrusters

Thrusters or reaction jets produce torque by expelling mass, and are potentially the largest source of force and torque on a spacecraft. They are highly active sources, and being external they will affect the total momentum. They can be used both for attitude and position control. In fact, they are the only actuators that can increase the altitude of a spacecraft in orbit. When used for attitude control a pair of thrusters on opposite sides of the spacecraft is activated to create a couple. The main advantage of using thrusters is that they can produce an accurate and well defined torque on demand, as well as being independent of altitude. The main disadvantage is that a spacecraft can only carry a limited amount of propellant.

Reaction wheels

Torquers associated with momentum storage such as reaction wheels are essentially active internal torquers, suitable for attitude control but not for controlling the angular momentum. By definition, a reaction wheel is a *flying wheel* with a body fixed axis designed to operate at zero bias. A flying wheel is any rotation wheel or disk used to store or transfer momentum. When the spacecraft is exposed to a perturbation or it is accelerated, so are the wheels mounted inside, and the result is generated torques from the wheels in the opposite direction, that is

$$\mathbf{I}_{rw}\dot{\boldsymbol{\omega}}_{rw} = -\mathbf{I}\dot{\boldsymbol{\omega}}_{ib}^b \tag{3.30}$$

As seen from (3.30) the wheels have to be accelerated in order to create a torque. Neglecting friction effects, the torque applied to a set of reaction wheels can be written as (Kaplan, 1976)

$$\tau_{bw} = \frac{d\mathbf{h}}{dt} + \boldsymbol{\omega}_{ib}^b \times \mathbf{h}$$
(3.31)

where $\mathbf{h} = \mathbf{I}_{rw} \boldsymbol{\omega}_{rw} \triangleq [h_1, h_2, h_3]^{\mathrm{T}}$ is the total angular momentum of the wheels, and τ_{bw} denote the torque applied to the wheels by the spacecraft body. By Newton's 3rd law, the torque employed to the spacecraft body from the wheels is therefore given by $\tau_{wb} = -\tau_{bw}$. By defining $-d\mathbf{h}/dt \triangleq [\tau_{wx}, \tau_{wy}, \tau_{wz}]^{\mathrm{T}}$ we get the following equation

$$\boldsymbol{\tau}_{wb} = \begin{bmatrix} \tau_{wx} - h_3\omega_2 + h_2\omega_3 \\ \tau_{wy} - h_1\omega_3 + h_3\omega_1 \\ \tau_{wz} - h_2\omega_1 + h_1\omega_2 \end{bmatrix}$$
(3.32)

This is consistent with the results derived earlier, as (3.12) can be obtained by substituting the torque in (3.32), in addition to some external torque, for the torque in (3.7).

Remark 3.2.6. When the wheels reach their maximum speed, the storage of momentum will be at its maximum as well. Therefore, it will be necessary to restore the nominal values, using external torques. This process is known as *momentum dumping*.

Momentum wheels

Momentum wheels are very similar to reaction wheels, but in contrast to the reaction wheels they are designed to operate at *biased*, or nonzero, momentum. As for the reaction wheels they need to be used in conjunction with other external actuators.

Magnetic actuators

An active magnetic actuator takes advantage of the natural torque caused by the magnetic field surrounding the Earth. The magnetic disturbance that was described earlier is exploited by installing magnetic coils or torquers inside the spacecraft. The principle can best be explained with the well known compass needle that attempts to align itself with the local field.

Magnetic actuators offer a cheap, reliable and robust way to control a spacecraft's attitude. Unfortunately they are only effective for low Earth orbit (LEO) spacecrafts and requires a complex model of the geomagnetic field surrounding the Earth.

Electromagnets may be used to provide an external torque, which can be modelled as

$$\boldsymbol{\tau}_{mag} = \mathbf{m} \times \mathbf{B} = niA \left(\hat{\mathbf{c}} \times \mathbf{B} \right) \tag{3.33}$$

where \mathbf{m} is the magnetic dipole moment generated by the coils in the magnet. The other parameters are listed in the following table.

Symbol	Explanation
В	Local geomagnetic field vector
ĉ	Unit vector in the direction of the coil's axis
i	Control current in the coil
n	Number of coil windings
A	Cross-sectional area of the coil

3.3 Celestial mechanics

As mentioned at the very beginning of this chapter, the study of astrodynamics can be divided into celestial mechanics and attitude dynamics. Sofar we have only considered the latter, in the sense of giving an overview of different topics considering the motion about the center of mass. Only a brief discussion on celestial mechanics will be given at this point. Even though the effects of the celestial mechanics are almost assumed negligible throughout this thesis, the motion of the center of mass of objects in space is highly relevant, and for a real-life system to be revealing, these effects should be taken into account when designing the control systems.

In general, the theory of celestial mechanics underlies all the dynamical aspects of the orbital motion of a spacecraft. Different approaches exist to provide the necessary equations needed

to calculate orbital elements from position and velocity, or vica verca, and to predict the future position and velocity of the spacecraft. In the case of circular orbits and spherical Earth it is relatively easy to determine these relations by the use of gradients, combined with rotation matrices. In a more general case the use of classical orbit elements, *Keplerian orbit elements* (COE), is adequate. Some information about COE can be found in Appendix E. Combined with perturbation theory this provides an excellent reference. Since the equations concerning perturbations are rather extensive, they are not considered here. For further details on this topic, the reader should refer to any textbook about spacecraft geodesy, e.g. Vallado (1997).

3.3.1 A simple orbit propagator

An orbit propagator is a mathematical algorithm for predicting the future position and velocity (or orbital elements) of an orbit, given some assumptions and initial conditions. There are many techniques and methods available, with widely different accuracy and applications.

In the following we describe a simple propagator, which is valid under the assumptions that there are no acting perturbations. Furthermore, the orbit is either circular or elliptic.

One way to describe the motion of a satellite is to use (E.2). An alternative approach is to describe the motion by means of Kepler's equation, that is

$$M = E - e \sin E = n (t - T), \text{ where } n = \sqrt{\frac{\mu}{a^3}}$$
 (3.34)

The mean anomaly M corresponds to the uniform angular motion on a circle, E is the eccentric anomaly, n is the mean motion, e is the eccentricity of the orbit, T is the time of periapsis passage (closest approach to the central body) and t is the time of flight.

There are several methods for finding the solution of E in (3.34), such as Newton-Raphson, which approximates the solution by using a root-finding iterative process until a desired convergence tolerance is reached. Based on this technique, it is shown in Vallado (1997) that the iteration can be written as

$$E_{k+1} = E_k + \frac{M - E_k + e \sin E_k}{1 - e \cos E_k}$$
(3.35)

The iterative approximation can then be utilized to determine the *true anomaly* ν and the distance to the satellite r. These relations are given as

$$\cos\nu = \frac{\cos E - e}{1 - e \cos E} \tag{3.36}$$

$$r = a(1 - e\cos E) \tag{3.37}$$

Based on the above mentioned equations, we can write an algorithm to solve for future COEs, given some initial conditions. From the calculated COEs we can then find the cartesian coordinates in \mathcal{F}_i by utilizing (E.7) and (E.9). The MATLAB code and Simulink blocks are given in Appendix C.

Chapter 4 Model and control properties

Sofar the main focus has been on different approaches to model the kinematic and dynamic differential equations for describing the configuration of a rigid body. The purpose of this chapter is to choose an adequate model based on these equations, and describe some of its most important properties. In our special case, the rigid body model is used to describe the ESEO satellite, as introduced in Chapter 1. The actuation will be by means of one reaction wheel about the satellite's principal y-axis, as well as thruster torque couples about all its axes. All physical parameters are taken from the SSETI/ESEO Phase B report.

4.1 Satellite model

To simplify the analysis it is important to choose a reasonable model that is not too complicated. Therefore, because of its useful properties, the Euler parameters presented in Chapter 2 are chosen to characterize the kinematics, while the dynamic equations are based on the Newton-Euler representation in Chapter 3. In the following the model to be used is derived.

A preliminary and rather complex model can be written as¹

$$\dot{\boldsymbol{\omega}}_{ib}^{b} = \mathbf{J}^{-1} \left[-\mathbf{S}(\boldsymbol{\omega}_{ib}^{b}) (\mathbf{I}\boldsymbol{\omega}_{ib}^{b} + \mathbf{A}\mathbf{I}_{s}\boldsymbol{\omega}_{s}) + \boldsymbol{\tau}_{e} \right] - \mathbf{J}^{-1}\mathbf{A}\boldsymbol{\tau}_{a}$$
(4.1a)

$$\dot{\boldsymbol{\omega}}_{s} = -\mathbf{A}^{\mathrm{T}}\mathbf{J}^{-1} \left[-\mathbf{S}(\boldsymbol{\omega}_{ib}^{b})(\mathbf{I}\boldsymbol{\omega}_{ib}^{b} + \mathbf{A}\mathbf{I}_{s}\boldsymbol{\omega}_{s}) + \boldsymbol{\tau}_{e} \right] + \left[\mathbf{A}^{\mathrm{T}}\mathbf{J}^{-1}\mathbf{A} + \mathbf{I}_{s}^{-1} \right] \boldsymbol{\tau}_{a} \quad (4.1b)$$

$$\dot{\eta} = -\frac{1}{2} \epsilon^{\mathrm{T}} \omega_{ob}^{b} \tag{4.1c}$$

$$\dot{\boldsymbol{\epsilon}} = \frac{1}{2} \left[\eta \mathbf{I} + \mathbf{S}(\boldsymbol{\epsilon}) \right] \boldsymbol{\omega}_{ob}^{b}$$
(4.1d)

where $\tau_e = \sum \tau_{control} + \sum \tau_{disturbance}$. The model arises immediately from (3.19) and (2.14), and both control and disturbance torques were discussed in Chapter 3. As mentioned earlier, it is important to emphasize that torques related to the wheels should not be included in τ_e since these are already accounted for. We also note that (4.1) represents a general N-wheel gyrostat.

$$\mathbf{Q}\mathbf{Q}^{-1} = \mathbf{Q}^{-1}\mathbf{Q} = \mathbf{I}$$

¹Let **Q** be an $m \times n$ matrix of rank k. If k = m = n, then **Q** is nonsingular and has a unique inverse, \mathbf{Q}^{-1} . The inverse is both left and right inverse, that is

As indicated, the model is later to be used to control and simulate the ESEO satellite, equipped with a reaction wheel and several thrusters, and also including the gravity gradient as a disturbance. The idea is to use thrusters to implement the torques for large and fast (slew) maneuvers during attitude initialization and target acquisition phases, while providing momentum management when necessary. The wheel could be used to supply reference torque in the sense of overcoming the disturbances. It would also be desirable to use the wheel as much as possible, since this would save propelant for the thrusters, hence increasing the lifetime of the satellite.

However, even though the model in (4.1) is somewhat simplified compared to a real-life scenario, it is still considered too rigorous for our purpose. For that reason, different assumptions will be regarded subsequently to obtain even further simplifications.

- a) All the states in the model are given in \mathcal{F}_b .
- b) The origin of \mathcal{F}_b coincides with the center of gravity of the satellite.
- c) The inertia matrix for the satellite is diagonal, $\mathbf{I} = \text{diag}\{i_{11}, i_{22}, i_{33}\}$.
- d) The satellite has one reaction wheel, which creates an internal control torque about the satellite's principal y-axis. The axial inertia of the wheel is $\mathbf{I}_s \triangleq i_s$, and the axis of rotation in \mathcal{F}_b is given as $\mathbf{A} = [0, 1, 0]^{\mathrm{T}}$.
- e) The external control torques $\sum \tau_{control} \triangleq \tau = [\tau_1, \tau_2, \tau_3]^T$ are provided by pairs of thrusters, and they can implement angular velocities about the satellite's principal axes directly.
- f) The only disturbance torque to be considered is the gravity gradient, as described in (3.25).
- g) Neither the wheel nor the thrusters are assumed to have any dominant dynamics, by means of having much quicker dynamics than the satellite body.

Table 4.1: Model assumptions

As can be seen from (4.1), the angular velocities are given in \mathcal{F}_b and relative to \mathcal{F}_i , while the kinematics are relative to \mathcal{F}_o . However, we would like our model to represent the attitude of \mathcal{F}_b relative to \mathcal{F}_o . As will be shown, this can easily be done by exploiting the relation

$$\boldsymbol{\omega}_{ib}^{b} = \boldsymbol{\omega}_{ob}^{b} + \mathbf{R}_{o}^{b}\boldsymbol{\omega}_{io}^{o} \quad \text{and} \quad \dot{\boldsymbol{\omega}}_{ib}^{b} = \dot{\boldsymbol{\omega}}_{ob}^{b} + \dot{\mathbf{R}}_{o}^{b}\boldsymbol{\omega}_{io}^{o} = \dot{\boldsymbol{\omega}}_{ob}^{b} - \mathbf{S}(\boldsymbol{\omega}_{ob}^{b})\mathbf{R}_{o}^{b}\boldsymbol{\omega}_{io}^{o}$$
(4.2)

where $\omega_{io}^o = [0, -\omega_0, 0]^{\mathrm{T}}$ is assumed constant, and equal to the mean angular velocity of \mathcal{F}_o , given in \mathcal{F}_i . This implies a circular orbit. By utilizing (4.2), we rewrite our nonlinear model as

$$\dot{\omega}_{ob}^b = \hat{f}_{inert} + \hat{f}_{\tau} + \hat{f}_g + \hat{f}_{add}$$
(4.3a)

$$\dot{\omega}_s = f_{inert} + f_\tau + f_g \tag{4.3b}$$

$$\dot{\eta} = -\frac{1}{2} \epsilon^{\mathrm{T}} \omega_{ob}^{b} \tag{4.3c}$$

$$\dot{\boldsymbol{\epsilon}} = \frac{1}{2} \left[\eta \mathbf{I} + \mathbf{S}(\boldsymbol{\epsilon}) \right] \boldsymbol{\omega}_{ob}^{b}$$
(4.3d)

The different terms in (4.3) are given as

$$\hat{f}_{inert} = \mathbf{J}^{-1} \left[-\mathbf{S} (\boldsymbol{\omega}_{ob}^{b} - \boldsymbol{\omega}_{o} \mathbf{c}_{2}) \left(\mathbf{I} \left[\boldsymbol{\omega}_{ob}^{b} - \boldsymbol{\omega}_{o} \mathbf{c}_{2} \right] + \mathbf{A} \mathbf{I}_{s} \boldsymbol{\omega}_{s} \right) \right]$$
(4.4a)

$$\bar{f}_{inert} = \mathbf{A}^{\mathrm{T}} \mathbf{J}^{-1} \left[\mathbf{S}(\boldsymbol{\omega}_{ob}^{b} - \boldsymbol{\omega}_{o} \mathbf{c}_{2}) \left(\mathbf{I} \left[\boldsymbol{\omega}_{ob}^{b} - \boldsymbol{\omega}_{o} \mathbf{c}_{2} \right] + \mathbf{A} \mathbf{I}_{s} \boldsymbol{\omega}_{s} \right) \right]$$
(4.4b)

$$f_{\tau} = \mathbf{J}^{-1}\boldsymbol{\tau} - \mathbf{J}^{-1}\mathbf{A}\tau_{a}$$
(4.4c)
$$\bar{f}_{\tau} = -\mathbf{A}^{\mathrm{T}}\mathbf{J}^{-1}\boldsymbol{\tau} + \left[\mathbf{A}^{\mathrm{T}}\mathbf{J}^{-1}\mathbf{A} + \mathbf{I}_{s}^{-1}\right]\tau_{a}$$
(4.4d)

$$\hat{f}_g = \mathbf{J}^{-1} \left[3\,\omega_0^2 \,\mathbf{S}(\mathbf{c}_3) \mathbf{I} \mathbf{c}_3 \right] \tag{4.4e}$$

$$\hat{f}_g = -\mathbf{A}^{\mathrm{T}} \mathbf{J}^{-1} \left[3 \,\omega_0^2 \, \mathbf{S}(\mathbf{c}_3) \mathbf{I} \mathbf{c}_3 \right]$$
(4.4f)

$$\hat{f}_{add} = \omega_o \dot{\mathbf{c}}_2 \tag{4.4g}$$

where $\mathbf{c}_j \triangleq [c_{1j}, c_{2j}, c_{3j}]^{\mathrm{T}}$ is the j'th column of the rotation matrix \mathbf{R}_o^b . In the following we also denote each element in \mathbf{R}_o^b by c_{ij} , i, j = 1, 2, 3. From the assumptions in Table 4.1, and by defining $\omega_{ob}^b \triangleq [\omega_1, \omega_2, \omega_3]^{\mathrm{T}}$, we can write the terms in (4.4) in their final form, that is

$$\hat{f}_{inert} = \begin{bmatrix} -\frac{(i_s \,\omega_s + (i_{22} - i_{33})(\omega_2 - \omega_0 c_{22}))(-\omega_3 + \omega_0 c_{32})}{i_{11}} \\ \frac{(i_{11} - i_{33})(\omega_1 - \omega_0 c_{12})(-\omega_3 + \omega_0 c_{32})}{i_{22} - i_s} \\ -\frac{(i_s \,\omega_s - (i_{11} - i_{22})(\omega_2 - \omega_0 c_{22}))(\omega_1 - \omega_0 c_{12})}{i_{33}} \end{bmatrix}$$
(4.5a)

$$\bar{f}_{inert} = \frac{(i_{11} - i_{33})(\omega_1 - \omega_0 c_{12})(\omega_3 - \omega_0 c_{32})}{i_{22} - i_s}$$
(4.5b)

$$\hat{f}_{\tau} = \begin{bmatrix} \frac{\tau_1}{i_{11}} \\ \frac{(i_{22} - i_s)\tau_2 - \tau_a}{(i_{22} - i_s)^2} \\ \frac{\tau_3}{i_{33}} \end{bmatrix}$$
(4.5c)

$$\bar{f}_{\tau} = \frac{i_{22} \tau_a - i_s \tau_2}{(i_{22} - i_s) i_s}$$
(4.5d)

$$\hat{f}_{g} = 3 \omega_{0}^{2} \begin{bmatrix} (i_{33} - i_{22})c_{23}c_{33} \\ (i_{11} - i_{33})c_{13}c_{33} \\ (i_{22} - i_{11})c_{13}c_{23} \end{bmatrix}$$
(4.5e)

$$\bar{f}_g = \frac{3\,\omega_0^2(i_{11}-i_{33})c_{13}c_{33}}{i_s-i_{22}} \tag{4.5f}$$

$$\hat{f}_{add} = \omega_0 \begin{bmatrix} \omega_3 c_{22} - \omega_2 c_{32} \\ \omega_1 c_{32} - \omega_3 c_{12} \\ \omega_2 c_{12} - \omega_1 c_{22} \end{bmatrix}$$
(4.5g)

4.1.1 Linearized model

To derive a linearized model of the satellite attitude, the nonlinear model in (4.3) has to be differentiated with respect to the total state vector, which for the the time being is chosen as $\mathbf{x} = [\omega_1, \omega_2, \omega_3, \omega_s, \eta, \epsilon_1, \epsilon_2, \epsilon_3]^{\mathrm{T}}$. Similar, the control vector is denoted by $\mathbf{u} = [\tau_1, \tau_2, \tau_3, \tau_a]^{\mathrm{T}}$. The linearized system can then be written as

$$\Delta \dot{\mathbf{x}} = \mathbf{A} \Delta \mathbf{x} + \mathbf{B} \Delta \mathbf{u} \tag{4.6}$$

where the matrices ${\bf A}$ and ${\bf B}$ are found from

$$\mathbf{A} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_8} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_8} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_8}{\partial x_1} & \frac{\partial f_8}{\partial x_2} & \cdots & \frac{\partial f_8}{\partial x_8} \end{bmatrix} \Big|_p \qquad \mathbf{B} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & \cdots & \frac{\partial f_1}{\partial u_4} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} & \cdots & \frac{\partial f_2}{\partial u_4} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_8}{\partial u_1} & \frac{\partial f_8}{\partial u_2} & \cdots & \frac{\partial f_8}{\partial u_4} \end{bmatrix} \Big|_p \qquad (4.7)$$

By letting the nonlinear model be denoted by $\mathbf{f}(\mathbf{x}, \mathbf{u}) = [\dot{\boldsymbol{\omega}}_{ob}^b, \dot{\omega}_s, \dot{\eta}, \dot{\boldsymbol{\epsilon}}]^{\mathrm{T}} \triangleq [f_1, \dots, f_8]^{\mathrm{T}}$, and choosing the equilibrium point p equal to $\mathbf{x}^p = [\mathbf{0}^4, 1, \mathbf{0}^3]^{\mathrm{T}}, \mathbf{u}^p = \mathbf{0}^4$, the matrices in (4.7) are

where we used the notation $k_x = \frac{i_{22} - i_{33}}{i_{11}}$, $k_y = \frac{i_{11} - i_{33}}{i_{22}}$, $k_z = \frac{i_{22} - i_{11}}{i_{33}}$ and $\kappa = i_{22} - i_s$.

4.2 Controllability

Investigating controllability is rather trivial when it comes to a linearized system, while for a nonlinear one it might be a rigorous task. The latter will therefore be omitted at this point, and we end the discussion by saying that we could have utilized *Lie brackets* and *Lie algebraic* rank conditions (LARC) to show accessibility of the nonlinear system. If accessibility was to be found, we could have investigated the system further to obtain a conclusion on controllability.

In case of a linear system we can easily conclude the same by using the following definition

Definition 4.1 (Controllability). The state and input matrix (\mathbf{A}, \mathbf{B}) must satisfy the controllability condition to ensure that there exists a control $\mathbf{u}(t)$ which can drive an arbitrary state $\mathbf{x}(t_0)$ to another arbitrary state $\mathbf{x}(t_1)$ for $t_1 > t_0$. The controllability condition requires that the $n \times n$ matrix

$$\boldsymbol{\mathcal{C}} = \begin{bmatrix} \mathbf{B} \, | \, \mathbf{A}\mathbf{B} \, | \, \dots \, | \, \mathbf{A}^{n-1}\mathbf{B} \end{bmatrix}$$
(4.9)

must be of rank n. A sufficient and necessary condition is that C has an inverse.

From the system matrix in (4.8a), we can immediately conclude that the linearized system is uncontrollable. This is a direct consequence in that all the terms corresponding to η are equal to zero. In fact, η turns out to be the only uncontrollable state, and the linearized system, omitting η , is controllable. As will be discussed subsequently, in order for the whole system to be *stabilizable*, the mode corresponding to the uncontrollable state needs to be stable.

Note that, even though η is uncontrollable, we can utilize the theory related to the Euler parameters, that were used earlier for describing the kinematics. From Chapter 2 we recall that they satisfy $\eta^2 + \epsilon^T \epsilon = 1$. This is an interesting observation, which makes us able to update η in an open-loop manner. In Simulink this is later done by utilizing a normalization block.

4.3 Stabilizability

As briefly mentioned above, for an uncontrollable system to be stabilizable, all the eigenvalues corresponding to the uncontrollable modes must be stable. A reasonable approach is to use a *Kalman canonical decomposition*, where the original linear system is decomposed into controllable and uncontrollable subspaces. A theorem on the topic is given in Chen (1999).

Theorem 4.1. Consider a n-dimensional linear state equation

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad \mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}$$
 (4.10)

with rank(\mathcal{C}) = $n_1 < n$, where \mathcal{C} is given as in (4.9). We form the $n \times n$ matrix $\mathbf{P}^{-1} \triangleq [\mathbf{p}_1, \dots, \mathbf{p}_{n_1}, \dots, \mathbf{p}_n]$, where the first n_1 columns are any n_1 linearly independent columns of \mathcal{C} , while the remaining columns can be arbitrarily chosen as long as \mathbf{P} is nonsingular. Then the similarity transformation $\mathbf{x} = \mathbf{P}^{-1}\bar{\mathbf{x}}$ transforms (4.10) into

$$\begin{bmatrix} \dot{\bar{\mathbf{x}}}_c \\ \dot{\bar{\mathbf{x}}}_{uc} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{A}}_c & \bar{\mathbf{A}}_{12} \\ \mathbf{0} & \bar{\mathbf{A}}_{uc} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{x}}_c \\ \bar{\mathbf{x}}_{uc} \end{bmatrix} + \begin{bmatrix} \bar{\mathbf{B}}_c \\ \mathbf{0} \end{bmatrix} \mathbf{u}, \quad \mathbf{y} = \begin{bmatrix} \bar{\mathbf{C}}_c & \bar{\mathbf{C}}_{uc} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{x}}_c \\ \bar{\mathbf{x}}_{uc} \end{bmatrix} + \mathbf{D}\mathbf{u}$$
(4.11)

where $\bar{\mathbf{A}}_c$ and $\bar{\mathbf{A}}_{uc}$ are $n_1 \times n_1$ and $(n - n_1) \times (n - n_1)$, and controllable and uncontrollable, respectively. The transfer function of the controllable subsystem is equal to that of (4.10).

Based on Theorem 4.1 we then investigate the complete linearized system, followed by the eigenvalue of the uncontrollable subsystem corresponding to η . The physical parameters used in this analysis, and for the remainder of this thesis are given in Table 5.1.

The outline described above is implemented in the function ctrbf in MATLAB, except that the order of the columns in \mathbf{P}^{-1} is reversed. After plugging in the model equations, we find that $\operatorname{eig}(\bar{\mathbf{A}}_{uc}) = 0$, hence being stable in sense of Lyapunov, or equally, marginally stable.

Based on this analysis we conclude that the complete linearized system is uncontrollable *but* stabilizable. This is for instance a necessary condition to guarantee a bounded performance index, or cost function, when implementing LQR.

4.4 Summary

In this chapter a reasonable nonlinear model for describing the SSETI/ESEO satellite was developed, based on theory discussed in earlier chapters. A linearized model was also derived, which is later to be used when designing and implementing various attitude control schemes.

General properties for the nonlinear model were not discussed at this point due to complexity. On the other hand, the linearized model was found to be uncontrollable, but stabilizable by means of the available control torques. Also, even though we do not have any direct control of η , the state can be updated by utilizing the properties of the Euler parameters. Observability was not discussed in this chapter, but it is assumed that this is fulfilled.

Finally we note that since all values corresponding to η turned out to be zero in the linearized model, we will for the remainder discard this state, hence reducing the order of the model to 7 states, as apposed to 8. The reduced size model is both controllable and observable.

Chapter 5 Attitude control and simulations

As mentioned in the introduction, the purpose of this thesis is to establish and investigate a reasonable model of a micro-satellite, and then finally propose a strategy with hopes of solving the problem of attitude control and stabilization. The system to be studied has three degrees of freedom and four available controls, which makes it an overactuated system. As discussed earlier, the model and its operation modes are based on the SSETI/ESEO satellite.

In the following we start off by giving a summary of some physical parameters for the satellite, as well as performing some open-loop simulations. Afterwards, we continue by discussing some well known control strategies, followed by the main topic of this thesis, attitude control by means of explicit MPC, via multi-parametric quadratic programming (mpQP). Since the actuating thrusters are on-off by nature, input modulation is an important implementation aspect. Some attempts are done in solving this problem, utilizing a simple bang-bang scheme.

The signal modulation scheme, and all the control strategies, are simulated together with the satellite plant in closed-loop. A thorough discussion of the results is given at the end.

5.1 SSETI/ESEO parameters

Some major physical parameters are summarized in Table 5.1. The values were used in earlier analysis, and are to be used throughout the remainder of this thesis. As earlier, all values are based on different documents in the Phase B report.

Short	Explanation	Value
$\mathbf{I} = \operatorname{diag}(i_{11}, i_{22}, i_{33})$ i_s $\tau_n = [\tau_{1n}, \tau_{2n}, \tau_{3n}]^{\mathrm{T}}$	Satellite inertia matrix Axial wheel inertia Nominal thruster torque	diag $(4.250, 4.337, 3.664)$ [kg m ²] $4 \cdot 10^{-5}$ [kg m ²] $[0.0484, 0.0484, 0.0398]^{T}$ [Nm] 527 [rod/g] ≈ 5022 mm
$\omega_{smax} \ \omega_o$	Maximum wheel velocity Mean angular velocity of \mathcal{F}_o	$527 \text{ [rad/s]} \approx 5032 \text{ rpm}$ Arbitrary, yet fixed

Table 5.1: SSETI/ESEO parameters

5.2 **Open-loop simulations**

Some simple open-loop simulations are given in the following with the purpose of showing the functionality of the orbit propagator, as well as discussing a simplification by means of using circular orbits, apposed to elliptical ones.

The distance from the origin of \mathcal{F}_i to \mathcal{F}_o is given in (E.8), in which we can easily find the average distance from the center of the Earth to the satellite by evaluating the integral

$$\bar{r} = \frac{1}{2\pi} \int_0^{2\pi} \frac{a(1+e^2)}{1+e\cos\nu} \, d\nu \quad \Rightarrow \quad \bar{r} = a\sqrt{1-e^2}$$
(5.1)

When we later do control design we will assume circular orbits, according to (5.1), while interpreting a varying distance as a disturbance when doing closed-loop simulations. In fact, when we derived our nonlinear model, we assumed that $\dot{\omega}_o = 0$, which implies circular orbits.

For the rest of this thesis, initial COE for SSETI/ESEO will be given as; i = 7 [deg], a = 24603.14 [km], e = 0.718, $\Omega_0 = -10$ [deg], $\varpi_0 = 178$ [deg] and $\nu_0 = 0$ [deg].

Using these initial conditions, the orbit propagator gives the results in Figure 5.1 and 5.2.



Figure 5.1: Results from orbit propagator, showing \mathcal{F}_o and \mathcal{F}_i

General open-loop simulations of the model are not included at this point. We also note that all the Simulink blocks that make up the system, as well as various MATLAB functions, are given in Appendix C.

5.3 Control design

The open-loop simulations of both the system and the orbit propagator gave reasonable results, and will therefore not be analyzed any further. The remainder of this section is on closed-loop control, where we begin with a discussion on the well known PD control scheme, as well as the Linear Quadratic Regulator (LQR). A more detailed discussion is given on explicit Model



Figure 5.2: Difference between original and mean orbit

Predictive Control, since this is a less known approach.

Simulations for all cases are given at the end of this chapter. For clarity, we note that when utilized, the term linearized model means the reduced size model, omitting η .

5.3.1 PD-control

As mentioned earlier, Euler parameters are widely used for describing attitude orientations, and they are also well suited for onboard real-time computations. A simple, yet reliable control scheme can be implemented using a linear *quaternion feedback control*, which is a realization of the more familiar PD control approach. The control torques can easily be found from

$$\boldsymbol{\tau} = -\begin{bmatrix} \mathbf{K}_p & \mathbf{K}_d \end{bmatrix} \begin{bmatrix} \mathbf{q}_e \\ \boldsymbol{\omega} \end{bmatrix}$$
(5.2)

where \mathbf{q}_e is the attitude error quaternion vector, and $\boldsymbol{\omega}$ is short for $\boldsymbol{\omega}_{ob}^b$. We recall that in the case of rotation matrices it does not make sense to subtract one matrix from another as the result would not be a valid rotation matrix. Therefore, as in (2.7) we define $\mathbf{R}(\tilde{\mathbf{q}}) \triangleq \mathbf{R}^{\mathrm{T}}(\mathbf{q}_d)\mathbf{R}(\mathbf{q}) = \mathbf{R}(\bar{\mathbf{q}}_d)\mathbf{R}(\mathbf{q})$, where $\mathbf{q} \triangleq [\eta, \boldsymbol{\epsilon}^{\mathrm{T}}]^{\mathrm{T}} = [\eta, \epsilon_1, \epsilon_2, \epsilon_3]^{\mathrm{T}}$ according to Section 2.3. Similarly, the desired Euler parameters are defined as $\mathbf{q}_d \triangleq [\eta_d, \boldsymbol{\epsilon}_d^{\mathrm{T}}]^{\mathrm{T}} = [\eta_d, \epsilon_{1d}, \epsilon_{2d}, \epsilon_{3d}]^{\mathrm{T}}$, while $\bar{\mathbf{q}}_d$ denotes the complex conjugate of \mathbf{q}_d . Further more, since we are dealing with successive rotations, we know that $\mathbf{R}(\bar{\mathbf{q}}_d)\mathbf{R}(\mathbf{q}) = \mathbf{R}(\bar{\mathbf{q}}_d \otimes \mathbf{q})$, where \otimes is the quaternion product operator. The error in Euler parameters can then be written as (Egeland and Gravdahl, 2002)

$$\tilde{\mathbf{q}} \triangleq \bar{\mathbf{q}}_d \otimes \mathbf{q} = \begin{bmatrix} \eta_d & \boldsymbol{\epsilon}_d^{\mathrm{T}} \\ -\boldsymbol{\epsilon}_d & \eta_d \mathbf{I}_{3\times 3} - \mathbf{S}(\boldsymbol{\epsilon}_d) \end{bmatrix} \mathbf{q}$$
(5.3)

or equally, if we define $\tilde{\mathbf{q}} \triangleq [\tilde{\eta}, \tilde{\boldsymbol{\epsilon}}^{\mathrm{T}}]^{\mathrm{T}} = [\tilde{\eta}, \tilde{\epsilon}_{1}, \tilde{\epsilon}_{2}, \tilde{\epsilon}_{3}]^{\mathrm{T}}$, in component form

$$\begin{bmatrix} \tilde{\eta} \\ \tilde{\epsilon}_1 \\ \tilde{\epsilon}_2 \\ \tilde{\epsilon}_3 \end{bmatrix} = \begin{bmatrix} \eta_d & \epsilon_{1d} & \epsilon_{2d} & \epsilon_{3d} \\ -\epsilon_{1d} & \eta_d & \epsilon_{3d} & -\epsilon_{2d} \\ -\epsilon_{2d} & -\epsilon_{3d} & \eta_d & \epsilon_{1d} \\ -\epsilon_{3d} & \epsilon_{2d} & -\epsilon_{1d} & \eta_d \end{bmatrix} \begin{bmatrix} \eta \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix}$$
(5.4)

The attitude error quaternion vector in (5.2) consists of the last three errors, i.e. $\mathbf{q}_e = [\tilde{\epsilon}_1, \tilde{\epsilon}_2, \tilde{\epsilon}_3]^{\mathrm{T}}$.

Remark 5.3.1. The error in the Euler parameters in (5.3) is a valid unit quaternion, that is, $\tilde{\mathbf{q}}$ satisfies $\tilde{\eta} + \tilde{\epsilon}^{\mathrm{T}} \tilde{\epsilon} = 1$. The latter would not have been the case if we were to define $\tilde{\mathbf{q}} = \mathbf{q} - \mathbf{q}_d$.

As is evident from (5.2), the PD control law does not utilize any information about the wheel. Also, the calculated control torques can be interpreted as generalized torques about the satellite's principal axes. Due to the fact that we have redundancy about the principal y-axis, the above mentioned issues lead us to a new problem in sense of doing torque allocation. A preliminary and simple way of solving this problem can be implemented using a simple dead-zone approach. In short, this means that the calculated torque about the y-axis is applied only to the wheel, whenever the torque is within a specific range, specified by the dead-zone. A more comprehensive discussion on control allocation for ships and marine vessels, though applicable for satellites, is given in Fossen (2002). It is shown that in its most simple form, a solution to the unconstrained¹ allocation problem can be found by solving a least-squares (LS) optimization problem, that is

$$J = \min_{\mathbf{f}} \left\{ \mathbf{f}^{\mathrm{T}} \mathbf{W} \mathbf{f} \right\}$$

subject to: $\boldsymbol{\tau} - \mathbf{T} \mathbf{f} = 0$ (5.5)

where \mathbf{W} is a positive definite matrix, weighting the control actuators, and the actuator configuration matrix \mathbf{T} is defined in terms of a set of column vectors, one for each actuator. Furthermore, it is shown that an explicit solution to (5.5) can be found as

$$\mathbf{f} = \mathbf{W}^{-1}\mathbf{T}^{\mathrm{T}}\left(\mathbf{T}\mathbf{W}^{-1}\mathbf{T}^{\mathrm{T}}
ight)^{-1}oldsymbol{ au} = \mathbf{T}_{\omega}^{\dagger}oldsymbol{ au}$$

where $\mathbf{T}_{\omega}^{\dagger}$ is recognized as the *generalized inverse*. The final control **u** is then applied to the different actuators according to the following expression

$$\mathbf{u} = \mathbf{K}^{-1} \mathbf{T}_{\omega}^{\dagger} \boldsymbol{\tau} \tag{5.6}$$

where K is the torque coefficient matrix, and τ is the generalized torque.

In industrial systems it may be necessary to minimize power consumptions or take actuator limitations into account. Other limitations may also exist, and in general this leads to a constrained allocation problem. This will not be pursued any further at this point, and the reader should refer to Fossen (2002) for details. Note that an explicit solution to latter has been developed by Johansen, Fossen and Tøndel (2003), based on multi-parametric quadratic programming. This is a similar approach, yet less complex, as will be used later in this thesis when doing explicit MPC. More information on this topic will be given subsequently.

¹Unconstrained, as apposed to constrained, means that there are no bounds on the elements to be investigated.

5.3.2 LQR

Using state feedback allows one to assign any closed-loop system eigenvalues if the system is controllable. This is indeed the case when solving the Linear Quadratic (LQ) optimal control problem. The LQ problem is one of the most frequently appearing optimal control problems, and is the basis of many modern robust control system design methods. A fundamental design issue is the regulator problem, where it is necessary to regulate the outputs or the states of the system to zero while ensuring that they exhibit desirable time-response characteristics. A Linear Quadratic Regulator (LQR) can be designed for this purpose.

The continuous-time *linear* system to be controlled is described by the state-space model

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) \text{ and } \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t)$$
 (5.7)

and the LQR is sought to minimize the quadratic performance index, or cost function, given as

$$J = \frac{1}{2} \mathbf{x}^{\mathrm{T}}(t_f) \, \mathbf{S} \, \mathbf{x}(t_f) + \frac{1}{2} \int_{t_0}^{t_f} \left\{ \mathbf{x}^{\mathrm{T}}(t) \, \mathbf{Q}(t) \, \mathbf{x}(t) + \mathbf{u}^{\mathrm{T}}(t) \, \mathbf{R}(t) \, \mathbf{u}(t) \right\} \, dt \tag{5.8}$$

where $\mathbf{S} = \mathbf{P}(t_f) \ge 0$, $\mathbf{Q} \ge 0$, $\mathbf{R} > 0$, and t_0 and t_f indicate initial and final time, respectively.

Remark 5.3.2. The problem statement is given for the general case where **A**, **B**, **Q** and **R** are time-varying. In most cases however, they are time-invariant, as is the case for our system.

It can easily be found that for J to be minimized the control must be given as

$$\mathbf{u}(t) = -\mathbf{K}(t)\mathbf{x}(t) = -\mathbf{R}^{-1}(t)\mathbf{B}^{\mathrm{T}}(t)\mathbf{P}(t)\mathbf{x}(t)$$
(5.9)

where $\mathbf{P}(t)$ is the solution of the Riccati differential equation, that is

$$\frac{d\mathbf{P}(t)}{dt} = \mathbf{A}^{\mathrm{T}}(t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{A}(t) - \mathbf{P}(t)\mathbf{B}(t)\mathbf{R}^{-1}(t)\mathbf{B}^{\mathrm{T}}(t)\mathbf{P}(t) + \mathbf{Q}(t), \quad \mathbf{P}(t_f) = \mathbf{S}$$

A practical and useful case is obtained by letting $t_f \to \infty$. If the system in (5.7) is controllable or stabilizable the performance index in (5.8) remains bounded, and $\mathbf{P}(t)$ converges to a stationary solution of the Riccati differential equation when integrated backward in time. If we now assume a time-invariant system, the stationary solution, \mathbf{P}_{∞} , is obtained by solving the continuous-time algebraic Riccati equation (ARE), that is

$$\mathbf{A}^{\mathrm{T}}\mathbf{P}_{\infty} + \mathbf{P}_{\infty}\mathbf{A} - \mathbf{P}_{\infty}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{P}_{\infty} + \mathbf{Q} = \mathbf{0}$$
(5.10)

The optimal control in (5.9) now simplifies to

$$\mathbf{u}(t) = -\mathbf{K}\mathbf{x}(t) = -\mathbf{R}^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{P}_{\infty}\mathbf{x}(t)$$
(5.11)

and the closed-loop system is given as

$$\dot{\mathbf{x}}(t) = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x}(t) = \mathbf{A}_c \,\mathbf{x}(t) \tag{5.12}$$

If the system is observable or detectable, the optimal state feedback is asymptotically stable, or equally, A_c is Hurwitz. The steady-state LQR feedback control law can be computed in MATLAB by applying the commands

Q = diag([q11,q22,...,qnn]); R = diag([r11,r22,...,rnn]); [K,P,E] = lqr(A,B,Q,R);

where K is the optimal gain, P is the stationary solution to the ARE and E contains the eigenvalues for the closed-loop system.

Given some weight matrices, the optimality of the state feedback control gain is measured in sense of the performance index. Whether it defines a good system in any engineering sense depends on the choice of \mathbf{Q} and \mathbf{R} . Even though finding these matrices is usually an iterative procedure, there are some rules of thumb. Without discussing these any further in this thesis, the reader should refer to Bryson and Ho (1975) for details.

To be shown later when doing simulations, the LQR approach achieves reasonable results. As mentioned above however, the performance is closely related to our choice of weight matrices. Also, even though the LQR is a robust control scheme, in sense of gain and phase margins, we can not directly include any constraints on the states nor the inputs for the system. As discussed earlier, a constrained allocation problem would be applicable, and together with the LQR, this would most likely provide a good solution. In fact, as will be shown later, the LQR is highly relevant when doing explicit MPC. Finally we note that, unlike the PD control, the LQR utilizes the information about the wheel, as well as rendering the angular velocity of the wheel to zero.

5.3.3 Explicit MPC

As already announced, the main topic of this thesis is on attitude control by means of *explicit Model Predictive Control*, from now on called eMPC. In the following we start by giving a brief review on the traditional MPC scheme, while continuing with an explanation on how this approach can be extended to eMPC, via multi-parametric quadratic programming (mpQP). Some pros and cons of eMPC, as well as implementation aspects, are also briefly discussed. To illustrate the whole procedure, a simple second order system is used, while when doing simulations at the end, the similar approach is applied for the linearized satellite model. The following discussion is based on Tøndel (2003) and Bemporad et al. (2002).

Model Predictive Control

Roughly speaking, MPC refers to a class of algorithms that compute a sequence of manipulated variable adjustments in order to optimize the future behavior of the plant. More specifically this means that the control action is obtained by computing an open-loop optimal sequence of control moves on a predefined horizon, once for each time sample. The first control input in the sequence is then applied to the plant, and the optimization is repeated with the new initial conditions and on the new horizon, shifted one step ahead. Due to the shifted horizon, the term *receding horizon control* is commonly used interchangeably with MPC. We also note that due to the repeated optimization, MPC is considered a closed-loop approach.

The ability to include process input and output constraints directly in the problem formulation, so that future constraint violations are anticipated and prevented, is probably the most important reason for the success of MPC in the industry. An excellent survey of successful use of traditional on-line MPC can be found in Qin and Badgwell (1997).

For the remainder of the discussion on MPC and eMPC, the process to be controlled can be described by a *discrete-time*, deterministic linear state-space model, that is

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k)$$
 (5.13)

where $\mathbf{x}(k) \in \mathbb{R}^n$ is the state variable, $\mathbf{u}(k) \in \mathbb{R}^m$ is the input variable, $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{m \times m}$, and (\mathbf{A}, \mathbf{B}) is a stabilizable pair. If we now consider the regulator problem, that is, the problem of rendering the state vector to the origin, the MPC solves the following optimization problem for the current $\mathbf{x}(k)$

$$\min_{\mathbf{U}} \{J(\mathbf{U}, \mathbf{x}(k))\}$$
subject to: $\mathbf{y}_{\min} \leq \mathbf{y}(k+i|k) \leq \mathbf{y}_{\max}, i = 1, \dots, N$
 $\mathbf{u}_{\min} \leq \mathbf{u}(k+i) \leq \mathbf{u}_{\max}, i = 1, \dots, M-1$
 $\mathbf{x}(k|k) = \mathbf{x}(k)$
 $\mathbf{x}(k+i+1|k) = \mathbf{A}\mathbf{x}(k+i|k) + \mathbf{B}\mathbf{u}(k+i), k \geq 0$
 $\mathbf{y}(k+i) = \mathbf{C}\mathbf{x}(k+i|k), k \geq 0$
 $\mathbf{u}(k+i) = \mathbf{K}\mathbf{x}(k+i|k), M \leq k \leq N-1$

$$(5.14)$$

where $\mathbf{U} \triangleq [\mathbf{u}^{\mathrm{T}}(k), \dots, \mathbf{u}^{\mathrm{T}}(k+M-1)]^{\mathrm{T}}$, $\mathbf{y}_{\min} < 0 < \mathbf{y}_{\max}$, $\mathbf{u}_{\min} < 0 < \mathbf{u}_{\max}$, $\mathbf{R} = \mathbf{R}^{\mathrm{T}} > 0$, $\mathbf{Q} = \mathbf{Q}^{\mathrm{T}} \ge 0$, $\mathbf{P} = \mathbf{P}^{\mathrm{T}} > 0$, $\mathbf{x}(k+i|k)$ is the prediction of $\mathbf{x}(k+i)$ at time k, M and N are input and constraint horizons, and the cost function that we try to minimize is given as

$$J = \mathbf{x}^{\mathrm{T}}(k+N|k)\mathbf{P}\mathbf{x}(k+N|k) + \sum_{i=0}^{N-1} \left\{ \mathbf{x}^{\mathrm{T}}(k+i|k)\mathbf{Q}\mathbf{x}(k+i|k) + \mathbf{u}^{\mathrm{T}}(k+i)\mathbf{R}\mathbf{u}(k+i) \right\}$$

As is evident from the cost function above, we see that it resembles the continuous-time cost function of the LQR in (5.8). In fact, when the final cost matrix **P** and gain matrix **K** in (5.14) are calculated from the *discrete-time* version of the ARE in (5.10), under the assumptions that the constraints are not active for $k \ge M$, (5.14) exactly solves the constrained infinite horizon LQR problem for (5.13).

To ensure stability the final state vector should enter an invariant set in which no constraints are active. That way, once $\mathbf{x}(k)$ is a member of the invariant set, the LQR is optimal for all time. This will not be investigated any further in thesis however, and we end the discussion by nothing that several modifications to MPC have been suggested to guarantee stability of the resulting controller. Also, as will be shown later, in sense of finding explicit solutions to the MPC problem, the solution turns out to be a piecewise affine (PWA) function. Consequently, a potential approach to guarantee stability in case of eMPC is to search for piecewise quadratic Lyapunov functions by solving a convex optimization problem. In Ferrari-Trecate et al. (2001) this was done by utilizing linear matrix inequalities (LMIs).

From linear MPC to mpQP

By substituting $\mathbf{x}(k+i|k) = \mathbf{A}^{i}\mathbf{x}(k) + \sum_{j=0}^{i-1}\mathbf{A}^{j}\mathbf{Bu}(k+i-1-j)$, it can be shown (Bemporad et al., 2002) that the linear MPC problem in (5.14) can be rewritten as

$$V(\mathbf{x}(k)) = \min_{\mathbf{U}} \left\{ \frac{1}{2} \mathbf{U}^{\mathrm{T}} \mathbf{H} \mathbf{U} + \mathbf{x}^{\mathrm{T}}(k) \mathbf{F} \mathbf{U} \right\}$$

subject to: $\mathbf{G} \mathbf{U} \leq \mathbf{W} + \mathbf{E} \mathbf{x}(k)$ (5.15)

The optimization problem in (5.15) is a quadratic program (QP). However, since we usually say that a mathematical program containing a vector of parameters, as apposed to a scalar value, is a multi-parametric program, we can reefer to (5.15) as a multi-parametric quadratic program (mpQP) in U.

From the relation $\mathbf{z} \triangleq \mathbf{U} + \mathbf{H}^{-1}\mathbf{F}^{T}\mathbf{x}(k)$, and by completing squares, (5.15) can be transformed further, into an equivalent mpQP in \mathbf{z} , that is

$$V_{z}(\mathbf{x}(k)) = \min_{\mathbf{z}} \left\{ \frac{1}{2} \mathbf{z}^{\mathrm{T}} \mathbf{H} \mathbf{z} \right\}$$

subject to: $\mathbf{G} \mathbf{z} \leq \mathbf{W} + \mathbf{S} \mathbf{x}(k)$ (5.16)

where $\mathbf{x}(k)$ is the current state vector, which can be treated as a vector of parameters. Note that $\mathbf{H} > 0$ since $\mathbf{R} > 0$. The latter is a strong result, in which the problem formulated in (5.16) is strictly convex, and the Karush-Kuhn-Tucker (KKT) conditions are necessary and sufficient conditions for optimality, giving an unique solution (Nocedal and Wright, 1999). As shown in Bemporad et al. (2002), the mpQP in (5.16) can be solved by applying the KKT conditions

$$\mathbf{Hz} + \mathbf{G}^{T} \lambda = 0, \quad \lambda \in \mathbb{R}^{q}$$

$$\lambda_{r} \left(\mathbf{G}^{r} \mathbf{z} - \mathbf{W}^{r} - \mathbf{S}^{r} \mathbf{x}(k)\right) = 0, \quad r = 1, \dots, q,$$

$$\lambda \ge 0,$$

$$\mathbf{Gz} - \mathbf{W} - \mathbf{Sx}(k) \le 0.$$

(5.17)

where the subscript r on some matrix denotes the r^{th} row, while q is the number of inequalities in the optimization problem, and the number of free variables is $n_z = m \cdot N$. Matrix dimensions are then given as $\mathbf{z} \in \mathbb{R}^{n_z}$, $\mathbf{H} \in \mathbb{R}^{n_z \times n_z}$, $\mathbf{G} \in \mathbb{R}^{q \times n_z}$, $\mathbf{W} \in \mathbb{R}^{q \times 1}$, and $\mathbf{S} \in \mathbb{R}^{q \times n}$.

Implementation aspects and properties of eMPC

Since the problem in (5.15) depends on the current state $\mathbf{x}(k)$, the implementation of a traditional MPC requires an on-line solution of a QP at each time step. Although efficient QP solvers are available, computing the input $\mathbf{u}(k)$ requires significant on-line computation effort. For this reason, the application of traditional MPC has only been suitable to processes with relatively slow dynamics.

Though limited to reasonable small problems, the idea behind eMPC is to divide the statespace or parameter space into a manageably small number of (convex) polyhedra or regions, in which one can pre-compute (off-line) different optimal control laws, to be applied according to a specific region. The key observation is that in the case of eMPC, the state $\mathbf{x}(k)$ is interpreted as a parameter and the problem in (5.15) is solved off-line for *all* \mathbf{x} . It is shown in Bemporad et al. (2002) that the solution $\mathbf{z}^*(\mathbf{x}(k))$ of (5.16), hence $\mathbf{U}^*(\mathbf{x}(k))$, is a continuous *piecewise affine* (PWA) function defined over a polyhedra partition. Consequently, the on-line effort is limited to evaluating this PWA function. In particular, the MPC algorithm to be implemented in real-time will then simply consist of reading the appropriate control law from a table look-up, depending on the current state estimate. This is particularly useful in high-bandwidth applications, when high control update rates are required.

In addition to the advantages mention above, eMPC can easily be implemented on inexpensive hardware, as fixed point arithmetics can be used. Also, the overhead in the table look-up approach is minimal, in that a few lines in software is sufficient. As mentioned earlier however, a disadvantage is that the method is limited to fairly small problems, since memory requirements and off-line computation times seems to increase more or less exponentially with problem dimension. It is reasonable to say that the number of polyhedra regions gives an indication of the complexity of the eMPC. The number of regions N_r is closely related to the number of constraints q, which in turn depends on the number of states n, the constraint horizon N, and the number of inputs and outputs, m and p, respectively. The higher value of q the higher value of N_r . Some complexity reduction techniques exist however, using for instance traditional input parametrization (input blocking). This approach is well known from traditional MPC. Another approach is to represent the calculated PWA optimal control law as a binary search tree for efficient on-line evaluation. For further information the reader is referred to Tøndel and Johansen (2002) and references therein.

We also note that even though the incorporation of input and output constraints in the problem formulation is the major advantage in MPC, is can also lead to infeasibility problems. This can happen because of unexpected disturbances, or due to model uncertainties. There are many ways in which the predictive control problem can become infeasible, and usually they are difficult to predict. Obviously, the consequence could be an unacceptable behavior and a situation where the controller is unable to find an input that keeps the plant within its constraints. Because of this, it is important to have a strategy for dealing with the possibility of infeasibility. A common approach is to *soften* the constraints, allowing one or more constraints to be violated if necessary for finding a solution to the optimization problem. A straightforward way to do this is to add new variables in the problem, so called *slack variables*, which are defined in such a way that they are non-zero only if the constraints are violated. On the other hand, if non-zero, they are heavily penalized in the cost function, so that the optimizer has a strong incentive to keep them zero if possible. When softening constraints we note that this is in most cases only possible with output or state variables, as input from actuators commonly suffer from hard constraints, e.g. saturation in hardware, and there is no way in which they can be softened. More information on slack variables will be given subsequently.

An mpQP toolbox in MATLAB

As explained above, explicit solutions to constrained linear MPC problems can be obtained solving multi-parametric quadratic programs, where the parameters are the components of the state vector. In the case of single parameter problems, as well as multi-parametric linear programs (mpLP), numerous work exist. When dealing with mpQP problems however, the contribution is limited. The work done by Bemporad et al. (2002) is one of the few dealing with general linear MPC problems. Another major contribution on the topic is given in Tøndel et al. (2003a), in which computation speed is greatly improved. For the remainder of this thesis, the

optimization solver to be used for solving the mpQP problem is based on Tøndel et al. (2003a), and the necessary algorithms are implemented in a non-commercial software package. The toolbox, which is implemented in MATLAB, was developed by Dr. Petter Tøndel and Professor Tor Arne Johansen, at the Department of Engineering Cybernetics, Norwegian University of Science and Technology. Without going into details, the toolbox contains functionality for completing the following tasks:

- **Formulation of mpQP**. By giving the linear discrete-time model, constraints, slack variable costs, cost matrices etc. of a linear MPC problem, the toolbox has functionality to reformulate this into an mpQP.
- mpQP solver. The mpQP problem is solved, giving a PWA control law.
- Generating binary search tree. The PWA control law can be represented as a binary search tree for efficient on-line evaluation.
- Generating C code. A stand alone application in ANCI C code is generated for the controller, based on the binary search tree.

Applying eMPC - an example

By means of visualizing the theory and procedure discussed above, a simply double integrator is considered. The exactly same approach is applicable for systems of higher order though, but visualization becomes harder in these cases. The eMPC controller will as mentioned earlier be generated using the mpQP toolbox by Tøndel and Johansen.

Consider the continuous-time double integrator (Johansen et al., 2000) $\dot{\mathbf{x}} = \mathbf{A}_c \mathbf{x} + \mathbf{B}_c u$, where

$$\mathbf{A}_{c} = \begin{bmatrix} 0 & 1\\ 0 & 0 \end{bmatrix}, \quad \mathbf{B}_{c} = \begin{bmatrix} 0\\ 1 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_{1}\\ x_{2} \end{bmatrix}$$
(5.18)

The continuous-time system in (5.18) can easily be converted into an equivalent discrete form by utilizing the MATLAB function c2d(sysc, T,method). For the remainder, the method used for discretizing the continuous-time system is a triangle approximation (modified firstorder hold), and control inputs are assumed piecewise linear over the sampling period T.

If x_1 is interpreted as position, x_2 as speed and u as force, the objective is to control the position under constraints on the speed and force, that is

If we chose the sampling time T equal to 0.05 and do not include any slack variables, the MPC problem in (5.14), over the horizon N = M = 2, with cost matrices

$$\mathbf{R} = 1, \quad \mathbf{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

is associated with the mpQP problem in (5.16) with

Some internal scaling is done within the mpQP toolbox, but this is not relevant in sense of understanding the procedure. Note that the matrices above are different from the scaled matrices used when doing the numerics. However, after finding the solutions, they are re-scaled such that they are representative for the original problem, as stated above.

The solution to the mpQP is shown in Figure 5.3, for different horizons. As can be seen, the complexity of the solution space increases with the optimization horizon. As discussed earlier though, the horizon is just one single factor that needs to be taken into consideration.



Figure 5.3: Regions in parameter space for the double integrator, using hard constraints

48 Attitude control and simulations

For the remainder of the discussion on the double integrator we use N = 2.

The explicit MPC controller that was computed above can be connected to the plant, establishing the closed-loop system. As we remember, the parameters used for determining which optimal control gain to use are equal to the state variables. By letting the initial conditions for the double integrator be $\mathbf{x}_0 = [-2, 0]^T$, we get the response shown in Figure 5.4.



(a) Simulation in state space

(b) Simulation in parameter space

Figure 5.4: Set point control of the double integrator, N = 2

We can investigate the solution space further, by looking at the data for each region, e.g.

```
>> region{1}
```

```
poly: [6x3 double]
    feas: 1
 actset: [0 0 0 0 0 0 0 0]
solution: [2x3 double]
lagrange: [0x3 double]
 chebyx: [2x1 double]
 chebyr: 3.0473
```

As can be seen from region {1}.actset there are no active constraints for this region. By looking at Figure 5.4(b) this makes sense, as the solution is closely related to the stationary discrete-time LQR. In fact, by investigating region $\{1\}$. solution further we find that the optimal control law in this region is identical to the stationary discrete-time LQR gain, -K.

As mentioned in the beginning, only the first input is applied to the plant. In the general case, the input corresponding to the first time step is given as $\mathbf{u}^*(k) = \mathbf{K}_r \mathbf{x}(k) + k_r$, where the index r indicates the active region. Clearly, the latter represents an affine function. As region $\{r\}$. solution holds \mathbf{K}_r and k_r for every time step on the horizon, the input above corresponds to the first time step. Omitting details, the complete input sequence, and solution to (5.15) is given as $\mathbf{U}^* = [(\mathbf{u}^*(k))^T, (\mathbf{u}^*(k+1))^T, \dots, (\mathbf{u}^*(k+N-1))^T]^T$.

We end this section by discussing the use of slack variables. As before the results in the following, using the double integrator as an example, are applicable for systems of higher order, and the double integrator is only used by means of visualizing the procedure.

Based on the theory discussed sofar, the MPC formulation in (5.14) can be rewritten to a new MPC problem, also including slack variables, that is

 $\min_{\mathbf{U},\mathbf{s}} \left\{ J(\mathbf{U}, \mathbf{x}(k)) + \boldsymbol{\rho} \| \mathbf{s} \|_{2}^{2} \right\}$ subject to: $\mathbf{y}_{\min} - \mathbf{s} \leq \mathbf{y}(k+i|k) \leq \mathbf{y}_{\max} + \mathbf{s}, i = 1, \dots, N$ $\mathbf{u}_{\min} \leq \mathbf{u}(k+i) \leq \mathbf{u}_{\max}, i = 1, \dots, M-1$ $\mathbf{x}(k|k) = \mathbf{x}(k)$ $\mathbf{x}(k+i+1|k) = \mathbf{A}\mathbf{x}(k+i|k) + \mathbf{B}\mathbf{u}(k+i), k \geq 0$ $\mathbf{y}(k+i) = \mathbf{C}\mathbf{x}(k+i|k), k \geq 0$ $\mathbf{u}(k+i) = \mathbf{K}\mathbf{x}(k+i|k), M \leq k \leq N-1$ (5.20)

where $\|\mathbf{s}\|_2$ is the \mathcal{L}_2 -norm of \mathbf{s} and $\boldsymbol{\rho}$ is the penalty weight of the slack variables. We note that using the \mathcal{L}_2 -norm is only one way of including slack variables. As mentioned earlier, $\boldsymbol{\rho}$ is chosen such that $\|\mathbf{s}\|_2$ is kept as small as possible. The solution to (5.15) is now given as $\mathbf{U}^* = [(\mathbf{u}^*(k))^T, (\mathbf{u}^*(k+1))^T, \dots, (\mathbf{u}^*(k+N-1))^T]^T, \mathbf{s}^* = [\mathbf{s}^T(k), \dots, \mathbf{s}^T(k+N-1)]^T$.

Applying eMPC - an example, continued

When we defined the eMPC problem for the double integrator the aim was to find optimal solutions for the parameter space $|x_1| \leq 3$ and $|x_2| \leq 3$. As can be seen from Figure 5.5(a) however, we did not obtain this goal in the case of using hard constraints. More precisely, the mpQP solver did succeed in solving the problem, but without including any slack variables, it turned out the be infeasible for part of the parameter space. The way to interpret this is that there exists no actuation, within its limit, such that x_2 is rendered within the chosen constraint for every time step, during the defined horizon. As discussed earlier this can in some cases lead to infeasibility. This can for instance be the case if the initial conditions are outside the feasible parameter space, if noise causes the output to go outside the solution space in the next time step, or if there are serious model uncertainties. Obviously this needs to be dealt with in real applications, and as mention above one way of doing this is to include slack variables. The direct consequence of this can be seen in 5.5(b), where when including softening techniques, the mpQP solver found feasible solutions for the whole parameter space.

Another interesting way to illustrate the effect of introducing slack variables can be seen in Figure 5.6. From the case when hard constraints are applied, we see that no input exists for $|x_2| \gtrsim 0.55$, while inputs exist for the desired parameter space in case of soft constraints. Also, we see that the input saturates, or is kept within the constraints at ± 1 , as defined in the problem.

Plotting an input as function of two states can of course be a reasonable way to represent the optimal control, wether we use slack variables or not.



Figure 5.5: Regions in parameter space for the double integrator, using hard and soft constraints



Figure 5.6: Input as function of states for the double integrator, using hard and soft constraints

5.4 Signal modulation

As mentioned in the introduction, it is important to keep in mind that the thrusters are on-off by nature. A fixed integer controller could have been useful in this case. However, since this approach is not considered in this thesis, and the fact that input sequences calculated from a wide range of controllers are continuous, this leads to problems in terms of implementation. An the other hand, sometimes a system utilizes both on-off actuators as well as continuous ones. Therefore, an attempt to solve this problem is done by keeping the continuous input to the wheel unchanged, while using a preliminary bang-bang approach for the thrusters. We note that it would also be interesting to investigate the use of Pulse-Width Pulse-Frequency (PWPF) modulation, since this is a more sophisticated technique than the simple bang-bang scheme. It has also in some cases been shown to be a more efficient approach in sense of fuel consumption. When we now do not pursue this any further this is due to time constraints, and we end the discussion by referring to the work on PWPF mentioned in the introduction. In its most simple form, the bang-bang control consists of a signum module, as shown in Figure 5.7(a). If we denote the continuous input u, we get a modified signal according to

$$u_* : \operatorname{signum}(u) = \begin{cases} -1 & \text{if } u < 0, \\ 0 & \text{if } u = 0, \\ 1 & \text{if } u > 0. \end{cases}$$
(5.21)

In sense of having a less aggressive actuation we can include a dead-zone together with the the signum function. The result of doing this is seen in Figure 5.7(b). The consequence is usually less accuracy, but more importantly, fuel usage is heavily reduced. Obviously, the accuracy can be adjusted by choosing an appropriate dead-zone.

Either we include a dead-zone or not, the modified signal is an integer, according to (5.21). As can be seen in Figure 5.7, we have also included a gain K_{nom} , which is simply the nominal thruster torques for SSETI/ESEO satellite, as described in Table 1.8.



Figure 5.7: Signal modulation

5.5 Simulations of the SSETI/ESEO satellite

We again turn our attention to the SSETI/ESEO satellite. The previous sections were included with purpose of explaining and visualizing some control techniques which are all applicable for our system. Since the focus in this thesis is on eMPC, the other schemes are only included by means of comparing the different methods and their performances.

A thorough discussion on the different results is given in the end of this chapter.

Hardware and software

The simulations to follow were all carried out on a Windows XP platform, using MATLAB version 6.5. As mentioned earlier, the toolbox by Tøndel and Johansen was used throughout. Note that instead of using the optimization toolbox in MATLAB, the Tomlab Optimization toolbox was utilized to solve LP/QP subproblems within the mpQP software. As far as hardware goes, the computer was equipped with a 2GHz Pentium 4 CPU and 512Mb RAM.

SSETI/ESEO - a recap

For clarity, a short recap on the satellite model that will be used when deriving the eMPC controller will be given. All the control schemes we will use in the following are linear control techniques, in contrast to our nonlinear model. As will be shown later though, the results obtained when simulating the closed-loop system are by all means satisfactory. The nonlinear model was derived in Chapter 4, together with its linearized version in (4.6). The linear model was then reduced from 8 to 7 states, forming the new state and input vectors $\mathbf{x} = [\omega_1, \omega_2, \omega_3, \omega_s, \epsilon_1, \epsilon_2, \epsilon_3]^{\mathrm{T}}$, $\mathbf{u} = [\tau_1, \tau_2, \tau_3, \tau_a]^{\mathrm{T}}$, where the terms are summarized as

	States		Input
ω_1	Angular velocity about the x-axis	$ au_1$	Thruster torque about the x-axis
	in \mathcal{F}_b relative to \mathcal{F}_o , [rad/s]		in \mathcal{F}_b , [Nm]
ω_2	Angular velocity about the y-axis	$ au_2$	Thruster torque about the y-axis
	in \mathcal{F}_b relative to \mathcal{F}_o , [rad/s]		in \mathcal{F}_b , [Nm]
ω_3	Angular velocity about the z-axis	$ au_3$	Thruster torque about the z-axis
	in \mathcal{F}_b relative to \mathcal{F}_o , [rad/s]		in \mathcal{F}_b , [Nm]
ω_s	Angular velocity of the wheel	$ au_a$	Torque applied on wheel from
	about its axial axis [rad/s]		DC motor, [Nm]
$\epsilon_1, \epsilon_2, \epsilon_3$	Euler parameters (3 out of 4)		

As before it is assumed that the output matrix C is equal to the identity matrix. Without any further explanation we claim that the states in the linear system vary in magnitude with a factor 10000, e.g. the wheel's angular velocity can be 400 [rad/s], while the angular velocity about the y-axis can be 0.04 [rad/s]. Due to numerical sensitivity in the mpQP solver, we therefore choose to scale the states, using a similarity transformation. We prefer to normalize the system using a diagonal scaling matrix N_x , where the nonzero elements are chosen based on their maximum deviation, that is

$$\mathbf{N}_{x} = \operatorname{diag}\left\{\max|\omega_{1}|, \max|\omega_{2}|, \max|\omega_{3}|, \max|\omega_{s}|, \max|\epsilon_{1}|, \max|\epsilon_{2}|, \max|\epsilon_{3}|\right\}$$
(5.22)

For the remainder we assume that $\max |\omega_1| = \max |\omega_2| = \max |\omega_3| = 0.04$ [rad/s], $\max |\omega_s| = 527$ [rad/s], and $\max |\epsilon_1| = \max |\epsilon_2| = \max |\epsilon_3| = 1$.

By defining $\mathbf{x}_n \triangleq \mathbf{N}_x^{-1}\mathbf{x}$, the normalized system is given as

$$\dot{\mathbf{x}}_n = \mathbf{A}_n \mathbf{x}_n + \mathbf{B}_n \mathbf{u} \tag{5.23}$$

where $\mathbf{A}_n = \mathbf{N}_x^{-1} \mathbf{A} \mathbf{N}_x$ and $\mathbf{B}_n = \mathbf{N}_x^{-1} \mathbf{B}$. The eigenvalues for \mathbf{A}_n are equal to those of \mathbf{A} .

As with the double integrator example, the continuous-time linear (normalized) model is converted into an equivalent discrete-time form by utilizing a modified first-order hold routine in MATLAB. The sampling time for the discretization is chosen as T = 0.1 [sec].

We finally note that the discretized model is only applied when designing the eMPC controller. When applying LQR, the stationary gain is found by utilizing the continuous-time model, though with the same weight matrices as in the discrete-time case.

5.5.1 Controller settings

Unless otherwise stated the controllers are designed according to Table 5.2. As will be shown later, the only parameter to be adjusted is the slack variable weight ρ . Note that the controllers were all based on either the *normalized* discrete-time linear model or its continuous-time version. They are all valid for the original linear model though, since the similarity transformation only changes the scaling of the states, and not the behavior of the model.

	Controller			
Parameter	eMPC	PD		
Q	diag $\{50, 50, 50, 1,$	-		
R	$diag\{200, 300$	-		
\mathbf{K}_p	-		$diag\{10, 10, 10\}$	
\mathbf{K}_d	-		$diag\{7, 7, 7\}$	
N (horizon)	2	∞	-	
ρ (slack)	∞	-	-	

Table 5.2: Summary of tuning parameters

When designing the eMPC controller, the parameter space, in which we search for feasible solutions of the mpQP, is chosen as $[-3, -3, -3, -3, -3, -3, -3]^T \leq \mathbf{x}_n \leq [3, 3, 3, 3, 3, 3, 3]^T$. Furthermore, when applicable, the constraints that will be used for the remainder of this thesis are $[-0.0484, -0.0484, -0.0398, -0.002]^T \leq \mathbf{u} \leq [0.0484, 0.0484, 0.0398, 0.002]^T$, and $-527 \leq \omega_s \leq 527$. A summary of states and units for the linear model was given above. Note that by means of visualizing the results, the Euler parameters will be transformed into Euler angles [deg], while the angular velocity of the body will be converted into [deg/s]. As mentioned earlier, torque allocation for the PD-control is done using the dead-zone approach, i.e. torque is applied to the wheel whenever the commanded torque about the y-axis is within some limits. The chosen limits are similar to the constraints on τ_a , when using eMPC.

5.5.2 eMPC compared to LQR and PD-control

The first simulations are done with purpose of comparing the performance of the different control schemes. All simulations are carried out on the nonlinear model, on an average orbit according to (5.1), and with the SSETI/ESEO parameters and initial orbit conditions described in Section 5.1 and 5.2, respectively. Initial conditions, and set-points for the simulations included in this section are summarized in Table 5.3.

Case 1	Initial condition	Set-point	Unit
ω^b_{ob}	$\{0, 0, 0\}$	$\{0, 0, 0\}$	deg/s
ω_s	0	0	rad/s
Euler angles (XYZ)	$\{60, -45, 30\}$	$\{0, 0, 0\}$	deg
Case 2	Initial condition	Set-point	Unit
$oldsymbol{\omega}^b_{ob}$	$\{0, 0, 0\}$	$\{0, 0, 0\}$	deg/s
ω_s	0	0	rad/s
Euler angles (XYZ)	$\{60, -45, 30\}$	$\{0, 0, 90\}$	deg
Case 3	Initial condition	Set-point	Unit
ω^b_{ob}	$\{1, -1, 1\}$	$\{0, 0, 0\}$	deg/s
ω_s	-500	0	rad/s
Euler angles (XYZ)	$\{60, -45, 30\}$	$\{0, 0, 0\}$	deg

Table 5.3: Summary of simulations

54 Attitude control and simulations

Simulations of the three cases summarized in Table 5.3 are given subsequently, and in the same order as they appear above. Through Case 1 and 2 we investigate the performance of the respective controllers for two different set-points. In both cases we assume no spin on neither the wheel nor the satellite body. In Case 3 we want to render all states to the origin, but in this case we have an initial spin for both wheel and body. This allows us to illustrate the efficiency and effect of including a hard constraint on a state in the eMPC formulation. Finally, in the modified case at the end, we have the same aim as in Case 3, but now we apply saturation on the actuation for both the PD-control and the LQR. The saturation values are similar to the input constraints for the eMPC.



Case 1: set-points at the origin, no spin

Figure 5.8: Case 1



Figure 5.8: Case 1, continued



Case 2: set-points different from the origin, no spin

(i) Wheel velocity: eMPC

Figure 5.9: Case 2



Figure 5.9: Case 2, continued



Case 3: set-points at the origin, spin, and hard constraints

Figure 5.10: Case 3



Figure 5.10: Case 3, continued



Case 3 (modified): including actuator saturation on PD and LQR

Figure 5.11: Case 3 (modified)

5.5.3 eMPC when exposed to extreme initial conditions

As mentioned earlier the mpQP solver may result in infeasible solutions, hence making the eMPC controller unable to find an applicable input. This can for instance be the case if the initial conditions are outside the feasible parameter space.

In the following, simulations are done to show the effects of including slack variables in the eMPC formulation. Only simulations of the eMPC approach will be included at this point. As before, simulations are carried out on the nonlinear model and with the SSETI/ESEO parameters and initial orbit conditions described in Section 5.1 and 5.2, respectively. Initial conditions, and set-points for the simulations included in this section are summarized in Table 5.4.

Case 4	Initial condition	Set-point	Unit
ω^b_{ob}	$\{-2, 2, 2\}$	$\{0, 0, 0\}$	deg/s
ω_s	1250	0	rad/s
Euler angles (XYZ)	$\{60, -45, 30\}$	$\{0, 0, 0\}$	deg
Case 5	Initial condition	Set-point	Unit
$oldsymbol{\omega}^b_{ob}$	$\{1, -1, 1\}$	$\{0, 0, 0\}$	deg/s
ω_s	-500	0	rad/s
Euler angles (XYZ)	$\{60, -45, 30\}$	$\{0, 0, 0\}$	deg

	Table 5.4:	Summary	of	simu	lations
--	------------	---------	----	------	---------

Through Case 4 we show how slack variables allow one to have extreme initial conditions, while Case 5 highlights the effects of penalizing the slack variables differently.

The eMPC tuning parameters for both cases are taken from Table 5.2, except for using either $\rho = 3000$ or $\rho = 500$ when including slack variables. We note that we have only included slack variables associated to the state ω_s , since this is the only state suffering from constraints. Since we have an optimization horizon N of two time-steps, we get two additional *free variables*, one for each step. Furthermore, we assume that the actuator constraints are hard constraints, while the velocity constraints of the wheel can be violated for a short period. This is a realistic assumption as the maximum allowed velocity is chosen based on energy consumption, and not on a physical limitation.

As was shown in the example using the double integrator we can visualize the polyhedra regions in the parameter space. Obviously, it is impossible to do this for more than three dimensions, and in fact it is hard to visualize the solution for more than two. If we do not regard the order, we know from basic mathematics that from our linear SSETI/ESEO state vector we can plot $\binom{7}{2} = 21$ different phase-planes, e.g. the $\omega_2 \omega_s$ -plane. As we do not have room for all of these, only a selection will be included here. When plotting one input as a function of two states this can be done in $\binom{7}{2}\binom{4}{1} = 84$ different ways. Different offsets can also be included in both, increasing the number of combinations arbitrarily.

Similar to the cases simulated earlier, the eMPC controller without slack variables is designed based on Section 5.5.1. A selection of the polyhedra regions can be seen in Figure 5.12. In the case of not including slack variables in the SSETI/ESEO eMPC controller, the mpQP solver

returned infeasible solutions for part of the desired solution space. Furthermore, as can be seen from Figure 5.12(c) and 5.12(d), only the phase-planes including ω_s show infeasible solutions in the parameter space. This is intuitive, since ω_s suffers from hard constraints.



Figure 5.12: Polyhedra regions for the SSETI/ESEO satellite, N=2 and $\rho = \infty$

The total number of regions in the parameter space when not including slack variables was 1205. However, we immediately see from Figure 5.12(c) and 5.12(d) that the eMPC controller without slack variables is not robust in terms of extreme initial conditions. If we let $\rho = 3000$ we obtain the robustness we are seeking, and since we use a relatively high penalty, potential violation of the constraints will remain low. As can be seen from Figure 5.13, the new controller is rather robust, in terms of disturbances and large start conditions. The total number of regions when using $\rho = 3000$ was 1725. It is emphasized that since we normalized the system, the values in the parameter space are different from the values of the original state vector. As mentioned earlier however, the results are valid for the original system as well. For instance, in Figure 5.13(a), $[\omega_2, \omega_s]_{normalized} = [1, 2] \Leftrightarrow [\omega_2, \omega_s]_{original} = [0.04, 1054]$. Recall that the constraints on ω_s in the original system was ± 527 [rad/s].

Similar to the double integrator, the effect of the slack variables can also easily be visualized by plotting an input as a function of two states. As can be seen from Figure 5.14(a), there do not exist inputs for the whole parameter space in the case of using hard constraints. The opposite is true, in the case of including softening techniques.


Figure 5.13: Polyhedra regions for the SSETI/ESEO satellite, N = 2 and $\rho = 3000$



Figure 5.14: Input as function of states for the SSETI/ESEO, using hard and soft constraints

Case 4: extreme initial conditions, $\rho = 3000$



Figure 5.15: Case 4



Figure 5.15: Case 4, continued





Figure 5.16: Case 5

5.5.4 eMPC in combination with bang-bang modulation

As mentioned in the introduction, it is important to keep in mind that the thrusters are restricted to on-off mode only. As will be shown shortly, one way of dealing with this problem is to use a bang-bang approach. This is only meant to be a preliminary design, and further analysis should be done. However, it is considered appropriate to include these simulations to illustrate how to apply a modulation scheme. The following simulations are done with a varying distance to the Earth, i.e. elliptical orbit. Initial conditions are equal to *Case 1*, as described earlier, and with controller parameters as in Table 5.2, except for having $\rho = 3000$. The first case to be studied is the bang-bang scheme in Figure 5.7(a). Afterwards is it shown that in sense of having a less aggressive actuation, we can include a dead-zone together with the the signum function. The latter is simulated for a duration of more than two full orbits to show efficiency of the controller, at least in sense of overcoming time-varying gravity gradient torques. Also to be illustrated, the weighting matrix \mathbf{Q} has a huge impact on the accuracy when the controller is implemented together with the modulation block.



Figure 5.17: Bang-bang without dead-zone

Obviously, as can be seen from the actuation in Figure 5.17(d), the inputs are not under any circumstances satisfactory. In Figure 5.18, a similar case is simulated, but now including the

dead-zone, as in Figure 5.7(b). The output from the bang-bang is in this case defined as

$$\tau_{1*} = \begin{cases} 0 & \text{if } |\tau_1| < 0.0035 \\ 1 & \text{if } \tau_1 \ge 0.0035 \\ -1 & \text{if } \tau_1 \le 0.0035 \end{cases}, \ \tau_{2*} = \begin{cases} 0 & \text{if } |\tau_1| < 0.001 \\ 1 & \text{if } \tau_1 \ge 0.001 \\ -1 & \text{if } \tau_1 \le 0.001 \end{cases}, \ \tau_{3*} = \begin{cases} 0 & \text{if } |\tau_1| < 0.0035 \\ 1 & \text{if } \tau_1 \ge 0.0035 \\ -1 & \text{if } \tau_1 \le 0.0035 \end{cases}$$

The resulting actuation from this change is clearly seen in Figure 5.18(e).



Figure 5.18: Bang-bang with dead-zone, simulated over two orbits

The final plots in Figure 5.19 show the response if we chose to weight the states differently. The controller is based on Table 5.2, except for having $\mathbf{R} = \text{diag}(50, 50, 50, 0.001, 100, 100, 100)$. In the latter we even changed to dead-zone slightly, to allow more actuation. Nevertheless, as can be seen from comparing Figure 5.19(c) and 5.18(d), the performance is different in sense of keeping the wheel velocity close to zero. In case of decreasing the weight of the state ω_s , we care less about its transient. We can interpret this as if the controller is more interested in driving the other states to the origin, and when this aim is achieved it is too costly to drive the wheel to zero as well, i.e. the other states are weighted much higher than the angular velocity of the wheel. This will discussed further in the next section, together with other design issues.



Figure 5.19: Bang-bang with dead-zone, different weight matrix

5.5.5 Discussion

Throughout this thesis focus has been on attitude control and the motion about the center of mass of a satellite. Part of the contents has been on different ways of modelling the spacecraft as a rigid body, as well as giving an overview of potential disturbances and various actuators. The equations describing the dynamics were derived based on the Newton-Euler formulation, making up the equations in (3.19). The reader may notice that these differ from the classical equations in (3.11). The reason for choosing the first alternative is that they were considered more intuitive and easier to use as a basis for the SSETI/ESEO structure. In (3.11) all the torques and the wheel angular momentum is generalized, which makes it necessary to calculate the net angular momentum due to the rotation of the wheels relative to the body, as well as the net torque applied to the wheels from the spacecraft. In case of only having one wheel this may not be a huge concern, but in general systems including thrusters and momentum exchange devices it may turn out to be a cumbersome task. Also, in this thesis it has been an aim to investigate an alternative approach as the standard formulation is already well understood. As far as describing the kinematics, it was chosen to use Euler parameters due to their nonsingular characteristics. The complete nonlinear model has been implemented in MATLAB and Simulink with purpose of doing various simulations.

The controllers that were utilized are all linear, hence making it necessary to linearize the nonlinear model about an equilibrium point. As described in the introduction the are many attitude mission modes. However, as the nominal mode is the most common, it was the only one considered in this thesis. Recall that it includes the task of maintaining a stable attitude, i.e. keeping the satellite nadir pointing. More specifically this means that the best obtainable

result is whenever \mathcal{F}_b coincides with \mathcal{F}_o , which resembles the equilibrium point at the origin. In some cases it may be reasonable to linearize about several equilibrium points, and then use a kind of scheduling between different modes. This was not considered here however, and the model was linearized at the origin only. Obviously, the linearized model was only used when deriving the controllers, and during final simulations the nonlinear plant was utilized.

As described, a similarity transformation was used for normalizing the linear model. Intuitively this may seem unreasonable, but due to numerics in the mpQP solver it turned out to be necessary. The obvious drawback from dealing with a scaled system is that there is no simple way of including noise in the model. A small contribution in the original system may become huge when scaled, or visa versa. This can easily be seen from Figure 5.20, were the noise was realized as uniform white noise, sampled at 10 Hz. It was also assumed that RMS estimation errors obtained from for instance an Ext. Kalman filter were given as in Table 5.5. Clearly, the undesired effect of noise will be higher in the the scaled case than for the unscaled.



Figure 5.20: Unscaled and scaled feedback for a random simulation

As mentioned earlier though, the transient response of both systems remains similar, and investigating one of them gives an indication of what to expect from the other. The results achieved in this thesis are therefore likely to be applicable for the original system as well. In fact, simulations were done based on the original (unscaled) system, and in the example given at the end, the results obtained indicate a desired behavior in closed-loop.

Estimates	Errors	Unit
$oldsymbol{\omega}_{ob}^b$	$\{0.2, 0.3, 0.2\}$	deg/s
ω_s	0.5	rad/s
Euler angles (XYZ)	$\{0.1, 0.1, 0.1\}$	deg

Table 5.5: RMS errors in state estimates

The results obtained through simulations show that there exists a range of regulator schemes suitable for controlling a satellite. Similar for the ones discussed here is that they are all linear techniques, and as both the LQR and PD-controller are well known, they were natural choices to use as references for the eMPC. As was shown through *Case 1-3*, the response of the schemes varied slightly. In terms of orienting the satellite in a specified direction, according to the quaternions, the performance did not differ. As is evident though, the major difference between the LQR, eMPC and the PD-controller is that in the latter no information about the

wheel is utilized, hence making it impossible to control. In Figure 5.9(g) for instance, the angular velocity of the wheel is way pass the physical saturation. Obviously, the stationary value of the wheel velocity is arbitrary in this case. For the LQR and eMPC the situation is different, as they both render the angular velocity of the wheel to zero. Not surprisingly, the eMPC was the only one keeping it within the desired limits, as evident in Figure 5.10(j). Another difference between the schemes is how the actuation occurs. Both the PD-controller and the LQR apply high torques initially, as apposed to the eMPC where the constraints on the actuation in most cases are active at the beginning. Physically this is desirable, as the high torques of the other schemes cannot be realized on a micro-satellite. Physical saturation on the actuation on both the PD-controller and the LQR was simulated in the modified version of Case 3. The actuation for the PD-controller became smoother in this case, but it was oscillating heavily. The reason for the oscillation, which in fact was even more severe without the saturation, is most likely due to the wheel, as it for the PD-controller only contributes as a passive damper. The LQR with saturation on the other hand gave an almost similar response to the eMPC. However, the torque applied to the wheel was actuating heavily, due to the low weighting in the R matrix, i.e. the wheel is almost applied without any cost.

As mentioned earlier, the ability to include input and state constraints directly in the problem formulation, so that future constraint violations are anticipated and prevented, is probably the most important reason for the success of MPC. Being able to solve the MPC problem explicit makes it even more applicable. In sense of comparing PD-control, LQR and eMPC, applied for attitude control, it is difficult to say which is the better as design issues have to be taken into consideration as well. This will be discussed shortly. However, from the results obtained in this thesis, the eMPC approach has shown to be an alternative that should be considered if constraints need to be taken into account. Also, as shown, the actuation was smooth, as apposed to the oscillating ones of the PD-controller and the LQR.

Further analysis was also done on the eMPC controller to highlight the effect of including slack variables in the problem formulation. In systems to be implemented in real life, this should be done, as it would be unacceptable to have a situation where no input exists. Depending on how the slack variables are penalized, i.e. how expensive it is to violate any of the constraints, the performance may differ. This can easily be seen in Figure 5.16(b). Also as shown, when including slack variables, the controller can allow extreme initial conditions. In the general the introduction of slack variables also increase the robustness in terms of measurement noise. As mentioned above, the latter is not applicable for a scaled system.

When it comes to general design issues, the first thing to sort out is wether it is desired or necessary to have the wheel converge to zero angular velocity. As it was discussed above, both the LQR and eMPC includes this functionality directly. The cost in fuel consumption for doing this was not investigated in this thesis though, but in the end, the amount of fuel used for doing this operation should be lower than fuel usage during detumbling. The latter is necessary whenever the wheel reaches saturation, and no more angular momentum can be provided. The second major design problem is to chose the weighting matrices in the LQR and eMPC controllers, since the optimality of the state feedback control is measured in sense of the performance index. Whether it defines a good system in any engineering sense depends on the choice of \mathbf{Q} and \mathbf{R} . Obviously, no universal solution exists for this problem, and usually it is an iterative procedure, as it was here. The weight matrices were equal for the LQR and the

eMPC. Additional choices need to be taken for the eMPC controller as well. If the constraints are too strict there may not be a solution to the problem, or if the prediction horizon is too long then the solution may become too complex. As for the weight matrices, the procedure of finding appropriate eMPC parameters is highly iterative. In the end the gains for the PD-controller were chosen sought to match the performance of the other two schemes.

The final issue to be discussed is the input modulation. As repeated earlier, the thrusters are onoff only. As was shown in Figure 5.18, using a bang-bang modulation scheme with dead-zone gave good results. Furthermore, the accuracy, and consequently the fuel usage can be adjusted by varying the limits of the dead-zone. However, in many cases it would be useful to have a Pulse-Width Pulse-Frequency (PWPF) modulation instead. This would especially be desirable when the input signals are heavily oscillating or if contaminated with noise. Without going into details, the PWPF consists of a *Schmitt-trigger*, which is similar to the bang-bang with dead-zone, but also implementing hysteresis. A feedback and a lead-lag filter is also utilized, allowing both frequency and width of the modulated pulses to change. Due to time constraints this topic was unfortunately not considered in this thesis.

Utilizing unscaled model for deriving eMPC

As mentioned above, most of the work done in thesis was based on a scaled model. In the following, it is shown that the results are applicable for the unscaled system as well. The horizon and constraints were similar to the ones used for the scaled system, while choosing $\mathbf{Q} = \text{diag}(200, 200, 200, 5 \cdot 10^{-7}, 1, 1, 1)$, $\mathbf{R} = \text{diag}(100, 200, 100, 1)$, and $\boldsymbol{\rho} = 8 \cdot 10^{-5}$. For the remainder measurement noise is included according to Table 5.5, and the orbit is assumed elliptical, hence causing additional disturbance. Initial conditions are equal to *Case 3* in Table 5.3. Note that further work has been carried out on the mpQP toolbox. However, due to time constraints, a thorough analysis of the outcome was not possible, and additional simulations to the ones in Figure 5.22 and 5.23 were therefore omitted in this thesis.

The parameter space, $[-1, -1, -1, -1500, -1, -1, -1]^T \leq x \leq [1, 1, 1, 1500, 1, 1, 1]^T$, was chosen to cover any physical values. As in the scaled case, a selection of the polyhedra regions can be seen in Figure 5.21.



Figure 5.21: Polyhedra regions for unscaled model, N=2 and $\rho = 8 \cdot 10^{-5}$



eMPC based on unscaled model, no input modulation

Figure 5.22: Case 3, unscaled model, no modulation



eMPC based on unscaled model, bang-bang with dead-zone

Figure 5.23: Case 3, unscaled model, bang-bang modulation with dead-zone

Chapter 6 Conclusions

A detailed and general model for a satellite has been derived, including topics related to disturbances and different actuators. In sense of representing the kinematics, the Euler parameters were chosen due their nonsingular characteristics. As for the dynamics, these were described by an alternative version of the Newton-Euler formulation.

After taking numerous simplifications into account, several properties of the satellite model have been presented. Among others, the most important being the fact that the model was uncontrollable, yet stabilizable. Also, due to the properties of the Euler parameters it is possible to keep track of and update the varying η , which otherwise would have remained constant.

In systems suffering from limited power supply, data storage, and computational resources, a reasonable approach is to evaluate a controller off-line, and consequently reducing the need for CPU speed drastically as real-time effort can be restricted to a table-lookup. Also, in many circumstances it may be desirable to prevent the different components from operating near their thresholds, either the design focus is to keep power consumption within some limits or to keep the rate of wear as low as possible. It has been illustrated that all these concerns can be dealt with by formulating a Model Predictive Control problem (MPC).

It has been shown that explicit solutions to constrained linear MPC problems can be computed by solving multi-parametric quadratic programs (mpQP), where the parameters are the components of the state vector. The solution to the mpQP, which is a piecewise affine (PWA) function, can be evaluated at each sample to obtain the optimal control law.

Using special made software, an explicit MPC (eMPC) controller has been derived for the SSETI/ESEO micro-satellite project, initiated by the European Space Agency (esa). Due to numerical sensitivity, most of the work was done on a normalized system. It was shown however, that the results were applicable for the original (unscaled) system as well, making it a very interesting control scheme for attitude control. The eMPC approach has shown to be an highly potential alternative to PD-control and LQR, and it should be considered if constraints need to be taken into account.

An important thing to keep in mind is that the thrusters on the satellite are on-off by nature. A preliminary solution to this problem was established using a bang-bang modulation scheme.

6.1 Further work

Even if the results in this thesis show that an explicit MPC controller can be applicable for attitude control of a satellite, there is definitely more to do on the subject. The first natural steps would be as follows;

- Continue the studies on an unscaled system with purpose of verifying the results obtained in this thesis. If necessary, work should be done on the mpQP solver to improve numerical robustness.
- A stability proof was not included in this thesis. Several modifications to the MPC problem have been suggested in the literature. However, as the solution to the mpQP problem is a piecewise affine function, a potential approach is to search for piecewise quadratic Lyapunov functions by solving a convex optimization problem using linear matrix inequalities (LMIs). The idea is to find a valid Lyapunov function for each region in the parameter space, separately yielding asymptotic stability.
- The weight matrices **R** and **Q** were picked using an iterative procedure. Based on a well defined mission mode, seek to find suitable costs matrices.
- As with the weight matrices, the tuning parameters for the MPC problem were picked for performance. Further studies should be done on the effects of increasing the prediction horizon, and of using a different sampling frequency when discretizing the nonlinear continuous-time model.
- Do a detailed study on fuel consumption for the thrusters with aim of defining suitable mission priorities. This could for instance be wether it is desired to render the angular velocity of the wheel to zero during the transient, or if subsequent detumbling should be applied instead.
- The problem of input modulation is still considered unsolved. As an alternative to bangbang modulation, utilize Pulse-Width Pulse Frequency (PWPF) modulation instead.

Bibliography

- Bemporad, A., Morari, M., Dua, V. and Pistikopoulos, E. N. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*, (38):pp. 3–20.
- Bryson, A. and Ho, Y.-C. (1975). Applied Optimal Control. John Wiley and Sons.
- Cavallo, A. and Maria, G. D. (1996). Attitude control for large angle maneuvers. pp. 232–237. IEEE International Workshop.
- Chen, C.-T. (1999). Linear System Theory and Design. Oxford University Press, third ed.
- Chen, Y.-P. and Lo, S.-C. (1993). Sliding-mode controller design for spacecraft attitude tracking maneuvers. *IEEE Transactions on Aerospace and Electronic Systems*, 29(4):pp. 1328– 1333.
- Conway, J. B. (1978). Functions of One Complex Variable. Springer Verlag, New York.
- Crouch, P. (1984). Spacecraft attitude control and stabilization: Applications of geometric control theory to rigid body models. *IEEE Transactions on Automatic Control*, 29(4):pp. 321–331.
- Egeland, O. and Gravdahl, J. T. (2002). *Modeling and Simulation for Automatic Control*. Marine Cybernetics, Trondheim, Norway.
- Ferrari-Trecate, G., Cuzzola, F. A., Mignore, D. and Morari, M. (2001). Analysis and control with performance of piecewise affine and hybrid systems. American Control Conferance.
- Fjellstad, O.-E. and Fossen, T. I. (1994). Comments on "The attitude control problem". *IEEE Transactions on Automatic Control*, 39(3):pp. 699–700.
- Fossen, T. I. (2002). Marine Control Systems Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles. Marine Cybernetics, Trondheim, Norway.
- Hall, C. D. (1995). Spinup Dynamics of Gyrostats. Journal of Guidance, Control, and Dynamics, 18(5):pp. 1177–1183.
- Hall, C. D. (1997). High speed flywheels for integrated energy storage and attitude control. vol. 3, pp. 1894–1898. American Control Conference, Albuquerque, NM.
- Hall, C. D. (2000). Attitude Dynamics of Orbiting Gyrostats. European Celestial Mechanics Workshop, Adam Mickiewicz University in Poznan, Poland.

- Hall, C. D., Tsiotras, P. and Shen, H. (1998). Tracking Rigid Body Motion Using Thrusters and Momentum Wheels. AIAA/AAS Astrodynamics Conference.
- Hughes, P. C. (1986). Spacecraft attitude dynamics. John Wiley & Sons, New York.
- Johansen, T. A., Fossen, T. I. and Tøndel, P. (2003). Efficient optimal contrained control allocation via multi-parametric programming. *Journal of Guidance, Control, and Dynamics*. Submitted.
- Johansen, T. A., Petersen, I. and Slupphaug, O. (2000). Explicit Subopitmal Linear Quadratic Regulation with Input and State Constraints. CDC, Sydney.
- Kane, T., Linkins, P. and Levinson, D. (1983). Spacecraft Dynamics. McGraw-Hill, New York.
- Kaplan, M. H. (1976). *Modern spacecraft dynamics and control*. John Wiley & Sons, New York.
- Murray, R., Li, Z. and Sastry, S. (1994). *A Mathematical Introduction to Robotic Manipulation*. CRC Press.
- Nocedal, J. and Wright, S. J. (1999). Numerical Optimization. Springer.
- Qin, S. J. and Badgwell, T. J. (1997). An overview of industrial model predictive control technology. pp. 232–256. Chemical Process Control-V, Tahoe, California.
- Schaub, H., Robinett, R. D. and Junkins, J. L. (1996). Globally Stable Feedback Laws for Near-Minimum-Fuel and Near-Minimum-Time Pointing Maneuvers for a Landmark-Tracking Spacecraft. *Journal of the Astronautical Sciences*, 44(4):pp. 443–466.
- Schaub, H., Tsiotras, P. and Junkins, J. L. (1995). Principal rotation representations of proper $N \times N$ orthogonal matrices. *International Journal of Engineering Science*, 33(15):pp. 2277–2295.
- Sellers, J. J. (2000). Understanding space. An introduction to astronautics. McGraw-Hill, second ed.
- Show, L.-L. and Juang, J.-C. (2003). Satellite large angle tracking control design: thruster control approach. vol. 2, pp. 1098–1103. American Control Conference.
- Show, L.-L., Juang, J.-C. and Jan, Y.-W. (2003). An LMI-based nonlinear attitude control approach. *IEEE Transactions on Control Systems Technology*, 11(1):pp. 73–83.
- Show, L.-L., Juang, J.-C. and Yang, C.-D. (2001). Nonlinear \mathcal{H}_{∞} robust control for satellite large angle attitude maneuvers. vol. 2, pp. 1357–1362. American Control Conference.
- Song, G. and Agrawal, B. N. (1999). Vibration reduction for flexible spacecraft attitude control using PWPF modulator and smart structures. vol. 2, pp. 161–172. American Control Conference.
- Sun, Y.-P. and Yang, C.-D. (2002). Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ attitude control of a LEO microsatellite in the presence of inertia matrix uncertainty. vol. 2, pp. 1354–1359. American Control Conference.

Teunissen, P. J. G. (1998). GPS for Geodesy. Springer.

- Tøndel, P. (2003). Constrained Optimal Control via Multiparametric Quadratic Programming. Ph.D. thesis, Norwegian University of Science and Technology.
- Tøndel, P. and Johansen, T. A. (2002). Complexity reduction in explicit model predictive control. IFAC World Congress, Barcelona.
- Tøndel, P., Johansen, T. A. and Bemporad, A. (2003a). An algorithm for multi-parametric programming and explicit MPC solutions. *Automatica*, (39):pp. 489–497.
- Tsiotras, P. (1994). New Control Laws for the Attitude Stabilization of Rigid Bodies. pp. 316–321. 13th IFAC Symposium on Automatic Control in Aerospace, Palo Alto, California.
- Tsiotras, P. and Doumtchenko, V. (2000). Control of Spacecraft Subject to Actuator Failures: State-of-the-Art and Open Problems. *Journal of the Astronautical Sciences*, 48(2):pp. 337–358.
- Tsiotras, P., Haijun, S. and Hall, C. D. (2001). Satellite Attitude Control and Power Tracking with Energy/Momentum Wheels. *Journal of Guidance, Control, and Dynamics*, 24(1):pp. 23–34.
- Tsiotras, P. and Longuski, J. (1995). A New Parameterization of the Attitude Kinematics. *Journal of the Astronautical Sciences*, 43(3):pp. 243–262.
- Tsiotras, P. and Longuski, J. M. (1996). Comments on a new parametrization of the attitude kinematics. In *Astrodynamics Specialists Conference*.
- Vallado, D. A. (1997). Fundamentals of Astrodynamics and Applications. McGraw-Hill, New York.
- Wen, J. T.-Y. and Kreutz-Delgado, K. (1991). The attitude control problem. *IEEE Transactions on Automatic Control*, 36(10):pp. 1148–1162.
- Wertz, J. R. (ed.) (1978). *Spacecraft Attitude Determination and Control*. Kluwer Academic Publishers, London, third ed.
- Wertz, J. R. (ed.) (1999). *Space mission analysis and design*. Kluwer Academic Publishers, London, third ed.
- Wie, B. and Barba, P. M. (1985). Quaternion feedback for spacecraft large angle maneuvers. *Journal of Guidance, Control, and Dynamics*, 8(3):pp. 360–365.
- Wie, B. and Plescia, C. T. (1984). Attitude Stabilization of Flexible Spacecraft During Stationkeeping Maneuvers. *Journal of Guidance, Control, and Dynamics*, 7(4):pp. 430–436.
- Wiśniewski, R. and Blanke, M. (1999). Fully magnetic attitude control for spacecraft subject to gravity gradient. *Automatica*, 35:pp. 1201–1214.

78 BIBLIOGRAPHY

Appendix A Paper submitted to the 24th American Control Conference (ACC)

Attitude Control by means of Explicit Model Predictive Control, via Multi-parametric Quadratic Programming

Øyvind Hegrenæs*[#] Jan Tommy Gravdahl*[†] Petter Tøndel*[‡]

*Department of Engineering Cybernetics, Norwegian University of Science and Technology 7491 Trondheim, Norway

Abstract—Explicit solutions to constrained linear MPC problems can be computed by solving multi-parametric quadratic programs (mpQP), where the parameters are the components of the state vector. The solution to the mpQP is a piecewise affine (PWA) function, which can be evaluated at each sample to obtain the optimal control law. The online effort is restricted to a table-lookup, and the controller can be implemented on inexpensive hardware as fixed-point arithmetics can be used. This is highly desirable on systems suffering from limited power supply and CPU resources, being for instance micro-satellites. In this paper the explicit MPC (eMPC) approach is applied to the SSET/LESEO microsatellite, initiated by the European Space Agency (ESA). The controller is connected in closed-loop with the nonlinear plant, and the effectiveness is demonstrated through simulations.

I. INTRODUCTION

The purpose of this paper is to establish a nonlinear model of a micro-satellite, utilizing thrusters and one reaction wheel, and then finally propose a strategy to solve the problem of attitude control. However, unlike preceding work, typically carried out using PD- or LQ-control [12], Lyapunov design procedures [2]-[3], sliding mode [4]-[5], adaptive- or quaternion feedback techniques [6]-[8], \mathcal{H}_{∞} or $\mathcal{H}_2/\mathcal{H}_{\infty}$ [9]-[11], the topic of this paper will be on explicit Model Predictive Control. It is shown in this paper to be a highly potential scheme, and it should be considered if constraints need to be taken into account, while real-time optimization is impossible due to computational limitations. As familiar to the authors, this approach has not yet been exploited for attitude control of spacecrafts.

Stability proofs are omitted at this point, but a potential approach is to search for piecewise quadratic Lyapunov functions by solving a convex optimization problem. In [13] this was done using linear matrix inequalities (LMIs).

When doing implementation, an important thing to keep in mind is that the actuating thrusters are on-off by nature. A preliminary bang-bang modulation scheme with dead-zone will be utilized to deal with this problem.

The structural data and satellite model is based on the SSETI micro-satellite project, initiated by the European Space Agency. Further information will be given shortly. The results in this paper are based on the work in [1]. A. Explicit Model Predictive Control

When solving a MPC problem the control action, or equally, the solution, is obtained by computing an open-loop optimal sequence of control moves on a predefined horizon, once for each time sample. The first control input in the sequence is then applied to the plant, and the optimization is repeated with the new initial conditions and on the new horizon, shifted one step ahead. Due to the shifted horizon, the term *receding horizon control* is commonly used interchangeably with MPC. For the remainder of this section, the process to be controlled can be described by a *discrete-time*, deterministic linear state-space model, that is

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k)$$
$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k)$$
(1)

where $\mathbf{x}(k) \in \mathbb{R}^n$ is the state variable, $\mathbf{u}(k) \in \mathbb{R}^m$ is the input variable, $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{m \times m}$, and (\mathbf{A}, \mathbf{B}) is a stabilizable pair. If we now consider the regulator problem, that is, the problem of rendering the state vector to the origin, the traditional MPC solves the following optimization problem for the current $\mathbf{x}(k)$

 $\min_{\mathbf{U},\mathbf{s}} \{J(\mathbf{U},\mathbf{s},\mathbf{x}(k))\} \text{ subject to:}$

$$\begin{aligned} \mathbf{y}_{\min} - \mathbf{s} &\leq \mathbf{y}_{k+i|k} \leq \mathbf{y}_{\max} + \mathbf{s}, \ i = 1, \dots, N \\ \mathbf{u}_{\min} &\leq \mathbf{u}_{k+i} \leq \mathbf{u}_{\max}, \ i = 1, \dots, M-1 \\ \mathbf{u}_{k+i} &= \mathbf{K}\mathbf{x}_{k+i|k}, \ M \leq k \leq N-1 \\ \mathbf{x}_{k|k} &= \mathbf{x}(k) \\ \mathbf{x}_{k+i+1|k} &= \mathbf{A}\mathbf{x}_{k+i|k} + \mathbf{B}\mathbf{u}_{k+i}, \ k \geq 0 \\ \mathbf{y}_{k+i} &= \mathbf{C}\mathbf{x}_{k+i|k}, \ k \geq 0 \end{aligned}$$
(2)

where the cost function we seek to minimize is given as

$$J = \boldsymbol{\rho} \|\mathbf{s}\|_{2}^{2} + \mathbf{x}_{k+N|k}^{\mathsf{T}} \mathbf{P} \mathbf{x}_{k+N|k} + \Sigma_{i=0}^{N-1} \left\{ \mathbf{x}_{k+i|k}^{\mathsf{T}} \mathbf{Q} \mathbf{x}_{k+i|k} + \mathbf{u}_{k+i}^{\mathsf{T}} \mathbf{R} \mathbf{u}_{k+i} \right\}$$
(3)

and $\mathbf{U} \triangleq [\mathbf{u}_{k}^{\mathsf{T}}, \dots, \mathbf{u}_{k+M-1}^{\mathsf{T}}]^{\mathsf{T}}$, $\mathbf{s} \triangleq [\mathbf{s}_{k}^{\mathsf{T}}, \dots, \mathbf{s}_{k+N-1}^{\mathsf{T}}]^{\mathsf{T}}$, $\mathbf{R} = \mathbf{R}^{\mathsf{T}} > 0$, $\mathbf{Q} = \mathbf{Q}^{\mathsf{T}} \ge 0$, $\mathbf{P} = \mathbf{P}^{\mathsf{T}} \ge 0$, $\mathbf{x}_{k+i|k}$ is the prediction of $\mathbf{x}(k+i)$ at time k, M and N are input and constraint horizons. When the final cost matrix \mathbf{P} and gain matrix \mathbf{K} are calculated from the algebraic Riccati equation, under the assumptions that the constraints are not active for $k \ge M$, (2) exactly solves the constrained infinite horizon LQR problem for (1), with weight matrices \mathbf{R} and

[#]Graduate student, e-mail: hegrenas@stud.ntnu.no

[†]Associate Professor, e-mail: tommy.gravdahl@itk.ntnu.no [‡]Postdoctoral Fellow, e-mail: petter.tondel@itk.ntnu.no

Q. The additional variable $\mathbf{s} \in \mathbb{R}^{n_s}$ is a vector containing slack variables, while the term $\|\mathbf{s}\|_2$ is the \mathcal{L}_2 -norm of s, and ρ is the penalty weight of the slack variables. Note that using the \mathcal{L}_2 -norm is only one way of including slack variables. The slack variables are defined in such a way that they are nonzero only if the output constraints are violated, yet heavily penalized in the cost function, so that the optimizer has a strong incentive to keep them zero if possible. If we have $\rho = \infty$, or equally s = 0, the MPC problem in (2) obtains its most simple form, only involving hard constraints. In some situations this may be necessary, but the optimization problem becomes more difficult so solve, and infeasibility may occur. This can for instance be the case if initial conditions are infeasible to start with, if noise causes the output to go outside the feasible solution space in the next time step, or if there are serious model uncertainties. Obviously this needs to be dealt with in real applications, and as mention above one way of doing this is to include slack variables.

1) From linear MPC to mpQP: It is shown in [14], in case of having $\rho = \infty$, that the MPC problem in (2) can by some algebraic manipulation be reformulated as

$$V_{z}(\mathbf{x}(k)) = \min_{\mathbf{z}} \left\{ \frac{1}{2} \mathbf{z}^{\mathsf{T}} \mathbf{H} \mathbf{z} \right\}$$
subject to: $\mathbf{G} \mathbf{z} \leq \mathbf{W} + \mathbf{S} \mathbf{x}(k)$
(4)

where $\mathbf{z} \triangleq \mathbf{U} + \mathbf{H}^{-1}\mathbf{F}^{\mathsf{T}}\mathbf{x}(k)$, \mathbf{U} is given as in (2), and $\mathbf{x}(k)$ is the current state, which can be treated as a vector of parameters. Dimensions are given as $\mathbf{z} \in \mathbb{R}^{n_z}$, $\mathbf{H} \in \mathbb{R}^{n_z \times n_z}$, $\mathbf{G} \in \mathbb{R}^{q \times n_z}$, $\mathbf{W} \in \mathbb{R}^{q \times 1}$, and $\mathbf{S} \in \mathbb{R}^{q \times n}$. Note that $\mathbf{H} > 0$ since $\mathbf{R} > 0$. The latter is a strong result, as the problem formulated in (4) is strictly convex, and the Karush-Kuhn-Tucker (KKT) conditions are necessary and sufficient conditions for optimality, giving an unique solution.

As shown in [14], the mpQP in (4) can be solved by applying the KKT conditions

$$\mathbf{Hz} + \mathbf{G}^{\mathrm{T}} \lambda = 0, \quad \lambda \in \mathbb{R}^{q}$$

$$\lambda_{r} \left(\mathbf{G}^{r} \mathbf{z} - \mathbf{W}^{r} - \mathbf{S}^{r} \mathbf{x}(k) \right) = 0, \quad r = 1, \dots, q,$$

$$\lambda \ge 0,$$

$$\mathbf{Gz} - \mathbf{W} - \mathbf{Sx}(k) \le 0.$$
 (5)

where the subscript r on some matrix denotes the r^{th} row, while q is the number of inequalities in the optimization problem, and the number of free variables is $n_z = m \cdot N$.

The key observation is that (4) is solved explicitly for all x. It is shown in [14] that the solution $z^*(x(k))$, hence $U^*(x(k))$, is a continuous piecewise affine (PWA) function defined over a polyhedra partition. Consequently, the online effort is limited to evaluating this PWA function.

Even though not derived for the case of including slack variables, both (4) and (5) can easily be extended to cover this situation, by defining the augmenting matrices $\widetilde{\mathbf{H}} \in \mathbb{R}^{\tilde{n}_z \times \tilde{n}_z}$, $\widetilde{\mathbf{G}} \in \mathbb{R}^{q \times \tilde{n}_z}$, and $\widetilde{\mathbf{z}} \triangleq [\mathbf{U}, \mathbf{s}]^T \in \mathbb{R}^{\tilde{n}_z}$. The number of *free variables* now becomes $\tilde{n}_z = n_z + n_s$.

B. SSETI/ESEO

We then turn our attention to the Student Space Exploration & Technology Initiative (SSETI). The specific satellite to be studied in this paper is the European Student Earth Orbiter (ESEO). Through the project, students from different European universities participate in designing, building and operating a micro-satellite. In addition to the satellite, the whole system consists of the payload carried by the spacecraft and the associated ground systems. The main author has been fortunate to participate on the work related to the attitude control system (ACS).

In terms of attitude and control requirements, these are specified according to the diversified situations the satellite is expected to face during its lifetime. Only the *nominal* mode will be considered in this paper when doing attitude control, which includes the task of maintaining a stable attitude. More specifically this means that the best obtainable result is whenever the body frame coincides with a defined orbit frame. This will be explained subsequently.

A short summary of structural data is given in Table I.

TABLE I SSETI/ESEO PARAMETERS

SSETI/ESEO PARAMETERS

Parameter	Value
Satellite inertia matrix, I	diag(4.250, 4.337, 3.664) [kg m2]
Axial wheel inertia, I_s	$4 \cdot 10^{-5} [\text{kg m}^2]$
Axial wheel placement, A	$[0, 1, 0]^{\mathrm{T}}$
Nominal thruster torque, τ_n	$[0.0484, 0.0484, 0.0398]^{T}$ [Nm]
Maximum wheel velocity	$527 \text{ [rad/s]} \approx 5032 \text{ rpm}$

II. MODELLING

Equations describing a satellite with thrusters and an *N*-wheel cluster are derived. The notation is based on [17].

A. Kinematics

The Euler parameters are chosen to represent the kinematics due to their nonsingular parametrization and linear differential equations if the angular velocities are known. The Euler parameters are defined in terms of the angle-axis parameters θ and k, and the mapping is defined as

$$=\cos\frac{\theta}{2}, \quad \epsilon = \mathbf{k}\sin\frac{\theta}{2}$$
 (6)

which gives the corresponding rotation matrix

 η

$$\mathbf{R}(\eta, \boldsymbol{\epsilon}) = \mathbf{1} + 2\eta \boldsymbol{\epsilon}^{\times} + 2\boldsymbol{\epsilon}^{\times} \boldsymbol{\epsilon}^{\times}$$
(7)

From the properties of the rotation matrix, it can be shown that its differential equation can be written as

$$\dot{\mathbf{R}}_{o}^{b} = \left(\boldsymbol{\omega}_{bo}^{b}\right)^{\times} \mathbf{R}_{o}^{b} = -\left(\boldsymbol{\omega}_{ob}^{b}\right)^{\times} \mathbf{R}_{o}^{b} \tag{8}$$

where ω_{ob}^{b} is defined as the angular velocity of the body frame \mathcal{F}_{b} relative the orbit frame \mathcal{F}_{o} , measured in \mathcal{F}_{b} , and \mathbf{R}_{o}^{b} is the rotation matrix from \mathcal{F}_{b} to \mathcal{F}_{o} . The orbit frame has its origin located at the center of mass of the satellite, and its z-axis is *always* nadir pointing (center of the Earth), while the the x-axis is pointing in the direction of the velocity. The y-axis completes the right-hand coordinate system. From (7) and (8) the kinematic differential equations for the Euler parameters can be found to be given as

$$\dot{\eta} = -\frac{1}{2} \epsilon^{\mathrm{T}} \omega_{ob}^{b} \tag{9a}$$

$$\dot{\boldsymbol{\epsilon}} = \frac{1}{2} \left[\eta \mathbf{1} + \boldsymbol{\epsilon}^{\times} \right] \boldsymbol{\omega}_{ob}^{b} \tag{9b}$$

B. Dynamics

The rotational equations of motion for a N-wheel gyrostat can be written as

$$\dot{\mathbf{h}}_b = \boldsymbol{\tau}_e - \left[\mathbf{J}^{-1}(\mathbf{h}_b - \mathbf{A}\mathbf{h}_a)\right] \times \mathbf{h}_b$$
 (10a)
 $\dot{\mathbf{h}}_a = \boldsymbol{\tau}_a$ (10b)

where \mathbf{h}_a is the $N \times 1$ vector of the axial angular momenta of the wheels, τ_e is the 3×1 vector of the external torque acting on the body, not including wheel torques, τ_a is the $N \times 1$ vector of the internal axial torques applied by the platform to the wheels, and \mathbf{A} is the $3 \times N$ matrix whose columns contain the axial unit vectors of the N momentum exchange wheels. If we let $\boldsymbol{\omega}_{bb}^b$ denote the angular velocity of the body frame \mathcal{F}_b relative an inertial frame \mathcal{F}_i , measured in \mathcal{F}_b , then the vector \mathbf{h}_b is the total angular momentum for the spacecraft in the body frame, given as

$$\mathbf{h}_b = \mathbf{J}\boldsymbol{\omega}_{ib}^b + \mathbf{A}\mathbf{h}_a \tag{11}$$

where \mathbf{J} is the inertialike matrix defined as

$$\mathbf{J} \triangleq \mathbf{I} - \mathbf{A} \mathbf{I}_s \mathbf{A}^{\mathrm{T}}$$
(12)

I is the moment of inertia matrix for the spacecraft, including wheels, and the matrix $I_s = \text{diag}\{I_{s1}, I_{s2}, ..., I_{sN}\}$ contains the axial moments of inertia of the wheels on the diagonal. The axial angular momenta of the wheels can be written in terms of the body angular velocity and the wheels' axial angular velocities relative to the body, ω_s , as

$$\mathbf{h}_a = \mathbf{I}_s \mathbf{A}^{\mathrm{T}} \boldsymbol{\omega}_{ib}^b + \mathbf{I}_s \boldsymbol{\omega}_s \tag{13}$$

Note that $\boldsymbol{\omega}_s = [\boldsymbol{\omega}_{s1}, \boldsymbol{\omega}_{s2}, ..., \boldsymbol{\omega}_{sN}]^{\mathrm{T}}$ is an $N \times 1$ vector, and that these relative angular velocities are those that would for instance be measured by tachometers fixed to the platform.

Equation (10) can also be written in terms of angular velocities. By defining $\mu \triangleq [\mathbf{h}_b, \mathbf{h}_a]^{\mathrm{T}}$ and $\upsilon \triangleq [\boldsymbol{\omega}_{bb}^{ib}, \boldsymbol{\omega}_s]^{\mathrm{T}}$ we can write (11) and (13) in the compact form

$$\mu = \Lambda \upsilon$$
, where $\Lambda = \begin{bmatrix} \mathbf{I} & \mathbf{A} \mathbf{I}_s \\ \mathbf{I}_s \mathbf{A}^{\mathrm{T}} & \mathbf{I}_s \end{bmatrix}$ (14)

Clearly, we can find ω_{ib}^b and ω_s from $\upsilon = \Lambda^{-1}\mu$, or equally, we can write $\dot{\upsilon} = \Lambda^{-1}\dot{\mu}$. By utilizing the matrix inversion lemma, together with (14), we get that

$$\begin{bmatrix} \dot{\boldsymbol{\omega}}_{ib}^{b} \\ \dot{\boldsymbol{\omega}}_{s} \end{bmatrix} = \begin{bmatrix} \mathbf{J}^{-1} & -\mathbf{J}^{-1}\mathbf{A} \\ -\mathbf{A}^{\mathrm{T}}\mathbf{J}^{-1} & \mathbf{A}^{\mathrm{T}}\mathbf{J}^{-1}\mathbf{A} + \mathbf{I}_{s}^{-1} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{h}}_{b} \\ \dot{\mathbf{h}}_{a} \end{bmatrix}$$
(15)

which can be written in full as

$$\begin{split} \dot{\boldsymbol{\omega}}_{ib}^{b} &= \mathbf{J}^{-1} \left[-(\boldsymbol{\omega}_{ib}^{b})^{\times} (\mathbf{I} \boldsymbol{\omega}_{ib}^{b} + \mathbf{A} \mathbf{I}_{s} \boldsymbol{\omega}_{s}) + \boldsymbol{\tau}_{e} \right] \\ &- \mathbf{A} \boldsymbol{\tau}_{a} \qquad (16a) \\ \dot{\boldsymbol{\omega}}_{s} &= \mathbf{A}^{\mathrm{T}} \mathbf{J}^{-1} \left[(\boldsymbol{\omega}_{ib}^{b})^{\times} (\mathbf{I} \boldsymbol{\omega}_{ib}^{b} + \mathbf{A} \mathbf{I}_{s} \boldsymbol{\omega}_{s}) - \boldsymbol{\tau}_{e} \right] \\ &+ \left[\mathbf{A}^{\mathrm{T}} \mathbf{J}^{-1} \mathbf{A} + \mathbf{I}_{s}^{-1} \right] \boldsymbol{\tau}_{a} \qquad (16b) \end{split}$$

As can be seen from (16), the angular velocities are given in \mathcal{F}_b and relative to \mathcal{F}_i , while the kinematics in (9) are relative to \mathcal{F}_o . However, it would be preferable if we in the model could represent the attitude of \mathcal{F}_b relative to \mathcal{F}_o . This can be done by exploiting the relation

$$\boldsymbol{\omega}_{ib}^{b} = \boldsymbol{\omega}_{ob}^{b} + \mathbf{R}_{o}^{b} \boldsymbol{\omega}_{io}^{o} \quad \text{and} \quad \dot{\boldsymbol{\omega}}_{ib}^{b} = \dot{\boldsymbol{\omega}}_{ob}^{b} + \dot{\mathbf{R}}_{o}^{b} \boldsymbol{\omega}_{io}^{o} \quad (17)$$

where $\boldsymbol{\omega}_{io}^{o} = [0, -\omega_{0}, 0]^{\mathrm{T}}$, and ω_{0} is assumed constant and equal to the mean angular velocity of \mathcal{F}_{o} , given in \mathcal{F}_{i} . This implies circular orbits. For the remainder we let \mathbf{c}_{i} denote the i'th column of the rotation matrix \mathbf{R}_{o}^{b} . If we also include the gravity gradient as a disturbance, that is $\tau_{e} = \boldsymbol{\tau} + \boldsymbol{\tau}_{g}$, where $\boldsymbol{\tau}$ is the torque provided from thrusters, while the gravity gradient is given as

$$\boldsymbol{\tau}_g = 3\omega_0^2 \left[\mathbf{c}_3 \times (\mathbf{I}\mathbf{c}_3) \right] \tag{18}$$

and by utilizing (8) and (17), we can rewrite (16) as

$$\dot{\omega}_{ob}^{b} = \hat{f}_{inert} + \hat{f}_{\tau} + \hat{f}_{g} + \hat{f}_{add}$$
(19a)
$$\dot{\omega}_{s} = \bar{f}_{inert} + \bar{f}_{\tau} + \bar{f}_{a}$$
(19b)

where the terms are given as

$$\begin{split} \hat{f}_{inert} &= \mathbf{J}^{-1} \left[-(\boldsymbol{\omega}_{ob}^{b} - \boldsymbol{\omega}_{o} \mathbf{c}_{2})^{\times} \\ & \left(\mathbf{I} \left[\boldsymbol{\omega}_{ob}^{b} - \boldsymbol{\omega}_{o} \mathbf{c}_{2} \right] + \mathbf{A} \mathbf{I}_{s} \boldsymbol{\omega}_{s} \right) \right] \\ \bar{f}_{inert} &= \mathbf{A}^{\mathsf{T}} \mathbf{J}^{-1} \left[(\boldsymbol{\omega}_{ob}^{b} - \boldsymbol{\omega}_{o} \mathbf{c}_{2})^{\times} \\ & \left(\mathbf{I} \left[\boldsymbol{\omega}_{ob}^{b} - \boldsymbol{\omega}_{o} \mathbf{c}_{2} \right] + \mathbf{A} \mathbf{I}_{s} \boldsymbol{\omega}_{s} \right) \right] \\ \hat{f}_{\tau} &= \mathbf{J}^{-1} \boldsymbol{\tau} - \mathbf{J}^{-1} \mathbf{A} \boldsymbol{\tau}_{a} \\ \bar{f}_{\tau} &= -\mathbf{A}^{\mathsf{T}} \mathbf{J}^{-1} \boldsymbol{\tau} + \left[\mathbf{A}^{\mathsf{T}} \mathbf{J}^{-1} \mathbf{A} + \mathbf{I}_{s}^{-1} \right] \boldsymbol{\tau}_{a} \\ \hat{f}_{g} &= \mathbf{J}^{-1} \left[3 \boldsymbol{\omega}_{0}^{2} \mathbf{c}_{3} \times (\mathbf{I} \mathbf{c}_{3}) \right] \\ \bar{f}_{g} &= -\mathbf{A}^{\mathsf{T}} \mathbf{J}^{-1} \left[3 \boldsymbol{\omega}_{0}^{2} \mathbf{c}_{3} \times (\mathbf{I} \mathbf{c}_{3}) \right] \\ \hat{f}_{add} &= \boldsymbol{\omega}_{o} \dot{\mathbf{c}}_{2} \end{split}$$

III. ATTITUDE CONTROL BY MEANS OF EXPLICIT MPC

In the following the explicit MPC controller is computed, and some aspects considering implementation are discussed. The complete nonlinear model throughout is given as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) = [\dot{\boldsymbol{\omega}}_{ob}^{b}, \dot{\boldsymbol{\omega}}_{s}, \dot{\boldsymbol{\eta}}, \dot{\boldsymbol{\epsilon}}]^{\mathrm{T}}$$
(21)

where $\boldsymbol{\omega}_{ob}^{b} \triangleq [\omega_{1}, \omega_{2}, \omega_{3}]^{\mathrm{T}}$, $\boldsymbol{\epsilon} \triangleq [\epsilon_{1}, \epsilon_{2}, \epsilon_{3}]^{\mathrm{T}}$, and $\mathbf{u} \triangleq [\tau_{1}, \tau_{2}, \tau_{3}, \tau_{a}]^{\mathrm{T}} = [\boldsymbol{\tau}^{\mathrm{T}}, \boldsymbol{\tau}_{a}^{\mathrm{T}}]^{\mathrm{T}}$ will be used for short.

A. Explicit MPC controller for the SSETI/ESEO satellite

As we are only considering the linear discrete-time MPC in this paper, it is necessary to linearize the model in (9) and (19) with respect to the total state vector. By choosing the equilibrium point p equal to $\mathbf{x}^p = [\mathbf{0}^4, 1, \mathbf{0}^3]^T, \mathbf{u}^p = \mathbf{0}^4$, which equals the scenario where \mathcal{F}_b coincides with \mathcal{F}_o and the angular velocity of the wheel is zero, it can be found that the linearized model can be given as

$$\Delta \dot{\mathbf{x}} = \mathbf{A}_c \Delta \mathbf{x} + \mathbf{B}_c \Delta \mathbf{u} \tag{21}$$

where the matrices \mathbf{A}_c and \mathbf{B}_c are given as

where we used that $\mathbf{I} = \text{diag}(i_{11}, i_{22}, i_{33}), k_x = \frac{i_{22}-i_{33}}{i_{11}}, k_y = \frac{i_{11}-i_{33}}{i_{22}}, k_z = \frac{i_{22}-i_{11}}{i_{33}} \text{ and } \kappa = i_{22} - \mathbf{I}_s.$

From the system matrix in (22a), we can immediately conclude that the linearized system is uncontrollable, as all the terms corresponding to η are equal to zero. However, the linearized system is found to be stabilizable, and omitting η , also controllable. Also note that we can utilize the fact that the Euler parameters satisfy $\eta^2 + \epsilon^T \epsilon = 1$, making us able to keep track of, and update η in an open-loop manner.

Before we can use the mpQP Algorithm, (21) is converted into an equivalent discrete-time form by utilizing a modified first-order hold approach. The sampling time is chosen as $T_s = 0.1$ [sec], and when deriving the controller, η is omitted, making up the new state vector $\tilde{\mathbf{x}}$ of 7th order.

TABLE II SUMMARY OF TUNING PARAMETERS

Parameter	Value	
Q	diag{200, 200, 200, $5 \cdot 10^{-7}$, 1, 1, 1}	
R	diag{100, 200, 100, 1}	
N (horizon)	2	
ρ (slack)	$8 \cdot 10^{-5}$	

The tuning parameters for the explicit MPC controller are summarized in Table II. The parameter space, in which we search for feasible solutions of the mpQP, the is chosen as

 $-[1, 1, 1, 1500, 1, 1, 1]^{T} \leq \widetilde{\mathbf{x}} \leq [1, 1, 1, 1500, 1, 1, 1]^{T}$ (23)

$$\mathbf{u}_{max} = -\mathbf{u}_{min} = \begin{bmatrix} 0.0484\\ 0.0484\\ 0.0398\\ 0.0020 \end{bmatrix}, \quad |\omega_s| \le 527 \quad (24)$$

The solution of the mpQP, obtained from the discrete-time version of (21), Table I and II, and (24), gave a polyhedral partition over the parameter space in (23), consisting of 2867 regions. If we denote each of these polyhedrons as \mathcal{X}_i , where *i* is the specific region, then $\mathcal{X}_i \subset \mathbb{R}^7$. Examples of planer intersections are shown in Fig. 1. Each polyhedron contains an optimal control law such that if $\tilde{\mathbf{x}}(k) \in \mathcal{X}_i$ then $\mathbf{u}(k) = \mathbf{K}_i \tilde{\mathbf{x}}(k) + \mathbf{k}_i$. The latter is clearly an affine function.



Fig. 1. Polyhedral partition, N=2 and $\pmb{\rho}=8\cdot 10^{-5}$

B. Bang-bang modulation

When doing implementation, an important thing to keep in mind is that the actuating thrusters are on-off by nature. A preliminary bang-bang modulation scheme can be applied for dealing with this problem. The technique is best explained through Fig. 2, where \mathbf{K}_{nom} are nominal thruster torques, and \mathbf{u}_* is given according to

$$\mathbf{u}_*: \operatorname{signum}(\mathbf{u}) = \begin{cases} -1 & \text{if } \mathbf{u} \le -dz, \\ 1 & \text{if } \mathbf{u} \ge dz. \end{cases}$$
(25)



Fig. 2. Bang-bang modulation with dead-zone

IV. SIMULATIONS

To demonstrate the aforementioned theory for the linear MPC controller, the following closed-loop simulations have been performed with the complete nonlinear model. Initial conditions for the dynamics and kinematics, as well as Keplerian orbital elements, are given in Table III.

The first case is simulated without noise and bang-bang modulation, while in the second, bang-bang is utilized, to-gether with measurement noise as in Table IV. By means of visualizing the results, the Euler parameters are transformed into Euler angles [deg].

TABLE III SUMMARY OF SIMULATIONS

Case 1	Initial condition	Set-point	Unit
ω_{ob}^{b}	$\{-0.05, 0.15, -0.08\}$	$\{0, 0, 0\}$	rad/s
ω_s	400	0	rad/s
Euler angles (XYZ)	$\{-25, 60, 90\}$	$\{0, 0, 0\}$	deg
Case 2	Initial condition	Set-point	Unit
ω_{ob}^{b}	$\{0.018, -0.018, 0.018\}$	$\{0, 0, 0\}$	rad/s
ω_s	-500	0	rad/s
Euler angles (XYZ)	$\{60, -45, 30\}$	$\{0, 0, 0\}$	deg
Keplerian elements	Initial condition		Unit
$[i, \omega, \Omega, \nu]$	$\{7, 178, -10, 0\}$		deg
a	24603.14		km
e	0.0		-

TABLE IV RMS ERRORS IN STATE ESTIMATES

Estimates	Errors	Unit
ω^{b}_{ob}	$\{0.0035, 0.0052, 0.0035\}$	rad/s
ω_s	0.5	rad/s
Euler angles (XYZ)	$\{0.1, 0.1, 0.1\}$	deg



Fig. 3. Case 1, N = 2 and $\rho = 8 \cdot 10^{-5}$



Fig. 4. Case 2, N = 2 and $\rho = 8 \cdot 10^{-5}$

V. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

It has been shown that explicit solutions to constrained linear MPC problems can be computed for the attitude control problem by solving multi-parametric quadratic programs (mpQP). The approach has shown to be an highly potential, and it should be considered if constraints need to be taken into account.

REFERENCES

- [1] Hegrenæs Ø., "Attitude Control by means of Explicit Model Predictive Control, via Multi-parametric Quadratic Programming", MSc. thesis, Norwegian University of Science and technology, (2004). Tsiotras P., "New Control Laws for the Attitude Stabilization of Rigid
- [2] Bodies", 13th IFAC Symposium on Automatic Control in Aerospace, (1994).
- Kall, C. D., P. Tsiotras and H. Shen, "Tracking Rigid Body Motion Using Thrusters and Momentum Wheels", *The Journal of the Astronautical Sciences*, Vol. 50, No. 3, pp. 311-323, (2002).
 Cavallo, A. and G. De Maria, "Attitude control for large angle and the product of the product
- maneuvers", *IEEE International Workshop*, pp. 232-237,(1996). Chen, Y-P. and S-C. Lo, "Sliding-mode controller design for space-
- [5] craft attitude tracking maneuvers", IEEE Transactions on Aerospace and Electronic Systems, Vol. 29, no. 4, pp. 1328-1333, (1993).
- [6] Wie, B. and P. M. Barba, "Quaternion feedback for spacecraft large angle maneuvers", Journal of Guidance, Control, and Dynamics, Vol. Joshi, S. M., A. G. Kelkar and J. T.-Y. Wen, "Robust attitude
- [7] stabilization of spacecraft using nonlinear quaternion feedback" IEEE Transactions on Automatic Control, Vol. 40, no. 10, pp. 1800-1803. (1995).
- Singh S. N.,"Nonlinear Adaptive Attitude Control for Spacecraft", [8] IEEE Transactions on Aerospace and Electronic Systems, Vol. 23, no. 3, pp. 371-380, (1987). Show, L.-L., J.-C. Juang and C.-D. Yang, "Nonlinear \mathcal{H}_∞ robust
- [9] control for satellite large angle attitude maneuvers", American Control Conference, Vol. 2, pp. 1357-1362, (2001). [10] Dalsmo, M. and O. Egeland, "State feedback \mathcal{H}_{\circ}
- o control of a rigid spacecraft", Proceedings of the 34th IEEE Conference on Decision spacetal , Proceedings of the S4m BLL competence on Decision and Control, Vol. 4, pp. 3968-3973, (1995). Sun, Y.-P. and C.-D. Yang, "Mixed $\mathcal{H}_2/\mathcal{H}_{\infty}$ attitude control of a
- [11] LEO microsatellite in the presence of inertia matrix uncertainty", American Control Conference, Vol. 2, pp. 1354-1359, (2002).
- [12] Show, L.-L., J.-C. Juang, C.-T. Lin and Y.-W. Jan, "Spacecraft robust attitude tracking design: PID control approach", Proceedings of the
- American Control Conference, Vol. 2, pp. 1360-1365, (2002).
 [13] Ferrari-Trecate, G., F. A. Cuzzola, D. Mignore and M. Morari, "Analysis and control with performance of piecewise affine and hybrid systems", American Control Conferance, Vol. 1, pp. 200-205, (2001).
- [14] Bemporad, A., M. Morari, V. Dua and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems", Automatica, Vol. 38, pp. 3-20, (2002).
 [15] Tøndel, P., T. A. Johansen and A. Bemporad, "An algorithm for
- multiparametric quadratic programming and explicit MPC solutions", *Automatica*, Vol. 39, pp. 489-497, (2003).
- - (1986).

Appendix B Communication and Workshops associated to ESA

Various communication and exchange of material has taken place during work on this thesis. A wide range of tools has been available for the SSETI project, and communication has been via e-mail, IRC, Newsgroup, ftp-servers, Workshops, and conferences. A short summary is given bellow.

Presentation on eMPC held at the 7th SSETI Workshop













Participation at the STEC 2004 conference, Lausanne, Switzerland



Appendix C MATLAB scripts and Simulink block diagrams

The Simulink block diagrams shown in this appendix were all used throughout our simulations, together with a few MATLAB scripts. Except for different plotting routines, all the block diagrams and scripts are included below. The source code for the mpQP toolbox is not included here, and the reader should refer to Petter Tøndel at the Norwegian University of Science and Technology for details.

MATLAB functions and scripts

```
init.m : (.) \rightarrow (defines various parameters)
```

```
응*
응*
    init.m
응*
응*
    Author: Oeyvind Hegrenaes and Morten Topland, NTNU 2004
응*
응*
    Initialization of different parameters used in Simulink
%*
% Earth Parameters by means of WGS84
mu = 398600.5; % [km^3/s^2]
re = 6378137e-3; % Equatorial Radius of the Earth [km]
f = 1/298.257223563; % Flattering of Earth ellipsoid
rp = re - f*re; % Polar Radius of the Earth [km]
% Orbit parameters for approximated circular orbit
% Initial orbit parameters given for SSETI/ESEO
% inclination: 7 [deg]
% semi-major axis: 24603.14 [km]
% eccentricity: 0.718
```

```
% argument of perigee: 178 [deg]
% 1st descending node: -10 [deg]
a = 24603.14; e = 0.718;
rc_mean = a*sqrt(1-e^2); % Mean radius of ellipse orbit [km]
omega0 = sqrt(mu/rc_mean^3);
% SSETI/ESEO Parameters
ill = 4.35; i22 = 4.337; i33 = 3.664;
Ib = [i11 0 0 ; 0 i22 0 ; 0 0 i33]; % Inertia matrix
a_vec = [0 1 0]'; % Normalized axis of relative rotation
A = a_vec;
Is = 4.0e-5; % Moment of inertia of the wheel about a_vec
J = Ib - A*Is*A'; % Inertialike matrix
Jinv=inv(J); AtJinv=A'*Jinv; AIs=A*Is;
AtJinvA=A'*Jinv*A; JinvA=Jinv*A; % Pre-calculated parameters
% Parameters for linearized model (the components
% corresponding to dot_eta have been omitted since
% they were only zeros)
kx=(i22-i33)/i11; ky=(i11-i33)/i22;
kz=(i22-i11)/i33; kappa=i22-Is;
A = [0, 0, (1-kx) * omega0, 0, -8 * kx * omega0^2, 0, 0;
     0,0,0,0,0,-6*ky*i22*omega0^2/kappa 0;
     (kz-1)*omega0,0,0,0,0,0,-2*kz*omega0^2;
     0,0,0,0,0,6*ky*i22*omega0^2/kappa,0;
     1/2,0,0,0,0,0,0;
     0,1/2,0,0,0,0,0;
     0,0,1/2,0,0,0,0 ];
B = [1/i11, 0, 0, 0;
     0,1/kappa,0,-1/kappa;
     0,0,1/i33,0;
     0,-1/kappa,0,i22/(kappa*Is);
     0,0,0,0;
     0,0,0,0;
     0,0,0,0];
C = eye(7,7); D = zeros(7,4);
% Nominal values from PROP document 24.02.2004 [Nm]
Tau_thrust_nom = [0.0484, 0.0484, 0.0398]';
% Reaction wheel
w_s_max = 5035*2*pi/60;
```

[K_scl,S_scl,E_scl]=LQR(A_scaled,B_scaled,Q_scaled,R_scaled);

```
solveKep.m : (i, n, e, \Omega_0, \varpi_0, \nu_0, \Delta t) \rightarrow (\Omega, \varpi, \nu)
```

```
function Orbit = solveKep(input)
응*
응*
   solveKep.m
                                                         *
%*
                                                         *
응*
   Author: Oeyvind Hegrenaes, NTNU 2004
                                                         *
응*
   Solves Keplers equation iteratively. The position
응*
                                                         *
   of the satellite in the geocentric inertal system
응*
   is found be means of classical Kepler elements at time t. *
응*
%*
                                                         *
   Input parameter: i [deg] (inclination)
응*
                                                         *
                     e (eccentricity)
응*
                                                         *
응*
                     a [km] (semimajor axis)
                     o [deg] (ascending node)
응*
                     w [deg] (perigee)
%*
%*
                     v [deg] (true anomaly)
                                                         *
                     t [sec] (time [sec])
응*
8*
응*
    Output parameter: Orbit (updated COE)
응*
i=(pi/180)*input(1); e=input(2); a=input(3); o=(pi/180)*input(4);
w=(pi/180)*input(5); v=(pi/180)*input(6); t=input(7);
mu = 398600.4418; % WGS84 [km^3/s^2]
% Orbit settings
n = sqrt(mu/a<sup>3</sup>); % mean motion
if(e^{\sim}=0)
   E0 = acos(((e + cos(v)) / (1 + e*cos(v))));
   if(pi < v & v < 2*pi)
       E0 = 2*pi-E0;
   end
else
   E0 = w + v;
end
MO = EO - e*sin(EO); M = MO + n*t;
if(-pi < M & M < 0 | pi < M)
   E0 = M - ei
else
```

```
E0 = M + e;
end
E_old = E0; E_new = E_old + (M - E_old + e*sin(E_old)) \dots
        / (1 - e*cos(E_old));
while(abs(E_new - E_old) > 10^{-8})
    E_old = E_new;
    E_new = E_old + (M - E_old + e*sin(E_old)) \dots
            / (1 - e*cos(E_old));
end
if(e^{-}=0)
    v_new = acos((cos(E_new)-e) / (1-e*cos(E_new)));
    if(pi < E_new & E_new < 2*pi)</pre>
        v_new = 2*pi-v_new;
    end
else
    v_new = E_new - w;
end
Orbit = [i,e,a,o,w,v_new];
```

```
kep2car.m : (COE) \rightarrow (\mathcal{F}_i)
```

```
function Orbit_ECI = kep2car(input)
응*
%* kep2car.m
응*
                                                     *
%* Author: Oeyvind Hegrenaes, NTNU 2004
응*
%* For COE, the function finds the corresponding position
%* and velocity of the orbit frame, given in the ECI frame.
응*
                                                     *
응*
    Input parameter: COE (Keplarian orbit elemets)
                                                     *
응*
                                                     *
    Output parameter: Orbit_ECI (pos. and vel. in ECI)
응*
                                                     *
응*
                   Abs. distance to center of the Earth
                                                    *
응*
i=input(1);e=input(2); a=input(3); Omega=input(4);
w=input(5);v=input(6);
% WGS84
mu = 398600.4418; % [km<sup>3</sup>/s<sup>2</sup>]
```

```
% Orbit settings
n = sqrt(mu/a^3);
                      % mean motion
% Find the ECI position and velocity from COE
r=a*(1-e^2)/(1+e*cos(v));
x=r*(cos(v+w)*cos(Omega)-sin(v+w)*sin(Omega)*cos(i));
y=r*(cos(v+w)*sin(Omega)+sin(v+w)*cos(Omega)*cos(i));
z=r*(sin(v+w)*sin(i));
b=a*sqrt(1-e^2);
ll=cos(Omega)*cos(w)-sin(Omega)*sin(w)*cos(i);
12 = -\cos(\operatorname{Omega}) * \sin(w) - \sin(\operatorname{Omega}) * \cos(w) * \cos(i);
ml=sin(Omega)*cos(w)+cos(Omega)*sin(w)*cos(i);
m2 = -sin(Omega) * sin(w) + cos(Omega) * cos(w) * cos(i);
nl=sin(w)*sin(i);
n2=\cos(w)*\sin(i); \cos E=(e+\cos(v))/(1+e*\cos(v));
sinE=(sqrt(1-e^2)*sin(v))/(1+e*cos(v));
x_dot=(n*a/r)*(b*l2*cosE-a*l1*sinE);
y_dot=(n*a/r)*(b*m2*cosE-a*m1*sinE);
z_dot=(n*a/r)*(b*n2*cosE-a*n1*sinE);
```

```
Orbit_ECI = [r,x,y,z,x_dot,y_dot,z_dot];
```

 $rot_o2i.m: (COE) \rightarrow (\mathbf{R}_o^i)$

```
function R_io = rot_o2i(input)
%*
%*
   rot_o2i.m
                                                *
응*
                                                *
   Author: Oeyvind Hegrenaes, NTNU 2004
8*
%∗
응*
   Given some COE, the function returns the rotation matrix *
   from o \rightarrow i, that is the rotation matrix R_io.
%*
                                                *
%*
                                                *
응*
   Input parameter: COE (Keplarian orbit elements)
                                                *
%*
응*
   Output parameter: R_io (rotation matrix from o -> i)
                                                *
%*
```

i=input(1); Omega=input(4); w=input(5); v=input(6); cos_i=cos(i); sin_i=sin(i); sin_u=sin(w+v); cos_u=cos(w+v);

```
sin_0=sin(Omega); cos_0=cos(Omega);
x_oi=[-cos_i*sin_0*cos_u-cos_0*sin_u , cos_i*cos_0*cos_u-...
sin_0*sin_u , sin_i*cos_u ]';
y_oi=[-sin_i*sin_0 , sin_i*cos_0 , -cos_i ]';
z_oi=[-cos_0*cos_u+cos_i*sin_0*sin_u , -sin_0*cos_u-...
cos_i*cos_0*sin_u , -sin_i*sin_u ]';
```

```
R_io = [x_oi, y_oi, z_oi]; \& Equation (3.5)
```

```
rot2quart : (\mathbf{R} \in SO(3)) \rightarrow (\mathbf{q} = [\eta, \epsilon_1, \epsilon_2, \epsilon_3])
```

```
function q = rot2quart(R)
8*
응*
    rot2quart.m
응*
    Author: Prof. Thor Inge Fossen
응*
응*
    Edited: Oeyvind Hegrenaes, NTNU 2004
응*
응*
    Given a rotation matrix, R in SO3, the function returns
응*
    the corrosponding quaternions.
응*
응*
    Input parameter: R (rotation matrix in SO3)
                                                       *
응*
                                                       *
응*
    Output parameter: q (quaternions)
                                                       *
%*
R(4,4)=trace(R); [Rmax,i]=max([R(1,1) R(2,2) R(3,3) R(4,4)]);
p_i = sqrt(1+2*R(i,i)-R(4,4));
if i = = 1,
  p1 = p_i;
  p2 = (R(2,1)+R(1,2))/p_i;
  p3 = (R(1,3)+R(3,1))/p_i;
  p4 = (R(3,2)-R(2,3))/p_i;
elseif i==2,
  p1 = (R(2,1)+R(1,2))/p_i;
  p2 = p_i;
  p3 = (R(3,2)+R(2,3))/p_i;
  p4 = (R(1,3)-R(3,1))/p_i;
elseif i==3,
  p1 = (R(1,3)+R(3,1))/p_i;
  p2 = (R(3,2)+R(2,3))/p_i;
  p3 = p_{i};
  p4 = (R(2,1)-R(1,2))/p_i;
```

```
else
    p1 = (R(3,2)-R(2,3))/p_i;
    p2 = (R(1,3)-R(3,1))/p_i;
```

```
p3 = (R(2,1)-R(1,2))/p_i;
p4 = p_i;
```

end

```
q=0.5*[p4 p1 p2 p3]'; q=[q/(q'*q)]';
```

```
ucalloc.m : (\boldsymbol{\tau}) \rightarrow (\mathbf{u})
```

```
function u = ucalloc(tau)
%*
                                                          *
   u = ucalloc(K,T,W,tau) unconstrained control allocation.
응*
                                                         *
%*
   The generalized force vector tau = T*K*u (dim n) is
                                                          *
   distributed to the input vector u (dim r) where r>=n by
8*
                                                          *
   minimizing the force f=K*u.
응*
%*
                                                          *
   An unconstrained solution u = inv(K) * inv(W) * T' * ...
%∗
                                                          *
응*
   inv(T*inv(W)*T') exists if T*T' is non-singular.
                                                          *
응*
      - K is a diagonal rxr matrix of force coefficients
응*
                                                          *
      - T is a nxr constant configuration matrix.
%*
                                                          *
      - W is a rxr positive diagonal matrix weighting
%*
                                                          *
        (prizing) the different control forces f = K*u.
%*
                                                          *
응*
    Author:
              Thor I. Fossen
8*
응*
    Date:
              3rd November 2001
%*
T = [1, 0, 0, 0; 0, 1, 0, -1; 0, 0, 1, 0];
K=[1,0,0,0 ; 0,1,0,-1 ;0,0,1,0];
W = [1,0,0,0; 0,1,0,-1; 0,0,1,0];
if det(T*T') == 0,
   error('T*T''is singular');
elseif det(W) == 0,
   error('W must be positive');
else
   Winv = inv(W);
   u = inv(K)*Winv*T'*inv(T*Winv*T')*tau;
end
```

Simulink block diagrams



Figure C.1: The ESEO satellite system



Figure C.2: Satellite model



Figure C.3: Nonlinear satellite subsystem


Figure C.4: Nonlinear kinematics



Figure C.5: Nonlinear dynamics



Figure C.6: Nonlinear dynamics - building blocks



Figure C.7: Linear satellite subsystem



Figure C.8: Rotation matrix manipulation



(b) mean R_c

Figure C.9: Orbit propagator



Figure C.10: Space Environment - gravity gradient



Figure C.11: Controller









(a) PD-controller



(b) Allocate torque

Figure C.14: PD-controller blocks



Figure C.15: Signal modulation

Appendix D Newton-Euler equations of motion

Equations of motion for a rigid body can be derived by summing up the equations of motion for individual mass elements dm with velocity $\vec{v_p}$. A rigid body B with a mass element dm is shown in Figure D.1. The point c is the center of mass, while o is the point where we want to express the equations of motion about. Later it will be assumed that the two points coincide. The material in this chapter is based on Egeland and Gravdahl (2002).



Figure D.1: Rigid body with mass element dm.

D.1 Translational motion

The translational equation of motion with reference to a point o can be written as

$$\vec{f_o} = m\vec{a}_c. \tag{D.1}$$

From Figure D.1 we have that $\vec{r_c} = \vec{r_o} + \vec{r_g}$, hence

$$\vec{v}_c = \vec{v}_o + \vec{\omega}_{ib} \times \vec{r}_g, \tag{D.2}$$

$$\vec{a}_c = \vec{a}_o + \vec{\omega}_{ib} \times \vec{r}_g + \vec{\omega}_{ib} \times (\vec{\omega}_{ib} \times \vec{r}_g), \tag{D.3}$$

where we have used that $\vec{r_g}$ is a constant in b.

Combining (D.1) and (D.3) gives the force equation with reference to the point o:

$$\vec{f}_o = m \left(\vec{a}_o + \dot{\vec{\omega}}_{ib} \times \vec{r}_g + \vec{\omega}_{ib} \times (\vec{\omega}_{ib} \times \vec{r}_g) \right).$$
(D.4)

The translational motion of a spacecraft can be controlled using thrusters. For a spacecraft in orbit the motion is governed by the laws of orbital mechanics. Such a law is the restricted two-body equation of motion:

$$\vec{a} = -\mu \frac{\vec{r}}{|r|^3} \tag{D.5}$$

where \vec{r} is the spacecraft's position and μ is the gravitational parameter for Earth. For more details see a textbook in orbital mechanics, for instance ?.

D.2 Angular motion

The Newton-Euler equations are derived from Euler's First and Second Axioms:

$$\vec{f_c} = m\vec{a_c}$$
 (D.6)

$$\vec{\tau}_c = \vec{h}_c \tag{D.7}$$

$$\vec{\tau}_o = \vec{\tau}_c + \vec{r}_g \times \vec{f}_c \tag{D.8}$$

where the angular momentum about c and o are defined as

$$\vec{h}_c = \int_B (\vec{r} \times \vec{v}_p) dm,$$
 (D.9)

$$\vec{h}_o = \int_B (\vec{r_d} \times \vec{v_p}) dm.$$
 (D.10)

By using that $\vec{v}_p = \vec{v}_o + \vec{\omega}_{ib} \times \vec{r}_d$ and $\vec{r}_d = \vec{r} + \vec{r}_g$, (D.10) can be written as

$$\vec{h}_o = m\vec{r}_g \times \vec{v}_o + \int_B \vec{r}_d \times (\boldsymbol{\omega}_{ib} \times \vec{r}_d) dm.$$
(D.11)

To simplify (D.11) the inertia dyadic

$$I_o = \int_B -S^2(\vec{r_d})dm \tag{D.12}$$

is introduced. The angular momentum about o can then be written as

$$\vec{h}_o = m\vec{r}_g \times \vec{v}_o + I_o \vec{\omega}_{ib}. \tag{D.13}$$

An alternative expression can be found by writing \vec{h}_o as

$$\vec{h}_{o} = \int_{B} (\vec{r} + \vec{r}_{g}) \times \vec{v}_{p} dm$$

$$= \vec{h}_{c} + \int_{B} (\vec{r}_{g} \times \vec{v}_{p}) dm$$

$$= \vec{h}_{c} + \vec{r}_{g} \times m\vec{r}_{c}, \qquad (D.14)$$

where we have used that $\vec{v_c} \equiv \frac{1}{m} \int_B \vec{v_p} dm$.

Time differentiation of \vec{h}_o with respect to the inertial frame yields ¹

$$\dot{\vec{h}}_o = \vec{v}_c \times m\vec{v}_o + \vec{r}_g \times m\dot{\vec{v}}_o + \vec{M}_o\dot{\vec{\omega}}_{ib} + \vec{\omega}_{ib} \times (\vec{M}_o\vec{\omega}_{ib}).$$
(D.15)

Equation (D.14) implies that

$$\vec{h}_o = \vec{h}_c + \vec{r}_g \times m\dot{\vec{v}_c} - \vec{v}_o \times m\vec{v}_c, \tag{D.16}$$

which combined with (D.15) gives

$$\vec{h}_c = \vec{\tau}_c = \vec{r}_g \times m(\dot{\vec{v}}_o - \dot{\vec{v}}_c) + \vec{M}_o \dot{\vec{\omega}}_{ib} + \vec{\omega}_{ib} \times (\vec{M}_o \vec{\omega}_{ib}).$$
 (D.17)

Insertion of (D.17) in (D.8) and using (D.7) gives the angular equation of motion

$$\vec{\tau}_o = \vec{r}_g \times m\vec{a}_o + \vec{M}_o \dot{\vec{\omega}}_{ib} + \vec{\omega}_{ib} \times (\vec{M}_o \vec{\omega}_{ib}).$$
(D.18)

D.3 Model summary

The equations (D.4) and (D.18) can be simplified by letting o coincide with the center of mass c, meaning $\vec{r_g} = \vec{0}$ and $\vec{M_o} = \vec{M_c}$. The simplified equations are

$$\vec{f} = m\vec{a},$$
 (D.19a)

$$\vec{\tau} = \vec{M} \dot{\vec{\omega}}_{ib} + \vec{\omega}_{ib} \times (\vec{M} \vec{\omega}_{ib}), \qquad (D.19b)$$

where the subscript c has been dropped for convenience.

Writing the equations of motion in coordinate form in the b frame yields

$$m\dot{\mathbf{v}}^b = \mathbf{f}^b,\tag{D.20}$$

$$\mathbf{M}\dot{\boldsymbol{\omega}}_{ib}^{b} + \mathbf{S}(\boldsymbol{\omega}_{ib}^{b})\mathbf{M}\boldsymbol{\omega}_{ib}^{b} = \boldsymbol{\tau}^{b}.$$
 (D.21)

At a first glance the translational and angular motion seems decoupled. A closer inspection reveals that this is not the case. The reason is that disturbance torques, $\vec{\tau}_d$, and forces, \vec{f}_d acting on a spacecraft are usually dependent of the spacecraft's position and attitude. However, for our purposes the translational and angular motion can be assumed decoupled.

¹For more details about the derivation refer to Egeland and Gravdahl (2002)

Appendix E Orbital mechanics

Johannes Kepler (1571-1630) formulated the three famous laws of planetary motion from an empirical study based on data collected by the astronomer *Tycho Brahe* (1546-1601). Kepler's laws describe the simplest form of motion of celestial bodies under the assumption that no external perturbing forces are present, and that the respective masses can be considered point masses.

The three laws gave a description of the motion but no explanation. Kepler himself was convinced that his empirically found laws followed a more general law. In 1687 *Isaac Newton* (1642-1727) published his three laws of motion and the law of universal gravitation in the *Mathematical Principles of Natural Philosophy*. By using these laws Kepler's laws can be derived.

E.1 The Two-Body problem

In celestial mechanics we are concerned with motions of celestial bodies under the influence of mutual mass attraction. This Keplerian motion is described by (E.1).

$$\mathbf{a} = -\mu \frac{M+m}{|\mathbf{r}|^3} \mathbf{r} \tag{E.1}$$

For an artificial Earth satellite the mass m can be neglected. The expression above then becomes

$$\mathbf{a} = -\frac{\mu M}{|\mathbf{r}|^3} \mathbf{r} \tag{E.2}$$

with r being the geocentric position of the satellite.

Equation (E.2) is a second order differential equation with solution on the form

$$\mathbf{r}(t) = \mathbf{r}(t; a_1 \cdots a_6)$$

$$\dot{\mathbf{r}}(t) = \dot{\mathbf{r}}(t; a_1 \cdots a_6)$$
 (E.3)

with $a_1 \cdots a_6$ being free selectable integration constants describing the orbit. Usually the six Keplerian orbital parameters $a, e, i, \Omega, \varpi, \nu$ are used.

E.2 Classical Orbital Elements COE

To describe a satellite orbit the six Keplerian elements $a, e, i, \Omega, \varpi, \nu$ can be used. See table E.1. For a good and pedagogic description of the classical orbital elements, see Sellers (2000). The COEs are best understood by looking at figure E.1.

Name	Symbol
Semimajor axis	a
Eccentricity	e
Inclination	i
Longitude of ascending node	Ω
Argument of perigee	$\overline{\omega}$
True anomaly	ν

Table E.1: The six classical orbital elements



Figure E.1: Orbital elements

Semimajor axis *a* The major axis of an elliptical orbit is the distance between the point of closest approach (perigee) and furthest point (apogee). Semimajor axis is one-half this distance.

Eccentricity e A circular orbit has an eccentricity of zero. Elliptical orbits has an eccentricity less than one, while hyperbolic orbits has an eccentricity greater than one.

Inclination *i* Describes the tilt of the orbital plane with the respect to the equatorial plane.

Longitude of ascending node Ω Also called *right ascension of the ascending node*. The ascending node is the point where the satellite crosses the equator moving south to north.

Argument of perigee ϖ Location of the perigee with respect to the ascending node.

True anomaly ν Location of satellite with respect to perigee

Mean anomaly M

Instead of the true anomaly ν the mean anomaly M can be used. M is defined by

$$M = E - e\sin E \tag{E.4}$$

where E is given by

$$\cos E = \frac{e + \cos \nu}{1 + e \cos \nu} \tag{E.5}$$

E.2.1 The orbital period

The orbital period, P is the time it takes for the satellite to revolve once around its orbit. The period can be derived from Kepler's Third Law (Sellers, 2000) as

$$P = 2\pi n^{-1} = 2\pi \sqrt{\frac{a^3}{\mu}}$$
(E.6)

where P is the period in seconds, a is the semimajor axis, while $\mu = 398600.4418 \ [km^3/s^2]$ is the gravitational parameter.

E.2.2 Conversion from COEs to the ECI frame

A satellite position given in COEs can be converted to the ECI frame by (Wertz, 1978)

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = r \cdot \begin{bmatrix} \cos(\nu + \varpi) \cos \Omega - \sin(\nu + \varpi) \sin \Omega \cos i \\ \cos(\nu + \varpi) \sin \Omega + \sin(\nu + \varpi) \cos \Omega \cos i \\ \sin(\nu + \varpi) \cos i \end{bmatrix}$$
(E.7)

where r is given by

$$r = \frac{a(1+e^2)}{1+e\cos\nu}$$
(E.8)

Similarly, we can find the velocity of the satellite in the ECI frame, that is

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \frac{n a}{r} \begin{bmatrix} b l_2 \cos E - a l_1 \sin E \\ b m_2 \cos E - a m_1 \sin E \\ b n_2 \cos E - a n_1 \sin E \end{bmatrix}$$
(E.9)

where n is the mean motion as in (E.6), r is defined in (E.8), and

$$b = a\sqrt{(1-e^2)}$$

$$l_1 = \cos\Omega\cos\varpi - \sin\Omega\sin\varpi\cos i$$

$$m_1 = \sin\Omega\cos\varpi + \cos\Omega\sin\varpi\cos i$$

$$n_1 = \sin\pi\sin i$$

$$l_2 = -\cos\Omega\sin\varpi - \sin\Omega\cos\varpi\cos i$$

$$m_2 = -\sin\Omega\sin\varpi + \cos\Omega\cos\varpi\cos i$$

$$n_2 = \cos\pi\sin i$$

E.3 Perturbed Satellite Motion

Equation (E.2) describes the ideal motion of a satellite around a body when only the mutual gravitational forces are considered. In reality a certain number of additional forces act on a satellite. The extended equation of motion is given by:

$$\vec{a} = -\mu \frac{M}{|\vec{r}|^3} \vec{r} + \vec{k}_s$$
 (E.10)

where

$$\vec{k}_s = \vec{a}_E + \vec{a}_m + \vec{a}_e + \vec{a}_o + \vec{a}_D + \vec{a}_{SP} + \vec{a}_A$$
(E.11)

Perturbing forces on a near-Earth satellite are in particular(Seeber, 1993)

- 1. accelerations due to the non-spherically and inhomogeneous mass distribution within the Earth, $\ddot{\mathbf{r}}_E$
- 2. accelerations due to other celestial bodies (Sun, Moon and planets), mainly $\ddot{\mathbf{r}}_s$, $\ddot{\mathbf{r}}_m$
- 3. accelerations due to Earth and oceanic tides $\ddot{\mathbf{r}}_e$, $\ddot{\mathbf{r}}_o$
- 4. accelerations due to the atmospheric drag $\ddot{\mathbf{r}}_D$
- 5. accelerations due to direct and Earth-reflected solar radiation pressure $\ddot{\mathbf{r}}_{SP}$, $\ddot{\mathbf{r}}_{A}$

Expressions for the perturbing forces can be found in Seeber (1993). An easy to understand description of the main perturbations and its implications can be found in Sellers (2000).

Appendix F The (w, z) parametrization

As mentioned in Chapter 2 the three-dimensional (w, z) parametrization is a relatively new formulation for describing the relative orientation of two reference frames using two perpendicular rotations. Although it uses three parameters to describe the motion, two of the parameters can be combined to a single complex variable. The complex variable is used to designate the second of the two rotations and it is derived using stereographic projection (Conway, 1978). The remaining parameter represents the initial rotation. As will be shown, the two rotations can be completely decoupled, which has important implications and advantages. Since this parametrization is quite new and not very well known it is here derived in detail. The material in this section is based on the work by Tsiotras and Longuski (1995, 1996).

Parametrization of the rotation matrix $\mathbf{R} \in SO(3)$

As mentioned above, we wish to derive a parametrization of the rotation matrix \mathbf{R} in section 2.1 using two successive rotations. The resulting rotation matrix can thus be decomposed as

$$\mathbf{R}(\mathbf{w}, z) \triangleq \mathbf{R}_i^b(\mathbf{w}, z) = \mathbf{R}_o^b(\mathbf{w}) \mathbf{R}_i^o(z) \tag{F.1}$$

This can be interpreted as follows. Consider a inertial frame *i*, defined by three orthogonal unit vectors $\vec{i_1}$, $\vec{i_2}$ and $\vec{i_3}$. Another frame *b* is fixed to the body and is defined by the three orthogonal unit vectors $\vec{b_1}$, $\vec{b_2}$ and $\vec{b_3}$. The rotation matrix **R** in (F.1) is then the rotation matrix from the the body frame to the inertial frame. The intermediate frame *o*, defined by the three orthogonal unit vectors $\vec{o_1}$, $\vec{o_2}$ and $\vec{o_3}$ is the result of the first rotation.

The first rotation is shown in Figure F.1 a) and represents a simple rotation about the inertial \vec{i}_3 -axis. This resulting rotation matrix can be written as

$$\mathbf{R}_{i}^{o}(z) = \begin{bmatrix} \cos z & \sin z & 0\\ -\sin z & \cos z & 0\\ 0 & 0 & 1 \end{bmatrix} \triangleq \mathbf{R}_{1}(z)$$
(F.2)

The next step is to find an expression for the second rotation matrix $\mathbf{R}_o^b(\mathbf{w}) \triangleq \mathbf{R}_2(\mathbf{w})$. Recall that a rotation matrix can be computed using an angle-axis description, which was shown in section 2.1 by using *Rodrigues' fomula* (2.2). The formula is repeated here for clarity;

$$\mathbf{R}(\mathbf{k},\theta) = \mathbf{I} + \mathbf{S}(\mathbf{k})\sin\theta + \mathbf{S}^{2}(\mathbf{k})(1 - \cos\theta)$$
(F.3)



Figure F.1: Rotation sequence of the (w, z) parametrization. Left hand side shows the initial rotation about the body \vec{i}_3 -axis. Right hand side shows the resulting coordinate system after the second rotation.

Assume that $\vec{o}_3 = a\vec{b}_1 + b\vec{b}_2 + c\vec{b}_3$, which is the \vec{o}_3 -axis in the intermediate frame, transformed to the body frame. In sense of the rotation matrix $\mathbf{R}_o^b(\mathbf{k},\theta)$ this can be written as

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \mathbf{R}_o^b(\mathbf{k}, \theta) \vec{o}_3 \tag{F.4}$$

The resulting vector $[a, b, c]^T$ is clearly the third column in the rotation matrix \mathbf{R}_2 . Also, it can be shown that (-a, -b, c) are the directed cosines of the \vec{b}_3 -axis in the *o* frame, that is

$$\vec{b}_3 = -a\vec{o}_1 - b\vec{o}_2 + c\vec{o}_3 \tag{F.5}$$

The angle between \vec{b}_3 and \vec{o}_3 is simply found from the vector dot product as

$$\theta = \cos^{-1} \frac{\vec{o}_3 \cdot \vec{b}_3}{|\vec{o}_3| |\vec{b}_3|} = \cos^{-1} c \tag{F.6}$$

The axis of rotation can be found from (note that \vec{k} has the same coordinates in both frames)

$$\vec{k} = \frac{\vec{o}_3 \times \vec{b}_3}{\|\vec{o}_3 \times \vec{b}_3\|} \tag{F.7}$$

Using $\vec{o}_3 = [0, 0, 1]^T$, when refereed to the intermediate reference frame, and (F.5), the axis of rotation can be calculated as

$$\vec{k} = \frac{b\vec{o}_1 - a\vec{o}_2}{\sqrt{a^2 + b^2}}$$
(F.8)

Now the angle-axis description of the rotation matrix can be used. Insertion of (F.8) and (F.6) into (F.3) gives the rotation matrix from the intermediate frame to the body frame, that is

$$\mathbf{R}_{b}^{o}(\mathbf{k},\theta) = \begin{bmatrix} c + \frac{b^{2}}{1+c} & -\frac{ab}{1+c} & -a\\ -\frac{ab}{1+c} & c + \frac{a^{2}}{1+c} & -b\\ a & b & c \end{bmatrix}$$
(F.9)



Figure F.2: Stereographic projection of a point (a, b, c) on a unit sphere on to a complex plane.

By taking the transpose of (F.9) we get the following expression for the rotation matrix $\mathbf{R}_2(w)$

$$\mathbf{R}_{o}^{b}(\mathbf{k},\theta) = \begin{bmatrix} c + \frac{b^{2}}{1+c} & -\frac{ab}{1+c} & a\\ -\frac{ab}{1+c} & c + \frac{a^{2}}{1+c} & b\\ -a & -b & c \end{bmatrix} \triangleq \mathbf{R}_{2}(\mathbf{w})$$
(F.10)

Expanding (F.1) with (F.2) and (F.10) gives the complicated matrix

$$\mathbf{R}(\mathbf{w},z) = \begin{bmatrix} \frac{c\cos z + ab\sin z + (b^2 + c^2)\cos z}{1+c} & \frac{c\sin z - ab\cos z + (b^2 + c^2)\sin z}{1+c} & a\\ -\frac{c\sin z + (c^2 + a^2)\sin z + ab\cos z}{1+c} z & \frac{c\cos z + (c^2 + a^2)\cos z - ab\sin z}{1+c} & b\\ -b\sin z - a\cos z & -b\cos z - a\sin z & c \end{bmatrix}$$
(F.11)

The foregoing matrix is redundant, in the sense that the elements a,b,c are not independent, but satisfy the constraint

$$a^2 + b^2 + c^2 = 1 \tag{F.12}$$

We can therefore eliminate one of the elements in (F.11) to obtain a simplified form. One way of doing this is to use a stereographic projection. From the constraint in (F.12) we introduce the set $S^2 = \{(x_1, x_2, x_3) \in \mathbb{R}^3 : x_1^2 + x_2^2 + x_3^2 = 1\} \in \mathbb{R}^3$, which is the unit sphere. For $(a, b, c) \in S^2$, we get the mapping $\sigma : S^2 \setminus \{(0, 0, -1)\} \to \mathbb{C}, (a, b, c) \mapsto w$. The stereographic projection is shown in Figure F.2 and is defined as

$$\mathbf{w} \triangleq \mathbf{w}_1 + i\mathbf{w}_2 = \frac{b - ia}{1 + c} \tag{F.13}$$

It can be verified that the inverse map $\sigma^{-1}: \mathbb{C} \to S^2 \setminus \{(0, 0, -1)\}, w \mapsto (a, b, c) \text{ is given by }$

$$a = \frac{i(\mathbf{w} - \bar{\mathbf{w}})}{1 + |\mathbf{w}|^2}, \quad b = \frac{\mathbf{w} + \bar{\mathbf{w}}}{1 + |\mathbf{w}|^2}, \quad c = \frac{1 - |\mathbf{w}|^2}{1 + |\mathbf{w}|^2}$$
(F.14)

where \bar{w} is the complex conjugate of w and $|w|^2 = w\bar{w} = w_1^2 + w_2^2$. The basis of the projection is the point (0, 0, -1), which is the south pole S of the unit sphere. Note that when c = -1the projection has a singularity and $w \to \infty$. The singularity corresponds to an upside-down orientation of the body. Using (F.14) we can express $\mathbf{R}_2(w)$ in terms of (F.13) as follows

$$\mathbf{R}_{2}(\mathbf{w}) = \frac{1}{1 + w_{1}^{2} + w_{2}^{2}} \begin{bmatrix} 1 + w_{1}^{2} - w_{2}^{2} & 2w_{1}w_{2} & -2w_{2} \\ 2w_{1}w_{2} & 1 - w_{1}^{2} + w_{2}^{2} & 2w_{1} \\ 2w_{2} & -2w_{1} & 1 - w_{1}^{2} - w_{2}^{2} \end{bmatrix}$$
(F.15)

When using complex notation, a more compact matrix can be given as

$$\mathbf{R}_{2}(\mathbf{w}) = \frac{1}{1+|\mathbf{w}|^{2}} \begin{bmatrix} 1 + \operatorname{Re}(\mathbf{w}^{2}) & \operatorname{Im}(\mathbf{w}^{2}) & -2\operatorname{Im}(\mathbf{w}) \\ \operatorname{Im}(\mathbf{w}^{2}) & 1 - \operatorname{Re}(\mathbf{w}^{2}) & 2\operatorname{Re}(\mathbf{w}) \\ 2\operatorname{Im}(\mathbf{w}) & -2\operatorname{Re}(\mathbf{w}) & 1 - |\mathbf{w}|^{2} \end{bmatrix}$$
(F.16)

The total rotation matrix $\mathbf{R}(w, z)$ is then given in terms of w and z as follows

$$\beta \cdot \begin{bmatrix} (1+w_1^2-w_2^2)cz - 2w_1w_2sz & (1+w_1^2-w_2^2)sz + 2w_1w_2cz & -2w_2\\ 2w_1w_2cz - (1-w_1^2+w_2^2)sz & 2w_1w_2sz + (1-w_1^2+w_2^2)cz & 2w_1\\ 2w_2cz + 2w_1sz & 2w_2sz - 2w_1cz & 1-w_1^2-w_2^2 \end{bmatrix}$$
(F.17)

where cz and sz denotes $\cos(z)$ and $\sin(z)$, respectively, and $\beta = 1/(1 + |w|^2)$.

The foregoing matrix can be written more compactly as

$$\mathbf{R}(\mathbf{w}, z) = \frac{1}{1 + |\mathbf{w}|^2} \begin{bmatrix} \operatorname{Re}(1 + \mathbf{w}^2)e^{iz} & \operatorname{Im}(1 + \mathbf{w}^2)e^{iz} & -2\operatorname{Im}(\mathbf{w}) \\ \operatorname{Im}(1 - \bar{\mathbf{w}}^2)e^{-iz} & \operatorname{Re}(1 - \bar{\mathbf{w}}^2)e^{-iz} & 2\operatorname{Re}(\mathbf{w}) \\ 2\operatorname{Im}(\mathbf{w}e^{iz}) & -2\operatorname{Re}(\mathbf{w}e^{iz}) & 1 - |\mathbf{w}|^2 \end{bmatrix}$$
(F.18)

Kinematic differential equations

From (2.6) we can derive the kinematic differential equations for the attitude motion of the rigid body. The differential equation for (F.1) becomes $\dot{\mathbf{R}}(\mathbf{w}, z) = \mathbf{S}(\bar{\boldsymbol{\omega}})\mathbf{R}(\mathbf{w}, z)$, where $\bar{\boldsymbol{\omega}} = \boldsymbol{\omega}_{bi}^b$ is the angular velocity of the inertial frame relative to the body frame, as seen from the body frame. It can be shown however, that $\boldsymbol{\omega}_{bi}^b = -\boldsymbol{\omega}_{ib}^b$ (Kane et al., 1983). This relation makes it possible to derive the kinematics using the more intuitive angular velocity $\boldsymbol{\omega}_{ib}^b$, which is the angular velocity of the body frame relative to the inertial frame, as seen from the body frame. For the reminder we let $\boldsymbol{\omega} = \boldsymbol{\omega}_{ib}^b \triangleq [\omega_1, \omega_2, \omega_3]^T$, and the redefined differential equation for (F.1) becomes $\dot{\mathbf{R}}(\mathbf{w}, z) = -\mathbf{S}(\boldsymbol{\omega})\mathbf{R}(\mathbf{w}, z)$, where $\mathbf{S}(\boldsymbol{\omega})$ was defined in (2.4).

It can easily be verified that the third column of $\dot{\mathbf{R}}(\mathbf{w}, z)$ must satisfy

$$\begin{bmatrix} \dot{a} \\ \dot{b} \\ \dot{c} \end{bmatrix} = -\mathbf{S}(\boldsymbol{\omega}) \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$
(F.19)

Recall from (F.13) that w is defined as

$$w = \frac{b - ia}{1 + c} \tag{F.20}$$

Differentiation of (F.20) gives

$$\dot{\mathbf{w}} = \frac{\dot{b} - i\dot{a} - \mathbf{w}\dot{c}}{1 + c} \tag{F.21}$$

Using the relations in (F.14) and (F.19) gives the differential equation for $w \in \mathbb{C}$

$$\dot{\mathbf{w}} = -i\omega_3 \mathbf{w} + \frac{\omega}{2} + \frac{\bar{\omega}}{2} \mathbf{w}^2 \tag{F.22}$$

where

$$\omega = \omega_1 + i\omega_2, \quad \bar{\omega} = \omega_1 - i\omega_2, \quad i = \sqrt{-1}$$
(F.23)

An alternative formulation, which will be used throughout, is given as

$$\dot{w}_1 = \omega_3 w_2 + \omega_2 w_1 w_2 + \frac{\omega_1}{2} (1 + w_1^2 - w_2^2)$$
 (F.24a)

$$\dot{\mathbf{w}}_2 = -\omega_3 \mathbf{w}_1 + \omega_1 \mathbf{w}_1 \mathbf{w}_2 + \frac{\omega_2}{2} (1 + \mathbf{w}_2^2 - \mathbf{w}_1^2)$$
 (F.24b)

To find the differential equation for z we start with the scalar form of the differential equation for a rotation matrix,

$$tr[\dot{\mathbf{R}}(\mathbf{w}, z)] = tr[-\mathbf{S}(\boldsymbol{\omega})\mathbf{R}(\mathbf{w}, z)]$$
(F.25)

where $tr(\cdot)$ denotes the trace of the matrix. Taking the trace of $\dot{\mathbf{R}}(\mathbf{w}, z)$ gives

$$tr[\dot{\mathbf{R}}(\mathbf{w},z)] = \frac{d}{dt} (tr[\mathbf{R}(\mathbf{w},z)]) = \frac{d}{dt} \left(\frac{2\cos z + 2}{1 + w_1^2 + w_2^2} - 1 \right)$$
$$= -\frac{2\dot{z}\sin z}{1 + w_1^2 + w_2^2} - \frac{4(1 + \cos z)(w_1\dot{w}_1 + w_2\dot{w}_2)}{(1 + w_1^2 + w_2^2)^2}$$
(F.26)

Combining (F.24a) and (F.24b) gives the relation

$$2\frac{\mathbf{w}_1\dot{\mathbf{w}}_1 + \mathbf{w}_2\dot{\mathbf{w}}_2}{(1 + \mathbf{w}_1^2 + \mathbf{w}_2^2)^2} = \omega_1\mathbf{w}_1 + \omega_2\mathbf{w}_2$$
(F.27)

Substituted (F.27) into (F.26) gives the expression

$$tr[\dot{\mathbf{R}}(\mathbf{w},z)] = \frac{2}{1 + w_1^2 + w_2^2} (\dot{z}\sin z + (1 + \cos z)(\omega_1 w_1 + \omega_2 w_2))$$
(F.28)

Expanding the right hand side of (F.25) we obtain

$$tr[-\mathbf{S}(\boldsymbol{\omega})\mathbf{R}(\mathbf{w},z)] = \frac{-2}{1+\mathbf{w}_1^2+\mathbf{w}_2^2} \left[(1+\cos z)(\omega_1\mathbf{w}_1+\omega_2\mathbf{w}_2) + (\omega_3-\omega_1\mathbf{w}_2+\omega_2\mathbf{w}_1)\sin z \right]$$
(E29)

Equating (F.29) with (F.28), we obtain the following differential equation for the angle z

$$\dot{z} = \omega_3 - \omega_1 \mathbf{w}_2 + \omega_2 \mathbf{w}_1 \tag{F.30}$$

or equivalently,

$$\dot{z} = \omega_3 + \frac{i}{2}(\bar{\omega}\mathbf{w} - \omega\bar{\mathbf{w}}) \tag{F.31}$$

To summarize the discussion above, the differential kinematic equations for the (w, z) parametrization are

$$\dot{w}_1 = \omega_3 w_2 + \omega_2 w_1 w_2 + \frac{\omega_1}{2} (1 + w_1^2 - w_2^2)$$
 (F.32a)

$$\dot{\mathbf{w}}_2 = -\omega_3 \mathbf{w}_1 + \omega_1 \mathbf{w}_1 \mathbf{w}_2 + \frac{\omega_2}{2} (1 + \mathbf{w}_2^2 - \mathbf{w}_1^2)$$
 (F.32b)

$$\dot{z} = \omega_3 - \omega_1 \mathbf{w}_2 + \omega_2 \mathbf{w}_1 \tag{F.32c}$$

Alternatively they can be written more compactly as

$$\dot{\mathbf{w}} = -i\omega_3\mathbf{w} + \frac{\omega}{2} + \frac{\bar{\omega}}{2}\mathbf{w}^2,$$
 (F.33a)

$$\dot{z} = \omega_3 + \frac{i}{2}(\bar{\omega}w - \omega\bar{w})$$
 (F.33b)

w	Kinematics	ω
$\frac{a+ib}{1+c}$	$\dot{\mathbf{w}} = -i\left(\omega_3\mathbf{w} - \frac{\omega}{2} + \frac{\bar{\omega}}{2}\mathbf{w}^2\right)$ $\dot{z} = \omega_3 + \frac{1}{2}\left(\omega\bar{\mathbf{w}} + \bar{\omega}\mathbf{w}\right)$	$\omega_1 + i\omega_2$
$\frac{b-ia}{1+c}$	$\dot{\mathbf{w}} = -i\omega_3\mathbf{w} + \frac{\omega}{2} + \frac{\bar{\omega}}{2}\mathbf{w}^2$ $\dot{z} = \omega_3 + \frac{i}{2}\left(\bar{\omega}\mathbf{w} - \omega\bar{\mathbf{w}}\right)$	$\omega_1 + i\omega_2$
$\frac{b+ic}{1+a}$	$\dot{\mathbf{w}} = -i\left(\omega_1\mathbf{w} - \frac{\omega}{2} + \frac{\bar{\omega}}{2}\mathbf{w}^2\right)$ $\dot{z} = \omega_1 + \frac{1}{2}\left(\omega\bar{\mathbf{w}} + \bar{\omega}\mathbf{w}\right)$	$\omega_2 + i\omega_3$
$\frac{c-ib}{1+a}$	$\dot{\mathbf{w}} = -i\omega_1\mathbf{w} + \frac{\omega}{2} + \frac{\bar{\omega}}{2}\mathbf{w}^2$ $\dot{z} = \omega_1 + \frac{i}{2}\left(\bar{\omega}\mathbf{w} - \omega\bar{\mathbf{w}}\right)$	$\omega_2 + i\omega_3$
$\frac{c+ia}{1+b}$	$\dot{\mathbf{w}} = -i\left(\omega_2 \mathbf{w} - \frac{\omega}{2} + \frac{\bar{\omega}}{2}\mathbf{w}^2\right)$ $\dot{z} = \omega_2 + \frac{1}{2}\left(\omega\bar{\mathbf{w}} + \bar{\omega}\mathbf{w}\right)$	$\omega_3 + i\omega_1$
$\frac{a-ic}{1+b}$	$\dot{\mathbf{w}} = -i\omega_2\mathbf{w} + \frac{\omega}{2} + \frac{\bar{\omega}}{2}\mathbf{w}^2$ $\dot{z} = \omega_2 + \frac{i}{2}\left(\bar{\omega}\mathbf{w} - \omega\bar{\mathbf{w}}\right)$	$\omega_3 + i\omega_1$

Table F.1: Stereographic coordinate w and corresponding kinematics

Remark F.0.1. It is straight forward to verify that the kinematic differential equations for the (w, z) parametrization can be written as

$$\frac{d}{dt}|\mathbf{w}|^2 = (1+|\mathbf{w}|^2)\operatorname{Re}(\omega\bar{\mathbf{w}})$$
(F.34a)

$$\dot{z} = \omega_3 + \operatorname{Im}(\omega \bar{w})$$
 (F.34b)

Remark F.0.2. Equation (F.13) is only one of the possible definitions of the parameter w. Other combinations will provide different kinematic equations for w and z. This is shown in Table F.1.